Handbook of Malware 2016 - A Wikipedia Book

Book · July 2016

DOI: 10.13140/RG.2.1.5039.5122

2 authors, including:

Reiner Creutzburg
Brandenburg University of Applied Sciences
489 PUBLICATIONS **472** CITATIONS

SEE PROFILE

Handbook of Malware 2016

A Wikipedia Book

By Wikipedians

Edited by:

Reiner Creutzburg
Technische Hochschule Brandenburg
Fachbereich Informatik und Medien
PF 2132
D-14737 Brandenburg
Germany

Email: creutzburg@th-brandenburg.de

Contents

Chapter 1

Malware - Introduction

1.1 Malware

Beast, a Windows-based backdoor Trojan horse.

Malware, short for **malicious software**, is any software used to disrupt computer operations, gather sensitive information, gain access to private computer systems, or display unwanted advertising.[1] Malicious software was called computer virus before the term malware was coined in 1990 by Yisrael Radai.[2] The first category of malware propagation concerns parasitic software fragments that attach themselves to some existing executable content. The fragment may be machine code that infects some existing application, utility, or system program, or even the code used to boot a computer system.[3] Malware is defined by its malicious intent, acting against the requirements of the computer user, and does not include software that causes unintentional harm due to some deficiency.

Malware may be stealthy, intended to steal information or spy on computer users for an extended period without their knowledge, as for example Regin, or it may be designed to cause harm, often as sabotage (e.g., Stuxnet), or to extort payment (CryptoLocker). 'Malware' is an umbrella term used to refer to a variety of forms of hostile or intrusive software,[4] including computer viruses, worms, trojan horses, ransomware, spyware, adware, scareware, and other malicious programs. It can take the form of executable code, scripts, active content, and other software.[5] Malware is often disguised as, or embedded in, non-malicious files. As of 2011 the majority of active malware threats were worms or trojans rather than viruses.[6]

In law, malware is sometimes known as a **computer contaminant**, as in the legal codes of several U.S. states.[7][8]

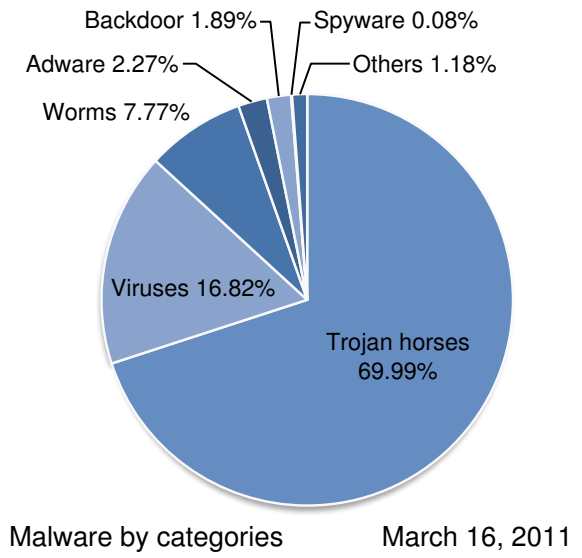
Spyware or other malware is sometimes found embedded in programs supplied officially by companies, e.g., downloadable from websites, that appear useful or attractive, but may have, for example, additional hidden tracking functionality that gathers marketing statistics. An example of such software, which was described as illegitimate, is the Sony rootkit, a Trojan embedded into CDs sold by Sony, which silently installed and concealed itself on purchasers' computers with the intention of preventing illicit copying; it also reported on users' listening habits, and unintentionally created vulnerabilities that were exploited by unrelated malware.[9]

Software such as anti-virus, anti-malware, and firewalls are used to protect against activity identified as malicious, and to recover from attacks.[10]

1.1.1 Purposes

Many early infectious programs, including the first Internet Worm, were written as experiments or pranks. Today, malware is used by both black hat hackers and governments, to steal personal, financial, or business information.[11][12]

Malware is sometimes used broadly against government or corporate websites to gather guarded information,[13] or to disrupt their operation in general. However, malware is often used against individuals to gain information such as personal identification numbers or details, bank or credit card numbers, and passwords. Left unguarded, personal and networked computers can be at considerable risk against these threats. (These are most frequently defended



Malware by categories on 16 March 2011.

against by various types of firewall, anti-virus software, and network hardware).[14]

Since the rise of widespread broadband Internet access, malicious software has more frequently been designed for profit. Since 2003, the majority of widespread viruses and worms have been designed to take control of users' computers for illicit purposes.[15] Infected "zombie computers" are used to send email spam, to host contraband data such as child pornography,[16] or to engage in distributed denial-of-service attacks as a form of extortion.[17]

Programs designed to monitor users' web browsing, display unsolicited advertisements, or redirect affiliate marketing revenues are called spyware. Spyware programs do not spread like viruses; instead they are generally installed by exploiting security holes. They can also be hidden and packaged together with unrelated user-installed software.[18]

Ransomware affects an infected computer in some way, and demands payment to reverse the damage. For example, programs such as CryptoLocker encrypt files securely, and only decrypt them on payment of a substantial sum of money.

Some malware is used to generate money by click fraud, making it appear that the computer user has clicked an advertising link on a site, generating a payment from the advertiser. It was estimated in 2012 that about 60 to 70% of all active malware used some kind of click fraud, and 22% of all ad-clicks were fraudulent.[19]

Malware is usually used for criminal purposes, but can be used for sabotage, often without direct benefit to the perpetrators. One example of sabotage was Stuxnet, used to destroy very specific industrial equipment. There have been politically motivated attacks that have spread over and shut down large computer networks, including massive deletion of files and corruption of master boot records, described as "computer killing". Such attacks were made on Sony Pictures Entertainment (25 November 2014, using malware known as Shamoon or W32.Disttrack) and Saudi Aramco (August 2012).[20][21]

1.1.2 Proliferation

Preliminary results from Symantec published in 2008 suggested that "the release rate of malicious code and other unwanted programs may be exceeding that of legitimate software applications."[22] According to F-Secure, "As much malware [was] produced in 2007 as in the previous 20 years altogether."[23] Malware's most common pathway from criminals to users is through the Internet: primarily by e-mail and the World Wide Web.[24]

The prevalence of malware as a vehicle for Internet crime, along with the challenge of anti-malware software to keep up with the continuous stream of new malware, has seen the adoption of a new mindset for individuals and businesses using the Internet. With the amount of malware currently being distributed, some percentage of computers are currently assumed to be infected. For businesses, especially those that sell mainly over the Internet, this means they need to find a way to operate despite security concerns. The result is a greater emphasis on back-office protection designed to protect against advanced malware operating on customers' computers.[25] A 2013 Webroot study shows that 64% of companies allow remote access to servers for 25% to 100% of their workforce and that companies with more than 25% of their employees accessing servers remotely have higher rates of malware threats.[26]

On 29 March 2010, Symantec Corporation named Shaoxing, China, as the world's malware capital.[27] A 2011 study from the University of California, Berkeley, and the Madrid Institute for Advanced Studies published an article in *Software Development Technologies*, examining how entrepreneurial hackers are helping enable the spread of malware by offering access to computers for a price. Microsoft reported in May 2011 that one in every 14 downloads from the Internet may now contain malware code. Social media, and Facebook in particular, are seeing a rise in the number of tactics used to spread malware to computers.[28]

A 2014 study found that malware is being increasingly aimed at mobile devices such as smartphones as they increase in popularity.[29]

1.1.3 Infectious malware: viruses and worms

Main articles: Computer virus and Computer worm

The best-known types of malware, viruses and worms, are known for the manner in which they spread, rather than any specific types of behavior. The term *computer virus* is used for a program that embeds itself in some other executable software (including the operating system itself) on the target system without the user's consent and when that is run causes the virus to spread to other executables. On the other hand, a *worm* is a stand-alone malware program that *actively* transmits itself over a network to infect other computers. These definitions lead to the observation that a virus requires the user to run an infected program or operating system for the virus to spread, whereas a worm spreads itself.[30]

1.1.4 Concealment: Viruses, trojan horses, rootkits, backdoors and evasion

These categories are not mutually exclusive, so malware may use multiple techniques.[31] This section only applies to malware designed to operate undetected, not sabotage and ransomware.

See also: Polymorphic packer

Viruses

Main article: Computer virus

A computer program usually hidden within another seemingly innocuous program that produces copies of itself and inserts them into other programs or files, and that usually performs a malicious action (such as destroying data).[32]

Trojan horses

Main article: Trojan horse (computing)

In computing, Trojan horse, or **Trojan**, is any malicious computer program which misrepresents itself to appear useful, routine, or interesting in order to persuade a victim to install it. The term is derived from the Ancient Greek story of the wooden horse that was used to help Greek troops invade the city of Troy by stealth.[33][34][35][36][37]

Trojans are generally spread by some form of social engineering, for example where a user is duped into executing an e-mail attachment disguised to be unsuspicious, (e.g., a routine form to be filled in), or by drive-by download. Although their payload can be anything, many modern forms act as a backdoor, contacting a controller which can then have unauthorized access to the affected computer.[38] While Trojans and backdoors are not easily detectable by themselves, computers may appear to run slower due to heavy processor or network usage.

Unlike computer viruses and worms, Trojans generally do not attempt to inject themselves into other files or otherwise propagate themselves.[39]

Rootkits

Main article: Rootkit

Once a malicious program is installed on a system, it is essential that it stays concealed, to avoid detection. Software packages known as *rootkits* allow this concealment, by modifying the host's operating system so that the malware is hidden from the user. Rootkits can prevent a malicious process from being visible in the system's list of processes, or keep its files from being read.[40]

Some malicious programs contain routines to defend against removal, not merely to hide themselves. An early example of this behavior is recorded in the Jargon File tale of a pair of programs infesting a Xerox CP-V time sharing system:

> Each ghost-job would detect the fact that the other had been killed, and would start a new copy of the recently stopped program within a few milliseconds. The only way to kill both ghosts was to kill them simultaneously (very difficult) or to deliberately crash the system.[41]

Backdoors

Main article: Backdoor (computing)

A backdoor is a method of bypassing normal authentication procedures, usually over a connection to a network such as the Internet. Once a system has been compromised, one or more backdoors may be installed in order to allow access in the future,[42] invisibly to the user.

The idea has often been suggested that computer manufacturers preinstall backdoors on their systems to provide technical support for customers, but this has never been reliably verified. It was reported in 2014 that US government agencies had been diverting computers purchased by those considered "targets" to secret workshops where software or hardware permitting remote access by the agency

was installed, considered to be among the most productive operations to obtain access to networks around the world.[43] Backdoors may be installed by Trojan horses, worms, implants, or other methods.[44][45]

Evasion

Since the beginning of 2015, a sizable portion of malware utilizes a combination of many techniques designed to avoid detection and analysis.[46]

- The most common evasion technique is when the malware evades analysis and detection by fingerprinting the environment when executed.[47]
- The second most common evasion technique is confusing automated tools' detection methods. This allows malware to avoid detection by technologies such as signature-based antivirus software by changing the server used by the malware.[48]
- The third most common evasion technique is timing-based evasion. This is when malware runs at certain times or following certain actions taken by the user, so it executes during certain vulnerable periods, such as during the boot process, while remaining dormant the rest of the time.
- The fourth most common evasion technique is done by obfuscating internal data so that automated tools do not detect the malware.[49]
- An increasingly common technique is adware that uses stolen certificates to disable anti-malware and virus protection; technical remedies are available to deal with the adware.[50]

Nowadays, one of the most sophisticated and stealthy ways of evasion is the use information hiding techniques, namely Stegomalware.

1.1.5 Vulnerability to malware

Main article: Vulnerability (computing)

- In this context, and throughout, what is called the "system" under attack may be anything from a single application, through a complete computer and operating system, to a large network.
- Various factors make a system more vulnerable to malware:

Security defects in software

Malware exploits security defects (security bugs or vulnerabilities) in the design of the operating system, in applications (such as browsers, e.g. older versions of Microsoft Internet Explorer supported by Windows XP[51]), or in vulnerable versions of browser plugins such as Adobe Flash Player, Adobe Acrobat or Reader, or Java SE.[52][53] Sometimes even installing new versions of such plugins does not automatically uninstall old versions. Security advisories from plug-in providers announce security-related updates.[54] Common vulnerabilities are assigned CVE IDs and listed in the US National Vulnerability Database. Secunia PSI[55] is an example of software, free for personal use, that will check a PC for vulnerable out-of-date software, and attempt to update it.

Malware authors target bugs, or loopholes, to exploit. A common method is exploitation of a buffer overrun vulnerability, where software designed to store data in a specified region of memory does not prevent more data than the buffer can accommodate being supplied. Malware may provide data that overflows the buffer, with malicious executable code or data after the end; when this payload is accessed it does what the attacker, not the legitimate software, determines.

Insecure design or user error

Early PCs had to be booted from floppy disks; when built-in hard drives became common the operating system was normally started from them, but it was possible to boot from another boot device if available, such as a floppy disk, CD-ROM, DVD-ROM, USB flash drive or network. It was common to configure the computer to boot from one of these devices when available. Normally none would be available; the user would intentionally insert, say, a CD into the optical drive to boot the computer in some special way, for example to install an operating system. Even without booting, computers can be configured to execute software on some media as soon as they become available, e.g. to autorun a CD or USB device when inserted.

Malicious software distributors would trick the user into booting or running from an infected device or medium; for example, a virus could make an infected computer add autorunnable code to any USB stick plugged into it; anyone who then attached the stick to another computer set to autorun from USB would in turn become infected, and also pass on the infection in the same way.[56] More generally, any device that plugs into a USB port-—"including gadgets like lights, fans, speakers, toys, even a digital microscope"—can be used to spread malware. Devices can be infected during manufacturing or supply if quality control is inadequate.[56]

This form of infection can largely be avoided by setting up computers by default to boot from the internal hard drive, if available, and not to autorun from devices.[56] Intentional booting from another device is always possible by pressing certain keys during boot.

Older email software would automatically open HTML email containing potentially malicious JavaScript code; users may also execute disguised malicious email attachments and infected executable files supplied in other ways.

Over-privileged users and over-privileged code

Main article: principle of least privilege

In computing, privilege refers to how much a user or program is allowed to modify a system. In poorly designed computer systems, both users and programs can be assigned more privileges than they should be, and malware can take advantage of this. The two ways that malware does this is through overprivileged users and overprivileged code.

Some systems allow all users to modify their internal structures, and such users today would be considered over-privileged users. This was the standard operating procedure for early microcomputer and home computer systems, where there was no distinction between an *administrator* or *root*, and a regular user of the system. In some systems, non-administrator users are over-privileged by design, in the sense that they are allowed to modify internal structures of the system. In some environments, users are over-privileged because they have been inappropriately granted administrator or equivalent status.

Some systems allow code executed by a user to access all rights of that user, which is known as over-privileged code. This was also standard operating procedure for early microcomputer and home computer systems. Malware, running as over-privileged code, can use this privilege to subvert the system. Almost all currently popular operating systems, and also many scripting applications allow code too many privileges, usually in the sense that when a user executes code, the system allows that code all rights of that user. This makes users vulnerable to malware in the form of e-mail attachments, which may or may not be disguised.

Use of the same operating system

- Homogeneity: e.g. when all computers in a network run the same operating system; upon exploiting one, one worm can exploit them all:[57] For example, Microsoft Windows or Mac OS X have such a large share of the market that concentrating on either could enable an exploited vulnerability to subvert a large number of systems. Instead, introducing diversity, purely for the sake of robustness, could increase short-term costs for training and maintenance. However, having a few diverse nodes could deter total shutdown of the network as long as all the nodes are not part of the same directory service for authentication, and allow those nodes to help with recovery of the infected nodes. Such separate, functional redundancy could avoid the cost of a total shutdown, at the cost of increased complexity and reduced usability in terms of single sign-on authentication.

1.1.6 Anti-malware strategies

Main article: Antivirus software

As malware attacks become more frequent, attention has begun to shift from viruses and spyware protection, to malware protection, and programs that have been specifically developed to combat malware. (Other preventive and recovery measures, such as backup and recovery methods, are mentioned in the computer virus article).

Anti-virus and anti-malware software

A specific component of anti-virus and anti-malware software, commonly referred to as an on-access or real-time scanner, hooks deep into the operating system's core or kernel and functions in a manner similar to how certain malware itself would attempt to operate, though with the user's informed permission for protecting the system. Any time the operating system accesses a file, the on-access scanner checks if the file is a 'legitimate' file or not. If the file is identified as malware by the scanner, the access operation will be stopped, the file will be dealt with by the scanner in a pre-defined way (how the anti-virus program was configured during/post installation), and the user will be notified. This may have a considerable performance impact on the operating system, though the degree of impact is dependent on how well the scanner was programmed. The goal is to stop any operations the malware may attempt on the system before they occur, including activities which might exploit bugs or trigger unexpected operating system behavior.

Anti-malware programs can combat malware in two ways:

1. They can provide real time protection against the installation of malware software on a computer. This type of malware protection works the same way as that of antivirus protection in that the anti-malware software scans all incoming network data for malware and blocks any threats it comes across.

2. Anti-malware software programs can be used solely for detection and removal of malware software that has already been installed onto a computer. This type of anti-malware software scans the contents of the Windows registry, operating system files, and installed programs on a computer and will provide a list of any threats found, allowing the user to choose which files to delete or keep, or to compare this list to a list of known malware components, removing files that match.[58]

Real-time protection from malware works identically to real-time antivirus protection: the software scans disk files at download time, and blocks the activity of components known to represent malware. In some cases, it may also intercept attempts to install start-up items or to modify browser settings. Because many malware components are installed as a result of browser exploits or user error, using security software (some of which are anti-malware, though many are not) to "sandbox" browsers (essentially isolate the browser from the computer and hence any malware induced change) can also be effective in helping to restrict any damage done.

Examples of Microsoft Windows antivirus and anti-malware software include the optional Microsoft Security Essentials[59] (for Windows XP, Vista, and Windows 7) for real-time protection, the Windows Malicious Software Removal Tool[60] (now included with Windows (Security) Updates on "Patch Tuesday", the second Tuesday of each month), and Windows Defender (an optional download in the case of Windows XP, incorporating MSE functionality in the case of Windows 8 and later).[61] Additionally, several capable antivirus software programs are available for free download from the Internet (usually restricted to non-commercial use).[62] Tests found some free programs to be competitive with commercial ones.[62] Microsoft's System File Checker can be used to check for and repair corrupted system files.

Some viruses disable System Restore and other important Windows tools such as Task Manager and Command Prompt. Many such viruses can be removed by rebooting the computer, entering Windows safe mode with networking,[63] and then using system tools or Microsoft Safety Scanner.[64]

Hardware implants can be of any type, so there can be no general way to detect them.

Website security scans

As malware also harms the compromised websites (by breaking reputation, blacklisting in search engines, etc.), some websites offer vulnerability scanning.[65][66][67][68] Such scans check the website, detect malware, may note outdated software, and may report known security issues.

"Air gap" isolation or "Parallel Network"

As a last resort, computers can be protected from malware, and infected computers can be prevented from disseminating trusted information, by imposing an "air gap" (i.e. completely disconnecting them from all other networks). However, malware can still cross the air gap in some situations. For example, removable media can carry malware across the gap. In December 2013 researchers in Germany showed one way that an apparent air gap can be defeated.[69]

"AirHopper",[70] "BitWhisper",[71] "GSMem" [72] and "Fansmitter" [73] are four techniques introduced by researchers that can leak data from air-gapped computers using electromagnetic, thermal and acoustic emissions.

1.1.7 Grayware

See also: Privacy-invasive software and Unwanted software bundling

Grayware is a term applied to unwanted applications or files that are not classified as malware, but can worsen the performance of computers and may cause security risks.[74]

It describes applications that behave in an annoying or undesirable manner, and yet are less serious or troublesome than malware. Grayware encompasses spyware, adware, fraudulent dialers, joke programs, remote access tools and other unwanted programs that harm the performance of computers or cause inconvenience. The term came into use around 2004.[75]

Another term, PUP, which stands for *Potentially Unwanted Program* (or PUA *Potentially Unwanted Application*),[76] refers to applications that would be considered unwanted despite often having been downloaded by the user, possibly after failing to read a download agreement. PUPs include spyware, adware, fraudulent dialers. Many security products classify unauthorised key generators as grayware, although they frequently carry true malware in addition to their ostensible purpose.

Software maker Malwarebytes lists several criteria for classifying a program as a PUP.[77] Some adware (using stolen certificates) disables anti-malware and virus protection; technical remedies are available.[50]

1.1.8 History of viruses and worms

Before Internet access became widespread, viruses spread on personal computers by infecting the executable boot

sectors of floppy disks. By inserting a copy of itself into the machine code instructions in these executables, a virus causes itself to be run whenever a program is run or the disk is booted. Early computer viruses were written for the Apple II and Macintosh, but they became more widespread with the dominance of the IBM PC and MS-DOS system. Executable-infecting viruses are dependent on users exchanging software or boot-able floppies and thumb drives so they spread rapidly in computer hobbyist circles.

The first worms, network-borne infectious programs, originated not on personal computers, but on multitasking Unix systems. The first well-known worm was the Internet Worm of 1988, which infected SunOS and VAX BSD systems. Unlike a virus, this worm did not insert itself into other programs. Instead, it exploited security holes (vulnerabilities) in network server programs and started itself running as a separate process.[78] This same behavior is used by today's worms as well.

With the rise of the Microsoft Windows platform in the 1990s, and the flexible macros of its applications, it became possible to write infectious code in the macro language of Microsoft Word and similar programs. These *macro viruses* infect documents and templates rather than applications (executables), but rely on the fact that macros in a Word document are a form of executable code.

Today, worms are most commonly written for the Windows OS, although a few like Mare-D[79] and the L10n worm[80] are also written for Linux and Unix systems. Worms today work in the same basic way as 1988's Internet Worm: they scan the network and use vulnerable computers to replicate. Because they need no human intervention, worms can spread with incredible speed. The SQL Slammer infected thousands of computers in a few minutes in 2003.[81]

1.1.9 Academic research

Main article: Malware research

The notion of a self-reproducing computer program can be traced back to initial theories about the operation of complex automata.[82] John von Neumann showed that in theory a program could reproduce itself. This constituted a plausibility result in computability theory. Fred Cohen experimented with computer viruses and confirmed Neumann's postulate and investigated other properties of malware such as detectability and self-obfuscation using rudimentary encryption. His doctoral dissertation was on the subject of computer viruses.[83] The combination of cryptographic technology as part of the payload of the virus, exploiting it for attack purposes was initialized and investigated from the mid 1990s, and includes initial ransomware and evasion ideas.[84]

1.1.10 See also

- Browser hijacking
- Command and control (malware)
- Comparison of antivirus software
- Computer security
- Cyber spying
- File binder
- Identity theft
- Industrial espionage
- Malvertising
- Phishing
- Riskware
- Security in Web applications
- Social engineering (security)
- Targeted threat
- Typosquatting
- Category:Web security exploits
- Web server overload causes
- Zombie (computer science)

1.1.11 References

[1] "Malware definition". techterms.com. Retrieved 27 September 2015.

[2] Christopher Elisan (5 September 2012). *Malware, Rootkits & Botnets A Beginner's Guide*. McGraw Hill Professional. pp. 10–. ISBN 978-0-07-179205-9.

[3] Stallings, William (2012). *Computer security : principles and practice*. Boston: Pearson. p. 182. ISBN 978-0-13-277506-9.

[4] "Defining Malware: FAQ". technet.microsoft.com. Retrieved 10 September 2009.

[5] "An Undirected Attack Against Critical Infrastructure" (PDF). United States Computer Emergency Readiness Team(Us-cert.gov). Retrieved 28 September 2014.

[6] "Evolution of Malware-Malware Trends". *Microsoft Security Intelligence Report-Featured Articles*. Microsoft.com. Retrieved 28 April 2013.

[7] "Virus/Contaminant/Destructive Transmission Statutes by State". National Conference of State Legislatures. 2012-02-14. Retrieved 26 August 2013.

[8] "§ 18.2-152.4:1 Penalty for Computer Contamination" (PDF). Joint Commission on Technology and Science. Retrieved 17 September 2010.

[9] Russinovich, Mark (2005-10-31). "Sony, Rootkits and Digital Rights Management Gone Too Far". *Mark's Blog*. Microsoft MSDN. Retrieved 2009-07-29.

[10] "Protect Your Computer from Malware". OnGuardOnline.gov. Retrieved 26 August 2013.

[11] "Malware". FEDERAL TRADE COMMISSION- CONSUMER INFORMATION. Retrieved 27 March 2014.

[12] Hernandez, Pedro. "Microsoft Vows to Combat Government Cyber-Spying". eWeek. Retrieved 15 December 2013.

[13] Kovacs, Eduard. "MiniDuke Malware Used Against European Government Organizations". Softpedia. Retrieved 27 February 2013.

[14] "South Korea network attack 'a computer virus'". *BBC*. Retrieved 20 March 2013.

[15] "Malware Revolution: A Change in Target". March 2007.

[16] "Child Porn: Malware's Ultimate Evil". November 2009.

[17] PC World – Zombie PCs: Silent, Growing Threat.

[18] "Peer To Peer Information". NORTH CAROLINA STATE UNIVERSITY. Retrieved 25 March 2011.

[19] "Another way Microsoft is disrupting the malware ecosystem". Retrieved 18 February 2015.

[20] "Shamoon is latest malware to target energy sector". Retrieved 18 February 2015.

[21] "Computer-killing malware used in Sony attack a wake-up call". Retrieved 18 February 2015.

[22] "Symantec Internet Security Threat Report: Trends for July–December 2007 (Executive Summary)" (PDF) **XIII**. Symantec Corp. April 2008: 29. Retrieved 11 May 2008.

[23] "F-Secure Reports Amount of Malware Grew by 100% during 2007" (Press release). F-Secure Corporation. 4 December 2007. Retrieved 11 December 2007.

[24] "F-Secure Quarterly Security Wrap-up for the first quarter of 2008". F-Secure. 31 March 2008. Retrieved 25 April 2008.

[25] "Continuing Business with Malware Infected Customers". Gunter Ollmann. October 2008.

[26] "New Research Shows Remote Users Expose Companies to Cybercrime". Webroot. April 2013.

[27] "Symantec names Shaoxing, China as world's malware capital". Engadget. Retrieved 15 April 2010.

[28] Rooney, Ben (2011-05-23). "Malware Is Posing Increasing Danger". Wall Street Journal.

[29] Suarez-Tangil, Guillermo; Juan E. Tapiador, Pedro Peris-Lopez, Arturo Ribagorda (2014). "Evolution, Detection and Analysis of Malware in Smart Devices" (PDF). *IEEE Communications Surveys & Tutorials*.

[30] "computer virus – Encyclopedia Britannica". Britannica.com. Retrieved 28 April 2013.

[31] All about Malware and Information Privacy

[32] "What are viruses, worms, and Trojan horses?". *Indiana University*. The Trustees of Indiana University. Retrieved 23 February 2015.

[33] Landwehr, C. E; A. R Bull; J. P McDermott; W. S Choi (1993). *A taxonomy of computer program security flaws, with examples*. DTIC Document. Retrieved 2012-04-05.

[34] "Trojan Horse Definition". Retrieved 2012-04-05.

[35] "Trojan horse". *Webopedia*. Retrieved 2012-04-05.

[36] "What is Trojan horse? – Definition from Whatis.com". Retrieved 2012-04-05.

[37] "Trojan Horse: [coined By MIT-hacker-turned-NSA-spook Dan Edwards] N.". Retrieved 2012-04-05.

[38] "What is the difference between viruses, worms, and Trojans?". Symantec Corporation. Retrieved 2009-01-10.

[39] "VIRUS-L/comp.virus Frequently Asked Questions (FAQ) v2.00 (Question B3: What is a Trojan Horse?)". 9 October 1995. Retrieved 2012-09-13.

[40] McDowell, Mindi. "Understanding Hidden Threats: Rootkits and Botnets". US-CERT. Retrieved 6 February 2013.

[41] "Catb.org". Catb.org. Retrieved 15 April 2010.

[42] Vincentas (11 July 2013). "Malware in SpyWareLoop.com". Spyware Loop. Retrieved 28 July 2013.

[43] Staff, SPIEGEL. "Inside TAO: Documents Reveal Top NSA Hacking Unit". SPIEGEL. Retrieved 23 January 2014.

[44] Edwards, John. "Top Zombie, Trojan Horse and Bot Threats". IT Security. Retrieved 25 September 2007.

[45] Appelbaum, Jacob. "Shopping for Spy Gear:Catalog Advertises NSA Toolbox". SPIEGEL. Retrieved 29 December 2013.

[46] Evasive malware

[47] Kirat, Dhilung; Vigna, Giovanni; Kruegel, Christopher (2014). *Barecloud: bare-metal analysis-based evasive malware detection*. ACM. pp. 287–301. ISBN 978-1-931971-15-7.

[48] The Four Most Common Evasive Techniques Used by Malware. April 27, 2015.

[49] Young, Adam; Yung, Moti (1997). "Deniable Password Snatching: On the Possibility of Evasive Electronic Espionage". *Symp. on Security and Privacy*. IEEE. pp. 224–235. ISBN 0-8186-7828-3.

[50] Casey, Henry T. (25 November 2015). "Latest adware disables antivirus software". *Tom's Guide*. Yahoo.com. Retrieved 25 November 2015.

[51] "Global Web Browser... Security Trends" (PDF). Kaspersky lab. November 2012.

[52] Rashid, Fahmida Y. (27 November 2012). "Updated Browsers Still Vulnerable to Attack if Plugins Are Outdated". pcmag.com.

[53] Danchev, Dancho (18 August 2011). "Kaspersky: 12 different vulnerabilities detected on every PC". pcmag.com.

[54] "Adobe Security bulletins and advisories". Adobe.com. Retrieved 19 January 2013.

[55] Rubenking, Neil J. "Secunia Personal Software Inspector 3.0 Review & Rating". PCMag.com. Retrieved 19 January 2013.

[56] "USB devices spreading viruses". *CNET*. CBS Interactive. Retrieved 18 February 2015.

[57] "LNCS 3786 – Key Factors Influencing Worm Infection", U. Kanlayasiri, 2006, web (PDF): SL40-PDF.

[58] "How Antivirus Software Works?". Retrieved 2015-10-16.

[59] "Microsoft Security Essentials". Microsoft. Retrieved 21 June 2012.

[60] "Malicious Software Removal Tool". Microsoft. Retrieved 21 June 2012.

[61] "Windows Defender". Microsoft. Retrieved 21 June 2012.

[62] Rubenking, Neil J. (8 January 2014). "The Best Free Antivirus for 2014". pcmag.com.

[63] "How do I remove a computer virus?". Microsoft. Retrieved 26 August 2013.

[64] "Microsoft Safety Scanner". Microsoft. Retrieved 26 August 2013.

[65] "An example of a website vulnerability scanner". Unmaskparasites.com. Retrieved 19 January 2013.

[66] "Redleg's File Viewer. Used to check a webpage for malicious redirects or malicious HTML coding". Aw-snap.info. Retrieved 19 January 2013.

[67] "Example Google.com Safe Browsing Diagnostic page". Google.com. Retrieved 19 January 2013.

[68] "Safe Browsing (Google Online Security Blog)". Retrieved 21 June 2012.

[69] Hanspach, Michael; Goetz, Michael (November 2013). "On Covert Acoustical Mesh Networks in Air". *Journal of Communications*. doi:10.12720/jcm.8.11.758-767.

[70] M. Guri, G. Kedma, A. Kachlon and Y. Elovici, "AirHopper: Bridging the air-gap between isolated networks and mobile phones using radio frequencies," *Malicious and Unwanted Software: The Americas (MALWARE), 2014 9th International Conference on*, Fajardo, PR, 2014, pp. 58-67.

[71] M. Guri, M. Monitz, Y. Mirski and Y. Elovici, "BitWhisper: Covert Signaling Channel between Air-Gapped Computers Using Thermal Manipulations," *2015 IEEE 28th Computer Security Foundations Symposium*, Verona, 2015, pp. 276-289.

[72] GSMem: Data Exfiltration from Air-Gapped Computers over GSM Frequencies. Mordechai Guri, Assaf Kachlon, Ofer Hasson, Gabi Kedma, Yisroel Mirsky, and Yuval Elovici, *Ben-Gurion University of the Negev; USENIX Security Symposium 2015*

[73] https://arxiv.org/ftp/arxiv/papers/1606/1606.05915.pdf

[74] Vincentas (11 July 2013). "Grayware in SpyWareLoop.com". Spyware Loop. Retrieved 28 July 2013.

[75] "Threat Encyclopedia – Generic Grayware". Trend Micro. Retrieved 27 November 2012.

[76] "Rating the best anti-malware solutions". Arstechnica. Retrieved 28 January 2014.

[77] "PUP Criteria". malwarebytes.org. Retrieved 13 February 2015.

[78] William A Hendric (4 September 2014). "Computer Virus history". *The Register*. Retrieved 29 March 2015.

[79] Nick Farrell (20 February 2006). "Linux worm targets PHP flaw". *The Register*. Retrieved 19 May 2010.

[80] John Leyden (28 March 2001). "Highly destructive Linux worm mutating". *The Register*. Retrieved 19 May 2010.

[81] "Aggressive net bug makes history". *BBC News*. 3 February 2003. Retrieved 19 May 2010.

[82] John von Neumann, "Theory of Self-Reproducing Automata", Part 1: Transcripts of lectures given at the University of Illinois, December 1949, Editor: A. W. Burks, University of Illinois, USA, 1966.

[83] Fred Cohen, "Computer Viruses", PhD Thesis, University of Southern California, ASP Press, 1988.

[84] Young, Adam; Yung, Moti (2004). *Malicious cryptography - exposing cryptovirology*. Wiley. pp. 1–392. ISBN 978-0-7645-4975-5.

1.1.12 External links

- Malicious Software at DMOZ
- Further Reading: Research Papers and Documents about Malware on IDMARCH (Int. Digital Media Archive)
- Advanced Malware Cleaning – a Microsoft video

Chapter 2

Infectious Malware

2.1 Computer virus

Not to be confused with computer worm or Trojan horse (computing).

Hex dump of the Blaster worm, showing a message left for Microsoft co-founder Bill Gates by the worm's programmer

A **computer virus** is a malware that, when executed, replicates by reproducing itself or infecting other programs by modifying them.[1] Infecting computer programs can include as well, data files, or the boot sector of the hard drive. When this replication succeeds, the affected areas are then said to be "infected".[2][3][4][5] The term computer virus was a misnomer until it was coined by Fred Cohen in 1985.[6] Viruses often perform some type of harmful activity on infected hosts, such as acquisition of hard disk space or CPU time, accessing private information, corrupting data, displaying political or humorous messages on the user's screen, spamming their contacts, logging their keystrokes, or even rendering the computer useless. However, not all viruses carry a destructive payload or attempt to hide themselves—the defining characteristic of viruses is that they are self-replicating computer programs which install themselves without user consent.

Virus writers use social engineering and exploit detailed knowledge of security vulnerabilities to gain access to their hosts' computing resources. The vast majority of viruses target systems running Microsoft Windows,[7][8][9] employing a variety of mechanisms to infect new hosts,[10] and often using complex anti-detection/stealth strategies to evade antivirus software.[11][12][13][14] Motives for creating viruses can include seeking profit, desire to send a political message, personal amusement, to demonstrate that a vulnerability exists in software, for sabotage and denial of service, or simply because they wish to explore artificial life and evolutionary algorithms.[15]

Computer viruses currently cause billions of dollars' worth of economic damage each year,[16] due to causing system failure, wasting computer resources, corrupting data, increasing maintenance costs, etc. In response, free, open-source antivirus tools have been developed, and an industry of antivirus software has cropped up, selling or freely distributing virus protection to users of various operating systems.[17] Even though no currently existing antivirus software is able to uncover all computer viruses (especially new ones), computer security researchers are actively searching for new ways to enable antivirus solutions to more effectively detect emerging viruses, before they have already become widely distributed.[18]

2.1.1 Historical development

See also: Timeline of notable computer viruses and worms

Early academic work on self-replicating programs

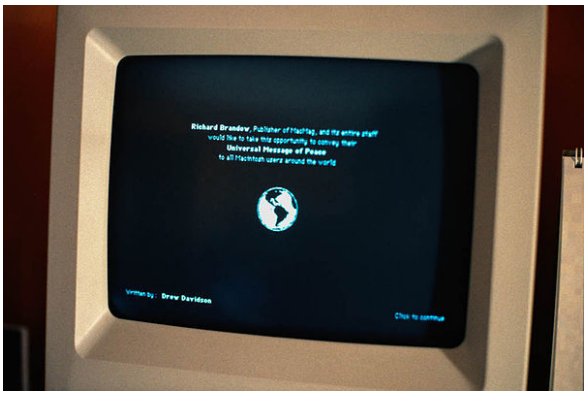
The first academic work on the theory of self-replicating computer programs[19] was done in 1949 by John von Neumann who gave lectures at the University of Illinois about the "Theory and Organization of Complicated Automata". The work of von Neumann was later published as the "Theory of self-reproducing automata". In his essay von Neumann described how a computer program could be designed to reproduce itself.[20] Von Neumann's design for a self-reproducing computer program is considered the world's

first computer virus, and he is considered to be the theoretical father of computer virology.[21]

In 1972 Veith Risak, directly building on von Neumann's work on self-replication, published his article "Selbstreproduzierende Automaten mit minimaler Informationsübertragung" (Self-reproducing automata with minimal information exchange).[22] The article describes a fully functional virus written in assembler language for a SIEMENS 4004/35 computer system.

In 1980 Jürgen Kraus wrote his diplom thesis "Selbstreproduktion bei Programmen" (Self-reproduction of programs) at the University of Dortmund.[23] In his work Kraus postulated that computer programs can behave in a way similar to biological viruses.

The first computer viruses



The MacMag virus 'Universal Peace', as displayed on a Mac in March of 1988

The Creeper virus was first detected on ARPANET, the forerunner of the Internet, in the early 1970s.[24] Creeper was an experimental self-replicating program written by Bob Thomas at BBN Technologies in 1971.[25] Creeper used the ARPANET to infect DEC PDP-10 computers running the TENEX operating system.[26] Creeper gained access via the ARPANET and copied itself to the remote system where the message, "I'm the creeper, catch me if you can!" was displayed. The *Reaper* program was created to delete Creeper.[27]

In fiction, the 1973 Michael Crichton movie *Westworld* made an early mention of the concept of a computer virus, being a central plot theme that causes androids to run amok.[28] Alan Oppenheimer's character summarizes the problem by stating that "...there's a clear pattern here which suggests an analogy to an infectious disease process, spreading from one...area to the next." To which the replies are stated: "Perhaps there are superficial similarities to disease" and, "I must confess I find it difficult to believe in a disease of machinery."[29] (Crichton's earlier work, the 1969 novel *The Andromeda Strain* and 1971 film were about a biological virus-like disease that threatened the human race.)

In 1982, a program called "Elk Cloner" was the first personal computer virus to appear "in the wild"—that is, outside the single computer or lab where it was created.[30] Written in 1981 by Richard Skrenta while in the ninth grade at Mount Lebanon High School near Pittsburgh, it attached itself to the Apple DOS 3.3 operating system and spread via floppy disk.[30][31] This virus, created as a practical joke when Skrenta was still in high school, was injected in a game on a floppy disk. On its 50th use the Elk Cloner virus would be activated, infecting the personal computer and displaying a short poem beginning "Elk Cloner: The program with a personality."

In 1984 Fred Cohen from the University of Southern California wrote his paper "Computer Viruses – Theory and Experiments".[32] It was the first paper to explicitly call a self-reproducing program a "virus", a term introduced by Cohen's mentor Leonard Adleman. In 1987, Fred Cohen published a demonstration that there is no algorithm that can perfectly detect all possible viruses.[33] Fred Cohen's theoretical compression virus[34] was an example of a virus which was not malware, but was putatively benevolent. However, antivirus professionals do not accept the concept of benevolent viruses, as any desired function can be implemented without involving a virus (automatic compression, for instance, is available under the Windows operating system at the choice of the user). Any virus will by definition make unauthorised changes to a computer, which is undesirable even if no damage is done or intended. On page one of *Dr Solomon's Virus Encyclopaedia*, the undesirability of viruses, even those that do nothing but reproduce, is thoroughly explained.[35]

An article that describes "useful virus functionalities" was published by J. B. Gunn under the title "Use of virus functions to provide a virtual APL interpreter under user control" in 1984.[36]

The first IBM PC virus in the wild was a boot sector virus dubbed (c)Brain,[37] created in 1986 by the Farooq Alvi Brothers in Lahore, Pakistan, reportedly to deter unauthorized copying of the software they had written.[38]

The first virus to specifically target Microsoft Windows, WinVir was discovered in April 1992, two years after the release of Windows 3.0.[39] The virus did not contain any Windows API calls, instead relying on DOS interrupts. A few years later, in February 1996, Australian hackers from the virus-writing crew Boza created the VLAD virus, which was the first known virus to target Windows 95. In late 1997 the encrypted, memory-resident stealth virus Win32.Cabanas was released—the first known virus that targeted Windows NT (it was also able to infect Windows

3.0 and Windows 9x hosts).[40]

Even home computers were affected by viruses. The first one to appear on the Commodore Amiga was a boot sector virus called SCA virus, which was detected in November 1987.[41]

The first social networking virus, Win32.5-0-1, Was created by Matt Larose on August 15, 2001.[42] The virus specifically targeted users of MSN Messenger and bulletin boards. Users would be required to click on a link to activate the virus, which would then send an email containing user data to an anonymous email address, which was later found to be owned by Larose. Data sent would contain items such as user IP and email addresses, contacts, site history, and commonly used phrases. In 2008, larger websites used part of the Win32.5-0-1 code to track web users ad related interests.

2.1.2 Operations & functions of a virus

Virus parts

A viable computer virus must contain a search routine, which locates new files or new disks which are worthwhile targets for infection. Secondly, every computer virus must contain a routine to copy itself into the program which the search routine locates.[43] The three main virus parts are:

Infection mechanism Infection mechanism (also called 'infection vector'), is how the virus spreads or propagates, a virus has a search routine, which locates new files or new disks for infection.[44]

Trigger Trigger, which is also known as logic bomb, is the compiled version that could be activated any time an executable with the virus is run that determines the event or condition for the payload to be activated or delivered[45] such as a particular date, a particular time, particular presence of another program, capacity of the disk exceeding some limit,[46] or a double-click that opens a particular file.[47]

Payload The payload is the actual body or data that perform the actual purpose of the virus. Payload activity might be noticeable, as most of the time it is the harmful activity,[44] or some times non-destructive but distributive, which is called Virus hoax.[48]

Virus phases

Virus phases is the life cycle of the computer virus, it can be divided into 4 phases:

Dormant Phase The virus is idle. The virus will eventually be activated by the trigger which states which event will execute the virus, such as a date, the presence of another program or file, or the capacity of the disk exceeding some limit. Not all viruses have this stage.[44]

Propagation Phase The virus starts propagating, that is multiplying itself. The virus places a copy of itself into other programs or into certain system areas on the disk. The copy may not be identical to the propagating version; viruses often morph to evade detection. Each infected program will now contain a clone of the virus, which will itself enter a propagation phase.[44]

Triggering Phase A dormant virus moves into this phase when it is activated, and will now perform the function for which it was intended. The triggering phase can be caused by a variety of system events, including a count of the number of times that this copy of the virus has made copies of itself.[44]

Execution Phase This is the actual work of the virus, where the payload will be released. It can be destructive such as deleting files on disk or harmless such as popping messages on screen.[44]

2.1.3 Infection targets and replication techniques

Computer viruses infect a variety of different subsystems on their hosts.[49] One manner of classifying viruses is to analyze whether they reside in binary executables (such as .EXE or .COM files), data files (such as Microsoft Word documents or PDF files), or in the boot sector of the host's hard drive (or some combination of all of these).[50][51]

Resident vs. non-resident viruses

A *memory-resident virus* (or simply "resident virus") installs itself as part of the operating system when executed, after which it remains in RAM from the time the computer is booted up to when it is shut down. Resident viruses overwrite interrupt handling code or other functions, and when the operating system attempts to access the target file or

disk sector, the virus code intercepts the request and redirects the control flow to the replication module, infecting the target. In contrast, a *non-memory-resident virus* (or "non-resident virus"), when executed, scans the disk for targets, infects them, and then exits (i.e. it does not remain in memory after it is done executing).[52][53][54]

Macro viruses

Many common applications, such as Microsoft Outlook and Microsoft Word, allow macro programs to be embedded in documents or emails, so that the programs may be run automatically when the document is opened. A *macro virus* (or "document virus") is a virus that is written in a macro language, and embedded into these documents so that when users open the file, the virus code is executed, and can infect the user's computer. This is one of the reasons that it is dangerous to open unexpected attachments in e-mails.[55][56]

Boot sector viruses

Boot sector viruses specifically target the boot sector/Master Boot Record[57] (MBR) of the host's hard drive or removable storage media (flash drives, floppy disks, etc.).[50][58][59]

2.1.4 Stealth strategies

In order to avoid detection by users, some viruses employ different kinds of deception. Some old viruses, especially on the MS-DOS platform, make sure that the "last modified" date of a host file stays the same when the file is infected by the virus. This approach does not fool antivirus software, however, especially those which maintain and date cyclic redundancy checks on file changes.[60]

Some viruses can infect files without increasing their sizes or damaging the files. They accomplish this by overwriting unused areas of executable files. These are called *cavity viruses*. For example, the CIH virus, or Chernobyl Virus, infects Portable Executable files. Because those files have many empty gaps, the virus, which was 1 KB in length, did not add to the size of the file.[61]

Some viruses try to avoid detection by killing the tasks associated with antivirus software before it can detect them (for example, Conficker).

As computers and operating systems grow larger and more complex, old hiding techniques need to be updated or replaced. Defending a computer against viruses may demand that a file system migrate towards detailed and explicit permission for every kind of file access.[62]

Read request intercepts

While some antivirus software employ various techniques to counter stealth mechanisms, once the infection occurs any recourse to clean the system is unreliable. In Microsoft Windows operating systems, the NTFS file system is proprietary. This leaves antivirus software little alternative but to send a read request to Windows OS files that handle such requests. Some viruses trick antivirus software by intercepting its requests to the OS. A virus can hide itself by intercepting the request to read the infected file, handling the request itself, and return an uninfected version of the file to the antivirus software. The interception can occur by code injection of the actual operating system files that would handle the read request. Thus, an antivirus software attempting to detect the virus will either not be given permission to read the infected file, or, the read request will be served with the uninfected version of the same file.[63]

The only reliable method to avoid stealth is to boot from a medium that is known to be clean. Security software can then be used to check the dormant operating system files. Most security software relies on virus signatures, or they employ heuristics.[64][65]

Security software may also use a database of file hashes for Windows OS files, so the security software can identify altered files, and request Windows installation media to replace them with authentic versions. In older versions of Windows, file hashes of Windows OS files stored in Windows—to allow file integrity/authenticity to be checked—could be overwritten so that the System File Checker would report that altered system files are authentic, so using file hashes to scan for altered files would not always guarantee finding an infection.[66]

Self-modification

See also: Self-modifying code

Most modern antivirus programs try to find virus-patterns inside ordinary programs by scanning them for so-called *virus signatures*. Unfortunately, the term is misleading, in that viruses do not possess unique signatures in the way that human beings do. Such a virus signature is merely a sequence of bytes that an antivirus program looks for because it is known to be part of the virus. A better term would be "search strings". Different antivirus programs will employ different search strings, and indeed different search methods, when identifying viruses. If a virus scanner finds such a pattern in a file, it will perform other checks to make sure that it has found the virus, and not merely a coincidental sequence in an innocent file, before it notifies the user that the file is infected. The user can then delete, or (in some

cases) “clean” or “heal” the infected file. Some viruses employ techniques that make detection by means of signatures difficult but probably not impossible. These viruses modify their code on each infection. That is, each infected file contains a different variant of the virus.

Encrypted viruses One method of evading signature detection is to use simple encryption to encipher the body of the virus, leaving only the encryption module and a static cryptographic key in cleartext which doesn't change from one infection to the next.[67] In this case, the virus consists of a small decrypting module and an encrypted copy of the virus code. If the virus is encrypted with a different key for each infected file, the only part of the virus that remains constant is the decrypting module, which would (for example) be appended to the end. In this case, a virus scanner cannot directly detect the virus using signatures, but it can still detect the decrypting module, which still makes indirect detection of the virus possible. Since these would be symmetric keys, stored on the infected host, it is in fact entirely possible to decrypt the final virus, but this is probably not required, since self-modifying code is such a rarity that it may be reason for virus scanners to at least flag the file as suspicious.[68] An old but compact way will be the use of arithmetic operation like addition or subtraction and the use of logical conditions such as XORing,[69] where each byte in a virus is with a constant, so that the exclusive-or operation had only to be repeated for decryption. It is suspicious for a code to modify itself, so the code to do the encryption/decryption may be part of the signature in many virus definitions.[68] An simpler older approach did not use a key, where the encryption consisted only of operations with no parameters, like incrementing and decrementing, bitwise rotation,arithmetic negation, and logical NOT.[69]

Some viruses will employ a means of encryption inside an executable in which the virus is encrypted under certain events, such as the virus scanner being disabled for updates or the computer being rebooted. This is called Cryptovirology. At said times, the executable will decrypt the virus and execute its hidden runtimes infecting the computer and sometimes disabling the antivirus software.

Polymorphic code Polymorphic code was the first technique that posed a serious threat to virus scanners. Just like regular encrypted viruses, a polymorphic virus infects files with an encrypted copy of itself, which is decoded by a decryption module. In the case of polymorphic viruses, however, this decryption module is also modified on each infection. A well-written polymorphic virus therefore has no parts which remain identical between infections, making it very difficult to detect directly using signatures.[70][71] Antivirus software can detect it by decrypting the viruses using an emulator, or by statistical pattern analysis of the encrypted virus body. To enable polymorphic code, the virus has to have a polymorphic engine (also called mutating engine or mutation engine) somewhere in its encrypted body. See polymorphic code for technical detail on how such engines operate.[72]

Some viruses employ polymorphic code in a way that constrains the mutation rate of the virus significantly. For example, a virus can be programmed to mutate only slightly over time, or it can be programmed to refrain from mutating when it infects a file on a computer that already contains copies of the virus. The advantage of using such slow polymorphic code is that it makes it more difficult for antivirus professionals to obtain representative samples of the virus, because bait files that are infected in one run will typically contain identical or similar samples of the virus. This will make it more likely that the detection by the virus scanner will be unreliable, and that some instances of the virus may be able to avoid detection.

There has also been virus called undetectable virus (proposed in Yongge Wang [73]). Undetectable virus is one kind of polymorphic virus that is static signature-free and whose dynamic signatures are hard to determine unless some cryptographic assumption fails.

Metamorphic code To avoid being detected by emulation, some viruses rewrite themselves completely each time they are to infect new executables. Viruses that utilize this technique are said to be metamorphic. To enable metamorphism, a metamorphic engine is needed. A metamorphic virus is usually very large and complex. For example, W32/Simile consisted of over 14,000 lines of assembly language code, 90% of which is part of the metamorphic engine.[74][75]

2.1.5 Vulnerabilities and infection vectors

Software bugs

Because software is often designed with security features to prevent unauthorized use of system resources, many viruses must exploit and manipulate security bugs (security defects) in a system or application software to spread and infect. Software development strategies that produce large numbers of bugs will generally also produce potential exploits.

Social engineering and poor security practices

In order to replicate itself, a virus must be permitted to execute code and write to memory. For this reason, many viruses attach themselves to executable files that may be

part of legitimate programs (see code injection). If a user attempts to launch an infected program, the virus' code may be executed simultaneously.[76]

In operating systems that use file extensions to determine program associations (such as Microsoft Windows), the extensions may be hidden from the user by default. This makes it possible to create a file that is of a different type than it appears to the user. For example, an executable may be created and named "picture.png.exe", in which the user sees only "picture.png" and therefore assumes that this file is an image and most likely is safe, yet when opened run the executable on the client machine.[77]

Vulnerability of different operating systems to viruses

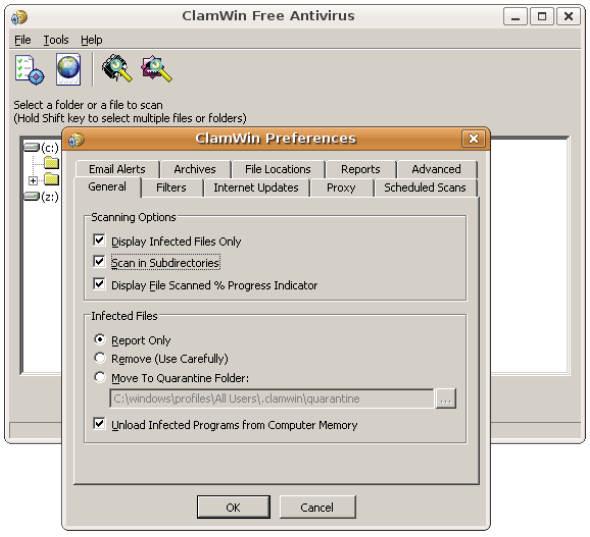
The vast majority of viruses target systems running Microsoft Windows. This is due to Microsoft's large market share of desktop users.[78] The diversity of software systems on a network limits the destructive potential of viruses and malware.[79] Open-source operating systems such as Linux allow users to choose from a variety of desktop environments, packaging tools, etc., which means that malicious code targeting any of these systems will only affect a subset of all users. Many Windows users are running the same set of applications, enabling viruses to rapidly spread among Microsoft Windows systems by targeting the same exploits on large numbers of hosts.[7][8][9][80]

While Linux and Unix in general have always natively prevented normal users from making changes to the operating system environment without permission, Windows users are generally not prevented from making these changes, meaning that viruses can easily gain control of the entire system on Windows hosts. This difference has continued partly due to the widespread use of administrator accounts in contemporary versions like XP. In 1997, researchers created and released a virus for Linux—known as "Bliss".[81] Bliss, however, requires that the user run it explicitly, and it can only infect programs that the user has the access to modify. Unlike Windows users, most Unix users do not log in as an administrator, or root user, except to install or configure software; as a result, even if a user ran the virus, it could not harm their operating system. The Bliss virus never became widespread, and remains chiefly a research curiosity. Its creator later posted the source code to Usenet, allowing researchers to see how it worked.[82]

2.1.6 Countermeasures

See also: Vulnerability to malware, Anti-malware strategies, and Browser hardening

Antivirus software



Screenshot of the open source ClamWin antivirus software running in Wine on Ubuntu Linux

Many users install antivirus software that can detect and eliminate known viruses when the computer attempts to download or run the executable (which may be distributed as an email attachment, or on USB flash drives, for example). Some antivirus software blocks known malicious web sites that attempt to install malware. Antivirus software does not change the underlying capability of hosts to transmit viruses. Users must update their software regularly to patch security vulnerabilities ("holes"). Antivirus software also needs to be regularly updated in order to recognize the latest threats. The German AV-TEST Institute publishes evaluations of antivirus software for Windows[83] and Android.[84]

Examples of Microsoft Windows anti virus and anti-malware software include the optional Microsoft Security Essentials[85] (for Windows XP, Vista and Windows 7) for real-time protection, the Windows Malicious Software Removal Tool[86] (now included with Windows (Security) Updates on "Patch Tuesday", the second Tuesday of each month), and Windows Defender (an optional download in the case of Windows XP).[87] Additionally, several capable antivirus software programs are available for free download from the Internet (usually restricted to non-commercial use).[88] Some such free programs are almost as good as commercial competitors.[89] Common security vulnerabilities are assigned CVE IDs and listed in the US National Vulnerability Database. Secunia PSI[90] is an example of software, free for personal use, that will check a PC for vulnerable out-of-date software, and attempt to update it. Ransomware and phishing scam alerts appear as press releases on the Internet Crime Complaint Center noticeboard.

Other commonly used preventative measures include timely operating system updates, software updates, careful Internet browsing, and installation of only trusted software.[91] Certain browsers flag sites that have been reported to Google and that have been confirmed as hosting malware by Google.[92][93]

There are two common methods that an antivirus software application uses to detect viruses, as described in the antivirus software article. The first, and by far the most common method of virus detection is using a list of virus signature definitions. This works by examining the content of the computer's memory (its RAM, and boot sectors) and the files stored on fixed or removable drives (hard drives, floppy drives, or USB flash drives), and comparing those files against a database of known virus "signatures". Virus signatures are just strings of code that are used to identify individual viruses; for each virus, the antivirus designer tries to choose a unique signature string that will not be found in a legitimate program. Different antivirus programs use different "signatures" to identify viruses. The disadvantage of this detection method is that users are only protected from viruses that are detected by signatures in their most recent virus definition update, and not protected from new viruses (see "zero-day attack").[94]

A second method to find viruses is to use a heuristic algorithm based on common virus behaviors. This method has the ability to detect new viruses for which antivirus security firms have yet to define a "signature", but it also gives rise to more false positives than using signatures. False positives can be disruptive, especially in a commercial environment.

Recovery strategies and methods

One may reduce the damage done by viruses by making regular backups of data (and the operating systems) on different media, that are either kept unconnected to the system (most of the time), read-only or not accessible for other reasons, such as using different file systems. This way, if data is lost through a virus, one can start again using the backup (which will hopefully be recent).[95]

If a backup session on optical media like CD and DVD is closed, it becomes read-only and can no longer be affected by a virus (so long as a virus or infected file was not copied onto the CD/DVD). Likewise, an operating system on a bootable CD can be used to start the computer if the installed operating systems become unusable. Backups on removable media must be carefully inspected before restoration. The Gammima virus, for example, propagates via removable flash drives.[96][97]

Virus removal Many websites run by antivirus software companies provide free online virus scanning, with limited cleaning facilities (the purpose of the sites is to sell antivirus products). Some websites—like Google subsidiary VirusTotal.com—allow users to upload one or more suspicious files to be scanned and checked by one or more antivirus programs in one operation.[98][99] Additionally, several capable antivirus software programs are available for free download from the Internet (usually restricted to non-commercial use).[100] Microsoft offers an optional free antivirus utility called Microsoft Security Essentials, a Windows Malicious Software Removal Tool that is updated as part of the regular Windows update regime, and an older optional anti-malware (malware removal) tool Windows Defender that has been upgraded to an antivirus product in Windows 8.

Some viruses disable System Restore and other important Windows tools such as Task Manager and CMD. An example of a virus that does this is CiaDoor. Many such viruses can be removed by rebooting the computer, entering Windows safe mode with networking, and then using system tools or Microsoft Safety Scanner.[101] System Restore on Windows Me, Windows XP, Windows Vista and Windows 7 can restore the registry and critical system files to a previous checkpoint. Often a virus will cause a system to hang, and a subsequent hard reboot will render a system restore point from the same day corrupt. Restore points from previous days should work provided the virus is not designed to corrupt the restore files and does not exist in previous restore points.[102][103]

Operating system reinstallation Microsoft's System File Checker (improved in Windows 7 and later) can be used to check for, and repair, corrupted system files.[104]

Restoring an earlier "clean" (virus-free) copy of the entire partition from a cloned disk, a disk image, or a backup copy is one solution—restoring an earlier backup disk image is relatively simple to do, usually removes any malware, and may be faster than disinfecting the computer—or reinstalling and reconfiguring the operating system and programs from scratch, as described below, then restoring user preferences.[95]

Reinstalling the operating system is another approach to virus removal. It may be possible to recover copies of essential user data by booting from a live CD, or connecting the hard drive to another computer and booting from the second computer's operating system, taking great care not to infect that computer by executing any infected programs on the original drive. The original hard drive can then be reformatted and the OS and all programs installed from original media. Once the system has been restored, precautions must be taken to avoid reinfection from any restored executable files.[105]

Viruses and the Internet

See also: Computer worm

Before computer networks became widespread, most viruses spread on removable media, particularly floppy disks. In the early days of the personal computer, many users regularly exchanged information and programs on floppies. Some viruses spread by infecting programs stored on these disks, while others installed themselves into the disk boot sector, ensuring that they would be run when the user booted the computer from the disk, usually inadvertently. Personal computers of the era would attempt to boot first from a floppy if one had been left in the drive. Until floppy disks fell out of use, this was the most successful infection strategy and boot sector viruses were the most common in the wild for many years.

Traditional computer viruses emerged in the 1980s, driven by the spread of personal computers and the resultant increase in bulletin board system (BBS), modem use, and software sharing. Bulletin board–driven software sharing contributed directly to the spread of Trojan horse programs, and viruses were written to infect popularly traded software. Shareware and bootleg software were equally common vectors for viruses on BBSs.[106][107][108] Viruses can increase their chances of spreading to other computers by infecting files on a network file system or a file system that is accessed by other computers.[109]

Macro viruses have become common since the mid-1990s. Most of these viruses are written in the scripting languages for Microsoft programs such as Word and Excel and spread throughout Microsoft Office by infecting documents and spreadsheets. Since Word and Excel were also available for Mac OS, most could also spread to Macintosh computers. Although most of these viruses did not have the ability to send infected email messages, those viruses which did take advantage of the Microsoft Outlook COM interface.[110][111]

Some old versions of Microsoft Word allow macros to replicate themselves with additional blank lines. If two macro viruses simultaneously infect a document, the combination of the two, if also self-replicating, can appear as a "mating" of the two and would likely be detected as a virus unique from the "parents".[112]

A virus may also send a web address link as an instant message to all the contacts on an infected machine. If the recipient, thinking the link is from a friend (a trusted source) follows the link to the website, the virus hosted at the site may be able to infect this new computer and continue propagating.[113]

Viruses that spread using cross-site scripting were first reported in 2002,[114] and were academically demonstrated in 2005.[115] There have been multiple instances of the cross-site scripting viruses in the wild, exploiting websites such as MySpace and Yahoo!.

2.1.7 See also

- Comparison of computer viruses
- Botnet
- Computer insecurity
- Crimeware
- Cryptovirology
- Keystroke logging
- Multipartite virus
- Spam (electronic)
- Trojan horse (computing)
- Virus hoax
- Windows 7 File Recovery
- Windows Action Center (Security Center)
- Command and control (malware)
- Zombie (computer science)
- Malware

2.1.8 References

[1] Stallings, William (2012). *Computer security : principles and practice*. Boston: Pearson. p. 182. ISBN 978-0-13-277506-9.

[2] Aycock, John (2006). *Computer Viruses and Malware*. Springer. p. 14. ISBN 978-0-387-30236-2.

[3] http://vx.netlux.org/lib/aas10.html

[4] "Alan Solomon 'All About Viruses' (VX heavens)". Web.archive.org. 2011-06-14. Archived from the original on January 17, 2012. Retrieved 2014-07-17.

[5] The term "virus" is also commonly, but erroneously, used to refer to other types of malware. "Malware" encompasses computer viruses along with many other forms of malicious software, such as computer worms, ransomware, trojan horses, keyloggers, rootkits, spyware, adware, malicious BHOs and other malicious software. The majority of active malware threats are actually trojans or worms rather than viruses.

[6] Ludwig, Mark (1998). *The giant black book of computer viruses*. Show Low, Ariz: American Eagle. p. 13. ISBN 978-0-929408-23-1.

[7] Mookhey, K.K. et al. (2005). *Linux: Security, Audit and Control Features*. ISACA. p. 128. ISBN 9781893209787.

[8] Toxen, Bob (2003). *Real World Linux Security: Intrusion Prevention, Detection, and Recovery*. Prentice Hall Professional. p. 365. ISBN 9780130464569.

[9] Noyes, Katherine (Aug 3, 2010). "Why Linux Is More Secure Than Windows". *PCWorld*.

[10] Skoudis, Edward (2004). "Infection mechanisms and targets". *Malware: Fighting Malicious Code*. Prentice Hall Professional. pp. 31–48. ISBN 9780131014053.

[11] Aycock, John (2006). *Computer Viruses and Malware*. Springer. p. 27. ISBN 978-0-387-30236-2.

[12] Ludwig, Mark A. (1996). *The Little Black Book of Computer Viruses: Volume 1, The Basic Technologies*. pp. 16–17. ISBN 0-929408-02-0.

[13] Harley, David et al. (2001). *Viruses Revealed*. McGraw-Hill. p. 6. ISBN 0-07-222818-0.

[14] Filiol, Eric (2005). *Computer viruses:from theory to applications*. Springer. p. 8. ISBN 978-2-287-23939-7.

[15] Bell, David J. et al, eds. (2004). "Virus". *Cyberculture: The Key Concepts*. Routledge. p. 154. ISBN 9780203647059.

[16] "Viruses that can cost you".

[17] Granneman, Scott. "Linux vs. Windows Viruses". The Register. Retrieved September 4, 2015.

[18] Kaspersky, Eugene (November 21, 2005). "The contemporary antivirus industry and its problems". SecureLight.

[19] The term "computer virus" was not used at that time.

[20] von Neumann, John (1966). "Theory of Self-Reproducing Automata" (PDF). *Essays on Cellular Automata* (University of Illinois Press): 66–87. Retrieved June 10, 2010.

[21] Éric Filiol, *Computer viruses: from theory to applications, Volume 1*, Birkhäuser, 2005, pp. 19–38 ISBN 2-287-23939-1.

[22] Risak, Veith (1972), "Selbstreproduzierende Automaten mit minimaler Informationsübertragung", *Zeitschrift für Maschinenbau und Elektrotechnik*

[23] Kraus, Jürgen (February 1980), *Selbstreproduktion bei Programmen* (PDF)

[24] "Virus list". Retrieved 2008-02-07.

[25] Thomas Chen, Jean-Marc Robert (2004). "The Evolution of Viruses and Worms". Retrieved 2009-02-16.

[26] Parikka, Jussi (2007). *Digital Contagions: A Media Archaeology of Computer Viruses*. New York: Peter Lang. p. 50. ISBN 978-0-8204-8837-0.

[27] Russell, Deborah & Gangemi, G.T. (1991). *Computer Security Basics*. O'Reilly. p. 86. ISBN 0-937175-71-4.

[28] http://www.imdb.com/title/tt0070909/synopsis: IMDB synopsis of Westworld. Retrieved November 28, 2015.

[29] Michael Crichton (November 21, 1973). *Westworld* (movie). 201 S. Kinney Road, Tucson, Arizona, USA: Metro-Goldwyn-Mayer. Event occurs at 32 minutes. And there's a clear pattern here which suggests an analogy to an infectious disease process, spreading from one resort area to the next." ... "Perhaps there are superficial similarities to disease." "I must confess I find it difficult to belief in a disease of machinery.

[30] Anick Jesdanun (1 September 2007). "School prank starts 25 years of security woes". CNBC. Retrieved April 12, 2013.

[31] "The anniversary of a nuisance".

[32] Cohen, Fred (1984), *Computer Viruses – Theory and Experiments*

[33] Cohen, Fred, An Undetectable Computer Virus, 1987, IBM

[34] Burger, Ralph, 1991. *Computer Viruses and Data Protection*, pp. 19–20

[35] *Dr. Solomon's Virus Encyclopedia*, 1995. ISBN 1-897661-00-2. Abstract. Archived August 4, 2008, at the Wayback Machine.

[36] Gunn, J.B. (June 1984). "Use of virus functions to provide a virtual APL interpreter under user control". *ACM SIGAPL APL Quote Quad archive* (ACM New York, NY, USA) **14** (4): 163–168. doi:10.1145/384283.801093. ISSN 0163-6006.

[37] "Boot sector virus repair". Antivirus.about.com. 2010-06-10. Retrieved 2010-08-27.

[38] "Amjad Farooq Alvi Inventor of first PC Virus post by Zagham". YouTube. Retrieved 2010-08-27.

[39] "winvir virus". Retrieved 10 June 2016.

[40] Grimes, Roger (2001). *Malicious Mobile Code: Virus Protection for Windows*. O'Reilly. pp. 99–100. ISBN 9781565926820.

[41] "SCA virus". Virus Test Center, University of Hamburg. 1990-06-05. Retrieved 2014-01-14.

[42] http://5-0-1.webs.com

[43] Ludwig, Mark (1998). *The giant black book of computer viruses*. Show Low, Ariz: American Eagle. p. 15. ISBN 978-0-929408-23-1.

[44] Stallings, William (2012). *Computer security : principles and practice*. Boston: Pearson. p. 183. ISBN 978-0-13-277506-9.

[45] Ludwig, Mark (1998). *The giant black book of computer viruses*. Show Low, Ariz: American Eagle. p. 292. ISBN 978-0-929408-23-1.

[46] "www.cs.colostate.edu" (PDF). Retrieved 2016-04-25.

[47] Gregory, Peter (2004). *Computer viruses for dummies* (in Danish). Hoboken, NJ: Wiley Pub. p. 210. ISBN 0-7645-7418-3.

[48] Szor, Peter (2005). *The art of computer virus research and defense*. Upper Saddle River, NJ: Addison-Wesley. p. 43. ISBN 0-321-30454-3.

[49] Serazzi, Giuseppe & Zanero, Stefano (2004). "Computer Virus Propagation Models". In Calzarossa, Maria Carla & Gelenbe, Erol. *Performance Tools and Applications to Networked Systems* (PDF). Lecture Notes in Computer Science. Vol. 2965. pp. 26–50.

[50] Avoine, Gildas et al. (2007). *Computer System Security: Basic Concepts and Solved Exercises*. EPFL Press / CRC Press. pp. 21–22. ISBN 9781420046205.

[51] Brain, Marshall; Fenton, Wesley. "How Computer Viruses Work". HowStuffWorks.com. Retrieved 16 June 2013.

[52] Grimes, Roger (2001). *Malicious Mobile Code: Virus Protection for Windows*. O'Reilly. pp. 37–38. ISBN 9781565926820.

[53] Salomon, David (2006). *Foundations of Computer Security*. Springer. pp. 47–48. ISBN 9781846283413.

[54] Polk, William T. (1995). *Antivirus Tools and Techniques for Computer Systems*. William Andrew (Elsevier). p. 4. ISBN 9780815513643.

[55] Grimes, Roger (2001). "Macro Viruses". *Malicious Mobile Code: Virus Protection for Windows*. O'Reilly. ISBN 9781565926820.

[56] Aycock, John (2006). *Computer Viruses and Malware*. Springer. p. 89. ISBN 9780387341880.

[57] "What is boot sector virus?". Retrieved 2015-10-16.

[58] Anonymous (2003). *Maximum Security*. Sams Publishing. pp. 331–333. ISBN 9780672324598.

[59] Skoudis, Edward (2004). "Infection mechanisms and targets". *Malware: Fighting Malicious Code*. Prentice Hall Professional. pp. 37–38. ISBN 9780131014053.

[60] editor-in-chief, Béla G. Lipták, (2002). *Instrument engineers' handbook*. (3rd ed.). Boca Raton: CRC Press. p. 874. ISBN 9781439863442. Retrieved September 4, 2015.

[61] "Computer Virus Strategies and Detection Methods" (PDF). Retrieved 2 September 2008.

[62] *Internet Communication*. PediaPress. pp. 163–. GGKEY: Y43AS5T4TFD. Retrieved 16 April 2016.

[63] Szor, Peter (2005). *The Art of Computer Virus Research and Defense*. Boston: Addison-Wesley. p. 285. ISBN 0-321-30454-3.

[64] Fox-Brewster, Thomas. "Netflix Is Dumping Anti-Virus, Presages Death Of An Industry". Forbes. Retrieved September 4, 2015.

[65] "How Anti-Virus Software Works". Stanford University. Retrieved September 4, 2015.

[66] "www.sans.org". Retrieved 2016-04-16.

[67] Bishop, Matt (2003). *Computer Security: Art and Science*. Addison-Wesley Professional. p. 620. ISBN 9780201440997.

[68] *Internet Communication*. PediaPress. pp. 165–. GGKEY: Y43AS5T4TFD.

[69] John Aycock (19 September 2006). *Computer Viruses and Malware*. Springer. pp. 35–36. ISBN 978-0-387-34188-0.

[70] Kizza, Joseph M. (2009). *Guide to Computer Network Security*. Springer. p. 341. ISBN 9781848009165.

[71] Eilam, Eldad (2011). *Reversing: Secrets of Reverse Engineering*. John Wiley & Sons. p. 216. ISBN 9781118079768.

[72] "Virus Bulletin : Glossary – Polymorphic virus". Virusbtn.com. 2009-10-01. Retrieved 2010-08-27.

[73] Yongge Wang (2000), "Using Mobile Agent Results to Create Hard-to-Detect Computer Viruses" (PDF), *Information Security for Global Information Infrastructures IFIP — The International Federation for Information Processing* (Springer Verlag): 161–17–, retrieved 2015-03-10

[74] Perriot, Fredrick; Peter Ferrie; Peter Szor (May 2002). "Striking Similarities" (PDF). Retrieved September 9, 2007.

[75] "Virus Bulletin : Glossary — Metamorphic virus". Virusbtn.com. Retrieved 2010-08-27.

[76] "Virus Basics". US-CERT.

[77] "Virus Notice: Network Associates' AVERT Discovers First Virus That Can Infect JPEG Files, Assigns Low-Profiled Risk". Retrieved 2002-06-13.

[78] "Operating system market share". *netmarketshare.com*. Retrieved 2015-05-16.

[79] This is analogous to how genetic diversity in a population decreases the chance of a single disease wiping out a population

[80] Raggi, Emilio et al. (2011). *Beginning Ubuntu Linux*. Apress. p. 148. ISBN 9781430236276.

[81] "McAfee discovers first Linux virus" (Press release). McAfee, via Axel Boldt. 5 February 1997.

[82] Boldt, Axel (19 January 2000). "Bliss, a Linux 'virus'".

[83] "Detailed test reports—(Windows) home user". AV-Test.org.

[84] "Detailed test reports — Android mobile devices". AV-Test.org.

[85] "Microsoft Security Essentials". Retrieved June 21, 2012.

[86] "Malicious Software Removal Tool". Retrieved June 21, 2012.

[87] "Windows Defender". Retrieved June 21, 2012.

[88] Rubenking, Neil J. (Feb 17, 2012). "The Best Free Antivirus for 2012". pcmag.com.

[89] Rubenking, Neil J. (Jan 10, 2013). "The Best Antivirus for 2013". pcmag.com.

[90] Rubenking, Neil J. "Secunia Personal Software Inspector 3.0 Review & Rating". PCMag.com. Retrieved 2013-01-19.

[91] "10 Step Guide to Protect Against Viruses". GrnLight.net. Retrieved 23 May 2014.

[92] "Google Safe Browsing".

[93] "Report malicious software (URL) to Google".

[94] Zhang, Yu et al. (2008). "A Novel Immune Based Approach For Detection of Windows PE Virus". In Tang, Changjie et al. *Advanced Data Mining and Applications: 4th International Conference, ADMA 2008, Chengdu, China, October 8-10, 2008, Proceedings*. Springer. p. 250. ISBN 9783540881919.

[95] "Good Security Habits | US-CERT". Retrieved 2016-04-16.

[96] "W32.Gammima.AG". Symantec. Retrieved 2014-07-17.

[97] Category: Computer Articles. "Viruses! In! Space!". GrnLight.net. Retrieved 2014-07-17.

[98] "VirusTotal.com (a subsidiary of Google)".

[99] "VirScan.org".

[100] Rubenking, Neil J. "The Best Free Antivirus for 2014". pcmag.com.

[101] "Microsoft Safety Scanner".

[102] "Virus removal -Help". Retrieved 2015-01-31.

[103] "W32.Gammima.AG Removal — Removing Help". Symantec. 2007-08-27. Retrieved 2014-07-17.

[104] "support.microsoft.com". Retrieved 2016-04-16.

[105] "www.us-cert.gov" (PDF). Retrieved 2016-04-16.

[106] David Kim; Michael G. Solomon (17 November 2010). *Fundamentals of Information Systems Security*. Jones & Bartlett Publishers. pp. 360–. ISBN 978-1-4496-7164-8.

[107] "1980s – Securelist – Information about Viruses, Hackers and Spam". Retrieved 2016-04-16.

[108] *Internet Communication*. PediaPress. pp. 160–. GGKEY: Y43AS5T4TFD.

[109] "What is a Computer Virus?". Actlab.utexas.edu. 1996-03-31. Retrieved 2010-08-27.

[110] Realtimepublishers.com (1 January 2005). *The Definitive Guide to Controlling Malware, Spyware, Phishing, and Spam*. Realtimepublishers.com. pp. 48–. ISBN 978-1-931491-44-0.

[111] Eli B. Cohen (2011). *Navigating Information Challenges*. Informing Science. pp. 27–. ISBN 978-1-932886-47-4.

[112] Vesselin Bontchev. "Macro Virus Identification Problems". *FRISK Software International*.

[113] "Facebook 'photo virus' spreads via email.". Retrieved 2014-04-28.

[114] Berend-Jan Wever. "XSS bug in hotmail login page". Retrieved 2014-04-07.

[115] Wade Alcorn. "The Cross-site Scripting Virus". bindshell.net. Retrieved 2015-10-13.

2.1.9 Further reading

- Burger, Ralf (16 February 2010) [1991]. *Computer Viruses and Data Protection*. Abacus. p. 353. ISBN 978-1-55755-123-8.
- Granneman, Scott (6 October 2003). "Linux vs. Windows Viruses". *The Register*.
- Ludwig, Mark (1993). *Computer Viruses, Artificial Life and Evolution*. Tucson, Arizona 85717: American Eagle Publications, Inc. ISBN 0-929408-07-1. Archived from the original on July 4, 2008.
- Mark Russinovich (November 2006). *Advanced Malware Cleaning **video*** (Web (WMV / MP4)). Microsoft Corporation. Retrieved 24 July 2011.
- Parikka, Jussi (2007). *Digital Contagions. A Media Archaeology of Computer Viruses*. Digital Formations. New York: Peter Lang. ISBN 978-0-8204-8837-0.

2.1.10 External links

- Viruses at DMOZ (DMOZ)
- Microsoft Security Portal
- US Govt CERT (Computer Emergency Readiness Team) site
- 'Computer Viruses – Theory and Experiments' – The original paper by Fred Cohen, 1984
- Hacking Away at the Counterculture by Andrew Ross (On hacking, 1990)
- VX Heaven - the biggest library computer viruses

2.2 Comparison of computer viruses

Not to be confused with Timeline of computer viruses and worms.

The compilation of a unified **list of computer viruses** is made difficult because of naming. To aid the fight against computer viruses and other types of malicious software, many security advisory organizations and developers of anti-virus software compile and publish lists of viruses. When a new virus appears, the rush begins to identify and understand it as well as develop appropriate countermeasures to stop its propagation. Along the way, a name is attached to the virus. As the developers of anti-virus software compete partly based on how quickly they react to the new threat, they usually study and name the viruses independently. By the time the virus is identified, many names denote the same virus.

Another source of ambiguity in names is that sometimes a virus initially identified as a completely new virus is found to be a variation of an earlier known virus, in which cases, it is often renamed. For example, the second variation of the Sobig worm was initially called “Palyh” but later renamed “Sobig.b”. Again, depending on how quickly this happens, the old name may persist.

2.2.1 Scope

In terms of scope, there are two major variants: the list of “in-the-wild” viruses, which list viruses in active circulation, and lists of all known viruses, which also contain viruses believed not to be in active circulation (also called “zoo viruses”). The sizes are vastly different, in-the-wild lists contain a hundred viruses but full lists contain tens of thousands.

2.2.2 Comparison of viruses and related programs

Related lists

- List of computer worms
- Timeline of computer viruses and worms

Unusual subtypes

- Palm OS Viruses
- Linux malware

Notable instances

- Conficker
- Creeper virus - The first malware that ran on ARPANET
- ILOVEYOU
- Leap - Mac OS X Trojan horse
- Storm Worm - A Windows trojan horse that forms the Storm botnet

Similar software

- Adware
- Malware
- Spamming
- Spyware
- Computer worm
- Trojan horse

Security topics

- Antivirus software
- Computer insecurity
- Cryptovirology
- Security through obscurity

2.2.3 See also

- Computer worm
- Spyware
- Virus hoax
- Zombie computer

2.2.4 References

[1] Vincentas (11 July 2013). "Computer Viruses in SpyWareLoop.com". Spyware Loop. Retrieved 28 July 2013.

2.2.5 External links

- The WildList, by WildList Organization International
- List of Computer Viruses - listing of the Latest Viruses by Symantec.
- List of all viruses All viruses cataloged in Panda Security's Collective Intelligence servers.

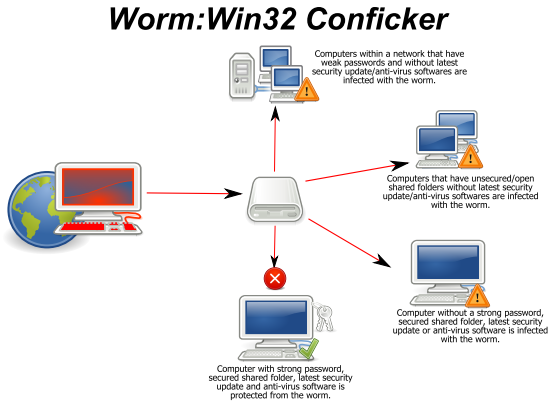
2.3 Computer worm

This article is about coding of a worm. For the data storage device, see Write Once Read Many. For other uses, see worm (disambiguation).

A **computer worm** is a standalone malware computer

Hex dump of the Blaster worm, showing a message left for Microsoft CEO Bill Gates by the worm programmer

program that replicates itself in order to spread to other computers.[1] Often, it uses a computer network to spread itself, relying on security failures on the target computer to access it. Unlike a computer virus, it does not need to attach itself to an existing program.[2] Worms almost always cause at least some harm to the network, even if only by consuming bandwidth, whereas viruses almost always corrupt or modify files on a targeted computer.

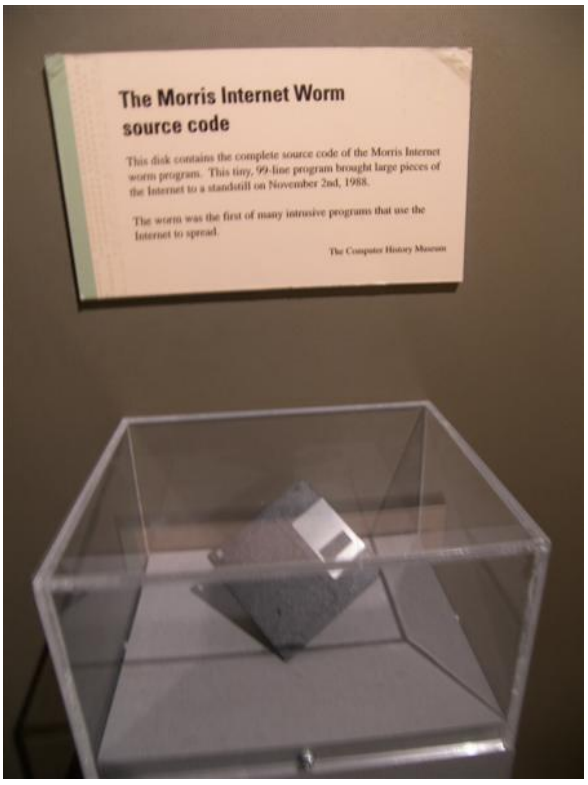


Spread of Conficker worm

Many worms that have been created are designed only to spread, and do not attempt to change the systems they pass through. However, as the Morris worm and Mydoom showed, even these "payload free" worms can cause major disruption by increasing network traffic and other unintended effects. A "payload" is code in the worm designed to do more than spread the worm—it might delete files on a host system (e.g., the ExploreZip worm), encrypt files in a ransomware attack, or send documents via e-mail. A very common payload for worms is to install a backdoor in the infected computer to allow the creation of a "zombie" computer under control of the worm author. Networks of such machines are often referred to as botnets and are very commonly used by spam senders for sending junk email or to cloak their website's address.[3] Spammers are therefore thought to be a source of funding for the creation of such worms,[4][5] and the worm writers have been caught selling lists of IP addresses of infected machines.[6] Others try to blackmail companies or schools with threatening DoS attacks.[7]

Users can minimize the threat posed by worms by keeping their computers' operating system and other software up to date, avoiding opening unrecognized or unexpected emails and running firewall and antivirus software.[8]

Backdoors can be exploited by other malware, including worms. Examples include Doomjuice, which can spread using the backdoor opened by Mydoom, and at least one instance of malware taking advantage of the rootkit and backdoor installed by the Sony/BMG DRM software utilized by millions of music CDs prior to late 2005.[9]



Morris worm source code disk at the Computer History Museum

2.3.1 History

The actual term “worm” was first used in John Brunner's 1975 novel, *The Shockwave Rider*. In that novel, Nichlas Haflinger designs and sets off a data-gathering worm in an act of revenge against the powerful men who run a national electronic information web that induces mass conformity. “You have the biggest-ever worm loose in the net, and it automatically sabotages any attempt to monitor it... There's never been a worm with that tough a head or that long a tail!"[10]

On November 2, 1988, Robert Tappan Morris, a Cornell University computer science graduate student, unleashed what became known as the Morris worm, disrupting a large number of computers then on the Internet, guessed at the time to be one tenth of all those connected[11] During the Morris appeal process, the U.S. Court of Appeals estimated the cost of removing the virus from each installation was in the range of $200–53,000, and prompting the formation of the CERT Coordination Center[12] and Phage mailing list.[13] Morris himself became the first person tried and convicted under the 1986 Computer Fraud and Abuse Act.[14]

2.3.2 Protecting against dangerous computer worms

Worms spread by exploiting vulnerabilities in operating systems. Vendors with security problems supply regular security updates[15] (see "Patch Tuesday"), and if these are installed to a machine then the majority of worms are unable to spread to it. If a vulnerability is disclosed before the security patch released by the vendor, a zero-day attack is possible.

Users need to be wary of opening unexpected email,[16][17] and should not run attached files or programs, or visit web sites that are linked to such emails. However, as with the ILOVEYOU worm, and with the increased growth and efficiency of phishing attacks, it remains possible to trick the end-user into running malicious code.

Anti-virus and anti-spyware software are helpful, but must be kept up-to-date with new pattern files at least every few days. The use of a firewall is also recommended.

In the April–June, 2008, issue of IEEE Transactions on Dependable and Secure Computing, computer scientists describe a potential new way to combat internet worms. The researchers discovered how to contain the kind of worm that scans the Internet randomly, looking for vulnerable hosts to infect. They found that the key is for software to monitor the number of scans that machines on a network sends out. When a machine starts sending out too many scans, it is a sign that it has been infected, allowing administrators to take it off line and check it for malware.[18][19] In addition, machine learning techniques can be used to detect new worms, by analyzing the behavior of the suspected computer.[20]

Mitigation techniques

- ACLs in routers and switches
- Packet-filters
- TCP Wrapper/ACL enabled network service daemons
- Nullroute

2.3.3 Worms with good intent

Main article: Helpful worm

Beginning with the very first research into worms at Xerox PARC, there have been attempts to create useful worms. Those worms allowed testing by John Shoch and Jon Hupp of the Ethernet principles on their network of Xerox Alto

computers. The Nachi family of worms tried to download and install patches from Microsoft's website to fix vulnerabilities in the host system—by exploiting those same vulnerabilities.[21] In practice, although this may have made these systems more secure, it generated considerable network traffic, rebooted the machine in the course of patching it, and did its work without the consent of the computer's owner or user. Regardless of their payload or their writers' intentions, most security experts regard all worms as malware.

Several worms, like XSS worms, have been written to research how worms spread. For example, the effects of changes in social activity or user behavior. One study proposed what seems to be the first computer worm that operates on the second layer of the OSI model (Data link Layer), it utilizes topology information such as Content-addressable memory (CAM) tables and Spanning Tree information stored in switches to propagate and probe for vulnerable nodes until the enterprise network is covered.[22]

2.3.4 See also

- Command and control (malware)
- Computer and network surveillance
- Computer virus
- Email spam
- Malware
- Timeline of computer viruses and worms
- Trojan horse (computing)
- XSS worm
- Zombie (computer science)

2.3.5 References

[1] Barwise, Mike. "What is an internet worm?". BBC. Retrieved 9 September 2010.

[2] "Difference between a computer virus and a computer worm". USCB SicienceLine.

[3] Ray, Tiernan (February 18, 2004). "Business & Technology: E-mail viruses blamed as spam rises sharply". *The Seattle Times*.

[4] McWilliams, Brian (October 9, 2003). "Cloaking Device Made for Spammers". *Wired*.

[5] "Unavailable".

[6] "Uncovered: Trojans as Spam Robots". *Hiese online*. 2004-02-21. Archived from the original on 2009-05-28. Retrieved 2012-11-02.

[7] "Hacker threats to bookies probed". *BBC News*. February 23, 2004.

[8] "Computer Worm Information and Removal Steps". Veracode. Retrieved 2015-04-04.

[9] "Sony Ships Sneaky DRM Software". Pcworld.com. 2005-11-01. Retrieved 2012-06-10.

[10] Brunner, John (1975). *The Shockwave Rider*. New York: Ballantine Books. ISBN 0-06-010559-3.

[11] "The Submarine".

[12] "Security of the Internet". *CERT/CC*.

[13] "Phage mailing list". securitydigest.org.

[14] Dressler, J. (2007). "United States v. Morris". *Cases and Materials on Criminal Law*. St. Paul, MN: Thomson/West. ISBN 978-0-314-17719-3.

[15] "USN list". Ubuntu. Retrieved 2012-06-10.

[16] Threat Description Email-Worm

[17] Threat Description Email-Worm: VBS/LoveLetter

[18] Sellke, S. H.; Shroff, N. B.; Bagchi, S. (2008). "Modeling and Automated Containment of Worms". *IEEE Transactions on Dependable and Secure Computing* **5** (2): 71–86. doi:10.1109/tdsc.2007.70230. Archived from the original on 25 May 2015.

[19] "A New Way to Protect Computer Networks from Internet Worms". *Newswise*. Retrieved July 5, 2011.

[20] Moskovitch R., Elovici Y., Rokach L. (2008), *Detection of unknown computer worms based on behavioral classification of the host*, Computational Statistics and Data Analysis, 52(9):4544–4566, DOI 10.1016/j.csda.2008.01.028

[21] "Virus alert about the Nachi worm". Microsoft.

[22] Al-Salloum, Z. S.; Wolthusen, S. D. (2010). "A link-layer-based self-replicating vulnerability discovery agent". *The IEEE symposium on Computers and Communications*. p. 704. doi:10.1109/ISCC.2010.5546723. ISBN 978-1-4244-7754-8.

2.3.6 External links

- Malware Guide – Guide for understanding, removing and preventing worm infections on Vernalex.com.
- "The 'Worm' Programs – Early Experience with a Distributed Computation", John Shoch and Jon Hupp, *Communications of the ACM*, Volume 25 Issue 3 (March 1982), pages 172–180.

- "The Case for Using Layered Defenses to Stop Worms", Unclassified report from the U.S. National Security Agency (NSA), 18 June 2004.
- Worm Evolution, paper by Jago Maniscalchi on Digital Threat, 31 May 2009.

2.4 List of computer worms

Not to be confused with Timeline of computer viruses and worms.
This list is incomplete; you can help by expanding it.

2.4.1 See also

- Timeline of notable computer viruses and worms
- Comparison of computer viruses
- List of trojan horses

2.5 Timeline of computer viruses and worms

Not to be confused with List of computer worms.

This **timeline of computer viruses and worms** presents a chronology of noteworthy computer viruses, computer worms, Trojan horses, similar malware, related research and events.

2.5.1 1949

- John von Neumann's article on the "Theory of self-reproducing automata" is published.[1] The article is based on lectures given by von Neumann at the University of Illinois about the "Theory and Organization of Complicated Automata" in 1949.

2.5.2 1970–1979

1971

- The Creeper system, an experimental self-replicating program, is written by Bob Thomas at BBN Technologies to test John von Neumann's theory.[2] Creeper infected DEC PDP-10 computers running the TENEX operating system. Creeper gained access via the ARPANET and copied itself to the remote system where the message "I'm the creeper, catch me if you can!" was displayed. The *Reaper* program was later created to delete Creeper.[3]

1973 (Fiction)

In fiction, the 1973 Michael Crichton movie *Westworld* made an early mention of the concept of a computer virus, being a central plot theme that causes androids to run amok.[4] Alan Oppenheimer's character summarizes the problem by stating that "...there's a clear pattern here which suggests an analogy to an infectious disease process, spreading from one...area to the next." To which the replies are stated: "Perhaps there are superficial similarities to disease" and, "I must confess I find it difficult to believe in a disease of machinery."[5] (Crichton's earlier work, the 1969 novel *The Andromeda Strain* and 1971 film were about a biological virus-like disease that threatened the human race.)

1974

- The Rabbit (or Wabbit) virus, more a fork bomb than a virus, is written. The Rabbit virus makes multiple copies of itself on a single computer (and was named "Rabbit" for the speed at which it did so) until it clogs the system, reducing system performance, before finally reaching a threshold and crashing the computer.[6]

1975

- April: ANIMAL is written by John Walker for the UNIVAC 1108.[7] ANIMAL asked a number of questions of the user in an attempt to guess the type of animal that the user was thinking of, while the related program PERVADE would create a copy of itself and ANIMAL in every directory to which the current user had access. It spread across the multi-user UNIVACs when users with overlapping permissions discovered the game, and to other computers when tapes were shared. The program was carefully written to avoid damage to existing file or directory structures, and not to copy itself if permissions did not exist or if damage could result. Its spread was therefore halted by an OS upgrade which changed the format of the file status tables that PERVADE used for safe copying. Though non-malicious, "Pervading Animal" represents the first Trojan "in the wild".[8]
- The novel *The Shockwave Rider* by John Brunner is published, coining the word "worm" to describe a pro-

gram that propagates itself through a computer network.[9]

2.5.3 1980–1989

1981

- A program called Elk Cloner, written for Apple II systems, was created by Richard Skrenta. The Apple II was seen as particularly vulnerable due to the storage of its operating system on floppy disk. Elk Cloner's design combined with public ignorance about what malware was and how to protect against it led to Elk Cloner being responsible for the first large-scale computer virus outbreak in history.[10]

1983

- November: The term 'virus' is coined by Frederick Cohen in describing self-replicating computer programs. In 1984 Cohen uses the phrase "computer virus" – as suggested by his teacher Leonard Adleman – to describe the operation of such programs in terms of "infection". He defines a 'virus' as "a program that can 'infect' other programs by modifying them to include a possibly evolved copy of itself." Cohen demonstrates a virus-like program on a VAX11/750 system at Lehigh University. The program could install itself in, or infect, other system objects.[11]

1984

- August: Ken Thompson publishes his seminal paper, *Reflections on Trusting Trust*, in which he describes how he modified a C compiler so that when used to compile a specific version of the Unix operating system, it inserted a backdoor into the login command, and when used to compile itself, it inserted the backdoor insertion code, even if neither the backdoor nor the backdoor insertion code were present in the source code.[12]

1986

- January: The Brain boot sector virus is released. Brain is considered the first IBM PC compatible virus, and the program responsible for the first IBM PC compatible virus epidemic. The virus is also known as Lahore, Pakistani, Pakistani Brain, and Pakistani flu as it was created in Lahore, Pakistan by 19-year-old Pakistani programmer, Basit Farooq Alvi, and his brother, Amjad Farooq Alvi.[13]
- December: Ralf Burger presented the Virdem model of programs at a meeting of the underground Chaos Computer Club in Germany. The Virdem model represented the first programs that could replicate themselves via addition of their code to executable DOS files in COM format.[14]

1987

- Appearance of the Vienna virus, which was subsequently neutralized—the first time this had happened on the IBM platform.[15]
- Appearance of Lehigh virus (discovered at its namesake university),[15] boot sector viruses such as Yale from USA, Stoned from New Zealand, Ping Pong from Italy, and appearance of first self-encrypting file virus, Cascade. Lehigh was stopped on campus before it spread to the wild, and has never been found elsewhere as a result. A subsequent infection of Cascade in the offices of IBM Belgium led to IBM responding with its own antivirus product development. Prior to this, antivirus solutions developed at IBM were intended for staff use only.
- October: The Jerusalem virus, part of the (at that time unknown) Suriv family, is detected in the city of Jerusalem. The virus destroys all executable files on infected machines upon every occurrence of Friday the 13th (except Friday 13 November 1987 making its first trigger date May 13, 1988). Jerusalem caused a worldwide epidemic in 1988.[15]
- November: The SCA virus, a boot sector virus for Amigas appears, immediately creating a pandemic virus-writer storm. A short time later, SCA releases another, considerably more destructive virus, the Byte Bandit.
- December: Christmas Tree EXEC was the first widely disruptive replicating network program, which paralyzed several international computer networks in December 1987. It was written in Rexx on the VM/CMS operating system and originated in what was then West Germany. It re-emerged in 1990.

1988

- March 1: The Ping-Pong virus (also called Boot, Bouncing Ball, Bouncing Dot, Italian, Italian-A or VeraCruz), an MS-DOS boot sector virus, is discovered at University of Turin in Italy.
- June: The CyberAIDS and Festering Hate Apple ProDOS viruses spreads from underground pirate BBS

systems and starts infecting mainstream networks. Festering Hate was the last iteration of the CyberAIDS series extending back to 1985 and 1986. Unlike the few Apple viruses that had come before which were essentially annoying, but did no damage, the Festering Hate series of viruses was extremely destructive, spreading to all system files it could find on the host computer (hard drive, floppy, and system memory) and then destroying everything when it could no longer find any uninfected files.

- November 2: The Morris worm, created by Robert Tappan Morris, infects DEC VAX and Sun machines running BSD UNIX that are connected to the Internet, and becomes the first worm to spread extensively "in the wild", and one of the first well-known programs exploiting buffer overrun vulnerabilities.

1989

- October: Ghostball, the first multipartite virus, is discovered by Friðrik Skúlason. It infects both executable .COM-files and boot sectors on MS-DOS systems. It captures certain information entered or saved by the user, with the corresponding threat to privacy, causes the loss of information stored on the computer, either specific files or data in general, affects the productivity of the computer, the network to which it's connected or other remote sites, decrease the security level of the computer, but does not automatically spread itself.
- December: Several thousand floppy disks containing the AIDS Trojan, the first known ransomware, are mailed to subscribers of PC Business World magazine and a WHO AIDS conference mailing list. This DOS Trojan lies dormant for 90 boot cycles, then encrypts all filenames on the system, displaying a notice asking for $189 to be sent to a post office box in Panama in order to receive a decryption program.

2.5.4 1990–1999

1990

- Mark Washburn, working on an analysis of the Vienna and Cascade viruses with Ralf Burger, develops the first family of polymorphic viruses, the Chameleon family. Chameleon series debuted with the release of 1260.[16][17][18]
- June: The Form computer virus is isolated in Switzerland. It would remain in the wild for almost 20 years and reappear afterwards; during the 1990s it tended to be the most common virus in the wild with 20 to more than 50 per cent of reported infections.

1992

- March: The Michelangelo virus was expected to create a digital apocalypse on March 6, with millions of computers having their information wiped, according to mass media hysteria surrounding the virus. Later assessments of the damage showed the aftermath to be minimal. John McAfee had been quoted by the media as saying that 5 million computers would be affected. He later said that, pressed by the interviewer to come up with a number, he had estimated a range from 5 thousand to 5 million, but the media naturally went with just the higher number.

1993

- "Leandro" or "Leandro & Kelly"[19] and "Freddy Krueger"[20] spread quickly due to popularity of BBS and shareware distribution.

1994

- April: OneHalf is a DOS-based polymorphic computer virus.

1995

- The first Macro virus, called "Concept", is created. It attacked Microsoft Word documents.[21]

1996

- "Ply" — DOS 16-bit based complicated polymorphic virus appeared with built-in permutation engine.
- Boza, the first virus designed specifically for Windows 95 files arrives.
- Laroux, the first Excel macro virus appears.
- Staog, the first Linux virus attacks Linux machines

1998

- June 2: The first version of the CIH virus appears. It is the first known virus able to erase flash ROM BIOS content.

1999

- January 20: The Happy99 worm first appeared. It invisibly attaches itself to emails, displays fireworks to hide the changes being made, and wishes the user a happy New Year. It modifies system files related to Outlook Express and Internet Explorer (IE) on Windows 95 and Windows 98.
- March 26: The Melissa worm was released, targeting Microsoft Word and Outlook-based systems, and creating considerable network traffic.
- June 6: The ExploreZip worm, which destroys Microsoft Office documents, was first detected.
- December 30: The Kak worm is a JavaScript computer worm that spread itself by exploiting a bug in Outlook Express.[22]

2.5.5 2000–2009

2000

- May 5: The ILOVEYOU worm, also known as Love Letter, or VBS, or Love Bug worm, is a computer worm purportedly created by a Filipino computer science student. Written in VBScript, it infected millions of Windows computers worldwide within a few hours of its release. Using social engineering techniques, it is considered to be one of the most damaging worms ever.
- June 28: The Pikachu virus is believed to be the first computer virus geared at children. It contains the character "Pikachu" from the Pokémon series, and is in the form of an e-mail titled “Pikachu Pokemon” with the message: “Pikachu is your friend.” The attachment to the email has “an image of a pensive Pikachu”, along with a message stating, “Between millions of people around the world I found you. Don’t forget to remember this day every time MY FRIEND.” Along with the image, there is a program, written in Visual Basic 6, called “pikachupokemon.exe” that modifies the AUTOEXEC.BAT file and adds a command for removing the contents of directories C:\Windows and C:\Windows\System at computer’s restart. But, a message would pop up during startup, asking the user if they would like to delete the contents. The affected operating systems are Windows 95, Windows 98 and Windows Me.

2001

- February 11: The Anna Kournikova virus hits e-mail servers hard by sending e-mail to contacts in the Microsoft Outlook addressbook.[23] Its creator, Dutchman Jan de Wit, was sentenced to 150 hours of community service.[24]
- May 8: The Sadmind worm spreads by exploiting holes in both Sun Solaris and Microsoft IIS.
- July: The Sircam worm is released, spreading through Microsoft systems via e-mail and unprotected network shares.
- July 13: The Code Red worm attacking the Index Server ISAPI Extension in Microsoft Internet Information Services is released.
- August 4: A complete re-write of the Code Red worm, Code Red II begins aggressively spreading onto Microsoft systems, primarily in China.
- September 18: The Nimda worm is discovered and spreads through a variety of means including vulnerabilities in Microsoft Windows and backdoors left by Code Red II and Sadmind worm.
- October 26: The Klez worm is first identified. It exploits a vulnerability in Microsoft Internet Explorer and Microsoft Outlook and Outlook Express.

2002

- February 11: The Simile virus is a metamorphic computer virus written in assembly.
- Beast is a Windows-based backdoor Trojan horse, more commonly known as a RAT (Remote Administration Tool). It is capable of infecting almost all versions of Windows. Written in Delphi and released first by its author Tataye in 2002, its most current version was released October 3, 2004
- March 7: Mylife is a computer worm that spread itself by sending malicious emails to all the contacts in Microsoft Outlook.[25]
- August 30: Optix Pro is a configurable remote access tool or trojan, similar to SubSeven or BO2K.[26]

2003

- January 24: The SQL Slammer worm, aka *Sapphire worm*, *Helkern* and other names, attacks vulnerabilities in Microsoft SQL Server and MSDE becomes the

fastest spreading worm of all time (measured by doubling time at the peak rate of growth),[27] crashing the Internet within 15 minutes of release.[28]

- April 2: Graybird is a trojan horse also known as Backdoor.Graybird.[29]
- June 13: ProRat is a Turkish-made Microsoft Windows based backdoor trojan horse, more commonly known as a RAT (Remote Administration Tool).[30]
- August 12: The Blaster worm, aka the *Lovesan* worm, rapidly spreads by exploiting a vulnerability in system services present on Windows computers.
- August 18: The Welchia (Nachi) worm is discovered. The worm tries to remove the blaster worm and patch Windows.
- August 19: The Sobig worm (technically the Sobig.F worm) spreads rapidly through Microsoft systems via mail and network shares.
- September 18: Swen is a computer worm written in C++.[31]
- October 24: The Sober worm is first seen on Microsoft systems and maintains its presence until 2005 with many new variants. The simultaneous attacks on network weakpoints by the Blaster and Sobig worms cause massive damage.
- November 10: Agobot is a computer worm that can spread itself by exploiting vulnerabilities on Microsoft Windows. Some of the vulnerabilities are MS03-026 and MS05-039.[32]
- November 20: Bolgimo is a computer worm that spread itself by exploiting a buffer overflow vulnerability at Microsoft Windows DCOM RPC Interface.[33]

2004

- January 18: Bagle is a mass-mailing worm affecting all versions of Microsoft Windows. There were 2 variants of Bagle worm, Bagle.A and Bagle.B. Bagle.B was discovered on February 17, 2004.
- January 23: The L10n worm (usually pronounced "lion") was a Linux worm that spread by exploiting a buffer overflow in the BIND DNS server. It was based on an earlier worm known as the Ramen worm (commonly, albeit incorrectly referred to as the Ramen Virus) which was written to target systems running versions 6.2 and 7.0 of the Red Hat Linux distribution.
- Late January: The MyDoom worm emerges, and currently holds the record for the fastest-spreading mass mailer worm.
- February 16: The Netsky worm is discovered. The worm spreads by email and by copying itself to folders on the local hard drive as well as on mapped network drives if available. Many variants of the Netsky worm appeared.
- March 19: The Witty worm is a record-breaking worm in many regards. It exploited holes in several Internet Security Systems (ISS) products. It was the fastest disclosure to worm, it was the first internet worm to carry a destructive payload and it spread rapidly using a prepopulated list of ground-zero hosts.
- May 1: The Sasser worm emerges by exploiting a vulnerability in the Microsoft Windows LSASS service and causes problems in networks, while removing MyDoom and Bagle variants, even interrupting business.
- June 15: Caribe or Cabir is a computer worm that is designed to infect mobile phones that run Symbian OS. It is the first computer worm that can infect mobile phones. It spread itself through Bluetooth. More information can be found on F-Secure[34] and Symantec.[35]
- August 16: Nuclear RAT (short for Nuclear Remote Administration Tool) is a backdoor trojan that infects Windows NT family systems (Windows 2000, Windows XP, Windows 2003).[36]
- August 20: Vundo, or the Vundo Trojan (also known as Virtumonde or Virtumondo and sometimes referred to as MS Juan) is a trojan known to cause popups and advertising for rogue antispyware programs, and sporadically other misbehaviour including performance degradation and denial of service with some websites including Google and Facebook.[37]
- October 12: Bifrost, also known as Bifrose, is a backdoor trojan which can infect Windows 95 through Vista. Bifrost uses the typical server, server builder, and client backdoor program configuration to allow a remote attack.[38]
- December: Santy, the first known "webworm" is launched. It exploited a vulnerability in phpBB and used Google in order to find new targets. It infected around 40000 sites before Google filtered the search query used by the worm, preventing it from spreading.

2005

- August 2005: Zotob
- October 2005: The copy protection rootkit deliberately and surreptitiously included on music CDs sold by Sony BMG is exposed. The rootkit creates vulnerabilities on affected computers, making them susceptible to infection by worms and viruses.
- Late 2005: The Zlob Trojan, is a Trojan horse program that masquerades as a required video codec in the form of the Microsoft Windows ActiveX component. It was first detected in late 2005.[39]
- Bandook or Bandook Rat (Bandook Remote Administration Tool) is a backdoor Trojan horse that infects the Windows family. It uses a server creator, a client and a server to take control over the remote computer. It uses process hijacking / kernel patching to bypass the firewall, and let the server component hijack processes and gain rights for accessing the Internet.

2006

- January 20: The Nyxem worm was discovered. It spread by mass-mailing. Its payload, which activates on the third of every month, starting on February 3, attempts to disable security-related and file sharing software, and destroy files of certain types, such as Microsoft Office files.
- February 16: discovery of the first-ever malware for Mac OS X, a low-threat trojan-horse known as OSX/Leap-A or OSX/Oompa-A, is announced.
- Late March: Brontok variant N was found in late March.[40] Brontok was a mass-email worm and the origin for the worm was from Indonesia.
- Late September: Stration or Warezov worm first discovered.

2007

- January 17: Storm Worm identified as a fast spreading email spamming threat to Microsoft systems. It begins gathering infected computers into the Storm botnet. By around June 30 it had infected 1.7 million computers, and it had compromised between 1 and 10 million computers by September.[41] Thought to have originated from Russia, it disguises itself as a news email containing a film about bogus news stories asking you to download the attachment which it claims is a film.
- July: Zeus is a trojan that targets Microsoft Windows to steal banking information by keystroke logging.

2008

- February 17: Mocmex is a trojan, which was found in a digital photo frame in February 2008. It was the first serious computer virus on a digital photo frame. The virus was traced back to a group in China.[42]
- March 3: Torpig, also known as Sinowal and Mebroot, is a Trojan horse that affects Windows, turning off anti-virus applications. It allows others to access the computer, modifies data, steals confidential information (such as user passwords and other sensitive data) and installs more malware on the victim's computer.[43]
- May 6: Rustock.C, a hitherto-rumoured spambot-type malware with advanced rootkit capabilities, was announced to have been detected on Microsoft systems and analyzed, having been in the wild and undetected since October 2007 at the very least.[44]
- July 6: Bohmini.A is a configurable remote access tool or trojan that exploits security flaws in Adobe Flash 9.0.115 with Internet Explorer 7.0 and Firefox 2.0 under Windows XP SP2.[45]
- July 31: The Koobface computer worm targets users of Facebook and Myspace. New variants constantly appear.[46]
- November 21: Computer worm Conficker infects anywhere from 9 to 15 million Microsoft server systems running everything from Windows 2000 to the Windows 7 Beta. The French Navy,[47] UK Ministry of Defence (including Royal Navy warships and submarines),[48] Sheffield Hospital network,[49] German Bundeswehr[50] and Norwegian Police were all affected. Microsoft sets a bounty of US$250,000 for information leading to the capture of the worm's author(s).[51] Five main variants of the Conficker worm are known and have been dubbed Conficker A, B, C, D and E. They were discovered 21 November 2008, 29 December 2008, 20 February 2009, 4 March 2009 and 7 April 2009, respectively. On December 16, 2008, Microsoft releases KB958644 [52] patching the server service vulnerability responsible for the spread of Conficker.

2009

- July 4: The July 2009 cyber attacks occur and the emergence of the W32.Dozer attack the United States and South Korea.

- July 15: Symantec discovered Daprosy Worm. Said trojan worm is intended to steal online-game passwords in internet cafes. It could, in fact, intercept all keystrokes and send them to its author which makes it potentially a very dangerous worm to infect B2B (business-to-business) systems.

- August 24: Source code for MegaPanzer is released by its author under GPLv3.[53] And appears to be apparently detected in the wild.[54]

2.5.6 2010 and later

2010

- January: The Waledac botnet sent spam emails. In February 2010, an international group of security researchers and Microsoft took Waledac down.[55]

- February 18: Microsoft announced that a BSoD problem on some Windows machines which was triggered by a batch of Patch Tuesday updates was caused by the Alureon Trojan.[56]

- June 17: Stuxnet, a Windows Trojan, was detected.[57] It is the first worm to attack SCADA systems.[58] There are suggestions that it was designed to target Iranian nuclear facilities.[59] It uses a valid certificate from Realtek.[60]

- September 9: The virus, called "here you have" or “VBMania”, is a simple Trojan horse that arrives in the inbox with the odd-but-suggestive subject line “here you have”. The body reads “This is The Document I told you about, you can find it Here” or “This is The Free Download Sex Movies, you can find it Here”.

- September 15: The virus called Kenzero is a virus that spreads online from Peer to peer (P2P) sites taking browsing history.[61]

2011

- SpyEye and Zeus merged code is seen.[62] New variants attack mobile phone banking information.[63]

- Anti-Spyware 2011, a Trojan horse that attacks Windows 9x, 2000, XP, Vista, and Windows 7, posing as an anti-spyware program. It actually disables security-related process of anti-virus programs, while also blocking access to the Internet, which prevents updates.[64]

- Summer 2011: The Morto worm attempts to propagate itself to additional computers via the Microsoft Windows Remote Desktop Protocol (RDP). Morto spreads by forcing infected systems to scan for Windows servers allowing RDP login. Once Morto finds an RDP-accessible system, it attempts to log in to a domain or local system account named 'Administrator' using a number of common passwords.[65] A detailed overview of how the worm works—along with the password dictionary Morto uses—was done by Imperva.[66]

- July 13: the ZeroAccess rootkit (also known as Sirefef or max++) was discovered.

- September 1: Duqu is a worm thought to be related to the Stuxnet worm. The Laboratory of Cryptography and System Security (CrySyS Lab)[67] of the Budapest University of Technology and Economics in Hungary discovered the threat, analysed the malware, and wrote a 60-page report naming the threat Duqu.[68][69] Duqu gets its name from the prefix "~DQ” it gives to the names of files it creates.[70]

2012

- May: Flame - also known as Flamer, sKyWIper, and Skywiper - a modular computer malware that attacks computers running Microsoft Windows. Used for targeted cyber espionage in Middle Eastern countries. Its discovery was announced on 28 May 2012 by MAHER Center of Iranian National Computer Emergency Response Team (CERT), Kaspersky Lab and CrySyS Lab of the Budapest University of Technology and Economics. CrySyS stated in their report that “sKyWIper is certainly the most sophisticated malware we encountered during our practice; arguably, it is the most complex malware ever found”.[71]

- August 16: Shamoon is a computer virus designed to target computers running Microsoft Windows in the energy sector. Symantec, Kaspersky Lab, and Seculert announced its discovery on August 16, 2012.

- September 20: NGRBot is a worm that uses the IRC network for file transfer, sending and receiving commands between zombie network machines and the attacker’s IRC server, and monitoring and controlling network connectivity and intercept. It employs a user-mode rootkit technique to hide and steal its victim’s information. This family of bot is also designed to infect HTML pages with inline frames ([HTML element#Frames|[iframes]]), causing redirections, blocking victims from getting updates from security/antimalware products, and killing those services. The bot is designed to connect via a prede-

fined IRC channel and communicate with a remote botnet.[72][73]

2013

- September: The CryptoLocker Trojan horse is discovered. Cryptolocker encrypts the files on a user's hard drive, then prompts them to pay a ransom to the developer in order to receive the decryption key. In the following months, a number of copycat ransomware Trojans are also discovered.
- December: The Gameover ZeuS Trojan is discovered. This type of virus steals one's login details on popular Web sites that involve monetary transactions. It works by detecting a login page, then proceeds to inject a malicious code into the page, keystroke logging the computer user's details.

2014

- November: The Regin Trojan horse is discovered. Regin is a dropper that is primarily spread via spoofed Web pages. Once downloaded, Regin quietly downloads extensions of itself, making it difficult to be detected via anti-virus signatures. It is suspected to have been created by the United States and United Kingdom over a period of months or years, as a tool for espionage and mass surveillance.

2016

- February: Ransomware Locky with its over 60 derivatives spread throughout Europe and infected several million computers. At the height of the spread over five thousand computers per hour were infected in Germany alone.[74] Although ransomware was not a new thing at the time, insufficient cyber security as well as a lack of standards in IT was responsible for the high number of infections.[75] Unfortunately even up to date antivirus and internet security software was unable to protect systems from early versions of Locky.[76]
- February: Tiny Banker Trojan makes headlines [77] Since its discovery, it has been found to have infected more than two dozen major banking institutions in the United States, including TD Bank, Chase, HSBC, Wells Fargo, PNC and Bank of America.[78][79] Tiny Banker Trojan uses HTTP injection to force the user's computer to believe that it is on the bank's website. This spoof page will look and function just as the real one. The user then enters their information to log on, at which point Tinba can launch the bank webpage's "incorrect login information" return, and redirect the user to the real website. This is to trick the user into thinking they had entered the wrong information and proceed as normal, although now Tinba has captured the credentials and sent them to its host.[80][81]

2.5.7 See also

- Helpful worm
- Multipartite virus
- Timeline of computer security hacker history

2.5.8 References

[1] von Neumann, John (1966). Arthur W. Burks, ed. *Theory of self-reproducing automata* (PDF). University of Illinois Press. Retrieved June 12, 2010.

[2] Chen, Thomas; Robert, Jean-Marc (2004). "The Evolution of Viruses and Worms". Retrieved 2009-02-16.

[3] Russell, Deborah; Gangemi, G T (1991). *Computer Security Basics*. O'Reilly. p. 86. ISBN 0-937175-71-4.

[4] http://www.imdb.com/title/tt0070909/synopsis: IMDB synopsis of Westworld. Retrieved November 28, 2015.

[5] Michael Crichton (November 21, 1973). *Westworld* (movie). 201 S. Kinney Road, Tucson, Arizona, USA: Metro-Goldwyn-Mayer. Event occurs at 32 minutes. And there's a clear pattern here which suggests an analogy to an infectious disease process, spreading from one resort area to the next." ... "Perhaps there are superficial similarities to disease." "I must confess I find it difficult to believe in a disease of machinery.

[6] "The very first viruses: Creeper, Wabbit and Brain", Daniel Snyder, InfoCarnivore, May 30, 2010

[7] "ANIMAL Source Code". Fourmilab.ch. 1996-08-13. Retrieved 2012-03-29.

[8] "The Animal Episode". Fourmilab.ch. Retrieved 2012-03-29.

[9] Craig E. Engler (1997). "The Shockwave Rider". *Classic Sci-Fi Reviews*. Archived from the original on 2008-07-03. Retrieved 2008-07-28.

[10] "First virus hatched as a practical joke", *Sydney Morning Herald* (AP), 3 September 2007. Retrieved 9 September 2013.

[11] "Fred Cohen 1984 "Computer Viruses – Theory and Experiments"". Eecs.umich.edu. 1983-11-03. Retrieved 2012-03-29.

[12] Communication of the ACM, Vol. 27, No. 8, August 1984, pp. 761-763.

[13] Leyden, John (January 19, 2006). "PC virus celebrates 20th birthday". *The Register*. Retrieved March 21, 2011.

[14] *The Art of Computer Virus Research and Defense*, Peter Szor, Symantec Press / Addison-Wesley Professional, 2005, ISBN 978-0-321-30454-4

[15] "Computer Virus!", Rob Wentworth, Reprinted from *The Digital Viking*, Twin Cities PC User Group, July 1996. Retrieved 9 September 2013.

[16] "Virus.DOS.Chameleon.1260 - Securelist". Viruslist.com. Retrieved 2010-07-10.

[17] "V2PX". Vil.nai.com. Retrieved 2010-07-10.

[18] "What we detect - Securelist". Viruslist.com. Retrieved 2010-07-10.

[19] "Leandro", *Threat Encyclopedia*, Trend Micro, 9 March 2000. Retrieved 9 September 2013.

[20] "Freddy Virus", Virus Information Summary List, December 1992. Retrieved 9 September 2013.

[21] "Glossary - Securelist". Viruslist.com. Retrieved 2010-07-10.

[22] "Wscript.KakWorm". Symantec. Retrieved 2012-03-29.

[23] "Kournikova computer virus hits hard". *BBC News*. February 13, 2001. Retrieved April 9, 2010.

[24] Evers, Joris (May 3, 2002). "Kournikova virus maker appeals sentence". Retrieved 20 November 2010.

[25] "MyLife Worm". Antivirus.about.com. 2002-03-07. Retrieved 2012-03-29.

[26] Sevcenco, Serghei (August 30, 2002). "Security Updates: Backdoor.OptixPro.12". Symantec. Retrieved 2009-03-01.

[27] "The Spread of the Sapphire/Slammer Worm". Retrieved 2012-12-14.

[28] "Slammed!". July 2003. Retrieved 2012-12-14.

[29] Sevcenco, Serghei (February 10, 2006). "Symantec Security Response: Backdoor.Graybird". Symantec. Retrieved 2009-03-01.

[30] "Backdoor.Prorat". Symantec. February 13, 2007. Retrieved 2009-03-01.

[31] "Threat Description: Worm:W32/Swen". F-secure.com. Retrieved 2012-03-29.

[32] "Backdoor.Win32.Agobot.gen". Securelist. Retrieved 2012-03-29.

[33] "W32.Bolgi.Worm". Symantec. Retrieved 2012-03-29.

[34] "Threat Description:Bluetooth-Worm:SymbOS/Cabir". F-secure.com. Retrieved 2012-03-29.

[35] "SymbOS.Cabir". Symantec. Retrieved 2012-03-29.

[36] "Spyware Detail Nuclear RAT 1.0b1". Computer Associates. August 16, 2004. Archived from the original on 2009-09-11. Retrieved 2009-03-01.

[37] "Vundo". McAfee. Retrieved 2009-03-01.

[38] "Backdoor.Bifrose". Symantec, Inc. October 12, 2004. Retrieved 2009-02-28.

[39] "The ZLOB Show: Trojan Poses as Fake Video Codec, Loads More Threats". Trend Micro. Retrieved 2009-02-28.

[40] "Threat Description: Email-Worm:W32/Brontok.N". F-secure.com. Retrieved 2012-03-29.

[41] Peter Gutmann (31 August 2007). "World's most powerful supercomputer goes online". Full Disclosure. Retrieved 2007-11-04.

[42] Gage, Deborah (February 17, 2005). "Chinese PC virus may have hidden agenda". SeatlePI. Retrieved 2009-03-01.

[43] Kimmo (March 3, 2008). "MBR Rootkit, A New Breed of". F-Secure. Retrieved 2009-03-01.

[44] "Win32.Ntldrbot (aka Rustock)". Dr. Web Ltd. Retrieved 2009-03-01.

[45] "Virus Total". virustotal.com. July 8, 2008. Archived from the original on 2009-04-01. Retrieved 2009-03-01.

[46] "Koobface malware makes a comeback". cnet.com. April 9, 2010. Retrieved 2009-04-13.

[47] Willsher, Kim (2009-02-07). "French fighter planes grounded by computer virus". London: The Daily Telegraph. Retrieved 2009-04-01.

[48] Williams, Chris (2009-01-20). "MoD networks still malware-plagued after two weeks". The Register. Retrieved 2009-01-20.

[49] Williams, Chris (2009-01-20). "Conficker seizes city's hospital network". The Register. Retrieved 2009-01-20.

[50] "Conficker-Wurm infiziert hunderte Bundeswehr-Rechner" (in German). PC Professionell. 2009-02-16. Archived from the original on 2009-03-21. Retrieved 2009-04-01.

[51] Neild, Barry (2009-02-13). "$250K Microsoft bounty to catch worm creator". CNN. Retrieved 2009-03-29.

[52] "MS08-067: Vulnerability in Server service could allow remote code execution". Microsoft Corporation.

[53] Dancho Danchev. "Source code for Skype eavesdropping trojan in the wild". *ZDNet*.

[54] "Code for Skype Spyware Released to Thwart Surveillance". *WIRED*. 31 August 2009.

[55] "Waledac Takedown Successful". honeyblog.org. February 25, 2010. Retrieved 16 November 2012.

[56] "Alureon trojan caused Windows 7 BSoD". microsoft.com. February 18, 2010. Retrieved 2010-02-18.

[57] "VirusBlokAda News". Anti-virus.by. Retrieved 2012-03-29.

[58] Gregg Keizer (16 September 2010). "Is Stuxnet the 'best' malware ever?". InfoWorld. Retrieved 16 September 2010.

[59] Stuxnet virus: worm 'could be aimed at high-profile Iranian targets', Telegraph, 23 Sep 2010

[60] "Possible New Rootkit Has Drivers Signed by Realtek". Kaspersky Labs. 15 July 2010.

[61] Harvison, Josh (September 27, 2010). "Blackmail virus infects computers, holds information ransom". kait8.com. Retrieved 20 November 2010.

[62] "Bastard child of SpyEye/ZeuS merger appears online". *The Register*. 2011. Retrieved April 11, 2011. Bastard child of SpyEye/ZeuS merger appears online

[63] "SpyEye mobile banking Trojan uses same tactics as ZeuS". *The Register*. 2011. Retrieved April 11, 2011. SpyEye mobile banking Trojan uses same tactics as ZeuS

[64] "XP AntiSpyware 2011 - Virus Solution and Removal". Precisesecurity.com. Retrieved 2012-03-29.

[65] "Morto Worm Spreads to Weak Systems". *blogs.appriver.com*. 2011.

[66] "Morto Post Mortem: Dissecting a Worm". *blog.imperva.com*. 2011.

[67] "Laboratory of Cryptography and System Security (CrySyS)". Retrieved 4 November 2011.

[68] "Duqu: A Stuxnet-like malware found in the wild, technical report" (PDF). Laboratory of Cryptography of Systems Security (CrySyS). 14 October 2011.

[69] "Statement on Duqu's initial analysis". Laboratory of Cryptography of Systems Security (CrySyS). 21 October 2011. Retrieved 25 October 2011.

[70] "W32.Duqu – The precursor to the next Stuxnet (Version 1.4)" (PDF). Symantec. 23 November 2011. Retrieved 30 December 2011.

[71] "sKyWIper: A Complex Malware for Targeted Attacks" (PDF). Budapest University of Technology and Economics. 28 May 2012. Archived from the original on 30 May 2012. Retrieved 29 May 2012.

[72] "NGRBot", Enigma Software Group, 15 October 2012. Retrieved 9 September 2013.

[73] "Dissecting the NGR bot framework: IRC botnets die hard", Aditya K. Sood and Richard J. Enbody, Michigan State University, USA, and Rohit Bansal, SecNiche Security, USA, with Helen Martin1 (ed.), January 2012. Retrieved 9 September 2013. (subscription required)

[74] "Ransomware: Erpresserische Schadprogramme", bsi-fuer-buerger.de, 9 February 2016. Retrieved 10 March 2016.

[75] "Locky ransomware on aggressive hunt for victims", Symantec.com, 18 February 2016. Retrieved 10 March 2016.

[76] "Antivirus scan for (Locky)", virustotal.com, 16 February 2016. Retrieved 10 March 2016.

[77] http://www.massivealliance.com/2014/09/19/tiny-banker-malware-attempted-customers-us-banks

[78] "Modified Tiny Banker Trojan Found Targeting Major U.S. Banks". *Entrust, Inc*.

[79] Jeremy Kirk (15 September 2014). "'Tiny banker' malware targets US financial institutions". *PCWorld*.

[80] "'Tiny Banker' Malware Targets Dozens of Major US Financial Institutions". *The State of Security*.

[81] "Tiny 'Tinba' Banking Trojan Is Big Trouble". *msnbc.com*.

2.5.9 External links

- Snopes — Compilation of viruses, worms, and Trojan horses at snopes.com.
- A short history of hacks, worms and cyberterror by Mari Keefe, Computerworld, April 2009

Chapter 3

Concealment

3.1 Trojan horse (computing)

For other uses, see Trojan horse (disambiguation).

In computing, **Trojan horse**, or **Trojan**, is any malicious computer program which is used to hack into a computer by misleading users of its true intent. The term is derived from the Ancient Greek story of the wooden horse that was used to help Greek troops invade the city of Troy by stealth.[1][2][3][4][5]

Trojans are generally spread by some form of social engineering, for example where a user is duped into executing an e-mail attachment disguised to be unsuspicious, (e.g., a routine form to be filled in), or by drive-by download. Although their payload can be anything, many modern forms act as a backdoor, contacting a controller which can then have unauthorized access to the affected computer.[6]

Unlike computer viruses and worms, Trojans generally do not attempt to inject themselves into other files or otherwise propagate themselves.[7]

3.1.1 Purpose and uses

If installed or run with elevated privileges a Trojan will generally have unlimited access. What it does with this power depends on the motives of the attacker.

Destructive

- Crashing the computer or device.
- Modification or deletion of files.
- Data corruption.
- Formatting disks, destroying all contents.
- Spreading malware across the network.
- Spying on user activities and access sensitive information.[8]

Use of resources or identity

- Use of the machine as part of a botnet (e.g. to perform automated spamming or to distribute Denial-of-service attacks)
- Using computer resources for mining cryptocurrencies [9]
- Using the infected computer as proxy for illegal activities and/or attacks on other computers.
- Infecting other connected devices on the network.

Money theft, ransom

- Electronic money theft
- Installing ransomware such as CryptoLocker

Data theft

- Data theft, including for industrial espionage
- User passwords or payment card information
- User personally identifiable information
- Trade secrets

Spying, surveilance or stalking

- Keystroke logging
- Watching the user's screen
- Viewing the user's webcam
- Controlling the computer system remotely

Trojan horses in this way may require interaction with a malicious controller (not necessarily distributing the Trojan horse) to fulfill their purpose. It is possible for those involved with Trojans to scan computers on a network to locate any with a Trojan horse installed, which the hacker can then control. .[10]

Some Trojans take advantage of a security flaw in older versions of Internet Explorer and Google Chrome to use the host computer as an anonymizer proxy to effectively hide Internet usage,[11] enabling the controller to use the Internet for illegal purposes while all potentially incriminating evidence indicates the infected computer or its IP address. The host's computer may or may not show the internet history of the sites viewed using the computer as a proxy. The first generation of anonymizer Trojan horses tended to leave their tracks in the page view histories of the host computer. Later generations of the Trojan horse tend to "cover" their tracks more efficiently. Several versions of Sub7 have been widely circulated in the US and Europe and became the most widely distributed examples of this type of Trojan horse.[10]

In German-speaking countries, spyware used or made by the government is sometimes called *govware*. Govware is typically a trojan horse software used to intercept communications from the target computer. Some countries like Switzerland and Germany have a legal framework governing the use of such software.[12][13] Examples of govware trojans include the Swiss MiniPanzer and MegaPanzer[14] and the German "state trojan" nicknamed R2D2.[12]

Due to the popularity of botnets among hackers and the availability of advertising services that permit authors to violate their users' privacy, Trojan horses are becoming more common. According to a survey conducted by BitDefender from January to June 2009, "Trojan-type malware is on the rise, accounting for 83-percent of the global malware detected in the world." Trojans have a relationship with worms, as they spread with the help given by worms and travel across the internet with them.[15] BitDefender has stated that approximately 15% of computers are members of a botnet, usually recruited by a Trojan infection.[16]

3.1.2 Notable examples

Private and governmental

- FinFisher – Lench IT solutions / Gamma International
- DaVinci / Galileo RCS – HT S.r.l. (*hacking team*)
- 0zapftis / r2d2 StaatsTrojaner – DigiTask
- TAO QUANTUM/FOXACID – NSA
- Magic Lantern – FBI
- WARRIOR PRIDE – GCHQ

Publicly available

- Netbus – 1998 (published)
- Sub7 – 1999 (published)
- Back Orifice – 1998 (published)
- Beast – 2002 (published)
- Bifrost Trojan – 2004 (published)
- DarkComet – 2008 (published)
- Blackhole exploit kit – 2012 (published)
- Gh0st RAT – 2009 (published)
- MegaPanzer BundesTrojaner – 2009 (published)[17][18]

Detected by security researchers

- Clickbot.A – 2006 (discovered)
- Zeus – 2007 (discovered)
- Flashback Trojan – 2011 (discovered)
- ZeroAccess – 2011 (discovered)
- Koobface – 2008 (discovered)
- Vundo – 2009 (discovered)
- Meredrop – 2010 (discovered)
- Coreflood – 2010 (discovered)
- Tiny Banker Trojan – 2012 (discovered)
- Shedun Android malware – 2015 (discovered) [19][20][21][22][23][24]

3.1.3 See also

- Computer security
- Remote administration
- Remote administration software
- Cyber spying
- Dancing pigs
- Exploit (computer security)
- Industrial espionage

- Malware
- Principle of least privilege
- Privacy-invasive software
- Reverse connection
- Rogue security software
- Social engineering (security)
- Spam
- Spyware
- Timeline of computer viruses and worms
- Command and control (malware)
- Zombie (computer science)

3.1.4 References

- Carnegie Mellon University (1999): "CERT Advisory CA-1999-02 Trojan Horses", ŸЦ

[1] Landwehr, C. E; A. R Bull; J. P McDermott; W. S Choi (1993). *A taxonomy of computer program security flaws, with examples*. DTIC Document. Retrieved 2012-04-05.

[2] "Trojan Horse Definition". Retrieved 2012-04-05.

[3] "Trojan horse". *Webopedia*. Retrieved 2012-04-05.

[4] "What is Trojan horse? – Definition from Whatis.com". Retrieved 2012-04-05.

[5] "Trojan Horse: [coined By MIT-hacker-turned-NSA-spook Dan Edwards] N.". Retrieved 2012-04-05.

[6] "What is the difference between viruses, worms, and Trojans?". Symantec Corporation. Retrieved 2009-01-10.

[7] "VIRUS-L/comp.virus Frequently Asked Questions (FAQ) v2.00 (Question B3: What is a Trojan Horse?)". 9 October 1995. Retrieved 2012-09-13.

[8] "Hackers, Spyware and Trojans – What You Need to Know". Comodo. Retrieved September 5, 2015.

[9] Robert McMillan (2013): Trojan Turns Your PC Into Bitcoin Mining Slave, Retrieved on 2015-02-01

[10] Jamie Crapanzano (2003): "Deconstructing SubSeven, the Trojan Horse of Choice", SANS Institute, Retrieved on 2009-06-11

[11] Vincentas (11 July 2013). "Trojan Horse in SpyWareLoop.com". Spyware Loop. Retrieved 28 July 2013.

[12] Basil Cupa, Trojan Horse Resurrected: On the Legality of the Use of Government Spyware (Govware), LISS 2013, pp. 419–428

[13] "Dokument nicht gefunden!". Federal Department of Justice and Police. Archived from the original on May 6, 2013.

[14] "Swiss coder publicises government spy Trojan – Techworld.com". News.techworld.com. Retrieved 2014-01-26.

[15] BitDefender.com Malware and Spam Survey

[16] Datta, Ganesh. "What are Trojans?". *SecurAid*.

[17] https://sourceforge.net/projects/mega-panzer/

[18] https://sourceforge.net/projects/mini-panzer/

[19] https://blog.lookout.com/blog/2015/11/19/shedun-trojanized-adware/

[20] http://www.theinquirer.net/inquirer/news/2435721/shedun-trojan-adware-is-hitting-the-android-accessibility-service

[21] https://blog.lookout.com/blog/2015/11/04/trojanized-adware/

[22] http://betanews.com/2015/11/05/shuanet-shiftybug-and-shedun-malware-could-auto-root-your-android/

[23] http://www.techtimes.com/articles/104373/20151109/new-family-of-android-malware-virtually-impossible-to-remove-say-hello-to-htm

[24] http://arstechnica.com/security/2015/11/android-adware-can-install-itself-even-when-users-explicitly-reject-it/

3.1.5 External links

- Trojan Horses at DMOZ

3.2 Rootkit

A **rootkit** is a collection of computer software, typically malicious, designed to enable access to a computer or areas of its software that would not otherwise be allowed (for example, to an unauthorized user) while at the same time masking its existence or the existence of other software.[1] The term *rootkit* is a concatenation of "root" (the traditional name of the privileged account on Unix-like operating systems) and the word "kit" (which refers to the software components that implement the tool). The term "rootkit" has negative connotations through its association with malware.[1]

Rootkit installation can be automated, or an attacker can install it once they've obtained root or Administrator access. Obtaining this access is a result of direct attack on a system

(i.e.), exploiting a known vulnerability (such as privilege escalation) or a password (obtained by cracking or social engineering). Once installed, it becomes possible to hide the intrusion as well as to maintain privileged access. The key is the root or administrator access. Full control over a system means that existing software can be modified, including software that might otherwise be used to detect or circumvent it.

Rootkit detection is difficult because a rootkit may be able to subvert the software that is intended to find it. Detection methods include using an alternative and trusted operating system, behavioral-based methods, signature scanning, difference scanning, and memory dump analysis. Removal can be complicated or practically impossible, especially in cases where the rootkit resides in the kernel; reinstallation of the operating system may be the only available solution to the problem.[2] When dealing with firmware rootkits, removal may require hardware replacement, or specialized equipment.

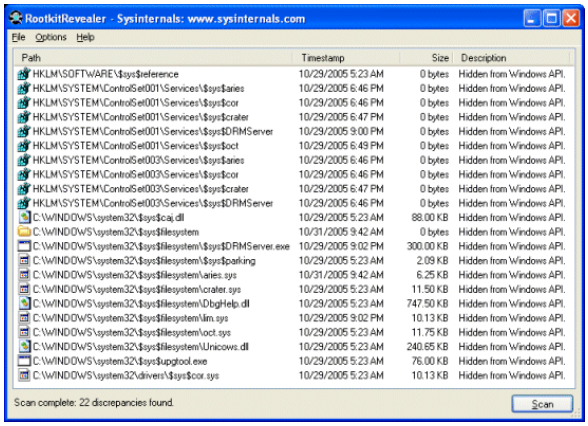
3.2.1 History

The term *rootkit* or *root kit* originally referred to a maliciously modified set of administrative tools for a Unix-like operating system that granted "root" access.[3] If an intruder could replace the standard administrative tools on a system with a rootkit, the intruder could obtain root access over the system whilst simultaneously concealing these activities from the legitimate system administrator. These first-generation rootkits were trivial to detect by using tools such as Tripwire that had not been compromised to access the same information.[4][5] Lane Davis and Steven Dake wrote the earliest known rootkit in 1990 for Sun Microsystems' SunOS UNIX operating system.[6] In the lecture he gave upon receiving the Turing award in 1983, Ken Thompson of Bell Labs, one of the creators of Unix, theorized about subverting the C compiler in a Unix distribution and discussed the exploit. The modified compiler would detect attempts to compile the Unix login command and generate altered code that would accept not only the user's correct password, but an additional "backdoor" password known to the attacker. Additionally, the compiler would detect attempts to compile a new version of the compiler, and would insert the same exploits into the new compiler. A review of the source code for the login command or the updated compiler would not reveal any malicious code.[7] This exploit was equivalent to a rootkit.

The first documented computer virus to target the personal computer, discovered in 1986, used cloaking techniques to hide itself: the Brain virus intercepted attempts to read the boot sector, and redirected these to elsewhere on the disk, where a copy of the original boot sector was kept.[1] Over time, DOS-virus cloaking methods became more sophisticated, with advanced techniques including the hooking of low-level disk INT 13H BIOS interrupt calls to hide unauthorized modifications to files.[1]

The first malicious rootkit for the Windows NT operating system appeared in 1999: a trojan called *NTRootkit* created by Greg Hoglund.[8] It was followed by *HackerDefender* in 2003.[1] The first rootkit targeting Mac OS X appeared in 2009,[9] while the Stuxnet worm was the first to target programmable logic controllers (PLC).[10]

Sony BMG copy protection rootkit scandal



Screenshot of RootkitRevealer, showing the files hidden by the Extended Copy Protection rootkit

Main article: Sony BMG copy protection rootkit scandal

In 2005, Sony BMG published CDs with copy protection and digital rights management software called Extended Copy Protection, created by software company First 4 Internet. The software included a music player but silently installed a rootkit which limited the user's ability to access the CD.[11]

Software engineer Mark Russinovich, who created the rootkit detection tool RootkitRevealer, discovered the rootkit on one of his computers.[1] The ensuing scandal raised the public's awareness of rootkits.[12]

To cloak itself, the rootkit hid from the user any file starting with "sys". Soon after Russinovich's report, malware appeared which took advantage of that vulnerability of affected systems.[1]

One BBC analyst called it a "public relations nightmare."[13] Sony BMG released patches to uninstall the rootkit, but it exposed users to an even more serious vulnerability.[14] The company eventually recalled the CDs. In the United States, a class-action lawsuit was brought against Sony BMG.[15]

Greek wiretapping case 2004–05

Main article: Greek wiretapping case 2004–05

The **Greek wiretapping case of 2004-05**, also referred to as **Greek Watergate**,[16] involved the illegal tapping of more than 100 mobile phones on the Vodafone Greece network belonging mostly to members of the Greek government and top-ranking civil servants. The taps began sometime near the beginning of August 2004 and were removed in March 2005 without discovering the identity of the perpetrators.

The intruders installed a rootkit targeting Ericsson's AXE telephone exchange. According to IEEE Spectrum, this was "the first time a rootkit has been observed on a special-purpose system, in this case an Ericsson telephone switch."[17] The rootkit was designed to patch the memory of the exchange while it was running, enable wiretapping while disabling audit logs, patch the commands that list active processes and active data blocks, and modify the data block checksum verification command. A backdoor allowed an operator with sysadmin status to deactivate the exchange's transaction log and alarms and access commands related to the surveillance capability.[17] The rootkit was discovered after the intruders installed a faulty update, which caused SMS texts to be undelivered, leading to an automated failure report being generated. Ericsson engineers were called in to investigate the fault and discovered the hidden data blocks containing the list of phone numbers being monitored, along with the rootkit and illicit monitoring software.

3.2.2 Uses

Modern rootkits do not elevate access,[3] but rather are used to make another software payload undetectable by adding stealth capabilities.[8] Most rootkits are classified as malware, because the payloads they are bundled with are malicious. For example, a payload might covertly steal user passwords, credit card information, computing resources, or conduct other unauthorized activities. A small number of rootkits may be considered utility applications by their users: for example, a rootkit might cloak a CD-ROM-emulation driver, allowing video game users to defeat anti-piracy measures that require insertion of the original installation media into a physical optical drive to verify that the software was legitimately purchased.

Rootkits and their payloads have many uses:

- Provide an attacker with full access via a backdoor, permitting unauthorized access to, for example, steal or falsify documents. One of the ways to carry this out is to subvert the login mechanism, such as the /bin/login program on Unix-like systems or GINA on Windows. The replacement appears to function normally, but also accepts a secret login combination that allows an attacker direct access to the system with administrative privileges, bypassing standard authentication and authorization mechanisms.
- Conceal other malware, notably password-stealing key loggers and computer viruses.[18]
- Appropriate the compromised machine as a zombie computer for attacks on other computers. (The attack originates from the compromised system or network, instead of the attacker's system.) "Zombie" computers are typically members of large botnets that can launch denial-of-service attacks, distribute e-mail spam, conduct click fraud, etc.
- Enforcement of digital rights management (DRM).

In some instances, rootkits provide desired functionality, and may be installed intentionally on behalf of the computer user:

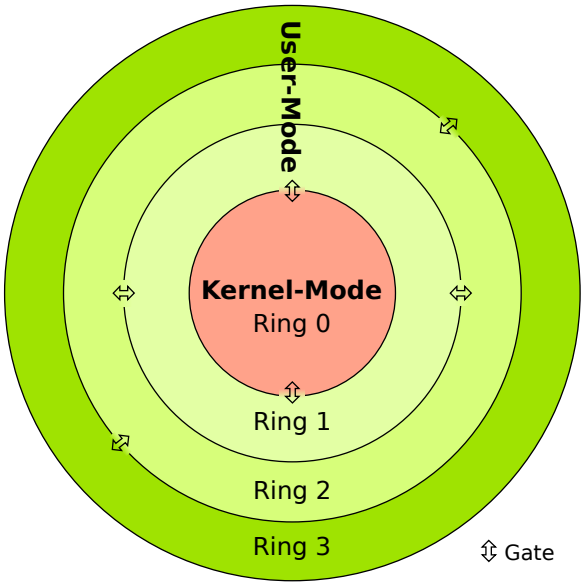
- Conceal cheating in online games from software like Warden.[19]
- Detect attacks, for example, in a honeypot.[20]
- Enhance emulation software and security software.[21] Alcohol 120% and Daemon Tools are commercial examples of non-hostile rootkits used to defeat copy-protection mechanisms such as SafeDisc and SecuROM. Kaspersky antivirus software also uses techniques resembling rootkits to protect itself from malicious actions. It loads its own drivers to intercept system activity, and then prevents other processes from doing harm to itself. Its processes are not hidden, but cannot be terminated by standard methods (It can be terminated with Process Hacker).
- Anti-theft protection: Laptops may have BIOS-based rootkit software that will periodically report to a central authority, allowing the laptop to be monitored, disabled or wiped of information in the event that it is stolen.[22]
- Bypassing Microsoft Product Activation[23]

3.2.3 Types

Further information: Ring (computer security)

There are at least five types of rootkit, ranging from those at the lowest level in firmware (with the highest privileges), through to the least privileged user-based variants that operate in Ring 3. Hybrid combinations of these may occur spanning, for example, user mode and kernel mode.[24]

User mode



Computer security rings (Note that Ring -1 is not shown)

User-mode rootkits run in Ring 3, along with other applications as user, rather than low-level system processes.[25] They have a number of possible installation vectors to intercept and modify the standard behavior of application programming interfaces (APIs). Some inject a dynamically linked library (such as a .DLL file on Windows, or a .dylib file on Mac OS X) into other processes, and are thereby able to execute inside any target process to spoof it; others with sufficient privileges simply overwrite the memory of a target application. Injection mechanisms include:[25]

- Use of vendor-supplied application extensions. For example, Windows Explorer has public interfaces that allow third parties to extend its functionality.
- Interception of messages.
- Debuggers.
- Exploitation of security vulnerabilities.
- Function hooking or patching of commonly used APIs, for example, to hide a running process or file that resides on a filesystem.[26]

> ...since user mode applications all run in their own memory space, the rootkit needs to perform this patching in the memory space of every running application. In addition, the rootkit needs to monitor the system for any new applications that execute and patch those programs' memory space before they fully execute.
> — Windows Rootkit Overview, Symantec[3]

Kernel mode

Kernel-mode rootkits run with the highest operating system privileges (Ring 0) by adding code or replacing portions of the core operating system, including both the kernel and associated device drivers. Most operating systems support kernel-mode device drivers, which execute with the same privileges as the operating system itself. As such, many kernel-mode rootkits are developed as device drivers or loadable modules, such as loadable kernel modules in Linux or device drivers in Microsoft Windows. This class of rootkit has unrestricted security access, but is more difficult to write.[27] The complexity makes bugs common, and any bugs in code operating at the kernel level may seriously impact system stability, leading to discovery of the rootkit.[27] One of the first widely known kernel rootkits was developed for Windows NT 4.0 and released in Phrack magazine in 1999 by Greg Hoglund.[28][29][30]

Kernel rootkits can be especially difficult to detect and remove because they operate at the same security level as the operating system itself, and are thus able to intercept or subvert the most trusted operating system operations. Any software, such as antivirus software, running on the compromised system is equally vulnerable.[31] In this situation, no part of the system can be trusted.

A rootkit can modify data structures in the Windows kernel using a method known as *direct kernel object manipulation* (DKOM).[32] This method can be used to hide processes. A kernel mode rootkit can also hook the System Service Descriptor Table (SSDT), or modify the gates between user mode and kernel mode, in order to cloak itself.[3] Similarly for the Linux operating system, a rootkit can modify the *system call table* to subvert kernel functionality.[33] It's common that a rootkit creates a hidden, encrypted filesystem in which it can hide other malware or original copies of files it has infected.[34]

Operating systems are evolving to counter the threat of kernel-mode rootkits. For example, 64-bit editions of Microsoft Windows now implement mandatory signing of all kernel-level drivers in order to make it more difficult for untrusted code to execute with the highest privileges in a

system.[35]

Bootkits A kernel-mode rootkit variant called a **bootkit** can infect startup code like the Master Boot Record (MBR), Volume Boot Record (VBR) or boot sector, and in this way, can be used to attack full disk encryption systems. An example is the "Evil Maid Attack", in which an attacker installs a bootkit on an unattended computer, replacing the legitimate boot loader with one under their control. Typically the malware loader persists through the transition to protected mode when the kernel has loaded, and is thus able to subvert the kernel.[36][37][38][39] For example, the "Stoned Bootkit" subverts the system by using a compromised boot loader to intercept encryption keys and passwords.[40] More recently, the Alureon rootkit has successfully subverted the requirement for 64-bit kernel-mode driver signing in Windows 7 by modifying the master boot record.[41] Although not malware in the sense of doing something the user doesn't want, certain "Vista Loader" or "Windows Loader" software works in a similar way by injecting an ACPI SLIC (System Licensed Internal Code) table in the RAM-cached version of the BIOS during boot, in order to defeat the Windows Vista and Windows 7 activation process.[42][43] This vector of attack was rendered useless in the (non-server) versions of Windows 8, which use a unique, machine-specific key for each system, that can only be used by that one machine.[44]

The only known defenses against bootkit attacks are the prevention of unauthorized physical access to the system—a problem for portable computers—or the use of a Trusted Platform Module configured to protect the boot path.[45]

Hypervisor level

Rootkits have been created as Type II Hypervisors in academia as proofs of concept. By exploiting hardware virtualization features such as Intel VT or AMD-V, this type of rootkit runs in Ring −1 and hosts the target operating system as a virtual machine, thereby enabling the rootkit to intercept hardware calls made by the original operating system.[5] Unlike normal hypervisors, they do not have to load before the operating system, but can load into an operating system before promoting it into a virtual machine.[5] A hypervisor rootkit does not have to make any modifications to the kernel of the target to subvert it; however, that does not mean that it cannot be detected by the guest operating system. For example, timing differences may be detectable in CPU instructions.[5] The "SubVirt" laboratory rootkit, developed jointly by Microsoft and University of Michigan researchers, is an academic example of a virtual machine–based rootkit (VMBR),[46] while Blue Pill is another.

In 2009, researchers from Microsoft and North Carolina State University demonstrated a hypervisor-layer anti-rootkit called Hooksafe, which provides generic protection against kernel-mode rootkits.[47]

Windows 10 introduces a new feature called "Device Guard", that takes advantage of virtualization to provide independent external protection of an operating system against rootkit-type malware.[48]

Firmware and hardware

A firmware rootkit uses device or platform firmware to create a persistent malware image in hardware, such as a router, network card,[49] hard drive, or the system BIOS.[25][50] The rootkit hides in firmware, because firmware is not usually inspected for code integrity. John Heasman demonstrated the viability of firmware rootkits in both ACPI firmware routines[51] and in a PCI expansion card ROM.[52]

In October 2008, criminals tampered with European credit-card-reading machines before they were installed. The devices intercepted and transmitted credit card details via a mobile phone network.[53] In March 2009, researchers Alfredo Ortega and Anibal Sacco published details of a BIOS-level Windows rootkit that was able to survive disk replacement and operating system re-installation.[54][55][56] A few months later they learned that some laptops are sold with a legitimate rootkit, known as Absolute CompuTrace or Absolute LoJack for Laptops, preinstalled in many BIOS images. This is an anti-theft technology system that researchers showed can be turned to malicious purposes.[22]

Intel Active Management Technology, part of Intel vPro, implements out-of-band management, giving administrators remote administration, remote management, and remote control of PCs with no involvement of the host processor or BIOS, even when the system is powered off. Remote administration includes remote power-up and power-down, remote reset, redirected boot, console redirection, pre-boot access to BIOS settings, programmable filtering for inbound and outbound network traffic, agent presence checking, out-of-band policy-based alerting, access to system information, such as hardware asset information, persistent event logs, and other information that is stored in dedicated memory (not on the hard drive) where it is accessible even if the OS is down or the PC is powered off. Some of these functions require the deepest level of rootkit, a second non-removable spy computer built around the main computer. Sandy Bridge and future chipsets have "the ability to remotely kill and restore a lost or stolen PC via 3G". Hardware rootkits built into the chipset can help recover stolen computers, remove data, or render them useless, but they also present privacy and security concerns of unde-

tectable spying and redirection by management or hackers who might gain control.

3.2.4 Installation and cloaking

Rootkits employ a variety of techniques to gain control of a system; the type of rootkit influences the choice of attack vector. The most common technique leverages security vulnerabilities to achieve surreptitious privilege escalation. Another approach is to use a Trojan horse, deceiving a computer user into trusting the rootkit's installation program as benign—in this case, social engineering convinces a user that the rootkit is beneficial.[27] The installation task is made easier if the principle of least privilege is not applied, since the rootkit then does not have to explicitly request elevated (administrator-level) privileges. Other classes of rootkits can be installed only by someone with physical access to the target system. Some rootkits may also be installed intentionally by the owner of the system or somebody authorized by the owner, e.g. for the purpose of employee monitoring, rendering such subversive techniques unnecessary.[57]

The installation of malicious rootkits is commercially driven, with a pay-per-install (PPI) compensation method typical for distribution.[58][59]

Once installed, a rootkit takes active measures to obscure its presence within the host system through subversion or evasion of standard operating system security tools and APIs used for diagnosis, scanning, and monitoring. Rootkits achieve this by modifying the behavior of core parts of an operating system through loading code into other processes, the installation or modification of drivers, or kernel modules. Obfuscation techniques include concealing running processes from system-monitoring mechanisms and hiding system files and other configuration data.[60] It is not uncommon for a rootkit to disable the event logging capacity of an operating system, in an attempt to hide evidence of an attack. Rootkits can, in theory, subvert *any* operating system activities.[61] The "perfect rootkit" can be thought of as similar to a "perfect crime": one that nobody realizes has taken place.

Rootkits also take a number of measures to ensure their survival against detection and cleaning by antivirus software in addition to commonly installing into Ring 0 (kernel-mode), where they have complete access to a system. These include polymorphism, stealth techniques, regeneration, disabling anti-malware software.[62] and not installing on virtual machines where it may be easier for researchers to discover and analyze them.

3.2.5 Detection

The fundamental problem with rootkit detection is that if the operating system has been subverted, particularly by a kernel-level rootkit, it cannot be trusted to find unauthorized modifications to itself or its components.[61] Actions such as requesting a list of running processes, or a list of files in a directory, cannot be trusted to behave as expected. In other words, rootkit detectors that work while running on infected systems are only effective against rootkits that have some defect in their camouflage, or that run with lower user-mode privileges than the detection software in the kernel.[27] As with computer viruses, the detection and elimination of rootkits is an ongoing struggle between both sides of this conflict.[61]

Detection can take a number of different approaches, including signatures (e.g. antivirus software), integrity checking (e.g. digital signatures), difference-based detection (comparison of expected vs. actual results), and behavioral detection (e.g. monitoring CPU usage or network traffic). For kernel-mode rootkits, detection is considerably more complex, requiring careful scrutiny of the System Call Table to look for hooked functions where the malware may be subverting system behavior,[63] as well as forensic scanning of memory for patterns that indicate hidden processes.

Unix rootkit detection offerings include Zeppoo,[64] chkrootkit, rkhunter and OSSEC. For Windows, detection tools include Microsoft Sysinternals RootkitRevealer,[65] Avast! Antivirus, Sophos Anti-Rootkit,[66] F-Secure,[67] Radix,[68] GMER,[69] and WindowsSCOPE. Any rootkit detectors that prove effective ultimately contribute to their own ineffectiveness, as malware authors adapt and test their code to escape detection by well-used tools.[Notes 1]

Detection by examining storage while the suspect operating system is not operational can miss rootkits not recognised by the checking software, as the rootkit is not active and suspicious behavior is suppressed; conventional anti-malware software running with the rootkit operational may fail if the rootkit hides itself effectively.

Alternative trusted medium

The best and most reliable method for operating-system-level rootkit detection is to shut down the computer suspected of infection, and then to check its storage by booting from an alternative trusted medium (e.g. a rescue CD-ROM or USB flash drive).[70] The technique is effective because a rootkit cannot actively hide its presence if it is not running.

Behavioral-based

The behavioral-based approach to detecting rootkits attempts to infer the presence of a rootkit by looking for rootkit-like behavior. For example, by profiling a system, differences in the timing and frequency of API calls or in overall CPU utilization can be attributed to a rootkit. The method is complex and is hampered by a high incidence of false positives. Defective rootkits can sometimes introduce very obvious changes to a system: the Alureon rootkit crashed Windows systems after a security update exposed a design flaw in its code.[71][72]

Logs from a packet analyzer, firewall, or intrusion prevention system may present evidence of rootkit behaviour in a networked environment.[24]

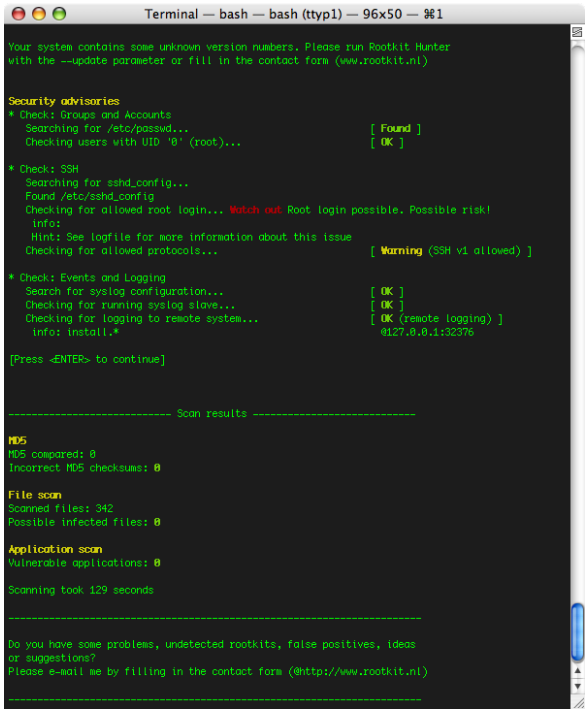
Signature-based

Antivirus products rarely catch all viruses in public tests (depending on what is used and to what extent), even though security software vendors incorporate rootkit detection into their products. Should a rootkit attempt to hide during an antivirus scan, a stealth detector may notice; if the rootkit attempts to temporarily unload itself from the system, signature detection (or "fingerprinting") can still find it. This combined approach forces attackers to implement counterattack mechanisms, or "retro" routines, that attempt to terminate antivirus programs. Signature-based detection methods can be effective against well-published rootkits, but less so against specially crafted, custom-root rootkits.[61]

Difference-based

Another method that can detect rootkits compares "trusted" raw data with "tainted" content returned by an API. For example, binaries present on disk can be compared with their copies within operating memory (in some operating systems, the in-memory image should be identical to the on-disk image), or the results returned from file system or Windows Registry APIs can be checked against raw structures on the underlying physical disks[61][73]—however, in the case of the former, some valid differences can be introduced by operating system mechanisms like memory relocation or shimming. A rootkit may detect the presence of a such difference-based scanner or virtual machine (the latter being commonly used to perform forensic analysis), and adjust its behaviour so that no differences can be detected. Difference-based detection was used by Russinovich's *RootkitRevealer* tool to find the Sony DRM rootkit.[1]

Integrity checking



The rkhunter utility uses SHA-1 hashes to verify the integrity of system files.

Code signing uses public-key infrastructure to check if a file has been modified since being digitally signed by its publisher. Alternatively, a system owner or administrator can use a cryptographic hash function to compute a "fingerprint" at installation time that can help to detect subsequent unauthorized changes to on-disk code libraries.[74] However, unsophisticated schemes check only whether the code has been modified since installation time; subversion prior to that time is not detectable. The fingerprint must be re-established each time changes are made to the system: for example, after installing security updates or a service pack. The hash function creates a *message digest*, a relatively short code calculated from each bit in the file using an algorithm that creates large changes in the message digest with even smaller changes to the original file. By recalculating and comparing the message digest of the installed files at regular intervals against a trusted list of message digests, changes in the system can be detected and monitored—as long as the original baseline was created before the malware was added. More-sophisticated rootkits are able to subvert the verification process by presenting an unmodified copy of the file for inspection, or by making code modifications only in memory, rather than on disk. The technique may therefore be effective only against unsophisticated rootkits—for example, those that replace Unix binaries like "ls" to hide the presence of a file.

Similarly, detection in firmware can be achieved by computing a cryptographic hash of the firmware and comparing it to a whitelist of expected values, or by extending the hash value into Trusted Platform Module (TPM) configuration registers, which are later compared to a whitelist of expected values.[75] The code that performs hash, compare, or extend operations must also be protected—in this context, the notion of an *immutable root-of-trust* holds that the very first code to measure security properties of a system must itself be trusted to ensure that a rootkit or bootkit does not compromise the system at its most fundamental level.[76]

Memory dumps

Forcing a complete dump of virtual memory will capture an active rootkit (or a kernel dump in the case of a kernel-mode rootkit), allowing offline forensic analysis to be performed with a debugger against the resulting dump file, without the rootkit being able to take any measures to cloak itself. This technique is highly specialized, and may require access to non-public source code or debugging symbols. Memory dumps initiated by the operating system cannot always be used to detect a hypervisor-based rootkit, which is able to intercept and subvert the lowest-level attempts to read memory[5]—a hardware device, such as one that implements a non-maskable interrupt, may be required to dump memory in this scenario.[77][78] Virtual machines also make it easier to analyze the memory of a compromised machine from the underlying hypervisor, so some rootkits will avoid infecting virtual machines for this reason.

3.2.6 Removal

Manual removal of a rootkit is often too difficult for a typical computer user,[25] but a number of security-software vendors offer tools to automatically detect and remove some rootkits, typically as part of an antivirus suite. As of 2005, Microsoft's monthly Windows Malicious Software Removal Tool is able to detect and remove some classes of rootkits.[79][80] Some antivirus scanners can bypass file system APIs, which are vulnerable to manipulation by a rootkit. Instead, they access raw filesystem structures directly, and use this information to validate the results from the system APIs to identify any differences that may be caused by a rootkit.[Notes 2][81][82][83][84]

There are experts who believe that the only reliable way to remove them is to re-install the operating system from trusted media.[85][86] This is because antivirus and malware removal tools running on an untrusted system may be ineffective against well-written kernel-mode rootkits. Booting an alternative operating system from trusted media can allow an infected system volume to be mounted and potentially safely cleaned and critical data to be copied off—or, alternatively, a forensic examination performed.[24] Lightweight operating systems such as Windows PE, Windows Recovery Console, Windows Recovery Environment, BartPE, or Live Distros can be used for this purpose, allowing the system to be cleaned.

Even if the type and nature of a rootkit is known, manual repair may be impractical, while re-installing the operating system and applications is safer, simpler and quicker.[85]

3.2.7 Public availability

Like much malware used by attackers, many rootkit implementations are shared and are easily available on the Internet. It is not uncommon to see a compromised system in which a sophisticated, publicly available rootkit hides the presence of unsophisticated worms or attack tools apparently written by inexperienced programmers.[24]

Most of the rootkits available on the Internet originated as exploits or as academic "proofs of concept" to demonstrate varying methods of hiding things within a computer system and of taking unauthorized control of it.[87] Often not fully optimized for stealth, such rootkits sometimes leave unintended evidence of their presence. Even so, when such rootkits are used in an attack, they are often effective. Other rootkits with keylogging features such as GameGuard are installed as part of online commercial games.

3.2.8 Defenses

System hardening represents one of the first layers of defence against a rootkit, to prevent it from being able to install.[88] Applying security patches, implementing the principle of least privilege, reducing the attack surface and installing antivirus software are some standard security best practices that are effective against all classes of malware.[89]

New secure boot specifications like Unified Extensible Firmware Interface have been designed to address the threat of bootkits, but even these are vulnerable if the security features they offer are not utilized.[50]

For server systems, remote server attestation using technologies such as Intel Trusted Execution Technology (TXT) provide a way of validating that servers remain in a known good state. For example, Microsoft Bitlocker encrypting data-at-rest validates servers are in a known "good state" on bootup. PrivateCore vCage is a software offering that secures data-in-use (memory) to avoid bootkits and rootkits by validating servers are in a known "good" state on bootup. The PrivateCore implementation works in concert with Intel TXT and locks down server system interfaces to avoid potential bootkits and rootkits.

3.2.9 See also

- Computer security conference
- Host-based intrusion detection system
- Man-in-the-middle attack
- *The Rootkit Arsenal: Escape and Evasion in the Dark Corners of the System*

3.2.10 Notes

[1] The process name of Sysinternals RootkitRevealer was targeted by malware; in an attempt to counter this countermeasure, the tool now uses a randomly generated process name.

[2] In theory, a sufficiently sophisticated kernel-level rootkit could subvert read operations against raw filesystem data structures as well, so that they match the results returned by APIs.

3.2.11 References

[1] "Rootkits, Part 1 of 3: The Growing Threat" (PDF). McAfee. 2006-04-17. Archived from the original (PDF) on 2006-08-23.

[2] http://www.technibble.com/how-to-remove-a-rootkit-from-a-windows-system/

[3] "Windows Rootkit Overview" (PDF). Symantec. 2006-03-26. Retrieved 2010-08-17.

[4] Sparks, Sherri; Butler, Jamie (2005-08-01). "Raising The Bar For Windows Rootkit Detection". *Phrack* **0xb** (0x3d).

[5] Myers, Michael; Youndt, Stephen (2007-08-07). "An Introduction to Hardware-Assisted Virtual Machine (HVM) Rootkits". Crucial Security. CiteSeerX: 10.1.1.90.8832.

[6] Andrew Hay; Daniel Cid; Rory Bray (2008). *OSSEC Host-Based Intrusion Detection Guide*. Syngress. p. 276. ISBN 1-59749-240-X.

[7] Thompson, Ken (August 1984). "Reflections on Trusting Trust" (PDF). *Communications of the ACM* **27** (8): 761. doi:10.1145/358198.358210.

[8] Greg Hoglund; James Butler (2006). *Rootkits: Subverting the Windows kernel*. Addison-Wesley. p. 4. ISBN 0-321-29431-9.

[9] Dai Zovi, Dino (2009-07-26). *Advanced Mac OS X Rootkits* (PDF). Blackhat. Endgame Systems. Retrieved 2010-11-23.

[10] "Stuxnet Introduces the First Known Rootkit for Industrial Control Systems". Symantec. 2010-08-06. Retrieved 2010-12-04.

[11] "Spyware Detail: XCP.Sony.Rootkit". Computer Associates. 2005-11-05. Archived from the original on 2010-08-18. Retrieved 2010-08-19.

[12] Russinovich, Mark (2005-10-31). "Sony, Rootkits and Digital Rights Management Gone Too Far". *TechNet Blogs*. Microsoft. Retrieved 2010-08-16.

[13] "Sony's long-term rootkit CD woes". BBC News. 2005-11-21. Retrieved 2008-09-15.

[14] Felton, Ed (2005-11-15). "Sony's Web-Based Uninstaller Opens a Big Security Hole; Sony to Recall Discs".

[15] Knight, Will (2005-11-11). "Sony BMG sued over cloaking software on music CD". *New Scientist* (Sutton, UK: Reed Business Information). Retrieved 2010-11-21.

[16] Kyriakidou, Dina (March 2, 2006). ""Greek Watergate" Scandal Sends Political Shockwaves". Reuters. Retrieved 2007-11-24.

[17] Vassilis Prevelakis; Diomidis Spinellis (July 2007). "The Athens Affair".

[18] Russinovich, Mark (June 2005). "Unearthing Root Kits". Windows IT Pro. Retrieved 2010-12-16.

[19] "World of Warcraft Hackers Using Sony BMG Rootkit". The Register. 2005-11-04. Retrieved 2010-08-23.

[20] Steve Hanna (September 2007). "Using Rootkit Technology for Honeypot-Based Malware Detection" (PDF). CCEID Meeting.

[21] Russinovich, Mark (6 February 2006). "Using Rootkits to Defeat Digital Rights Management". *Winternals*. SysInternals. Archived from the original on 31 August 2006. Retrieved 2006-08-13.

[22] Ortega, Alfredo; Sacco, Anibal (2009-07-24). *Deactivate the Rootkit: Attacks on BIOS anti-theft technologies* (PDF). Black Hat USA 2009 (PDF). Boston, MA: Core Security Technologies. Retrieved 2014-06-12.

[23] Kleissner, Peter (2009-09-02). "Stoned Bootkit: The Rise of MBR Rootkits & Bootkits in the Wild" (PDF). Retrieved 2010-11-23.

[24] Anson, Steve; Bunting, Steve (2007). *Mastering Windows Network Forensics and Investigation*. John Wiley and Sons. pp. 73–74. ISBN 0-470-09762-0.

[25] "Rootkits Part 2: A Technical Primer" (PDF). McAfee. 2007-04-03. Archived from the original (PDF) on 2008-12-05. Retrieved 2010-08-17.

[26] Kdm. "NTIllusion: A portable Win32 userland rootkit". *Phrack* **62** (12).

[27] "Understanding Anti-Malware Technologies" (PDF). Microsoft. 2007-02-21. Retrieved 2010-08-17.

[28] Hoglund, Greg (1999-09-09). "A *REAL* NT Rootkit, Patching the NT Kernel". *Phrack* **9** (55). Retrieved 2010-11-21.

[29] Shevchenko, Alisa (2008-09-01). "Rootkit Evolution". *Help Net Security*. Help Net Security.

[30] Chuvakin, Anton (2003-02-02). An Overview of Unix Rootkits (PDF) (Report). Chantilly, Virginia: iDEFENSE. Retrieved 2010-11-21.

[31] Butler, James; Sparks, Sherri (2005-11-16). "Windows Rootkits of 2005, Part Two". *Symantec Connect*. Symantec. Retrieved 2010-11-13.

[32] Butler, James; Sparks, Sherri (2005-11-03). "Windows Rootkits of 2005, Part One". *Symantec Connect*. Symantec. Retrieved 2010-11-12.

[33] Burdach, Mariusz (2004-11-17). "Detecting Rootkits And Kernel-level Compromises In Linux". Symantec. Retrieved 2010-11-23.

[34] Marco Giuliani (11 April 2011). "ZeroAccess – An Advanced Kernel Mode Rootkit" (PDF). Webroot Software. Retrieved 10 August 2011.

[35] "Driver Signing Requirements for Windows". Microsoft. Retrieved 2008-07-06.

[36] Soeder, Derek; Permeh, Ryan (2007-05-09). "Bootroot". eEye Digital Security. Archived from the original on 2013-08-17. Retrieved 2010-11-23.

[37] Schneier, Bruce (2009-10-23). "'Evil Maid' Attacks on Encrypted Hard Drives". Retrieved 2009-11-07.

[38] Kumar, Nitin; Kumar, Vipin (2007). *Vbootkit: Compromising Windows Vista Security* (PDF). Black Hat Europe 2007.

[39] "BOOT KIT: Custom boot sector based Windows 2000/XP/2003 Subversion". *NVlabs*. 2007-02-04. Archived from the original on June 10, 2010. Retrieved 2010-11-21.

[40] Kleissner, Peter (2009-10-19). "Stoned Bootkit". Peter Kleissner. Retrieved 2009-11-07.

[41] Goodin, Dan (2010-11-16). "World's Most Advanced Rootkit Penetrates 64-bit Windows". The Register. Retrieved 2010-11-22.

[42] Peter Kleissner, "The Rise of MBR Rootkits And Bootkits in the Wild", Hacking at Random (2009) - text; slides

[43] Windows Loader - Software Informer. This is the loader application that's used by millions of people worldwide

[44] Microsoft tightens grip on OEM Windows 8 licensing

[45] Scambray, Joel; McClure, Stuart (2007). *Hacking Exposed Windows: Windows Security Secrets & Solutions*. McGraw-Hill Professional. pp. 371–372. ISBN 0-07-149426-X.

[46] King, Samuel T.; Chen, Peter M.; Wang, Yi-Min; Verbowski, Chad; Wang, Helen J.; Lorch, Jacob R. (2006-04-03). International Business Machines (ed.), ed. *SubVirt: Implementing malware with virtual machines* (PDF). 2006 IEEE Symposium on Security and Privacy. Institute of Electrical and Electronics Engineers. doi:10.1109/SP.2006.38. ISBN 0-7695-2574-1. Retrieved 2008-09-15.

[47] Wang, Zhi; Jiang, Xuxian; Cui, Weidong; Ning, Peng (2009-08-11). "Countering Kernel Rootkits with Lightweight Hook Protection" (PDF). In Al-Shaer, Ehab (General Chair). *Proceedings of the 16th ACM Conference on Computer and Communications Security*. CCS 2009: 16th ACM Conference on Computer and Communications Security. Jha, Somesh; Keromytis, Angelos D. (Program Chairs). New York: ACM New York. doi:10.1145/1653662.1653728. ISBN 978-1-60558-894-0. Retrieved 2009-11-11.

[48] https://msdn.microsoft.com/en-us/library/dn986865(v=vs.85).aspx

[49] Delugré, Guillaume (2010-11-21). *Reversing the Broacom NetExtreme's Firmware* (PDF). hack.lu. Sogeti. Retrieved 2010-11-25.

[50] http://blog.trendmicro.com/trendlabs-security-intelligence/hacking-team-uses-uefi-bios-rootkit-to-keep-rcs-9-agent-in-target-systems/

[51] Heasman, John (2006-01-25). *Implementing and Detecting an ACPI BIOS Rootkit* (PDF). Black Hat Federal 2006. NGS Consulting. Retrieved 2010-11-21.

[52] Heasman, John (2006-11-15). "Implementing and Detecting a PCI Rootkit" (PDF). Next Generation Security Software. CiteSeerX: 10.1.1.89.7305. Retrieved 2010-11-13.

[53] Modine, Austin (2008-10-10). "Organized crime tampers with European card swipe devices: Customer data beamed overseas". *The Register*. Situation Publishing. Retrieved 2008-10-13.

[54] Sacco, Anibal; Ortéga, Alfredo (2009). *Persistent BIOS infection* (PDF). CanSecWest 2009. Core Security Technologies. Retrieved 2010-11-21.

[55] Goodin, Dan (2009-03-24). "Newfangled rootkits survive hard disk wiping". *The Register*. Situation Publishing. Retrieved 2009-03-25.

[56] Sacco, Anibal; Ortéga, Alfredo (2009-06-01). "Persistent BIOS Infection: The Early Bird Catches the Worm". *Phrack* **66** (7). Retrieved 2010-11-13.

[57] Ric Vieler (2007). *Professional Rootkits*. John Wiley & Sons. p. 244. ISBN 9780470149546.

[58] Matrosov, Aleksandr; Rodionov, Eugene (2010-06-25). "TDL3: The Rootkit of All Evil?" (PDF). Moscow: ESET. p. 3. Retrieved 2010-08-17.

[59] Matrosov, Aleksandr; Rodionov, Eugene (2011-06-27). "The Evolution of TDL: Conquering x64" (PDF). ESET. Retrieved 2011-08-08.

[60] Brumley, David (1999-11-16). "Invisible Intruders: rootkits in practice". *USENIX*. USENIX.

[61] Davis, Michael A.; Bodmer, Sean; LeMasters, Aaron (2009-09-03). "Chapter 10: Rootkit Detection" (PDF). *Hacking Exposed Malware & Rootkits: Malware & rootkits security secrets & solutions* (PDF). New York: McGraw Hill Professional. ISBN 978-0-07-159118-8. Retrieved 2010-08-14.

[62] Trlokom (2006-07-05). "Defeating Rootkits and Keyloggers" (PDF). Trlokom. Retrieved 2010-08-17.

[63] Dai Zovi, Dino (2011). "Kernel Rootkits". Archived from the original on September 10, 2012. Retrieved 13 Sep 2012.

[64] "Zeppoo". SourceForge. 18 July 2009. Retrieved 8 August 2011.

[65] Cogswell, Bryce; Russinovich, Mark (2006-11-01). "RootkitRevealer v1.71". Microsoft. Retrieved 2010-11-13.

[66] "Sophos Anti-Rootkit". Sophos. Retrieved 8 August 2011.

[67] "BlackLight". F-Secure. Retrieved 8 August 2011.

[68] "Radix Anti-Rootkit". usec.at. Retrieved 8 August 2011.

[69] "GMER". Retrieved 8 August 2011.

[70] Harriman, Josh (2007-10-19). "A Testing Methodology for Rootkit Removal Effectiveness" (PDF). Dublin, Ireland: Symantec Security Response. Retrieved 2010-08-17.

[71] Cuibotariu, Mircea (2010-02-12). "Tidserv and MS10-015". Symantec. Retrieved 2010-08-19.

[72] "Restart Issues After Installing MS10-015". Microsoft. 2010-02-11. Retrieved 2010-10-05.

[73] "Strider GhostBuster Rootkit Detection". Microsoft Research. 2010-01-28. Retrieved 2010-08-14.

[74] "Signing and Checking Code with Authenticode". Microsoft. Retrieved 2008-09-15.

[75] "Stopping Rootkits at the Network Edge" (PDF). Beaverton, Oregon: Trusted Computing Group. January 2007. Retrieved 2008-07-11.

[76] "TCG PC Specific Implementation Specification, Version 1.1" (PDF). Trusted Computing Group. 2003-08-18. Retrieved 2010-11-22.

[77] "How to generate a complete crash dump file or a kernel crash dump file by using an NMI on a Windows-based system". Microsoft. Retrieved 2010-11-13.

[78] Seshadri, Arvind; et al. (2005). "Pioneer: Verifying Code Integrity and Enforcing Untampered Code Execution on Legacy Systems". Carnegie Mellon University.

[79] Dillard, Kurt (2005-08-03). "Rootkit battle: Rootkit Revealer vs. Hacker Defender".

[80] "The Microsoft Windows Malicious Software Removal Tool helps remove specific, prevalent malicious software from computers that are running Windows 7, Windows Vista, Windows Server 2003, Windows Server 2008, or Windows XP". Microsoft. 2010-09-14.

[81] Hultquist, Steve (2007-04-30). "Rootkits: The next big enterprise threat?". *InfoWorld* (IDG). Retrieved 2010-11-21.

[82] "Security Watch: Rootkits for fun and profit". CNET Reviews. 2007-01-19. Archived from the original on 2012-10-08. Retrieved 2009-04-07.

[83] Bort, Julie (2007-09-29). "Six ways to fight back against botnets". PCWorld. San Francisco: PCWorld Communications. Retrieved 2009-04-07.

[84] Hoang, Mimi (2006-11-02). "Handling Today's Tough Security Threats: Rootkits". *Symantec Connect*. Symantec. Retrieved 2010-11-21.

[85] Danseglio, Mike; Bailey, Tony (2005-10-06). "Rootkits: The Obscure Hacker Attack". Microsoft.

[86] Messmer, Ellen (2006-08-26). "Experts Divided Over Rootkit Detection and Removal". *NetworkWorld.com* (Framingham, Mass.: IDG). Retrieved 2010-08-15.

[87] Stevenson, Larry; Altholz, Nancy (2007). *Rootkits for Dummies*. John Wiley and Sons Ltd. p. 175. ISBN 0-471-91710-9.

[88] Skoudis, Ed; Zeltser, Lenny (2004). *Malware: Fighting Malicious Code*. Prentice Hall PTR. p. 335. ISBN 0-13-101405-6.

[89] Hannel, Jeromey (2003-01-23). "Linux RootKits For Beginners - From Prevention to Removal". SANS Institute. Archived from the original (PDF) on October 24, 2010. Retrieved 2010-11-22.

3.2.12 Further reading

- Blunden, Bill (2009). *The Rootkit Arsenal: Escape and Evasion in the Dark Corners of the System*. Wordware. ISBN 978-1-59822-061-2.
- Hoglund, Greg; Butler, James (2005). *Rootkits: Subverting the Windows Kernel*. Addison-Wesley Professional. ISBN 0-321-29431-9.
- Grampp, F. T.; Morris, Robert H., Sr. (October 1984). "The UNIX System: UNIX Operating System Security". *AT&T Bell Laboratories Technical Journal* (AT&T) **62** (8): 1649–1672.
- Kong, Joseph (2007). *Designing BSD Rootkits*. No Starch Press. ISBN 1-59327-142-5.

- Veiler, Ric (2007). *Professional Rootkits*. Wrox. ISBN 978-0-470-10154-4.

3.2.13 External links

- Rootkit Analysis: Research and Analysis of Rootkits
- Even Nastier: Traditional RootKits
- Sophos Podcast about rootkit removal
- Rootkit research in Microsoft
- Testing of antivirus/anti-rootkit software for the detection and removal of rootkits, Anti-Malware Test Lab, January 2008
- Testing of anti-rootkit software, InformationWeek, January 2007
- Security Now! Episode 9, Rootkits, Podcast by Steve Gibson/GRC explaining Rootkit technology, October 2005

3.3 Backdoor (computing)

Beast, a Windows-based backdoor Trojan horse.

A **backdoor** is a method, often secret, of bypassing normal authentication in a product, computer system, cryptosystem or algorithm etc. Backdoors are often used for securing unauthorized remote access to a computer, or obtaining access to plaintext in cryptographic systems.

A backdoor may take the form of a hidden part of a program,[1] a separate program (e.g. Back Orifice may subvert the system through a rootkit), or may be a hardware feature.[2] Although normally surreptitiously installed, in some cases backdoors are deliberate and widely known. These kinds of backdoors might have "legitimate" uses such as providing the manufacturer with a way to restore user passwords.

Default passwords can function as backdoors if they are not changed by the user. Some debugging features can also act as backdoors if they are not removed in the release version.[3]

In 1993 the United States government attempted to deploy an encryption system, the Clipper chip, with an explicit backdoor for law enforcement and national security access. The chip was unsuccessful internationally and in business.

3.3.1 Overview

The threat of backdoors surfaced when multiuser and networked operating systems became widely adopted. Petersen and Turn discussed computer subversion in a paper published in the proceedings of the 1967 AFIPS Conference.[4] They noted a class of active infiltration attacks that use "trapdoor" entry points into the system to bypass security facilities and permit direct access to data. The use of the word *trapdoor* here clearly coincides with more recent definitions of a backdoor. However, since the advent of public key cryptography the term *trapdoor* has acquired a different meaning (see trapdoor function), and thus the term "backdoor" is now preferred. More generally, such security breaches were discussed at length in a RAND Corporation task force report published under ARPA sponsorship by J.P. Anderson and D.J. Edwards in 1970.[5]

A backdoor in a login system might take the form of a hard coded user and password combination which gives access to the system. A famous example of this sort of backdoor was used as a plot device in the 1983 film *WarGames*, in which the architect of the "WOPR" computer system had inserted a hardcoded password (his dead son's name) which gave the user access to the system, and to undocumented parts of the system (in particular, a video game-like simulation mode and direct interaction with the artificial intelligence).

Although the number of backdoors in systems using proprietary software (software whose source code is not publicly available) is not widely credited, they are nevertheless frequently exposed. Programmers have even succeeded in secretly installing large amounts of benign code as Easter eggs in programs, although such cases may involve official forbearance, if not actual permission.

Examples

Many computer worms, such as Sobig and Mydoom, install a backdoor on the affected computer (generally a PC

on broadband running Microsoft Windows and Microsoft Outlook). Such backdoors appear to be installed so that spammers can send junk e-mail from the infected machines. Others, such as the Sony/BMG rootkit distributed silently on millions of music CDs through late 2005, are intended as DRM measures—and, in that case, as data gathering agents, since both surreptitious programs they installed routinely contacted central servers.

A sophisticated attempt to plant a backdoor in the Linux kernel, exposed in November 2003, added a small and subtle code change by subverting the revision control system.[6] In this case, a two-line change appeared to *check* root access permissions of a caller to the sys_wait4 function, but because it used assignment = instead of equality checking ==, it actually *granted* permissions to the system. This difference is easily overlooked, and could even be interpreted as an accidental typographical error, rather than an intentional attack.[7]

```
ROM:0013DC50                LDR             R0, =aSCtUUnSSipSDip ; ">>> %s(ct=%u, un='%s',
ROM:0013DC54                LDR             R1, =aAuth_admin_int ; "auth_admin_internal"
ROM:0013DC58                BL              sub_558F74
ROM:0013DC5C
ROM:0013DC5C loc_13DC5C                             ; CODE XREF: auth_admin_internal+2C↑j
ROM:0013DC5C                ADD             R0, R5, #0x44
ROM:0013DC60                LDR             R1, =aSUnSU ; "<<< %s(un='%s') = %u"
ROM:0013DC64                BL              strcmp
ROM:0013DC68                CMP             R0, #0
ROM:0013DC6C                BNE             loc_13DC78
ROM:0013DC70                MOV             R0, #0xFFFFFFFD
ROM:0013DC74                LDMDB           R11, {R4-R8,R11,SP,PC}
ROM:0013DC78 ; ---------------------------------------------------------------------------
```

Marked in yellow: backdoor admin password hidden in the code

In January 2014, a backdoor was discovered in certain Samsung Android products, like the Galaxy devices. The Samsung proprietary Android versions are fitted with a backdoor that provides remote access to the data stored on the device. In particular, the Samsung Android software that is in charge of handling the communications with the modem, using the Samsung IPC protocol, implements a class of requests known as remote file server (RFS) commands, that allows the backdoor operator to perform via modem remote I/O operations on the device hard disk or other storage. As the modem is running Samsung proprietary Android software, it is likely that it offers over-the-air remote control that could then be used to issue the RFS commands and thus to access the file system on the device.[8]

Object code backdoors

Harder to detect backdoors involve modifying object code, rather than source code – object code is much harder to inspect, as it is designed to be machine-readable, not human-readable. These backdoors can be inserted either directly in the on-disk object code, or inserted at some point during compilation, assembly linking, or loading – in the latter case the backdoor never appears on disk, only in memory. Object code backdoors are difficult to detect by inspection of the object code, but are easily detected by simply checking for changes (differences), notably in length or in checksum, and in some cases can be detected or analyzed by disassembling the object code. Further, object code backdoors can be removed (assuming source code is available) by simply recompiling from source.

Thus for such backdoors to avoid detection, all extant copies of a binary must be subverted, and any validation checksums must also be compromised, and source must be unavailable, to prevent recompilation. Alternatively, these other tools (length checks, diff, checksumming, disassemblers) can themselves be compromised to conceal the backdoor, for example detecting that the subverted binary is being checksummed and returning the expected value, not the actual value. To conceal these further subversions, the tools must also conceal the changes in themselves – for example, a subverted checksummer must also detect if it is checksumming itself (or other subverted tools) and return false values. This leads to extensive changes in the system and tools being needed to conceal a single change.

Because object code can be regenerated by recompiling (reassembling, relinking) the original source code, making a persistent object code backdoor (without modifying source code) requires subverting the compiler itself – so that when it detects that it is compiling the program under attack it inserts the backdoor – or alternatively the assembler, linker, or loader. As this requires subverting the compiler, this in turn can be fixed by recompiling the compiler, removing the backdoor insertion code. This defense can in turn be subverted by putting a source meta-backdoor in the compiler, so that when it detects that it is compiling itself it then inserts this meta-backdoor generator, together with the original backdoor generator for the original program under attack. After this is done, the source meta-backdoor can be removed, and the compiler recompiled from original source with the compromised compiler executable: the backdoor has been bootstrapped. This attack dates to Karger & Schell (1974), and was popularized in Thompson's 1984 article, entitled "Reflections on Trusting Trust";[9] it is hence colloquially known as the "Trusting Trust" attack. See compiler backdoors, below, for details. Analogous attacks can target lower levels of the system, such as the operating system, and can be inserted during the system booting process; these are also mentioned in Karger & Schell (1974), and now exist in the form of boot sector viruses.[10]

Asymmetric backdoors

A traditional backdoor is a symmetric backdoor: anyone that finds the backdoor can in turn use it. The notion of an asymmetric backdoor was introduced by Adam Young and Moti Yung in the *Proceedings of Advances in Cryptology:*

Crypto '96. An asymmetric backdoor can only be used by the attacker who plants it, even if the full implementation of the backdoor becomes public (e.g., via publishing, being discovered and disclosed by reverse engineering, etc.). Also, it is computationally intractable to detect the presence of an asymmetric backdoor under black-box queries. This class of attacks have been termed kleptography; they can be carried out in software, hardware (for example, smartcards), or a combination of the two. The theory of asymmetric backdoors is part of a larger field now called cryptovirology. Notably, NSA inserted a kleptographic backdoor into the Dual_EC_DRBG standard.[2][11][12]

There exists an experimental asymmetric backdoor in RSA key generation. This OpenSSL RSA backdoor was designed by Young and Yung, utilizes a twisted pair of elliptic curves, and has been made available.[13]

3.3.2 Compiler backdoors

A sophisticated form of black box backdoor is a **compiler backdoor**, where not only is a compiler subverted (to insert a backdoor in some other program, such as a login program), but it is further modified to detect when it is compiling itself and then inserts both the backdoor insertion code (targeting the other program) and the code modifying self-compilation, like the mechanism how retroviruses infect their host. This can be done by modifying the source code, and the resulting compromised compiler (object code) can compile the original (unmodified) source code and insert itself: the exploit has been boot-strapped.

This attack was originally presented in Karger & Schell (1974, p. 52, section 3.4.5: “Trap Door Insertion”), which was a United States Air Force security analysis of Multics, where they described such an attack on a PL/I compiler, and call it a “compiler trap door"; they also mention a variant where the system initialization code is modified to insert a backdoor during booting, as this is complex and poorly understood, and call it an “initialization trapdoor"; this is now known as a boot sector virus.[10]

This attack was then actually implemented and popularized by Ken Thompson, in his Turing Award acceptance speech in 1983 (published 1984), “Reflections on Trusting Trust”,[9] which points out that trust is relative, and the only software one can truly trust is code where every step of the bootstrapping has been inspected. This backdoor mechanism is based on the fact that people only review source (human-written) code, and not compiled machine code (object code). A program called a compiler is used to create the second from the first, and the compiler is usually trusted to do an honest job.

Thompson's paper describes a modified version of the Unix C compiler that would:

- Put an invisible backdoor in the Unix login command when it noticed that the login program was being compiled, and as a twist
- Also add this feature undetectably to future compiler versions upon *their* compilation as well.

Because the compiler itself was a compiled program, users would be extremely unlikely to notice the machine code instructions that performed these tasks. (Because of the second task, the compiler's source code would appear “clean”.) What's worse, in Thompson's proof of concept implementation, the subverted compiler also subverted the analysis program (the disassembler), so that anyone who examined the binaries in the usual way would not actually see the real code that was running, but something else instead.

An updated analysis of the original exploit is given in Karger & Schell (2002, Section 3.2.4: Compiler trap doors), and a historical overview and survey of the literature is given in Wheeler (2009, Section 2: Background and related work).

Occurrences

Thompson's version was, officially, never released into the wild. It is believed, however, that a version was distributed to BBN and at least one use of the backdoor was recorded.[14] There are scattered anecdotal reports of such backdoors in subsequent years.[15]

This attack was recently (August 2009) discovered by Sophos labs: The W32/Induc-A virus infected the program compiler for Delphi, a Windows programming language. The virus introduced its own code to the compilation of new Delphi programs, allowing it to infect and propagate to many systems, without the knowledge of the software programmer. An attack that propagates by building its own Trojan horse can be especially hard to discover. It is believed that the Induc-A virus had been propagating for at least a year before it was discovered.[16]

Countermeasures

Once a system has been compromised with a backdoor or Trojan horse, such as the *Trusting Trust* compiler, it is very hard for the “rightful” user to regain control of the system – typically one should rebuild a clean system and transfer data (but not executables!) over. However, several practical weaknesses in the *Trusting Trust* scheme have been suggested. For example, a sufficiently motivated user could painstakingly review the machine code of the untrusted compiler before using it. As mentioned above, there are

ways to hide the Trojan horse, such as subverting the disassembler; but there are ways to counter that defense, too, such as writing your own disassembler from scratch.

A generic method to counter trusting trust attacks is called Diverse Double-Compiling (DDC). The method requires a different compiler and the source code of the compiler-under-test. That source, compiled with both compilers, results in two different stage-1 compilers, which however should have the same behavior. Thus the same source compiled with both stage-1 compilers must then result in two identical stage-2 compilers. A formal proof is given that the latter comparison guarantees that the purported source code and executable of the compiler-under-test correspond, under some assumptions. This method was applied by its author to verify that the C compiler of the GCC suite (v. 3.0.4) contained no trojan, using icc (v. 11.0) as the different compiler.[17]

In practice such verifications are not done by end users, except in extreme circumstances of intrusion detection and analysis, due to the rarity of such sophisticated attacks, and because programs are typically distributed in binary form. Removing backdoors (including compiler backdoors) is typically done by simply rebuilding a clean system. However, the sophisticated verifications are of interest to operating system vendors, to ensure that they are not distributing a compromised system, and in high-security settings, where such attacks are a realistic concern.

3.3.3 List of known backdoors

- Back Orifice was created in 1998 by hackers from Cult of the Dead Cow group as a remote administration tool. It allowed Windows computers to be remotely controlled over a network and exploited the name similarity with Microsoft BackOffice.
- The Dual_EC_DRBG cryptographically secure pseudorandom number generator was revealed in 2013 to possibly have a kleptographic backdoor deliberately inserted by NSA, who also had the private key to the backdoor.[2][12]
- Several backdoors in the unlicensed copies of WordPress plug-ins were discovered in March 2014.[18] They were inserted as obfuscated JavaScript code and silently created, for example, an admin account in the website database. A similar scheme was later exposed in the Joomla plugin.[19]
- Borland Interbase versions 4.0 through 6.0 had a hard-coded backdoor, put there by the developers. The server code contains a compiled-in backdoor account (username: *politically*, password: *correct*), which could be accessed over a network connection, and once a user logged in with it, he could take full control over all Interbase databases. The backdoor was detected in 2001 and a patch was released.[20][21]
- Juniper Networks backdoor inserted in the year 2008 into the versions of firmware ScreenOS from 6.2.0r15 to 6.2.0r18 and from 6.3.0r12 to 6.3.0r20[22] that gives any user administrative access when using a special master password.[23]

3.3.4 See also

- Backdoor:Win32.Hupigon
- Backdoor.Win32.Seed

3.3.5 References

[1] Chris Wysopal, Chris Eng. "Static Detection of Application Backdoors" (PDF). Veracode. Retrieved 2015-03-14.

[2] wired.com: "How a Crypto 'Backdoor' Pitted the Tech World Against the NSA" (Zetter) 24 Sep 2013

[3] http://blog.erratasec.com/2012/05/bogus-story-no-chinese-backdoor-in.html

[4] H.E. Petersen, R. Turn. "System Implications of Information Privacy". *Proceedings of the AFIPS Spring Joint Computer Conference*, vol. 30, pages 291–300. AFIPS Press: 1967.

[5] *Security Controls for Computer Systems*, Technical Report R-609, WH Ware, ed, Feb 1970, RAND Corp.

[6] Larry McVoy (November 5, 2003) Linux-Kernel Archive: Re: BK2CVS problem. ussg.iu.edu

[7] Thwarted Linux backdoor hints at smarter hacks; Kevin Poulsen; *SecurityFocus*, 6 November 2003.

[8] replicant.us: "Samsung Galaxy Back-door" 28 Jan 2014

[9] "Reflections on Trusting Trust" (PDF). *http://www.ece.cmu.edu/~{}ganger*. External link in |website= (help)

[10] Karger & Schell 2002.

[11] G+M: "The strange connection between the NSA and an Ontario tech firm" 20 Jan 2014

[12] nytimes.com: "N.S.A. Able to Foil Basic Safeguards of Privacy on Web" (Perlroth et al.) 5 Sep 2013

[13] cryptovirology.com page on OpenSSL RSA backdoor

[14] Jargon File entry for "backdoor" at catb.org, describes Thompson compiler hack

[15] Mick Stute's answer to "What is a coder's worst nightmare?", *Quora* – describes a case in 1989.

[16] Compile-a-virus — W32/Induc-A Sophos labs on the discovery of the Induc-A virus

[17] Wheeler 2009.

[18] "Unmasking "Free" Premium WordPress Plugins". *Sucuri Blog*. Retrieved 3 March 2015.

[19] Sinegubko, Denis. "Joomla Plugin Constructor Backdoor". *Securi*. Retrieved 13 March 2015.

[20] "Vulnerability Note VU#247371". *Vulnerability Note Database*. Retrieved 13 March 2015.

[21] "Interbase Server Contains Compiled-in Back Door Account". *http://www.cert.org/*. Retrieved 13 March 2015. External link in |website= (help)

[22] "Researchers confirm backdoor password in Juniper firewall code". *Ars Technica*. Retrieved 2016-01-16.

[23] "Zagrożenia tygodnia 2015-W52 - Spece.IT". *Spece.IT* (in Polish). Retrieved 2016-01-16.

- Karger, Paul A.; Schell, Roger R. (June 1974). *Multics Security Evaluation: Vulnerability Analysis* (PDF). Vol II.
- Karger, Paul A.; Schell, Roger R. (September 18, 2002). *Thirty Years Later: Lessons from the Multics Security Evaluation* (PDF). *Computer Security Applications Conference, 2002. Proceedings. 18th Annual* (IEEE). pp. 119–126. doi:10.1109/CSAC.2002.1176285. Retrieved 2014-11-08.
- Wheeler, David A. (7 December 2009). *Fully Countering Trusting Trust through Diverse Double-Compiling* (Ph.D.). Fairfax, VA: George Mason University. Retrieved 2014-11-09.

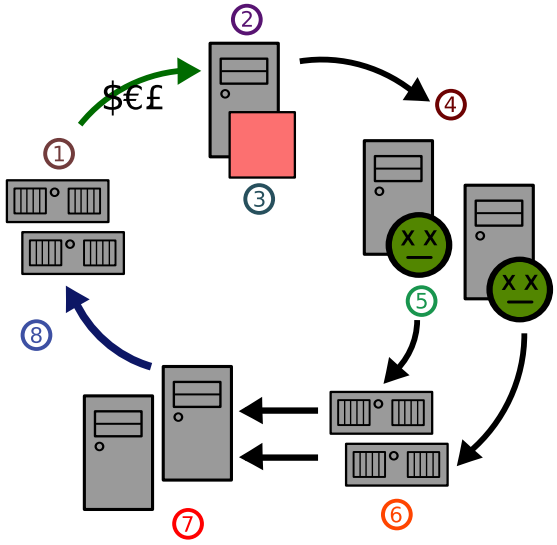
3.3.6 External links

- Three Archaic Backdoor Trojan Programs That Still Serve Great Pranks
- Backdoors removal — List of backdoors and their removal instructions.
- FAQ Farm's Backdoors FAQ: wiki question and answer forum
- List of backdoors and Removal —
- David A. Wheeler's Page on "Fully Countering Trusting Trust through Diverse Double-Compiling"—Author's 2009 Ph.D. thesis at George Mason University

3.4 Zombie (computer science)

"Zombie virus" redirects here. For the use of the term in science fiction, see Zombie apocalypse.

In computer science, a **zombie** is a computer connected to the Internet that has been compromised by a hacker, computer virus or trojan horse and can be used to perform malicious tasks of one sort or another under remote direction. Botnets of zombie computers are often used to spread e-mail spam and launch denial-of-service attacks. Most owners of zombie computers are unaware that their system is being used in this way. Because the owner tends to be unaware, these computers are metaphorically compared to zombies. A coordinated DDoS attack by multiple botnet machines also resembles a zombie horde attack.



(1) Spammer's web site (2) Spammer (3) Spamware (4) Infected computers (5) Virus or trojan (6) Mail servers (7) Users (8) Web traffic

3.4.1 History

Zombies have been used extensively to send e-mail spam; as of 2005, an estimated 50–80% of all spam worldwide was sent by zombie computers.[1] This allows spammers to avoid detection and presumably reduces their bandwidth costs, since the owners of zombies pay for their own bandwidth. This spam also greatly furthers the spread of Trojan horses, as Trojans are not self-replicating. They rely on the movement of e-mails or spam to grow, whereas worms can spread by other means.[2]

For similar reasons zombies are also used to commit click

fraud against sites displaying pay per click advertising. Others can host phishing or money mule recruiting websites.

Zombies can be used to conduct distributed denial-of-service attacks, a term which refers to the orchestrated flooding of target websites by large numbers of computers at once. The large number of Internet users making simultaneous requests of a website's server is intended to result in crashing and the prevention of legitimate users from accessing the site.[3] A variant of this type of flooding is known as distributed degradation-of-service. Committed by "pulsing" zombies, distributed degradation-of-service is the moderated and periodical flooding of websites, done with the intent of slowing down rather than crashing a victim site. The effectiveness of this tactic springs from the fact that intense flooding can be quickly detected and remedied, but pulsing zombie attacks and the resulting slowdown in website access can go unnoticed for months and even years.[4]

Notable incidents of distributed denial- and degradation-of-service attacks in past include the attack upon the SPEWS service in 2003, and the one against Blue Frog service in 2006. In 2000, several prominent Web sites (Yahoo, eBay, etc.) were clogged to a standstill by a distributed denial of service attack mounted by 'MafiaBoy', a Canadian teenager. An attack on grc.com is discussed at length, and the perpetrator, a 13-year-old probably from Kenosha, Wisconsin, was identified on the Gibson Research Web site. Steve Gibson disassembled a 'bot' which was a zombie used in the attack, and traced it to its distributor. In his account about his research, he describes the operation of a 'bot'-controlling IRC channel.[5]

Beginning in July 2009, similar botnet capabilities have also emerged for the growing smartphone market. Examples include the July 2009 in the wild release of the Sexy Space text message worm, the world's first botnet capable SMS worm, which targeted the Symbian operating system in Nokia smartphones. Later that month, Charlie Miller revealed a proof of concept text message worm for the iPhone at Black Hat. Also in July, United Arab Emirates consumers were targeted by the Etisalat BlackBerry spyware program. At the present time, the security community is divided as to the real world potential of mobile botnets. But in an August 2009 interview with The New York Times, cyber security consultant Michael Gregg summarized the issue this way: "We are about at the point with phones that we were with desktops in the '80s."

3.4.2 See also

- Malware
- Trojan horse (computing)
- Command and control (malware)
- Botnet
- Low Orbit Ion Cannon

3.4.3 References

[1] Tom Spring (2005-06-20). "Spam Slayer: Slaying Spam-Spewing Zombie PCs". PC World. Retrieved 2015-12-19.

[2] White, Jay D. (2007). *Managing Information in the Public Sector*. M.E. Sharpe. p. 221. ISBN 0-7656-1748-X.

[3] Weisman, Steve (2008). *The Truth about Avoiding Scams*. FT Press. p. 201. ISBN 0-13-233385-6.

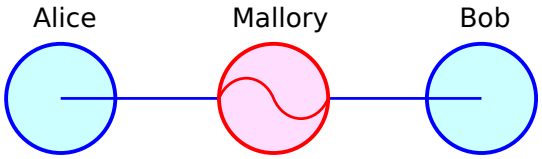
[4] Schwabach, Aaron (2006). *Internet and the Law*. ABC-CLIO. p. 325. ISBN 1-85109-731-7.

[5] Steve Gibson, The Attacks on GRC.COM, *Gibson Research Corporation*, first: 2001-05-04, last: 2009-08-12

3.4.4 External links

- Study by IronPort finds 80% of e-mail spam sent by Zombie PCs. June 28, 2006
- Botnet operation controlled 1.5 million PCs
- Is Your PC a Zombie? on About.com
- Intrusive analysis of a web-based proxy zombie network
- A detailed account of what a zombie machine looks like and what it takes to "fix" it
- Correspondence between Steve Gibson and Wicked
- Zombie networks, comment spam, and referer [sic] spam
- The New York Times: Phone Hacking Threat is Low, But It Exists
- Hackers Target Cell Phones, WPLG-TV/ABC-10 Miami
- Researcher: BlackBerry Spyware Wasn't Ready for Prime Time
- Forbes: How to Hijack Every iPhone in the World
- Hackers Plan to Clobber the Cloud, Spy on Blackberries
- SMobile Systems release solution for Etisalat BlackBerry spyware

- LOIC IRC-0 - An Open-Source IRC Botnet for Network Stress Testing
- An Open-Source IRC and Webpage Botnet for Network Stress Testing



3.5 Man-in-the-middle attack

This article is about the form of communication interception. For the decryption optimisation process, see Meet-in-the-middle attack.

In cryptography and computer security, a **man-in-the-middle attack** (often abbreviated **MitM**, **MiM attack**, **MitMA** or the same using all capital letters) is an attack where the attacker secretly relays and possibly alters the communication between two parties who believe they are directly communicating with each other. Man-in-the-middle attacks can be thought about through a chess analogy. Mallory, who barely knows how to play chess, claims that she can play two grandmasters simultaneously and either win one game or draw both. She waits for the first grandmaster to make a move and then makes this same move against the second grandmaster. When the second grandmaster responds, Mallory makes the same play against the first. She plays the entire game this way and cannot lose. A man-in-the-middle attack is a similar strategy and can be used against many cryptographic protocols.[1] One example of man-in-the-middle attacks is active eavesdropping, in which the attacker makes independent connections with the victims and relays messages between them to make them believe they are talking directly to each other over a private connection, when in fact the entire conversation is controlled by the attacker. The attacker must be able to intercept all relevant messages passing between the two victims and inject new ones. This is straightforward in many circumstances; for example, an attacker within reception range of an unencrypted Wi-Fi wireless access point can insert himself as a man-in-the-middle.[2]

As an attack that aims at circumventing mutual authentication, or lack thereof, a man-in-the-middle attack can succeed only when the attacker can impersonate each endpoint to their satisfaction as expected from the legitimate other end. Most cryptographic protocols include some form of endpoint authentication specifically to prevent MITM attacks. For example, TLS can authenticate one or both parties using a mutually trusted certificate authority.[3]

An illustration of the man-in-the-middle attack

3.5.1 Example

Suppose Alice wishes to communicate with Bob. Meanwhile, Mallory wishes to intercept the conversation to eavesdrop and optionally to deliver a false message to Bob.

First, Alice asks Bob for his public key. If Bob sends his public key to Alice, but Mallory is able to intercept it, a man-in-the-middle attack can begin. Mallory sends a forged message to Alice that purports to come from Bob, but instead includes Mallory's public key.

Alice, believing this public key to be Bob's, encrypts her message with Mallory's key and sends the enciphered message back to Bob. Mallory again intercepts, deciphers the message using her private key, possibly alters it if she wants, and re-enciphers it using the public key Bob originally sent to Alice. When Bob receives the newly enciphered message, he believes it came from Alice.

1. Alice sends a message to Bob, which is intercepted by Mallory:

   Alice *"Hi Bob, it's Alice. Give me your key."* → Mallory Bob

2. Mallory relays this message to Bob; Bob cannot tell it is not really from Alice:

   Alice Mallory *"Hi Bob, it's Alice. Give me your key."* → Bob

3. Bob responds with his encryption key:

   Alice Mallory ← *[Bob's key]* Bob

4. Mallory replaces Bob's key with her own, and relays this to Alice, claiming that it is Bob's key:

   Alice ← *[Mallory's key]* Mallory Bob

5. Alice encrypts a message with what she believes to be Bob's key, thinking that only Bob can read it:

   > Alice *"Meet me at the bus stop!" [encrypted with Mallory's key]* → Mallory Bob

6. However, because it was actually encrypted with Mallory's key, Mallory can decrypt it, read it, modify it (if desired), re-encrypt with Bob's key, and forward it to Bob:

   > Alice Mallory *"Meet me at the van down by the river!" [encrypted with Bob's key]* → Bob

7. Bob thinks that this message is a secure communication from Alice.

This example[4][5][6] shows the need for Alice and Bob to have some way to ensure that they are truly using each other's public keys, rather than the public key of an attacker. Otherwise, such attacks are generally possible, in principle, against any message sent using public-key technology. Fortunately, a variety of techniques can help defend against MITM attacks.

3.5.2 Defences against the attack

All cryptographic systems that are secure against MITM attacks require an additional exchange or transmission of information over some kind of secure channel. Many key agreement methods have been developed, with different security requirements for the *secure* channel. Interlock Protocol attempts to address this.

Various defenses against MITM attacks use authentication techniques that include:

- DNSSEC: Secure DNS extensions
- Public key infrastructures: Transport Layer Security is an example of implementing public key infrastructure over Transmission Control Protocol. This is used to prevent Man-in-the-middle attack over a secured HTTP connection on internet. Client and server exchange PKI certificates issued and verified by a common certificate authority.
  - PKI mutual authentication: The main defense in a PKI scenario is mutual authentication. In this case applications from both client and server mutually validates their certificates issued by a common root certificate authority. Virtual private networks do mutual authentication before sending data over the created secure tunnel, however mutual authentication over internet for HTTP connections are seldom enforced.
- Certificate pinning
- A recorded media attestment (assuming that the user's identity can be recognized from the recording), which can either be:
  - A verbal communication of a shared value for each session (as in ZRTP)
  - An audio/visual communication of the public key hash (which can be easily distributed via PKI)[7]
- Stronger mutual authentication, such as:
  - Secret keys (which are usually high information entropy secrets, and thus more secure), or
  - Passwords (which are usually low information entropy secrets, and thus less secure)
- Latency examination, such as with long cryptographic hash function calculations that lead into tens of seconds; if both parties take 20 seconds normally, and the calculation takes 60 seconds to reach each party, this can indicate a third party
- Second (secure) channel verification
- Testing is being carried out on deleting compromised certificates from issuing authorities on the actual computers and compromised certificates are being exported to sandbox area before removal for analysis
- Quantum Cryptography

The integrity of public keys must generally be assured in some manner, but need not be secret. Passwords and shared secret keys have the additional secrecy requirement. Public keys can be verified by a certificate authority, whose public key is distributed through a secure channel (for example, with a web browser or OS installation). Public keys can also be verified by a web of trust that distributes public keys through a secure channel (for example by face-to-face meetings).

See key-agreement protocol for a classification of protocols that use various forms of keys and passwords to prevent man-in-the-middle attacks.

3.5.3 Forensic analysis

Captured network traffic from what is suspected to be a MITM attack can be analyzed in order to determine if it really was a MITM attack or not. Important evidence to analyze when doing network forensics of a suspected TLS MITM attack include:[8]

- IP address of the server
- DNS name of the server
- X.509 certificate of the server
  - Is the certificate self signed?
  - Is the certificate signed by a trusted CA?
  - Has the certificate been revoked?
  - Has the certificate been changed recently?
  - Do other clients, elsewhere on the Internet, also get the same certificate?

3.5.4 Quantum cryptography

Quantum cryptography protocols typically authenticate part or all of their classical communication with an unconditionally secure authentication scheme e.g. Wegman-Carter authentication.[9]

3.5.5 Beyond cryptography

A notable non-cryptographic man-in-the-middle attack was perpetrated by a Belkin wireless network router in 2003. Periodically, it would take over an HTTP connection being routed through it: this would fail to pass the traffic on to destination, but instead itself respond as the intended server. The reply it sent, in place of the web page the user had requested, was an advertisement for another Belkin product. After an outcry from technically literate users, this 'feature' was removed from later versions of the router's firmware.[10]

In 2013, the Nokia's Xpress Browser was revealed to be decrypting HTTPS traffic on Nokia's proxy servers, giving the company clear text access to its customers' encrypted browser traffic. Nokia responded by saying that the content was not stored permanently, and that the company had organizational and technical measures to prevent access to private information.[11]

3.5.6 Implementations

Notable man-in-the-middle attack implementations include the following:

- DSniff – the first public implementation of MITM attacks against SSL and SSH
- Fiddler2 HTTP(S) diagnostic tool
- NSA impersonation of Google[12]
- Opendium Iceni Content-control software, used to perform inspection of HTTPS traffic at the gateway.
- Subterfuge – a framework to launch multiple MITM attacks
- Superfish malware
- Websense Content Gateway – used to perform inspection of SSL traffic at the proxy
- wsniff – a tool for 802.11 HTTP/HTTPS based MITM attacks

3.5.7 See also

- Aspidistra transmitter – a British radio transmitter used for World War II "intrusion" operations, an early man-in-the-middle attack.
- Babington Plot – the plot against Elizabeth I of England, where Francis Walsingham intercepted the correspondence.
- Boy-in-the-browser – a simpler type of web browser MITM
- Computer security – the design of secure computer systems.
- Cryptanalysis – the art of deciphering encrypted messages with incomplete knowledge of how they were encrypted.
- Digital signature – a cryptographic guarantee of the authenticity of a text, usually the result of a calculation only the author is expected to be able to perform.
- Evil Maid Attack – attack used against full disk encryption systems
- Interlock protocol – a specific protocol to circumvent a man-in-the-middle attack when the keys may have been compromised.
- Key management – how to manage cryptographic keys, including generation, exchange and storage.
- Key-agreement protocol – a cryptographic protocol for establishing a key in which both parties can have confidence.
- Man-in-the-browser – a type of web browser MITM
- Mutual authentication – how communicating parties establish confidence in one another's identities.
- Password-authenticated key agreement – a protocol for establishing a key using a password.
- Quantum cryptography – the use of quantum mechanics to provide security in cryptography (while older methods rely on one-way functions).

- Secure channel – a way of communicating resistant to interception and tampering.
- Spoofing attack

3.5.8 References

[1] "What is Man in the Middle Attack". internetofthings. Retrieved 27 May 2016.

[2] Tanmay Patange (November 10, 2013). "How to defend yourself against MITM or Man-in-the-middle attack".

[3] Callegati, Franco; Cerroni, Walter; Ramilli, Marco (2009). "IEEE Xplore - Man-in-the-Middle Attack to the HTTPS Protocol". *ieeexplore.ieee.org*: 78–81. Retrieved 13 April 2016.

[4] MiTM on RSA public key encryption

[5] How Encryption Works

[6] Public-key cryptography

[7] Heinrich, Stuart (2013). "Public Key Infrastructure based on Authentication of Media Attestments". arXiv:1311.7182v1.

[8] "Network Forensic Analysis of SSL MITM Attacks". *NETRESEC Network Security Blog*. Retrieved March 27, 2011.

[9] "5. Unconditionally secure authentication". *liu.se*.

[10] Leyden, John (2003-11-07). "Help! my Belkin router is spamming me". *The Register*.

[11] Meyer, David (10 January 2013). "Nokia: Yes, we decrypt your HTTPS data, but don't worry about it". Gigaom, Inc. Retrieved 13 June 2014.

[12] "NSA disguised itself as Google to spy, say reports". CNET. 12 Sep 2013. Retrieved 15 Sep 2013.

3.5.9 External links

- Finding Hidden Threats by Decrypting SSL, SANS Institute

3.6 Man-in-the-browser

Man-in-the-Browser (**MITB**, **MitB**, **MIB**, **MiB**), a form of Internet threat related to man-in-the-middle (MITM), is a proxy Trojan horse[1] that infects a web browser by taking advantage of vulnerabilities in browser security to modify web pages, modify transaction content or insert additional transactions, all in a completely covert fashion invisible to both the user and host web application. A MitB attack will be successful irrespective of whether security mechanisms such as SSL/PKI and/or two or three-factor Authentication solutions are in place. A MitB attack may be countered by utilising out-of-band transaction verification, although SMS verification can be defeated by **man-in-the-mobile** (**MitMo**) malware infection on the mobile phone. Trojans may be detected and removed by antivirus software[2] with a 23% success rate against Zeus in 2009,[3] and still low rates in 2011.[4] The 2011 report concluded that additional measures on top of antivirus were needed.[4] A related, more simple attack is the **boy-in-the-browser** (**BitB**, **BITB**). The majority of financial service professionals in a survey considered MitB to be the greatest threat to online banking.

3.6.1 Description

The MitB threat was demonstrated by Augusto Paes de Barros in his 2005 presentation about backdoor trends "The future of backdoors - worst of all worlds".[5] The name "Man-in-the-Browser" was coined by Philipp Gühring on 27 January 2007.[6]

A MitB Trojan works by utilising common facilities provided to enhance browser capabilities such as Browser Helper Objects (a feature limited to Internet Explorer), browser extensions and user scripts (for example in JavaScript) etc.[6] Antivirus software can detect some of these methods.[2]

In a nutshell example exchange between user and host, such as an Internet banking funds transfer, the customer will always be shown, via confirmation screens, the exact payment information as keyed into the browser. The bank, however, will receive a transaction with materially altered instructions, i.e. a different destination account number and possibly amount. The use of strong authentication tools simply creates an increased level of misplaced confidence on the part of both customer and bank that the transaction is secure. Authentication, by definition, is concerned with the validation of identity credentials. This should not be confused with transaction verification.

3.6.2 Examples

Examples of MitB threats on different operating systems and web browsers:

3.6.3 Protection

Antivirus

Known Trojans may be detected, blocked and removed by antivirus software.[2] In a 2009 study, the effectiveness of

antivirus against Zeus was 23%,[3] and again low success rates were reported in a separate test in 2011.[4] The 2011 report concluded that additional measures on top of antivirus were needed.[4]

Hardened software

- Browser security software: MitB attacks may be blocked by in-browser security software such as Trusteer Rapport for Microsoft Windows and Mac OS X which blocks the APIs from browser extensions and controls communication.[11][12][15]

- Alternative software: Reducing or eliminating the risk of malware infection by using portable applications or using alternatives to Microsoft Windows like Mac OS X, Linux, or mobile OSes Android, iOS, Chrome OS, Windows Mobile, Symbian etc., and/or browsers Chrome, Opera.[26] Further protection can be achieved by running this alternative OS, like Linux, from a non-installed live CD, or Live USB.[27]

- Secure Web Browser: Several vendors can now provide a two-factor security solution where a Secure Web Browser is part of the solution. In this case MitB attacks are avoided as the user executes a hardened browser from their two-factor security device rather than executing the "infected" browser from their own machine.

Out-of-band transaction verification

A theoretically effective method of combating any MitB attack is through an out-of-band (OOB) transaction verification process. This overcomes the MitB trojan by verifying the transaction details, as received by the host (bank), to the user (customer) over a channel other than the browser; for example an automated telephone call, SMS, or a dedicated mobile app with graphical cryptogram.[28] OOB transaction verification is ideal for mass market use since it leverages devices already in the public domain (e.g. landline, mobile phone, etc.) and requires no additional hardware devices yet enables three-factor authentication (utilising voice biometrics), transaction signing (to non-repudiation level) and transaction verification. The downside is that the OOB transaction verification adds to the level of the end-user's frustration with more and slower steps.

Man-in-the-Mobile Mobile phone mobile Trojan spyware **man-in-the-mobile** (**MitMo**)[29] can defeat OOB SMS transaction verification.[30]

- ZitMo (Zeus-In-The-Mobile) is not a MitB Trojan itself (although it performs a similar proxy function on the incoming SMSes), but is mobile malware suggested for installation on a mobile phone by a Zeus infected computer. By intercepting all incoming SMSes, it defeats SMS-based banking OOB two-factor authentication on Windows Mobile, Android, Symbian, BlackBerry.[30] ZitMo may be detected by Antivirus running on the mobile device.

- SpitMo (SpyEye-In-The-Mobile, SPITMO), is similar to ZitMo.[31]

Web fraud detection

Web Fraud Detection can be implemented at the bank to automatically check for anomalous behaviour patterns in transactions.[32]

3.6.4 Related attacks

Proxy trojans

Keyloggers are the most primitive form of **proxy trojans**, followed by browser-session recorders which capture more data, and lastly MitBs are the most sophisticated type.[1]

Man-in-the-middle

Main article: Man-in-the-middle

SSL/PKI etc. may offer protection in a man-in-the-middle attack, but offers no protection in a man-in-the-browser attack.

Boy-in-the-Browser

A related attack that is simpler and quicker for malware authors to set up is termed **boy-in-the-Browser** (**BitB** or **BITB**). Malware is used to change the client's computer network routing to perform a classic man-in-the-middle attack. Once the routing has been changed, the malware may completely remove itself, making detection more difficult.[33]

Clickjacking

Main article: Clickjacking

Clickjacking tricks a web browser user into clicking on something different from what the user perceives, by means of malicious code in the webpage.

3.6.5 See also

- Browser security
- Form grabbing
- IT risk
- Threat (computer)
- Timeline of computer viruses and worms
- Online banking
- Security token
- Transaction authentication number
- DNS hijacking

3.6.6 References

[1] Bar-Yosef, Noa (2010-12-30). "The Evolution of Proxy Trojans". Retrieved 2012-02-03.

[2] F-Secure (2007-02-11). "Threat Description: Trojan-Spy:W32/Nuklus.A". Retrieved 2012-02-03.

[3] Trusteer (2009-09-14). "Measuring the in-the-wild effectiveness of Antivirus against Zeus" (PDF). Archived from the original (PDF) on November 6, 2011. Retrieved 2012-02-05.

[4] Quarri Technologies, Inc (2011). "Web Browsers: Your Weak Link in Achieving PCI Compliance" (PDF). Retrieved 2012-02-05.

[5] Paes de Barros, Augusto (15 September 2005). "O futuro dos backdoors - o pior dos mundos" (PDF) (in Portuguese). Sao Paulo, Brazil: Congresso Nacional de Auditoria de Sistemas, Segurança da Informação e Governança - CNASI. Archived from the original (PDF) on July 6, 2011. Retrieved 2009-06-12.

[6] Gühring, Philipp (27 January 2007). "Concepts against Man-in-the-Browser Attacks" (PDF). Retrieved 2008-07-30.

[7] Dunn, John E (2010-07-03). "Trojan Writers Target UK Banks With Botnets". Retrieved 2012-02-08.

[8] Dunn, John E (2010-10-12). "Zeus not the only bank Trojan threat, users warned". Retrieved 2012-02-03.

[9] Curtis, Sophie (2012-01-18). "Facebook users targeted in Carberp man-in-the-browser attack". Retrieved 2012-02-03.

[10] Marusceac Claudiu Florin (2008-11-28). "Trojan.PWS.ChromeInject.B Removal Tool". Retrieved 2012-02-05.

[11] Nattakant Utakrit, School of Computer and Security Science, Edith Cowan University (2011-02-25). "Review of Browser Extensions, a Man-in-theBrowser Phishing Techniques Targeting Bank Customers". Retrieved 2012-02-03.

[12] Symantec Marc Fossi (2010-12-08). "ZeuS-style banking Trojans seen as greatest threat to online banking: Survey". Retrieved 2012-02-03.

[13] Ted Samson (2011-02-22). "Crafty OddJob malware leaves online bank accounts open to plunder". Retrieved 2012-02-06.

[14] Symantec Marc Fossi (2008-01-23). "Banking with Confidence". Retrieved 2008-07-30.

[15] Trusteer. "Trusteer Rapport". Retrieved 2012-02-03.

[16] CEO of Trusteer Mickey Boodaei (2011-03-31). "Man-in-the-Browser attacks target the enterprise". Retrieved 2012-02-03.

[17] www.net-security.org (2011-05-11). "Explosive financial malware targets Windows". Retrieved 2012-02-06.

[18] Jozsef Gegeny; Jose Miguel Esparza (2011-02-25). "Tatanga: a new banking trojan with MitB functions". Retrieved 2012-02-03.

[19] "Tiny 'Tinba' Banking Trojan Is Big Trouble". *msnbc.com*. Retrieved 2016-02-28.

[20] Borean, Wayne (2011-05-24). "The Mac OS X Virus That Wasn't". Retrieved 2012-02-08.

[21] Fisher, Dennis (2011-05-02). "Crimeware Kit Emerges for Mac OS X". Archived from the original on September 5, 2011. Retrieved 2012-02-03.

[22] F-secure. "Threat DescriptionTrojan-Spy:W32/Zbot". Retrieved 2012-02-05.

[23] Hyun Choi; Sean Kiernan (2008-07-24). "Trojan.Wsnpoem Technical Details". Symantec. Retrieved 2012-02-05.

[24] Microsoft (2010-04-30). "Encyclopedia entry: Win32/Zbot - Learn more about malware - Microsoft Malware Protection Center". Symantec. Retrieved 2012-02-05.

[25] Richard S. Westmoreland (2010-10-20). "Antisource - ZeuS". Retrieved 2012-02-05.

[26] Horowitz, Michael (2012-02-06). "Online banking: what the BBC missed and a safety suggestion". Retrieved 2012-02-08.

[27] Purdy, Kevin (2009-10-14). "Use a Linux Live CD/USB for Online Banking". Retrieved 2012-02-04.

[28] Finextra Research (2008-11-13). "Commerzbank to deploy Cronto mobile phone-based authentication technology". Retrieved 2012-02-08.

[29] Chickowski, Ericka (2010-10-05). "'Man In The Mobile' Attacks Highlight Weaknesses In Out-Of-Band Authentication". Retrieved 2012-02-09.

[30] Schwartz, Mathew J. (2011-07-13). "Zeus Banking Trojan Hits Android Phones". Retrieved 2012-02-04.

[31] Balan, Mahesh (2009-10-14). "Internet Banking & Mobile Banking users beware – ZITMO & SPITMO is here !!". Retrieved 2012-02-05.

[32] Sartain, Julie (2012-02-07). "How to protect online transactions with multi-factor authentication". Retrieved 2012-02-08.

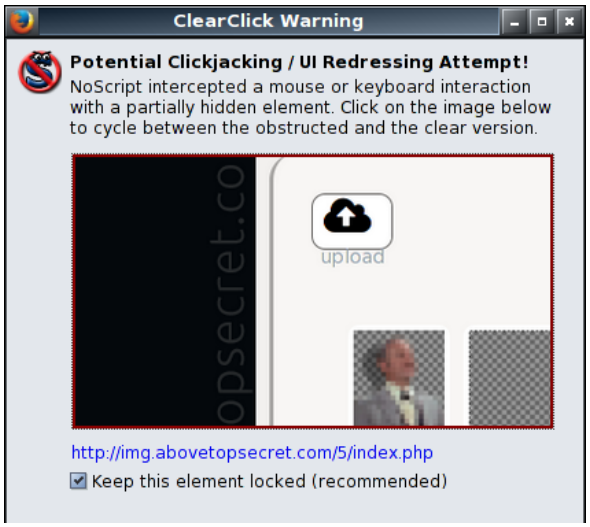
[33] Imperva (2010-02-14). "Threat Advisory Boy in the Browser". Retrieved 2015-03-12.

3.6.7 External links

- Virus attack on HSBC Transactions with OTP Device
- Virus attack on ICICI Bank Transactions
- Virus attack on Citibank Transactions
- Hackers outwit online banking identity security systems BBC Click
- Antisource - ZeuS A summary of ZeuS as a Trojan and Botnet, plus vector of attacks
- Man-In-The-Browser Video on YouTube Entrust President and CEO Bill Conner
- Zeus: King of crimeware toolkits Video on YouTube The Zeus toolkit, Symantec Security Response
- How safe is online banking? Audio BBC Click
- Boy-in-the-Browser Cyber Attack Video on YouTube Imperva

3.7 Clickjacking

Clickjacking (**User Interface redress attack**, **UI redress attack**, **UI redressing**) is a malicious technique of tricking a Web user into clicking on something different from what the user perceives they are clicking on, thus potentially revealing confidential information or taking control of their computer while clicking on seemingly innocuous web pages.[1][2][3][4] It is a browser security issue that is a vulnerability across a variety of browsers and platforms. A clickjack takes the form of embedded code or a script that can execute without the user's knowledge, such as clicking on a button that appears to perform another function.[5] The term "clickjacking" was coined by Jeremiah Grossman and Robert Hansen in 2008.[6][7] Clickjacking can be understood as an instance of the confused deputy problem, a term used to describe when a computer is innocently fooled into misusing its authority.[8]



"potential clickjacking" warning from the "NoScript" internet-browser addon

3.7.1 Description

Clickjacking is possible because seemingly harmless features of HTML web pages can be employed to perform unexpected actions.

A clickjacked page tricks a user into performing undesired actions by clicking on a concealed link. On a clickjacked page, the attackers load another page over it in a transparent layer. The users think that they are clicking visible buttons, while they are actually performing actions on the hidden/invisible page. The hidden page may be an authentic page; therefore, the attackers can trick users into performing actions which the users never intended. There is no way of tracing such actions to the attackers later, as the users would have been genuinely authenticated on the hidden page.

3.7.2 Examples

A user might receive an email with a link to a video about a news item, but another webpage, say a product page on Amazon.com, can be "hidden" on top or underneath the "PLAY" button of the news video. The user tries to "play" the video but actually "buys" the product from Amazon.

The hacker can only send a single click, so they rely on the fact that the visitor is both logged into Amazon.com and has 1-click ordering enabled.

Other known exploits include:

- Tricking users into enabling their webcam and microphone through Flash (though this has since been fixed since originally reported)
- Tricking users into making their social networking profile information public
- Downloading and running a malware (malicious software) allowing to a remote attacker to take control of others computers[9][10][11]
- Making users follow someone on Twitter[12]
- Sharing or liking links on Facebook[13][14]
- Getting likes on Facebook fan page[15] or +1 on Google Plus
- Clicking Google Adsense ads to generate pay per click revenue
- Playing YouTube videos to gain views
- Following someone on Facebook

While technical implementation of these attacks may be challenging due to cross-browser incompatibilities, a number of tools such as BeEF or Metasploit Project offer almost fully automated exploitation of clients on vulnerable websites. Clickjacking may be facilitated by - or may facilitate - other web attacks, such as XSS.[16][17]

Likejacking

Likejacking is a malicious technique of tricking users of a website into "liking" a Facebook page that they did not intentionally mean to "like".[18] The term "likejacking" came from a comment posted by Corey Ballou in the article *How to "Like" Anything on the Web (Safely)*,[19] which is one of the first documented postings explaining the possibility of malicious activity regarding Facebook's "like" button.[20]

According to an article in *IEEE Spectrum*, a solution to likejacking was developed at one of Facebook's hackathons.[21] A "Like" bookmarklet is available that avoids the possibility of likejacking present in the Facebook Like Button.[22]

Cursorjacking

Cursorjacking is a UI redressing technique to change the cursor from the location the user perceives, discovered in 2010 by Eddy Bordi, a researcher at Vulnerability.fr, Marcus Niemietz demonstrated this with a custom cursor icon, and in 2012 Mario Heiderich by hiding the cursor.[23][24]

Jordi Chancel, a researcher at Alternativ-Testing.fr, discovered a cursorjacking vulnerability using Flash, HTML and JavaScript code in Mozilla Firefox on Mac OS X systems (fixed in Firefox 30.0) which can lead to arbitrary code execution and webcam spying.[25]

A second CursorJacking vulnerability was again discovered by Jordi Chancel in Mozilla Firefox on Mac OS X systems (fixed in Firefox 37.0) using once again Flash, HTML and JavaScript code which can lead also to the spying of the webcam and the execution of a malicious addon allowing the execution of a malware on the computer of the trapped user.[26]

Password manager attack

A 2014 paper from researcher at the Carnegie Mellon University found that whilst browsers refuse to autofill if the protocol on the current login page is different from the protocol at the time the password was saved, some password managers would insecurely fill in passwords for the http version of https-saved passwords. Most managers did not protect against iFrame and redirection based attacks and exposed additional passwords where password synchronization had been used between multiple devices.[27]

3.7.3 Prevention

Client-side

NoScript Protection against clickjacking (including likejacking) can be added to Mozilla Firefox desktop and mobile[28] versions by installing the NoScript add-on: its ClearClick feature, released on 8 October 2008, prevents users from clicking on invisible or "redressed" page elements of embedded documents or applets.[29] According to Google's "Browser Security Handbook" from year 2008, NoScript's ClearClick is "the only freely available product that offers a reasonable degree of protection" against Clickjacking.[30] Protection from the newer cursorjacking attack was added to NoScript 2.2.8 RC1.[23]

GuardedID GuardedID (a commercial product) includes client-side clickjack protection for users of Internet Explorer and Firefox[31] without interfering with the opera-

tion of legitimate iFrames. GuardedID clickjack protection forces all frames to become visible.

Gazelle Gazelle is a Microsoft Research project secure web browser based on IE, that uses an OS-like security model, and has its own limited defenses against clickjacking.[32] In Gazelle, a window of different origin may only draw dynamic content over another window's screen space if the content it draws is opaque.

Server-side

Framekiller Web site owners can protect their users against UI redressing (frame based clickjacking) on the server side by including a framekiller JavaScript snippet in those pages they do not want to be included inside frames from different sources.[30]

Such JavaScript-based protection, unfortunately, is not always reliable. This is especially true on Internet Explorer,[30] where this kind of countermeasure can be circumvented "by design" by including the targeted page inside an <IFRAME SECURITY=restricted> element.[33]

X-Frame-Options Introduced in 2009 in Internet Explorer 8 was a new HTTP header X-Frame-Options which offered a partial protection against clickjacking[34][35] and was shortly after adopted by other browsers (Safari,[36] Firefox,[37] Chrome,[38] and Opera[39]). The header, when set by website owner, declares its preferred framing policy: values of DENY, SAMEORIGIN, or ALLOW-FROM *origin* will prevent any framing, framing by external sites, or allow framing only by the specified site, respectively. In addition to that, some advertising sites return a non-standard ALLOWALL value with the intention to allow framing their content on any page (equivalent of not setting X-Frame-Options at all).

In 2013 the X-Frame-Options header has been officially published as RFC 7034,[40] but is not an internet standard. The document is provided for informational purposes only.

Content Security Policy The frame-ancestors directive of Content Security Policy (introduced in version 1.1) can allow or disallow embedding of content by potentially hostile pages using iframe, object, etc. This directive obsoletes the X-Frame-Options directive. If a page is served with both headers, the frame-ancestors policy should be preferred by the browser.[41]—although some popular browsers disobey this requirement.[42]

Example frame-ancestors policies:

Disallow embedding. All iframes etc. will be blank, or contain a browser specific error page. Content-Security-Policy: frame-ancestors 'none' # Allow embedding of own content only. Content-Security-Policy: frame-ancestors 'self' # Allow specific origins to embed this content Content-Security-Policy: frame-ancestors example.com wikipedia.org

3.7.4 See also

- Browser security
- Internet security
- Internet safety
- Hacker (computer security)
- Cross-site scripting
- Phishing
- Ghostery
- Social jacking

3.7.5 References

[1] Robert McMillan (17 September 2008). "At Adobe's request, hackers nix 'clickjacking' talk". PC World. Retrieved 2008-10-08.

[2] Megha Dhawan (29 September 2008). "Beware, clickjackers on the prowl". India Times. Retrieved 2008-10-08.

[3] Dan Goodin (7 October 2008). "Net game turns PC into undercover surveillance zombie". *The Register*. Retrieved 2008-10-08.

[4] Fredrick Lane (8 October 2008). "Web Surfers Face Dangerous New Threat: 'Clickjacking'". newsfactor.com. Archived from the original on 13 October 2008. Retrieved 2008-10-08.

[5] Sumner Lemon (30 September 2008). "Business Center: Clickjacking Vulnerability to Be Revealed Next Month". Retrieved 2008-10-08.

[6] You don't know (click)jack Robert Lemos, October 2008

[7] Naveen, Sharanya. "Clickjacking". Retrieved 7 June 2016.

[8] The Confused Deputy rides again!, Tyler Close, October 2008

[9] "select element persistance allows for attacks". Retrieved 2012-10-09.

[10] "UI selection timeout missing on download prompts". Retrieved 2014-02-04.

[11] "Delay following click events in file download dialog too short on OS X". Retrieved 2016-03-08.

[12] Daniel Sandler (12 February 2009). "Twitter's "Don't Click" prank, explained (dsandler.org)". Retrieved 2009-12-28.

[13] Krzysztof Kotowicz (21 December 2009). "New Facebook clickjacking attack in the wild". Retrieved 2009-12-29.

[14] BBC (3 June 2010). "Facebook "clickjacking" spreads across site". *BBC News*. Retrieved 2010-06-03.

[15] Josh MacDonald. "Facebook Has No Defence Against Black Hat Marketing". Retrieved 2016-02-03.

[16] "The Clickjacking meets XSS: a state of art". Exploit DB. 2008-12-26. Retrieved 2015-03-31.

[17] Krzysztof Kotowicz. "Exploiting the unexploitable XSS with clickjacking". Retrieved 2015-03-31.

[18] Cohen, Richard (31 May 2010). "Facebook Work - "Likejacking"". Sophos. Retrieved 2010-06-05.

[19] Ballou, Corey (2 June 2010). ""Likejacking" Term Catches On". jqueryin.com. Archived from the original on 5 June 2010. Retrieved 2010-06-08.

[20] Perez, Sarah (2 June 2010). ""Likejacking" Takes Off on Facebook". ReadWriteWeb. Retrieved 2010-06-05.

[21] Kushner, David (June 2011). "Facebook Philosophy: Move Fast and Break Things". spectrum.ieee.org. Retrieved 2011-07-15.

[22] Perez, Sarah (23 April 2010). "How to "Like" Anything on the Web (Safely)". *ReadWriteWeb*. Retrieved 24 August 2011.

[23] Krzysztof Kotowicz (18 January 2012). "Cursorjacking Again". Retrieved 2012-01-31.

[24] Aspect Security. "Cursor-jacking attack could result in application security breaches". Retrieved 2012-01-31.

[25] "Mozilla Foundation Security Advisory 2014-50". Mozilla. Retrieved 17 August 2014.

[26] "Mozilla Foundation Security Advisory 2015-35". Mozilla. Retrieved 25 October 2015.

[27] "Password Managers: Attacks and Defenses" (PDF). Retrieved 26 July 2015.

[28] Giorgio Maone (24 June 2011). "NoScript Anywhere". hackademix.net. Retrieved 2011-06-30.

[29] Giorgio Maone (8 October 2008). "Hello ClearClick, Goodbye Clickjacking". hackademix.net. Retrieved 2008-10-27.

[30] Michal Zalevski (10 December 2008). "Browser Security Handbook, Part 2, UI Redressing". Google Inc. Retrieved 2008-10-27.

[31] Robert Hansen (4 February 2009). "Clickjacking and GuardedID ha.ckers.org web application security lab". Retrieved 2011-11-30.

[32] Wang, Helen J.; Grier, Chris; Moschchuk, Alexander; King, Samuel T.; Choudhury, Piali; Venter, Herman (August 2009). "The Multi-Principal OS Construction of the Gazelle Web Browser" (PDF). 18th Usenix Security Symposium, Montreal, Canada. Retrieved 2010-01-26.

[33] Giorgio Maone (27 October 2008). "Hey IE8, I Can Has Some Clickjacking Protection". hackademix.net. Retrieved 2008-10-27.

[34] Eric Lawrence (27 January 2009). "IE8 Security Part VII: ClickJacking Defenses". Retrieved 2010-12-30.

[35] Eric Lawrence (30 March 2010). "Combating ClickJacking With X-Frame-Options". Retrieved 2010-12-30.

[36] Ryan Naraine (8 June 2009). "Apple Safari jumbo patch: 50+ vulnerabilities fixed". Retrieved 2009-06-10.

[37] https://developer.mozilla.org/en/The_X-FRAME-OPTIONS_response_header The X-Frame-Options response header — MDC

[38] Adam Barth (26 January 2010). "Security in Depth: New Security Features". Retrieved 2010-01-26.

[39] "Web specifications support in Opera Presto 2.6". 12 October 2010. Retrieved 2012-01-22.

[40] "HTTP Header Field X-Frame-Options". IETF. 2013.

[41] "Content Security Policy Level 2". *w3.org*. 2014-07-02. Retrieved 2015-01-29.

[42] "Clickjacking Defense Cheat Sheet". Retrieved 2016-01-15.

3.7.6 External links

- Original paper on clickjacking

Chapter 4

Malware for Profit

4.1 Privacy-invasive software

See also: Greyware

Privacy-invasive software is a category of computer software that ignores users' privacy and that is distributed with a specific intent, often of a commercial nature. Three typical examples of privacy-invasive software are adware, spyware and content hijacking programs.

4.1.1 Background

In a digital setting, such as the Internet, there are a wide variety of privacy threats. These vary from the tracking of user activity (sites visited, items purchased etc.), to mass marketing based on the retrieval of personal information (spam offers and telemarketing calls are more common than ever), to the distribution of information on lethal technologies used for, e.g., acts of terror.

Today, software-based privacy-invasions occur in numerous aspects of Internet usage. Spyware programs set to collect and distribute user information secretly download and execute on users' workstations. Adware displays advertisements and other commercial content often based upon personal information retrieved by spyware programs. System monitors record various actions on computer systems. Keyloggers record users' keystrokes in order to monitor user behavior. Self-replicating malware downloads and spreads disorder in systems and networks. Data-harvesting software programmed to gather e-mail addresses have become conventional features of the Internet, which among other things results in that spam e-mail messages fill networks and computers with unsolicited commercial content. With those threats in mind, we hereby define privacy-invasive software as:

4.1.2 Definition

In this context, *ignoring users' right to be left alone* means that the software is unsolicited and that it does not permit users to determine for themselves when, how and to what extent personally identifiable data is gathered, stored or processed by the software. *Distributed* means that it has entered the computer systems of users from (often unknown) servers placed on the Internet infrastructure. *Often of a commercial nature* means that the software (regardless of type or quality) is used as a tool in some sort of a commercial plan to gain revenues.

4.1.3 Problem with the spyware concept

In early 2000, Steve Gibson formulated the first description of spyware after realizing software that stole his personal information had been installed on his computer (Gibson Research Corporation). His definition reads as follows:

This definition was valid in the beginning of the spyware evolution. However, as the spyware concept evolved over the years it attracted new kinds of behaviours. As these behaviours grew both in number and in diversity, the term spyware became hollowed out. This evolution resulted in that a great number of synonyms sprang up, e.g. thiefware, scumware, trackware, and badware. It is believed that the lack of a single standard definition of spyware depends on the diversity in all these different views on what really should be included, or as Aaron Weiss put it (Weiss 2005):

Despite this vague comprehension of the essence in spyware, all descriptions include two central aspects. The degree of associated user consent, and the level of negative impact they impair on the user and their computer system (further discussed in Section 2.3 and Section 2.5 in (Boldt 2007a)). Because of the diffuse understanding in the spyware concept, recent attempts to define it have

been forced into compromises. The Anti-Spyware Coalition (ASC) which is constituted by public interest groups, trade associations, and anti-spyware companies, have come to the conclusion that the term spyware should be used at two different abstraction levels (Anti-Spyware Coalition). At the low level they use the following definition, which is similar to Steve Gibson's original one:

However, since this definition does not capture all the different types of spyware available they also provide a wider definition, which is more abstract in its appearance:

Difficulties in defining spyware, forced the ASC to define what they call *Spyware (and Other Potentially Unwanted Technologies)* instead. In this term they include any software that does not have the users' appropriate consent for running on their computers. Another group that has tried to define spyware is a group called StopBadware.org, which consists of actors such as Harvard Law School, Oxford University, Google, Lenovo, and Sun Microsystems (StopBadware.org). Their result is that they do not use the term spyware at all, but instead introduce the term *badware*. Their definition thereof span over seven pages, but the essence looks as follows (StopBadware.org Guidelines):

Both definitions from ASC and StopBadware.org show the difficulty with defining spyware. We therefore regard the term spyware at two different abstraction levels. On the lower level it can be defined according to Steve Gibsons original definition. However, in its broader and in a more abstract sense the term spyware is hard to properly define, as concluded above.

4.1.4 Introducing the term, "privacy-invasive software"

A joint conclusion is that it is important, for both software vendors and users, that a clear separation between acceptable and unacceptable software behaviour is established (Bruce 2005)(Sipior 2005). The reason for this is the subjective nature of many spyware programs included, which result in inconsistencies between different users beliefs, i.e. what one user regards as legitimate software could be regarded as a spyware by others. As the spyware concept came to include increasingly more types of programs, the term got hollowed out, resulting in several synonyms, such as trackware, evilware and badware, all negatively emotive. We therefore choose to introduce the term *privacy-invasive software* to encapsulate all such software. We believe this term to be more descriptive than other synonyms without having as negative connotation. Even if we use the word *invasive* to describe such software, we believe that an invasion of privacy can be both desired and beneficial for the user as long as it is fully transparent, e.g. when implementing specially user-tailored services or when including personalization features in software.

	Tolerable negative consequences	**Moderate negative consequences**	**Severe negative consequences**
High consent	Legitimate software	Adverse software	Double agents
Medium consent	Semi-transparent software	Unsolicited software	Semi-parasites
Low consent	Covert software	Trojans	Parasites

Legitimate software with high user consent *and* tolerable negative consequences

Spyware with medium user consent *or* moderate negative consequences

Malicious software with low user consent *or* severe negative consequences

A three-by-three matrix classification of privacy-invasive software showing legitimate, spyware and malicious software. (Boldt 2010, p. 110)

The work by Warkentins et al. (described in Section 7.3.1 in (Boldt 2007a)) can be used as a starting point when developing a classification of privacy-invasive software, where we classify privacy-invasive software as a combination between *user consent* and *direct negative consequences*. User consent is specified as either *low*, *medium* or *high*, while the degree of direct negative consequences span between *tolerable*, *moderate*, and *severe*. This classification allows us to first make a distinction between legitimate software and spyware, and secondly between spyware and malicious software. All software that has a low user consent, *or* which impairs severe direct negative consequences should be regarded as malware. While, on the other hand, any software that has high user consent, *and* which results in tolerable direct negative consequences should be regarded as legitimate software. By this follows that spyware constitutes the remaining group of software, i.e. those that have medium user consent or which impair moderate direct negative consequences. This classification is described in further detail in Chapter 7 in (Boldt 2007a).

In addition to the direct negative consequences, we also introduce *indirect negative consequences*. By doing so our classification distinguishes between any negative behaviour

a program has been designed to carry out (direct negative consequences) and security threats introduced by just having that software executing on the system (indirect negative consequences). One example of an indirect negative consequence is the exploitation risk of software vulnerabilities in programs that execute on users' systems without their knowledge (Saroiu 2004).

4.1.5 Comparison to malware

The term privacy-invasive software is motivated in that software types such as adware and spyware are essentially often defined according to their actions instead of their distribution mechanisms (as with most malware definitions, which also rarely correspond to motives of, e.g., business and commerce). The overall intention with the concept of privacy-invasive software is consequently to convey the commercial aspect of unwanted software contamination. The threats of privacy-invasive software consequently do not find their roots in totalitarianism, malice or political ideas, but rather in the free market, advanced technology and the unbridled exchange of electronic information. By the inclusion of purpose in its definition, the term privacy-invasive software is a contribution to the research community of privacy and security.

4.1.6 History

Internet goes commercial

In the mid-1990s, the development of the Internet increased rapidly due to the interest from the general public. One important factor behind this accelerating increase was the 1993 release of the first browser, called Mosaic (Andreessen 1993). This marked the birth of the graphically visible part of the Internet known as the World Wide Web (WWW). Commercial interests became well aware of the potential offered by the WWW in terms of electronic commerce, and soon companies selling goods over the Internet emerged, i.e. pioneers such as book dealer Amazon.com and CD retailer CDNOW.com, which both were founded in 1994 (Rosenberg 2004).

During the following years, personal computers and broadband connections to the Internet became more commonplace. Also, the increased use of the Internet resulted in that e-commerce transactions involved considerable amounts of money (Abhijit 2002). As competition over customers intensified, some e-commerce companies turned to questionable methods in their battle to entice customers into completing transactions with them (CDT 2006) and (Shukla 2005). This opened ways for illegitimate actors to gain revenues by stretching the limits used with methods for collecting personal information and for propagating commercial advertisements. Buying such services allowed for some e-commerce companies to get an advantage over their competitors, e.g. by using advertisements based on unsolicited commercial messages (also known as spam) (Jacobsson 2004).

Commercially motivated adverse software

The use of questionable techniques, such as Spam, were not as destructive as the more traditional malicious techniques, e.g. computer viruses or trojan horses. Compared to such malicious techniques the new ones differed in two fundamental ways. First, they were not necessarily illegal, and secondly, their main goal was gaining money instead of creating publicity for the creator by reaping digital havoc. Therefore, these techniques grouped as a "grey" area next to the already existing "dark" side of the Internet.

Behind this development stood advertisers that understood that Internet was a "merchant's utopia", offering huge potential in global advertising coverage at a relatively low cost. By using the Internet as a global notice board, e-commerce companies could market their products through advertising agencies that delivered online ads to the masses. In 2004, online advertisement yearly represented between $500 million and $2 billion markets, which in 2005 increased to well over $6 billion-a-year (McFedries 2005) and (Zhang 2005)]. The larger online advertising companies report annual revenues in excess of $50 million each (CNET 2005). In the beginning of this development such companies distributed their ads in a broadcast-like manner, i.e. they were not streamlined towards individual users' interests. Some of these ads were served directly on Web sites as banner ads, but dedicated programs, called adware, soon emerged. Adware were used to display ads through pop-up windows without depending on any Internet access or Web pages.

The birth of spyware

In the search for more effective advertising strategies, these companies soon discovered the potential in ads that were targeted towards user interests. Once targeted online ads started to appear, the development took an unfortunate turn. Now, some advertisers developed software that became known as spyware, collecting users' personal interests, e.g. through their browsing habits. Over the coming years spyware would evolve into a significant new threat to Internet-connected computers, bringing along reduced system performance and security. The information gathered by spyware were used for constructing user profiles, including personal interests, detailing what users could be persuaded to buy. The introduction of online advertisements

also opened a new way to fund software development by having the software display advertisements to its users. By doing so the software developer could offer their software "free of charge", since they were paid by the advertising agency. Unfortunately, many users did not understand the difference between "free of charge" and a "free gift", where difference is that a free gift is given without any expectations of future compensation, while something provided free of charge expects something in return. A dental examination that is provided free of charge at a dentist school is not a free gift. The school expects gained training value and as a consequence the customer suffers increased risks. As adware were combined with spyware, this became a problem for computer users. When downloading software described as "free of charge" the users had no reason to suspect that it would report on for instance their Internet usage, so that presented advertisements could be targeted towards their interests.

Some users probably would have accepted to communicate their browsing habits because of the positive feedback, e.g. "offers" relevant to their interests. However, the fundamental problem was that users were not properly informed about neither the occurrence nor the extent of such monitoring, and hence were not given a chance to decide on whether to participate or not. As advertisements became targeted, the borders between adware and spyware started to dissolve, combining both these programs into a single one, that both monitored users and delivered targeted ads. The fierce competition soon drove advertisers to further "enhance" the ways used for serving their ads, e.g. replacing user-requested content with sponsored messages instead, before showing it to the users.

The arms-race between spyware vendors

As the chase for faster financial gains intensified, several competing advertisers turned to use even more illegitimate methods in an attempt to stay ahead of their competitors. This targeted advertising accelerated the whole situation and created a "gray" between conventional adds that people chose to see, such as subscribing to an Internet site & adds pushed on users through "pop-up adds" or downloaded adds displayed in a program itself. [1] This practice pushed Internet advertising closer to the "dark" side of Spam & other types of invasive, privacy compromising advertising (Görling 2004). During this development, users experienced infections from unsolicited software that crashed their computers by accident, change application settings, harvested personal information, and deteriorated their computer experience (Pew 2005). Over time these problems led to the introduction of countermeasures in the form of anti-spyware tools.

These tools purported to clean computers from spyware, adware, and any other type of shady software located in that same "gray" area. This type of software can lead to false positives as some types of legitimate software came to be branded by some users as "Spyware" (i.e. Spybot: Search & Destroy identifies the ScanSpyware program as a Spybot.) These tools were designed similarly to anti-malware tools, such as Antivirus software. Anti-spyware tools identify programs using signatures (semantics, program code, or other identifying attributes). The process only works on known programs, which can lead to the false positives mentioned earlier & leave previously unknown spyware undetected. To further aggravate the situation, a few especially illegitimate companies distributed fake anti-spyware tools in their search for a larger piece of the online advertising market. These fake tools claimed to remove spyware, but instead installed their own share of adware and spyware on unwitting users' computers. Sometimes even accompanied by the functionality to remove adware and spyware from competing vendors. Anti-Spyware has become a new area of online vending with fierce competition.

New spyware programs are being added to the setting in what seems to be a never-ending stream, although the increase has levelled out somewhat over the last years. However, there still does not exist any consensus on a common spyware definition or classification, which negatively affects the accuracy of anti-spyware tools. As mentioned above, some spyware programs remain undetected on users' computers (Good et al. 2006) and (MTL 2006). Developers of anti-spyware programs officially state that the fight against spyware is more complicated than the fight against viruses, trojan horses, and worms (Webroot 2006).

4.1.7 Predicted future development

There are several trends integrating computers and software into people's daily lives. One example is traditional media-oriented products which are being integrated into a single device, called media centres. These media centres include the same functionality as conventional television, DVD-players, and stereo equipment, but combined with an Internet connected computer. In a foreseeable future these media centres are anticipated to reach vast consumer impact (CES) (Newman 2006). In this setting, spyware could monitor and surveillance for instance what television channels are being watched, when/why users change channel or what DVD movies users have purchased and watch. This is information that is highly attractive for any advertising or media-oriented corporation to obtain. This presents us with a probable scenario where spyware is tailored towards these new platforms; the technology needed is to a large extent the same as is used in spyware today.

Another interesting area for spyware vendors is the increasing amount of mobile devices being shipped. Distributors of advertisements have already turned their eyes to these devices. So far this development have not utilized the geographic position data stored in these devices. However, during the time of this writing companies are working on GPS-guided ads and coupons destined for mobile phones and hand-held devices (Business 2.0 Magazine). In other words, development of location-based marketing that allow advertising companies to get access to personal geographical data so that they can serve geographically dependent ads and coupons to their customers. Once such geographic data is being harvested and correlated with already accumulated personal information, another privacy barrier has been crossed.

4.1.8 References

[1] Vincentas (11 July 2013). "Privacy Invasive Software in SpyWareLoop.com". Spyware Loop. Retrieved 27 July 2013.

General

- Abhijit, C.; Kuilboer, J.P. (2002), *E-Business & E-Commerce Infrastructure: Technologies Supporting the E-Business Initiative*, Columbus, USA: McGraw Hill.
- Andreessen, M. (1993), *NCSA Mosaic Technical Summary*, USA: National Center for Supercomputing Applications.
- ASC (2006-10-05), *Anti-Spyware Coalition*, http://www.antispywarecoalition.org External link in |publisher= (help).
- Boldt, M. (2007a), *Privacy-Invasive Software - Exploring Effects and Countermeasures* (PDF), School of Engineering, Blekinge Institute of Technology, Sweden: *Licentiate Thesis Series No. 2007:01*.
- Boldt, M. (2010), *Privacy-Invasive Software* (PDF), Blekinge, Sweden: School of Computing, Blekinge Institute of Technology
- Boldt, M.; Carlsson, B.; Larsson, T.; Lindén, N. (2007b), *Preventing Privacy-Invasive Software using Online Reputations*, Springer Verlag, Berlin Germany: in *Lecture Notes in Computer Science* series, Volume 4721 PDF External link in |publisher= (help).
- Boldt, M.; Carlsson, B. (2006a), *Privacy-Invasive Software and Preventive Mechanisms*, Papeete French, Polynesia: in Proceedings *of IEEE International Conference on Systems and Networks Communications (ICSNC 2006)* PDF External link in |publisher= (help).
- Boldt, M.; Carlsson, B. (2006b), *Analysing Privacy-Invasive Software Countermeasures*, Papeete, French Polynesia: in *Proceedings of IEEE International Conference on Systems and Networks Communications (ICSEA 2006)*.
- Boldt, M.; Jacobsson, A.; Carlsson, B. (2004), "Exploring Spyware Effects" (PDF), *Proceedings of the Eighth Nordic Workshop on Secure IT Systems* (NordSec2004) *(Helsinki, Finland)*.
- Bruce, J. (2005), "Defining Rules for Acceptable Adware", *Proceedings of the 15th Virus Bulletin Conference* (Dublin, Ireland).
- Business 2.0 Magazine, *20 Smart Companies to Start Now*, http://money.cnn.com/magazines/business2/business2_archive/2006/09/01/8384349/index.htm?source=yahoo_quote External link in |publisher= (help).
- CDT (2006), *Following the Money*, http://www.cdt.org: Center for Democracy & Technology.
- CES, *International Consumer Electronics Association*, http://www.cesweb.org External link in |publisher= (help).
- CNET (2005), *The Money Game: How Adware Works and How it is Changing*, CNET Anti Spyware Workshop, San Francisco, USA.
- Gibson, *Gibson Research Corporation*, http://www.grc.com/optout.htm External link in |publisher= (help).
- Good, N.; et al. (2006), *User Choices and Regret: Understanding Users' Decision Process about Consentually Acquired Spyware*, Columbus, USA: I/S: A Journal of Law and Policy for the Information Society, Volume 2, Issue 2.
- Görling, S. (2004), *An Introduction to the Parasite Economy*, Luxemburg: In Proceedings of EICAR.
- Jacobsson, A. (2007), *Security in Information Networks - from Privacy-Invasive Software to Plug and Play Business*, School of Engineering, Blekinge Institute of Technology, Sweden: Doctoral Thesis.
- Jacobsson, A. (2004), *Exploring Privacy Risks in Information Networks*, School of Engineering, Blekinge Institute of Technology, Sweden: Licentiate Thesis Series No. 2004:11.
- Jacobsson, A.; Boldt, M.; Carlsson, B. (2004), *Privacy-Invasive Software in File-Sharing Tools*, Kluwer Academic Publishers, Dordrecht NL, pp.

281-296: Deswarte, F. Cuppens, S. Jajodia and L. Wang (Eds.) *Information Security Management, Education and Privacy* PDF External link in |publisher= (help).

- McFedries, P. (2005), *The Spyware Nightmare*, Nebraska, USA: in IEEE Spectrum, Volume 42, Issue 8.
- MTL (2006), *AntiSpyware Comparison Reports*, http://www.malware-test.com/antispyware.html: Malware-Test Lab.
- Newman, M.W. (2006), "Recipes for Digital Living", *IEEE Computer* (Vol. 39, Issue 2).
- Pew, Internet (2005), "The Threat of Unwanted Software Programs is Changing the Way People use the Internet" (PDF), *PIP Spyware Report July 05* (Pew Internet & American Life Project), archived from the original (PDF) on July 13, 2007.
- Rosenberg, R.S. (2004), *The Social Impact of Computers* (3rd ed.), Place=Elsevier Academic Press, San Diego CA.
- Saroiu, S.; Gribble, S.D.; Levy, H.M. (2004), "Measurement and Analysis of Spyware in a University Environment", *Proceedings of the 1st Symposium on Networked Systems Design and Implementation (NSDI)* (San Francisco, USA).
- Shukla, S.; Nah, F.F. (2005), "Web Browsing and Spyware Intrusion", *Communications of the ACM* (New York, USA) **48** (8).
- Sipior, J.C. (2005), "A United States Perspective on the Ethical and Legal Issues of Spyware", *Proceedings of 7th International Conference on Electronic Commerce* (Xian, China).
- StopBadware.org, *StopBadware.org*, http://www.stopbadware.org External link in |publisher= (help).
- StopBadware.org Guidelines, "StopBadware.org Software Guidelines", *StopBadware.org*, archived from the original on September 28, 2007.
- Webroot (2006), "Differences between Spyware and Viruses", *Spysweeper.com* (Webroot Software).
- Weiss, A. (2005), "Spyware Be Gone", *ACM netWorker* (ACM Press, New York, USA) **9** (1).
- Zhang, X. (2005), "What Do Consumers Really Know About Spyware?", *Communications of the ACM* (ACM) **48** (8).

4.2 Adware

For the Lavasoft anti-virus program, see Ad-Aware.

Adware, or **advertising-supported software**, is any software package that automatically renders advertisements in order to generate revenue for its author. The advertisements may be in the user interface of the software or on a screen presented to the user during the installation process. The functions may be designed to analyze which Internet sites the user visits and to present advertising pertinent to the types of goods or services featured there. The term is sometimes used to refer to software that displays unwanted advertisements known as malware.[1]

4.2.1 Advertising-supported software

In legitimate software, the advertising functions are integrated into or bundled with the program. Adware is usually seen by the developer as a way to recover development costs, and in some cases, it may allow the software to be provided to the user free of charge or at a reduced price. The income derived from presenting advertisements to the user may allow or motivate the developer to continue to develop, maintain and upgrade the software product.[2] The use of advertising-supported software in business is becoming increasingly popular, with a third of IT and business executives in a 2007 survey by McKinsey & Company planning to be using ad-funded software within the following two years.[3] Advertisement-funded software is also one of the business models for open-source software.

In application software

Some software is offered in both an advertising-supported mode and a paid, advertisement-free mode. The latter is usually available by an online purchase of a license or registration code for the software that unlocks the mode, or the purchase and download of a separate version of the software.[lower-alpha 1]

Some software authors offer advertising-supported versions of their software as an alternative option to business organizations seeking to avoid paying large sums for software licenses, funding the development of the software with higher fees for advertisers.[7]

Examples of advertising-supported software include the Windows version of the Internet telephony application Skype,[8] and the Amazon Kindle 3 family of e-book readers, which has versions called "Kindle with Special Offers" that display advertisements on the home page and in sleep mode in exchange for substantially lower pricing.[9]

In 2012, Microsoft and its advertising division, Microsoft Advertising,[lower-alpha 2] announced that Windows 8, the major release of the Microsoft Windows operating system, would provide built-in methods for software authors to use advertising support as a business model.[11][12] The idea had been considered since as early as 2005.[13]

In software as a service

Support by advertising is a popular business model of software as a service (SaaS) on the Web. Notable examples include the email service Gmail[2][14] and other Google Apps products,[3] and the social network Facebook.[15][16] Microsoft has also adopted the advertising-supported model for many of its social software SaaS offerings.[17] The Microsoft Office Live service was also available in an advertising-supported mode.[3]

In the view of Federal Trade Commission staff,[18] there appears to be general agreement that software should be considered "spyware" only if it is downloaded or installed on a computer without the user's knowledge and consent. However, unresolved issues remain concerning how, what, and when consumers need to be told about software installed on their computers for consent to be adequate. For instance, distributors often disclose in an end-user license agreement that there is additional software bundled with primary software, but some panelists and commenters did not view such disclosure as sufficient to infer consent to the installation of the bundled software.

4.2.2 As malware

The term *adware* is frequently used to describe a form of malware (malicious software)[19][20] which presents unwanted advertisements to the user of a computer.[21][22] The advertisements produced by adware are sometimes in the form of a pop-up or sometimes in an "unclosable window".[23]

When the term is used in this way, the severity of its implication varies. While some sources rate adware only as an "irritant",[24] others classify it as an "online threat"[25] or even rate it as seriously as computer viruses and trojans.[26] The precise definition of the term in this context also varies.[lower-alpha 3] Adware that observes the computer user's activities without their consent and reports it to the software's author is called spyware.[28] However most adware operates legally and some adware manufacturers have even sued antivirus companies for blocking adware.[29]

Programs that have been developed to detect, quarantine, and remove advertisement-displaying malware, include: Ad-Aware, Malwarebytes' Anti-Malware, Spyware Doctor and Spybot - Search & Destroy. In addition, almost all commercial antivirus software currently detect adware and spyware, or offer a separate detection module.[30]

A new wrinkle is adware (using stolen certificates) that disables anti-malware and virus protection; technical remedies are available.[29]

Adware has also been discovered in certain low-cost Android devices, particularly those made by small Chinese firms running on Allwinner systems-on-chip. There are even cases where adware code is embedded deep into files stored on the /system and boot partitions, to which removal involves extensive (and complex) modifications to the firmware.[31]

4.2.3 See also

- Amonetize
- Malvertising
- Typhoid adware

4.2.4 Notes

[1] For example, in 2007 Microsoft changed its productivity suite Microsoft Works to be advertising-supported.[4] Works was subsequently replaced with the Microsoft Office 2010 software suite operating in a "starter" mode that included advertisements.[5] As of 2012, this product is also being phased out and replaced with Office Web Apps.[6]

[2] Formed in 2008 following Microsoft's acquisition of digital marketing company aQuantive.[10]

[3] A workshop held by the Federal Trade Commission in 2005 asked representatives of the computer, electronic advertising, and anti-spyware product industries, as well as representatives of trade associations, government agencies, consumer and privacy advocacy groups, to try and define adware and its relation to spyware, and did not find a clear consensus.[27]

4.2.5 References

[1] Tulloch, Mitch (2003). Koch, Jeff; Haynes, Sandra, eds. *Microsoft Encyclopedia of Security*. Redmond, Washington: Microsoft Press. p. 16. ISBN 0-7356-1877-1. Any software that installs itself on your system without your knowledge and displays advertisements when the user browses the Internet.

[2] Braue, David (4 September 2008). "Feature: Ad-supported software". ZDNet. Retrieved 4 December 2012.

[3] Hayes Weier, Mary (5 May 2007). "Businesses Warm To No-Cost, Ad-Supported Software". *Information Week*. Retrieved 4 December 2012.

[4] Foley, Mary Jo (30 July 2007). "Microsoft Works to become a free, ad-funded product". Retrieved 4 December 2012.

[5] Foley, Mary Jo (9 October 2009). "Microsoft adds an 'Office Starter' edition to its distribution plans". ZDNet. Retrieved 4 December 2012.

[6] Foley, Mary Jo (21 June 2012). "Microsoft begins phasing out Starter edition of its Office suite". ZDNet. Retrieved 4 December 2012.

[7] Levy, Ari (23 April 2012). "Ad-supported software reaches specialized audience". *SF Gate*. Retrieved 4 December 2012.

[8] Tung, Liam (11 March 2011). "Skype now free ad-supported software". iT News for Australian Business. Retrieved 4 December 2012.

[9] "Kindle, Wi-Fi, Graphite, 6" Display with New E Ink Pearl Technology — includes Special Offers & Sponsored Screensavers". *Amazon.com*. Amazon.com, Inc. Retrieved 4 August 2011.

[10] "Microsoft Advertising Historical Timeline". Microsoft Advertising. September 2008. Retrieved 20 November 2012.

[11] "Windows 8 Ads in Apps". Microsoft Advertising. Retrieved 20 November 2012.

[12] Kim, Stephen (1 October 2012). "Microsoft Advertising Unveils New Windows 8 Ads in Apps Concepts with Agency Partners at Advertising Week 2012". Microsoft. Retrieved 20 November 2012.

[13] Fried, Ina. "Microsoft eyes making desktop apps free". *CNET*. Archived from the original on 14 November 2005. Retrieved 20 November 2012.

[14] Teeter, Ryan; Karl Barksdale (9 February 2011). *Google Apps For Dummies*. pp. 3–27. ISBN 1-118-05240-4.

[15] 17 January 2011 by Jolie O'Dell 203 (17 January 2011). "Facebook's Ad Revenue Hit $1.86B for 2010". Mashable.com. Retrieved 21 December 2011.

[16] Womack, Brian (20 September 2011). "Facebook Revenue Will Reach $4.27 Billion, EMarketer Says". Bloomberg. Retrieved 21 December 2011.

[17] Foley, Mary Jo (3 May 2007). "Meet Microsoft, the advertising company". ZDNet. Retrieved 20 November 2012.

[18] Majoras, Deborah Platt. "FTC Staff Report. Monitoring Software on Your PC: Spyware, Adware, and Other Software" (PDF). Federal Trade Commission. Retrieved 4 April 2005.

[19] National Cyber Security Alliance. "Malware & Botnets". StaySafeOnline.org. Retrieved 4 December 2012. The terms 'spyware' and 'adware' apply to several different [malware] technologies...

[20] "Viruses and other forms of malicious software". Princeton University Office of Information Technology. 5 July 2012. Retrieved 4 December 2012. malware also includes worms, spyware and adware.

[21] Vincentas (11 July 2013). "Adware in SpyWareLoop.com". Spyware Loop. Retrieved 27 July 2013.

[22] "Malware from A to Z". Lavasoft. Retrieved 4 December 2012. [Adware] delivers advertising content potentially in a manner or context that may be unexpected and unwanted by users.

[23] National Cyber Security Alliance. "Data Privacy Day Glossary". StaySafeOnline.org. Retrieved 4 December 2012. Adware: type of malware that allows popup ads on a computer system, ultimately taking over a user's Internet browsing.

[24] "Spyware, Adware and Malware — Advice for networks and network users". RM Education. Retrieved 4 December 2012. [Adware] tend[s] to be more of an irritant than do actual damage to your system, but [is] an unwanted presence nonetheless.

[25] "McAfee, Inc. Names Most Dangerous Celebrities in Cyberspace". McAfee. Retrieved 4 December 2012. online threats, such as spyware, spam, phishing, adware, viruses and other malware... Copy available at Bloomberg.

[26] Stern, Jerry. "Spyware, Adware, Malware, Thief: Creating Business Income from Denial of Service and Fraud" (PDF). *ASPects, Newsletter of the Association of Shareware Professionals*. Association of Software Professionals. Archived from the original (PDF) on 2012-09-17. Adware has become a bad word, linked to spyware and privacy violations by everyone except the publishers of the products... [it was] a good thing ten or fifteen years ago, and [is] bad now... [t]he lines for adware are even being blended into virus and trojan territory.

[27] *Spyware Workshop: Monitoring Software On Your Personal Computer: Spyware, Adware and Other Software*. Federal Trade Commission. March 2005. p. 2. Retrieved 4 December 2012.

[28] Schwabach, Aaron (2005). *Internet and the Law: Technology, Society, and Compromises*. ABC-CLIO. p. 10. ISBN 978-1-85109-731-9. Retrieved 4 December 2012.

[29] Casey, Henry T. (25 November 2015). "Latest adware disables antivirus software". *Tom's Guide*. Yahoo.com. Retrieved 25 November 2015.

[30] Honeycutt, Jerry (20 April 2004). "How to protect your computer from Spyware and Adware". *Microsoft.com*. Microsoft corporation. Archived from the original on 7 February 2006.

[31] "Decompile: Technical analysis of the Trojan". Cheetah Mobile. 9 November 2015. Retrieved 7 December 2015.

4.3 Spyware

Spyware is software that aims to gather information about a person or organization without their knowledge and that may send such information to another entity without the consumer's consent, or that asserts control over a computer without the consumer's knowledge.[1]

"Spyware" is mostly classified into four types: system monitors, trojans, adware, and tracking cookies.[2] Spyware is mostly used for the purposes of tracking and storing Internet users' movements on the Web and serving up pop-up ads to Internet users.

Whenever spyware is used for malicious purposes, its presence is typically hidden from the user and can be difficult to detect. Some spyware, such as keyloggers, may be installed by the owner of a shared, corporate, or public computer intentionally in order to monitor users.

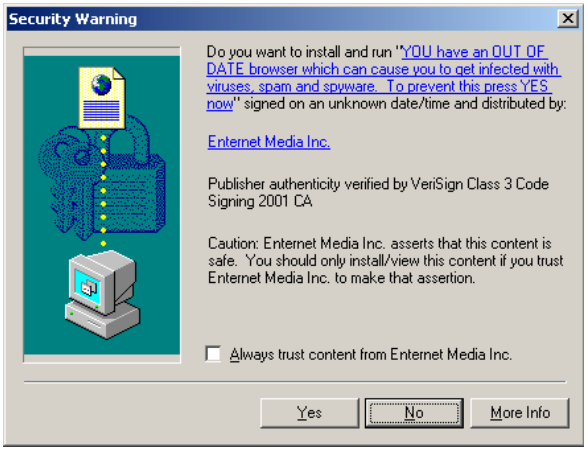
While the term *spyware* suggests software that monitors a user's computing, the functions of spyware can extend beyond simple monitoring. Spyware can collect almost any type of data, including personal information like internet surfing habits, user logins, and bank or credit account information. Spyware can also interfere with user control of a computer by installing additional software or redirecting web browsers. Some spyware can change computer settings, which can result in slow Internet connection speeds, un-authorized changes in browser settings, or changes to software settings.

Sometimes, spyware is included along with genuine software, and may come from a malicious website or may have been added to the intentional functionality of genuine software (see the paragraph about Facebook below). In response to the emergence of spyware, a small industry has sprung up dealing in anti-spyware software. Running anti-spyware software has become a widely recognized element of computer security practices, especially for computers running Microsoft Windows. A number of jurisdictions have passed anti-spyware laws, which usually target any software that is surreptitiously installed to control a user's computer.

In German-speaking countries, spyware used or made by the government is called *govware* by computer experts (in common parlance: *Regierungstrojaner*, literally 'Government Trojan'). Govware is typically a trojan horse software used to intercept communications from the target computer. Some countries like Switzerland and Germany have a legal framework governing the use of such software.[3][4] In the US, the term policeware has been used for similar purposes.[5]

Use of the term "spyware" has eventually declined as the practice of tracking users has been pushed ever further into the mainstream by major websites and data mining companies; these generally break no known laws and compel users to be tracked, not by fraudulent practices per se, but by the default settings created for users and the language of terms-of-service agreements. As one documented example, on March 7, 2011, CBS/Cnet News reported on a Wall Street Journal analysis revealing the practice of Facebook and other websites of tracking users' browsing activity, linked to their identity, far beyond users' visit and activity within the Facebook site itself. The report stated "Here's how it works. You go to Facebook, you log in, you spend some time there, and then ... you move on without logging out. Let's say the next site you go to is New York Times. Those buttons, without you clicking on them, have just reported back to Facebook and Twitter that you went there and also your identity within those accounts. Let's say you moved on to something like a site about depression. This one also has a tweet button, a Google widget, and those, too, can report back who you are and that you went there." The WSJ analysis was researched by Brian Kennish, founder of Disconnect, Inc.[6]

4.3.1 Routes of infection



Malicious websites attempt to install spyware on readers' computers.

Spyware does not necessarily spread in the same way as a virus or worm because infected systems generally do not attempt to transmit or copy the software to other computers. Instead, spyware installs itself on a system by deceiving the user or by exploiting software vulnerabilities.

Most spyware is installed without users' knowledge, or by using deceptive tactics. Spyware may try to deceive users by bundling itself with desirable software. Other common tactics are using a Trojan horse, spy gadgets that look like normal devices but turn out to be something else, such as a USB Keylogger. These devices actually are connected to the device as memory units but are capable of recording

each stroke made on the keyboard. Some spyware authors infect a system through security holes in the Web browser or in other software. When the user navigates to a Web page controlled by the spyware author, the page contains code which attacks the browser and forces the download and installation of spyware.

The installation of spyware frequently involves Internet Explorer. Its popularity and history of security issues have made it a frequent target. Its deep integration with the Windows environment make it susceptible to attack into the Windows operating system. Internet Explorer also serves as a point of attachment for spyware in the form of Browser Helper Objects, which modify the browser's behavior to add toolbars or to redirect traffic.

4.3.2 Effects and behaviors

A spyware program is rarely alone on a computer: an affected machine usually has multiple infections. Users frequently notice unwanted behavior and degradation of system performance. A spyware infestation can create significant unwanted CPU activity, disk usage, and network traffic. Stability issues, such as applications freezing, failure to boot, and system-wide crashes are also common. Spyware, which interferes with networking software, commonly causes difficulty connecting to the Internet.

In some infections, the spyware is not even evident. Users assume in those situations that the performance issues relate to faulty hardware, Windows installation problems, or another infection. Some owners of badly infected systems resort to contacting technical support experts, or even buying a new computer because the existing system "has become too slow". Badly infected systems may require a clean reinstallation of all their software in order to return to full functionality.

Moreover, some types of spyware disable software firewalls and anti-virus software, and/or reduce browser security settings, which further open the system to further opportunistic infections. Some spyware disables or even removes competing spyware programs, on the grounds that more spyware-related annoyances make it even more likely that users will take action to remove the programs.[7]

Keyloggers are sometimes part of malware packages downloaded onto computers without the owners' knowledge. Some keyloggers software is freely available on the internet while others are commercial or private applications. Most keyloggers allow not only keyboard keystrokes to be captured but also are often capable of collecting screen captures from the computer.

A typical Windows user has administrative privileges, mostly for convenience. Because of this, any program the user runs has unrestricted access to the system. As with other operating systems, Windows users are able to follow the principle of least privilege and use non-administrator accounts. Alternatively, they can also reduce the privileges of specific vulnerable Internet-facing processes such as Internet Explorer.

Since Windows Vista, by default, a computer administrator runs everything under limited user privileges. When a program requires administrative privileges, a User Account Control pop-up will prompt the user to allow or deny the action. This improves on the design used by previous versions of Windows.

4.3.3 Remedies and prevention

See also: Computer virus § Virus removal

As the spyware threat has worsened, a number of techniques have emerged to counteract it. These include programs designed to remove or block spyware, as well as various user practices which reduce the chance of getting spyware on a system.

Nonetheless, spyware remains a costly problem. When a large number of pieces of spyware have infected a Windows computer, the only remedy may involve backing up user data, and fully reinstalling the operating system. For instance, some spyware cannot be completely removed by Symantec, Microsoft, PC Tools.

Anti-spyware programs

See also: Category:Spyware removal

Many programmers and some commercial firms have released products dedicated to remove or block spyware. Programs such as PC Tools' Spyware Doctor, Lavasoft's *Ad-Aware SE* and Patrick Kolla's *Spybot - Search & Destroy* rapidly gained popularity as tools to remove, and in some cases intercept, spyware programs. On December 16, 2004, Microsoft acquired the *GIANT AntiSpyware* software,[8] rebranding it as *Windows AntiSpyware beta* and releasing it as a free download for Genuine Windows XP and Windows 2003 users. (In 2006 it was renamed Windows Defender).

Major anti-virus firms such as Symantec, PC Tools, McAfee and Sophos have also added anti-spyware features to their existing anti-virus products. Early on, anti-virus firms expressed reluctance to add anti-spyware functions, citing lawsuits brought by spyware authors against the authors of web sites and programs which described their products as "spyware". However, recent versions of these ma-

jor firms' home and business anti-virus products do include anti-spyware functions, albeit treated differently from viruses. Symantec Anti-Virus, for instance, categorizes spyware programs as "extended threats" and now offers real-time protection against these threats.

How anti-spyware software works

Anti-spyware programs can combat spyware in two ways:

1. They can provide real-time protection in a manner similar to that of anti-virus protection: they scan all incoming network data for spyware and blocks any threats it detects.

2. Anti-spyware software programs can be used solely for detection and removal of spyware software that has already been installed into the computer. This kind of anti-spyware can often be set to scan on a regular schedule.

Such programs inspect the contents of the Windows registry, operating system files, and installed programs, and remove files and entries which match a list of known spyware. Real-time protection from spyware works identically to real-time anti-virus protection: the software scans disk files at download time, and blocks the activity of components known to represent spyware. In some cases, it may also intercept attempts to install start-up items or to modify browser settings. Earlier versions of anti-spyware programs focused chiefly on detection and removal. Javacool Software's SpywareBlaster, one of the first to offer real-time protection, blocked the installation of ActiveX-based spyware.

Like most anti-virus software, many anti-spyware/adware tools require a frequently updated database of threats. As new spyware programs are released, anti-spyware developers discover and evaluate them, adding to the list of known spyware, which allows the software to detect and remove new spyware. As a result, anti-spyware software is of limited usefulness without regular updates. Updates may be installed automatically or manually.

A popular generic spyware removal tool used by those that requires a certain degree of expertise is HijackThis, which scans certain areas of the Windows OS where spyware often resides and presents a list with items to delete manually. As most of the items are legitimate windows files/registry entries it is advised for those who are less knowledgeable on this subject to post a HijackThis log on the numerous antispyware sites and let the experts decide what to delete.

If a spyware program is not blocked and manages to get itself installed, it may resist attempts to terminate or uninstall it. Some programs work in pairs: when an anti-spyware scanner (or the user) terminates one running process, the other one respawns the killed program. Likewise, some spyware will detect attempts to remove registry keys and immediately add them again. Usually, booting the infected computer in safe mode allows an anti-spyware program a better chance of removing persistent spyware. Killing the process tree may also work.

Security practices

To detect spyware, computer users have found several practices useful in addition to installing anti-spyware programs. Many users have installed a web browser other than Internet Explorer, such as Mozilla Firefox or Google Chrome. Though no browser is completely safe, Internet Explorer is at a greater risk for spyware infection due to its large user base as well as vulnerabilities such as ActiveX.

Some ISPs—particularly colleges and universities—have taken a different approach to blocking spyware: they use their network firewalls and web proxies to block access to Web sites known to install spyware. On March 31, 2005, Cornell University's Information Technology department released a report detailing the behavior of one particular piece of proxy-based spyware, *Marketscore*, and the steps the university took to intercept it.[9] Many other educational institutions have taken similar steps.

Individual users can also install firewalls from a variety of companies. These monitor the flow of information going to and from a networked computer and provide protection against spyware and malware. Some users install a large hosts file which prevents the user's computer from connecting to known spyware-related web addresses. Spyware may get installed via certain shareware programs offered for download. Downloading programs only from reputable sources can provide some protection from this source of attack.[10]

4.3.4 Applications

"Stealware" and affiliate fraud

A few spyware vendors, notably 180 Solutions, have written what the *New York Times* has dubbed "stealware", and what spyware researcher Ben Edelman terms *affiliate fraud*, a form of click fraud. Stealware diverts the payment of affiliate marketing revenues from the legitimate affiliate to the spyware vendor.

Spyware which attacks affiliate networks places the spyware operator's affiliate tag on the user's activity – replacing any other tag, if there is one. The spyware operator

is the only party that gains from this. The user has their choices thwarted, a legitimate affiliate loses revenue, networks' reputations are injured, and vendors are harmed by having to pay out affiliate revenues to an "affiliate" who is not party to a contract.[11] Affiliate fraud is a violation of the terms of service of most affiliate marketing networks. As a result, spyware operators such as 180 Solutions have been terminated from affiliate networks including LinkShare and ShareSale. Mobile devices can also be vulnerable to chargeware, which manipulates users into illegitimate mobile charges.

Identity theft and fraud

In one case, spyware has been closely associated with identity theft.[12] In August 2005, researchers from security software firm Sunbelt Software suspected the creators of the common CoolWebSearch spyware had used it to transmit "chat sessions, user names, passwords, bank information, etc.";[13] however it turned out that "it actually (was) its own sophisticated criminal little trojan that's independent of CWS."[14] This case is currently under investigation by the FBI.

The Federal Trade Commission estimates that 27.3 million Americans have been victims of identity theft, and that financial losses from identity theft totaled nearly $48 billion for businesses and financial institutions and at least $5 billion in out-of-pocket expenses for individuals.[15]

Digital rights management

Some copy-protection technologies have borrowed from spyware. In 2005, Sony BMG Music Entertainment was found to be using rootkits in its XCP digital rights management technology[16] Like spyware, not only was it difficult to detect and uninstall, it was so poorly written that most efforts to remove it could have rendered computers unable to function. Texas Attorney General Greg Abbott filed suit,[17] and three separate class-action suits were filed.[18] Sony BMG later provided a workaround on its website to help users remove it.[19]

Beginning on April 25, 2006, Microsoft's Windows Genuine Advantage Notifications application[20] was installed on most Windows PCs as a "critical security update". While the main purpose of this deliberately uninstallable application is to ensure the copy of Windows on the machine was lawfully purchased and installed, it also installs software that has been accused of "phoning home" on a daily basis, like spyware.[21][22] It can be removed with the RemoveWGA tool.

Personal relationships

Spyware has been used to monitor electronic activities of partners in intimate relationships. At least one software package, Loverspy, was specifically marketed for this purpose. Depending on local laws regarding communal/marital property, observing a partner's online activity without their consent may be illegal; the author of Loverspy and several users of the product were indicted in California in 2005 on charges of wiretapping and various computer crimes.[23]

Browser cookies

Anti-spyware programs often report Web advertisers' HTTP cookies, the small text files that track browsing activity, as spyware. While they are not always inherently malicious, many users object to third parties using space on their personal computers for their business purposes, and many anti-spyware programs offer to remove them.[24]

4.3.5 Examples

These common spyware programs illustrate the diversity of behaviors found in these attacks. Note that as with computer viruses, researchers give names to spyware programs which may not be used by their creators. Programs may be grouped into "families" based not on shared program code, but on common behaviors, or by "following the money" of apparent financial or business connections. For instance, a number of the spyware programs distributed by Claria are collectively known as "Gator". Likewise, programs that are frequently installed together may be described as parts of the same spyware package, even if they function separately.

- **CoolWebSearch**, a group of programs, takes advantage of Internet Explorer vulnerabilities. The package directs traffic to advertisements on Web sites including *coolwebsearch.com*. It displays pop-up ads, rewrites search engine results, and alters the infected computer's hosts file to direct DNS lookups to these sites.[25]
- **FinFisher**, sometimes called **FinSpy** is a high-end surveillance suite sold to law enforcement and intelligence agencies. Support services such as training and technology updates are part of the package.[26]
- **HuntBar**, aka **WinTools** or **Adware.Websearch**, was installed by an ActiveX drive-by download at affiliate Web sites, or by advertisements displayed by other spyware programs—an example of how spyware can install more spyware. These programs add toolbars to IE, track aggregate browsing behavior, redirect affiliate references, and display advertisements.[27][28]

- **Internet Optimizer**, also known as **DyFuCa**, redirects Internet Explorer error pages to advertising. When users follow a broken link or enter an erroneous URL, they see a page of advertisements. However, because password-protected Web sites (HTTP Basic authentication) use the same mechanism as HTTP errors, Internet Optimizer makes it impossible for the user to access password-protected sites.[29]

- Spyware such as **Look2Me** hides inside system-critical processes and start up even in safe mode. With no process to terminate they are harder to detect and remove, which is a combination of both spyware and a rootkit. Rootkit technology is also seeing increasing use,[30] as newer spyware programs also have specific countermeasures against well known anti-malware products and may prevent them from running or being installed, or even uninstall them.

- **Movieland**, also known as **Moviepass.tv** and **Popcorn.net**, is a movie download service that has been the subject of thousands of complaints to the Federal Trade Commission (FTC), the Washington State Attorney General's Office, the Better Business Bureau, and other agencies. Consumers complained they were held hostage by a cycle of oversized pop-up windows demanding payment of at least $29.95, claiming that they had signed up for a three-day free trial but had not cancelled before the trial period was over, and were thus obligated to pay.[31][32] The FTC filed a complaint, since settled, against Movieland and eleven other defendants charging them with having "engaged in a nationwide scheme to use deception and coercion to extract payments from consumers."[33]

- **WeatherStudio** has a plugin that displays a window-panel near the *bottom* of a browser window. The official website notes that it is easy to remove (uninstall) WeatherStudio from a computer, using its own uninstall-program, such as under C:\Program Files\WeatherStudio. Once WeatherStudio is removed, a browser returns to the prior display appearance, without the need to modify the browser settings.

- **Zango** (formerly **180 Solutions**) transmits detailed information to advertisers about the Web sites which users visit. It also alters HTTP requests for affiliate advertisements linked from a Web site, so that the advertisements make unearned profit for the 180 Solutions company. It opens pop-up ads that cover over the Web sites of competing companies (as seen in their [Zango End User License Agreement]).[11]

- **Zlob trojan**, or just **Zlob**, downloads itself to a computer via an ActiveX codec and reports information back to **Control Server**. Some information can be the search-history, the Websites visited, and even keystrokes. More recently, Zlob has been known to hijack routers set to defaults.[34]

4.3.6 History and development

The first recorded use of the term spyware occurred on October 16, 1995 in a Usenet post that poked fun at Microsoft's business model.[35] *Spyware* at first denoted *software* meant for espionage purposes. However, in early 2000 the founder of Zone Labs, Gregor Freund, used the term in a press release for the ZoneAlarm Personal Firewall.[36] Later in 2000, a parent using ZoneAlarm was alerted to the fact that "Reader Rabbit," educational software marketed to children by the Mattel toy company, was surreptitiously sending data back to Mattel.[37] Since then, "spyware" has taken on its present sense.

According to a 2005 study by AOL and the National Cyber-Security Alliance, 61 percent of surveyed users' computers were infected with form of spyware. 92 percent of surveyed users with spyware reported that they did not know of its presence, and 91 percent reported that they had not given permission for the installation of the spyware.[38] As of 2006, spyware has become one of the preeminent security threats to computer systems running Microsoft Windows operating systems. Computers on which Internet Explorer (IE) is the primary browser are particularly vulnerable to such attacks, not only because IE is the most widely used,[39] but because its tight integration with Windows allows spyware access to crucial parts of the operating system.[39][40]

Before Internet Explorer 6 SP2 was released as part of Windows XP Service Pack 2, the browser would automatically display an installation window for any ActiveX component that a website wanted to install. The combination of user ignorance about these changes, and the assumption by Internet Explorer that all ActiveX components are benign, helped to spread spyware significantly. Many spyware components would also make use of exploits in JavaScript, Internet Explorer and Windows to install without user knowledge or permission.

The Windows Registry contains multiple sections where modification of key values allows software to be executed automatically when the operating system boots. Spyware can exploit this design to circumvent attempts at removal. The spyware typically will link itself from each location in the registry that allows execution. Once running, the spyware will periodically check if any of these links are removed. If so, they will be automatically restored. This ensures that the spyware will execute when the operating system is booted, even if some (or most) of the registry links are removed.

4.3.7 Programs distributed with spyware

- Kazaa[41]
- Morpheus[42]
- WeatherBug[43]
- WildTangent[44][45]

Programs formerly distributed with spyware

- AOL Instant Messenger[44] (AOL Instant Messenger still packages Viewpoint Media Player, and WildTangent)
- DivX[46]
- FlashGet[47][48][49][50][51][52]
- magicJack[53]

4.3.8 Rogue anti-spyware programs

See also: List of rogue security software, List of fake anti-spyware programs, and Rogue software

Malicious programmers have released a large number of rogue (fake) anti-spyware programs, and widely distributed Web banner ads can warn users that their computers have been infected with spyware, directing them to purchase programs which do not actually remove spyware—or else, may add more spyware of their own.[54][55]

The recent proliferation of fake or spoofed antivirus products that bill themselves as antispyware can be troublesome. Users may receive popups prompting them to install them to protect their computer, when it will in fact add spyware. This software is called rogue software. It is recommended that users do not install any freeware claiming to be anti-spyware unless it is verified to be legitimate. Some known offenders include:

- AntiVirus 360
- Antivirus 2009
- AntiVirus Gold
- ContraVirus
- MacSweeper
- Pest Trap
- PSGuard
- Spy Wiper
- Spydawn
- Spylocked
- Spysheriff
- SpyShredder
- Spyware Quake
- SpywareStrike
- UltimateCleaner
- WinAntiVirus Pro 2006
- Windows Police Pro
- WinFixer[56]
- WorldAntiSpy

Fake antivirus products constitute 15 percent of all malware.[57]

On January 26, 2006, Microsoft and the Washington state attorney general filed suit against Secure Computer for its Spyware Cleaner product.[58]

4.3.9 Legal issues

Criminal law

Unauthorized access to a computer is illegal under computer crime laws, such as the U.S. Computer Fraud and Abuse Act, the U.K.'s Computer Misuse Act, and similar laws in other countries. Since owners of computers infected with spyware generally claim that they never authorized the installation, a *prima facie* reading would suggest that the promulgation of spyware would count as a criminal act. Law enforcement has often pursued the authors of other malware, particularly viruses. However, few spyware developers have been prosecuted, and many operate openly as strictly legitimate businesses, though some have faced lawsuits.[59][60]

Spyware producers argue that, contrary to the users' claims, users do in fact give consent to installations. Spyware that comes bundled with shareware applications may be described in the legalese text of an end-user license agreement (EULA). Many users habitually ignore these purported contracts, but spyware companies such as Claria say these demonstrate that users have consented.

Despite the ubiquity of EULAs agreements, under which a single click can be taken as consent to the entire text, relatively little caselaw has resulted from their use. It has been established in most common law jurisdictions that this type

of agreement can be a binding contract *in certain circumstances.*[61] This does not, however, mean that every such agreement is a contract, or that every term in one is enforceable.

Some jurisdictions, including the U.S. states of Iowa[62] and Washington,[63] have passed laws criminalizing some forms of spyware. Such laws make it illegal for anyone other than the owner or operator of a computer to install software that alters Web-browser settings, monitors keystrokes, or disables computer-security software.

In the United States, lawmakers introduced a bill in 2005 entitled the Internet Spyware Prevention Act, which would imprison creators of spyware.[64]

Administrative sanctions

US FTC actions The US Federal Trade Commission has sued Internet marketing organizations under the "unfairness doctrine"[65] to make them stop infecting consumers' PCs with spyware. In one case, that against Seismic Entertainment Productions, the FTC accused the defendants of developing a program that seized control of PCs nationwide, infected them with spyware and other malicious software, bombarded them with a barrage of pop-up advertising for Seismic's clients, exposed the PCs to security risks, and caused them to malfunction. Seismic then offered to sell the victims an "antispyware" program to fix the computers, and stop the popups and other problems that Seismic had caused. On November 21, 2006, a settlement was entered in federal court under which a $1.75 million judgment was imposed in one case and $1.86 million in another, but the defendants were insolvent[66]

In a second case, brought against CyberSpy Software LLC, the FTC charged that CyberSpy marketed and sold "RemoteSpy" keylogger spyware to clients who would then secretly monitor unsuspecting consumers' computers. According to the FTC, Cyberspy touted RemoteSpy as a "100% undetectable" way to "Spy on Anyone. From Anywhere." The FTC has obtained a temporary order prohibiting the defendants from selling the software and disconnecting from the Internet any of their servers that collect, store, or provide access to information that this software has gathered. The case is still in its preliminary stages. A complaint filed by the Electronic Privacy Information Center (EPIC) brought the RemoteSpy software to the FTC's attention.[67]

Netherlands OPTA An administrative fine, the first of its kind in Europe, has been issued by the Independent Authority of Posts and Telecommunications (OPTA) from the Netherlands. It applied fines in total value of Euro 1,000,000 for infecting 22 million computers. The spyware concerned is called DollarRevenue. The law articles that have been violated are art. 4.1 of the Decision on universal service providers and on the interests of end users; the fines have been issued based on art. 15.4 taken together with art. 15.10 of the Dutch telecommunications law.[68]

Civil law

Former New York State Attorney General and former Governor of New York Eliot Spitzer has pursued spyware companies for fraudulent installation of software.[69] In a suit brought in 2005 by Spitzer, the California firm Intermix Media, Inc. ended up settling, by agreeing to pay US$7.5 million and to stop distributing spyware.[70]

The hijacking of Web advertisements has also led to litigation. In June 2002, a number of large Web publishers sued Claria for replacing advertisements, but settled out of court.

Courts have not yet had to decide whether advertisers can be held liable for spyware that displays their ads. In many cases, the companies whose advertisements appear in spyware pop-ups do not directly do business with the spyware firm. Rather, they have contracted with an advertising agency, which in turn contracts with an online subcontractor who gets paid by the number of "impressions" or appearances of the advertisement. Some major firms such as Dell Computer and Mercedes-Benz have sacked advertising agencies that have run their ads in spyware.[71]

Libel suits by spyware developers

Litigation has gone both ways. Since "spyware" has become a common pejorative, some makers have filed libel and defamation actions when their products have been so described. In 2003, Gator (now known as Claria) filed suit against the website PC Pitstop for describing its program as "spyware".[72] PC Pitstop settled, agreeing not to use the word "spyware", but continues to describe harm caused by the Gator/Claria software.[73] As a result, other anti-spyware and anti-virus companies have also used other terms such as "**potentially unwanted programs**" or greyware to denote these products.

WebcamGate

Main article: Robbins v. Lower Merion School District

In the 2010 WebcamGate case, plaintiffs charged two suburban Philadelphia high schools secretly spied on students by surreptitiously and remotely activating webcams embedded in school-issued laptops the students were using at home, and therefore infringed on their privacy rights. The

school loaded each student's computer with LANrev's remote activation tracking software. This included the now-discontinued "TheftTrack". While TheftTrack was not enabled by default on the software, the program allowed the school district to elect to activate it, and to choose which of the TheftTrack surveillance options the school wanted to enable.[74]

TheftTrack allowed school district employees to secretly remotely activate a tiny webcam embedded in the student's laptop, above the laptop's screen. That allowed school officials to secretly take photos through the webcam, of whatever was in front of it and in its line of sight, and send the photos to the school's server. The LANrev software disabled the webcams for all other uses (*e.g.*, students were unable to use Photo Booth or video chat), so most students mistakenly believed their webcams did not work at all. In addition to webcam surveillance, TheftTrack allowed school officials to take screenshots, and send them to the school's server. In addition, LANrev allowed school officials to take snapshots of instant messages, web browsing, music playlists, and written compositions. The schools admitted to secretly snapping over 66,000 webshots and screenshots, including webcam shots of students in their bedrooms.[74][75][76]

4.3.10 See also

- Cyber spying
- Employee monitoring software
- Industrial espionage
- Malware
- Spy-phishing

4.3.11 Specific Variants

- Trojan:Win32/Meredrop
- Superfish

4.3.12 References

[1] FTC Report (2005). ""

[2] SPYWARE ""

[3] Basil Cupa, Trojan Horse Resurrected: On the Legality of the Use of Government Spyware (Govware), LISS 2013, pp. 419–428

[4] FAQ – Häufig gestellte Fragen

[5] Jeremy Reimer (July 20, 2007). "The tricky issue of spyware with a badge: meet 'policeware'". Ars Technica.

[6] Cooley, Brian (March 7, 2011). "'Like,' 'tweet' buttons divulge sites you visit: CNET News Video". *CNet News*. Retrieved March 7, 2011.

[7] Edelman, Ben; December 7, 2004 (updated February 8, 2005); Direct Revenue Deletes Competitors from Users' Disks; benedelman.com. Retrieved November 28, 2006.

[8] "http://www.microsoft.com/presspass/press/2004/dec04/12-16GIANTPR.mspx"

[9] Schuster, Steve. ""Blocking Marketscore: Why Cornell Did It". Archived from the original on February 14, 2007.". Cornell University, Office of Information Technologies. March 31, 2005.

[10] Vincentas (July 11, 2013). "Information About Spyware in SpyWareLoop.com". Spyware Loop. Retrieved July 27, 2013.

[11] Edelman, Ben (2004). "The Effect of 180solutions on Affiliate Commissions and Merchants". *Benedelman.org*. Retrieved November 14, 2006.

[12] Ecker, Clint (2005). *Massive spyware-based identity theft ring uncovered*. Ars Technica, August 5, 2005.

[13] Eckelberry, Alex. "Massive identity theft ring", *SunbeltBLOG,* August 4, 2005.

[14] Eckelberry, Alex. "Identity Theft? What to do?", *SunbeltBLOG,* August 8, 2005.

[15] FTC Releases Survey of Identity Theft in U.S. 27.3 Million Victims in Past 5 Years, Billions in Losses for Businesses and Consumers. Federal Trade Commission, September 3, 2003.

[16] Russinovich, Mark. "Sony, Rootkits and Digital Rights Management Gone Too Far,", *Mark's Blog,* October 31, 2005. Retrieved November 22, 2006.

[17] Press release from the Texas Attorney General's office, November 21, 2005; Attorney General Abbott Brings First Enforcement Action In Nation Against Sony BMG For Spyware Violations. Retrieved November 28, 2006.

[18] "Sony sued over copy-protected CDs; Sony BMG is facing three lawsuits over its controversial anti-piracy software", *BBC News,* November 10, 2005. Retrieved November 22, 2006.

[19] Information About XCP Protected CDs. Retrieved November 29, 2006.

[20] Microsoft.com – Description of the Windows Genuine Advantage Notifications application. Retrieved June 13, 2006.

[21] Weinstein, Lauren. Windows XP update may be classified as 'spyware', *Lauren Weinstein's Blog,* June 5, 2006. Retrieved June 13, 2006.

[22] Evers, Joris. Microsoft's antipiracy tool phones home daily, *CNET,* June 7, 2006. Retrieved August 31, 2014.

[23] "Creator and Four Users of Loverspy Spyware Program Indicted". Department of Justice. August 26, 2005. Retrieved November 21, 2014.

[24] "Tracking Cookie". Symantec. Retrieved 2013-04-28.

[25] ""CoolWebSearch". *Parasite information database*. Archived from the original on January 6, 2006. Retrieved September 4, 2008.

[26] Nicole Perlroth (August 30, 2012). "Software Meant to Fight Crime Is Used to Spy on Dissidents". *The New York Times*. Retrieved August 31, 2012.

[27] "CA Spyware Information Center – HuntBar". .ca.com. Retrieved September 11, 2010.

[28] "What is Huntbar or Search Toolbar?". Pchell.com. Retrieved September 11, 2010.

[29] ""InternetOptimizer". *Parasite information database*. Archived from the original on January 6, 2006. Retrieved September 4, 2008.

[30] Roberts, Paul F. "Spyware meets Rootkit Stealth". *eweek.com*. June 20, 2005.

[31] "FTC, Washington Attorney General Sue to Halt Unfair Movieland Downloads". Federal Trade Commission. August 15, 2006.

[32] "Attorney General McKenna Sues Movieland.com and Associates for Spyware". Washington State Office of the Attorney General. August 14, 2006.

[33] "Complaint for Permanent Injunction and Other Equitable Relief (PDF, 25 pages)" (PDF). Federal Trade Commission. August 8, 2006.

[34] PCMAG, New Malware changes router settings, *PC Magazine*, June 13, 2008. Archived July 15, 2011, at the Wayback Machine.

[35] Vossen, Roland (attributed); October 21, 1995; Win 95 Source code in c!! posted to rec..programmer; retrieved from groups.google.com November 28, 2006.

[36] Wienbar, Sharon. "The Spyware Inferno". *News.com*. August 13, 2004.

[37] Hawkins, Dana; "Privacy Worries Arise Over Spyware in Kids' Software". *U.S. News & World Report*. June 25, 2000 Archived November 3, 2013, at the Wayback Machine.

[38] "AOL/NCSA Online Safety Study". *America Online & The National Cyber Security Alliance*. 2005. Archived December 13, 2005, at the Wayback Machine.

[39] Spanbauer, Scott. "Is It Time to Ditch IE?". *Pcworld.com*. September 1, 2004

[40] Keizer, Gregg. "Analyzing IE At 10: Integration With OS Smart Or Not?". *TechWeb Technology News*. August 25, 2005. Archived September 29, 2007, at the Wayback Machine.

[41] Edelman, Ben (2004). "Claria License Agreement Is Fifty Six Pages Long". Retrieved July 27, 2005.

[42] Edelman, Ben (2005). "Comparison of Unwanted Software Installed by P2P Programs". Retrieved July 27, 2005.

[43] ""WeatherBug". *Parasite information database*. Archived from the original on February 6, 2005. Retrieved September 4, 2008.

[44] "Adware.WildTangent". Sunbelt Malware Research Labs. June 12, 2008. Retrieved September 4, 2008.

[45] "Winpipe". Sunbelt Malware Research Labs. June 12, 2008. Retrieved September 4, 2008. It is possible that this spyware is distributed with the adware bundler WildTangent or from a threat included in that bundler.

[46] "How Did I Get Gator?". *PC Pitstop*. Retrieved July 27, 2005.

[47] "eTrust Spyware Encyclopedia – FlashGet". *Computer Associates*. Retrieved July 27, 2005. Archived May 5, 2007, at the Wayback Machine.

[48] "Jotti's malware scan of FlashGet 3". Virusscan.jotti.org. Retrieved September 11, 2010.

[49] VirusTotal scan of FlashGet 3.

[50] "Jotti's malware scan of FlashGet 1.96". Virusscan.jotti.org. Retrieved September 11, 2010.

[51] VirusTotal scan of FlashGet 1.96.

[52] Some caution is required since FlashGet 3 EULA makes mention of Third Party Software, but does not name any third party producer of software. However, a scan with SpyBot Search & Destroy, performed on November 20, 2009 after installing FlashGet 3 did not show any malware on an already anti-spyware immunized system (by SpyBot and SpywareBlaster).

[53] "Gadgets boingboing.net, "MagicJack's EULA says it will spy on you and force you into arbitration"". Gadgets.boingboing.net. April 14, 2008. Retrieved September 11, 2010.

[54] Roberts, Paul F. (May 26, 2005). "Spyware-Removal Program Tagged as a Trap". *eWeek*. Retrieved September 4, 2008.

[55] Howes, Eric L. "The Spyware Warrior List of Rogue/Suspect Anti-Spyware Products & Web Sites". Retrieved July 10, 2005.

[56] Also known as WinAntiVirusPro, ErrorSafe, SystemDoctor, WinAntiSpyware, AVSystemCare, WinAntiSpy, Windows Police Pro, Performance Optimizer, StorageProtector, PrivacyProtector, WinReanimator, DriveCleaner, WinspywareProtect, PCTurboPro, FreePCSecure, ErrorProtector, SysProtect, WinSoftware, XPAntivirus, Personal Antivirus, Home Antivirus 20xx, VirusDoctor, and ECsecure

[57] Elinor Mills (April 27, 2010). "Google: Fake antivirus is 15 percent of all malware". CNET. Retrieved 2011-11-05.

[58] McMillan, Robert. Antispyware Company Sued Under Spyware Law. *PC World,* January 26, 2006.

[59] "Lawsuit filed against 180solutions". *zdnet.com* September 13, 2005

[60] Hu, Jim. "180solutions sues allies over adware". *news.com* July 28, 2004

[61] Coollawyer; 2001–2006; Privacy Policies, Terms and Conditions, Website Contracts, Website Agreements; coollawyer.com. Retrieved November 28, 2006.

[62] "CHAPTER 715 Computer Spyware and Malware Protection". *nxtsearch.legis.state.ia.us.* Retrieved May 11, 2011.

[63] Chapter 19.270 RCW: Computer spyware. *apps.leg.wa.gov*. Retrieved November 14, 2006.

[64] Gross, Grant. US lawmakers introduce I-Spy bill. *InfoWorld,* March 16, 2007. Retrieved March 24, 2007.

[65] See Federal Trade Commission v. Sperry & Hutchinson Trading Stamp Co.

[66] *FTC Permanently Halts Unlawful Spyware Operations* (FTC press release with links to supporting documents; archived copy); see also *FTC cracks down on spyware and PC hijacking, but not true lies*, Micro Law, IEEE MICRO (Jan.-Feb. 2005), also available at IEEE Xplore.

[67] See *Court Orders Halt to Sale of Spyware* (FTC press release November 17, 2008, with links to supporting documents).

[68] OPTA, "Besluit van het college van de Onafhankelijke Post en Telecommunicatie Autoriteit op grond van artikel 15.4 juncto artikel 15.10 van de Telecommunicatiewet tot oplegging van boetes ter zake van overtredingen van het gestelde bij of krachtens de Telecommunicatiewet" from November 5, 2007, http://opta.nl/download/202311+boete+verspreiding+ongewenste+software.pdf

[69] "State Sues Major "Spyware" Distributor" (Press release). Office of New York State Attorney General. April 28, 2005. Archived from the original on January 10, 2009. Retrieved September 4, 2008. Attorney General Spitzer today sued one of the nation's leading internet marketing companies, alleging that the firm was the source of "spyware" and "adware" that has been secretly installed on millions of home computers.

[70] Gormley, Michael. "Intermix Media Inc. says it is settling spyware lawsuit with N.Y. attorney general". *Yahoo! News*. June 15, 2005. Archived from the original on June 22, 2005.

[71] Gormley, Michael (June 25, 2005). "Major advertisers caught in spyware net". *USA Today*. Retrieved September 4, 2008.

[72] Festa, Paul. "See you later, anti-Gators?". *News.com*. October 22, 2003.

[73] "Gator Information Center". *pcpitstop.com* November 14, 2005.

[74] "Initial LANrev System Findings", LMSD Redacted Forensic Analysis, L-3 Services – prepared for Ballard Spahr (LMSD's counsel), May 2010. Retrieved August 15, 2010. Archived June 15, 2010, at the Wayback Machine.

[75] Doug Stanglin (February 18, 2010). "School district accused of spying on kids via laptop webcams". *USA Today*. Retrieved February 19, 2010.

[76] "Suit: Schools Spied on Students Via Webcam". *CBS NEWS*. March 8, 2010.

4.3.13 External links

- Home Computer Security – Carnegie Mellon Software Institute
- OnGuard Online.gov – How to Secure Your Computer

4.3.14 Categories

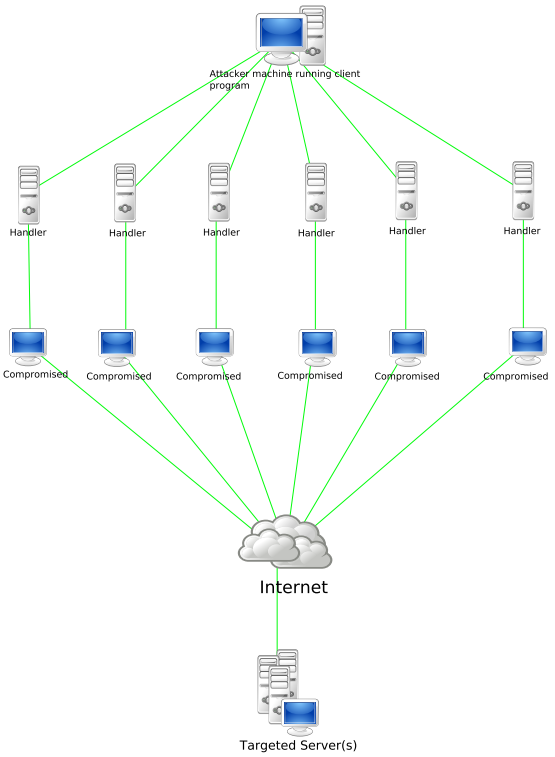
4.4 Botnet

A **botnet** is a number of Internet-connected computers communicating with other similar machines in which components located on networked computers communicate and coordinate their actions by command and control (C&C) or by passing messages to one another (C&C might be built into the botnet as P2P).[1] Botnets have been used many times to send spam email or participate in distributed denial-of-service attacks. The word botnet is a combination of the words robot and network. The term is usually used with a negative or malicious connotation.

4.4.1 Applications of botnets

Legal applications

Most of the time when botnets are in the legal area are commonly used for Distributed computing which is a field of



Stacheldraht botnet diagram showing a DDoS attack. (Note this is also an example of a type of client-server model of a botnet.)

computer science that studies distributed systems. A distributed system is a software system in which components located on networked computers communicate and coordinate their actions by passing messages. The components interact with each other in order to achieve a common goal. Three significant characteristics of distributed systems are: concurrency of components, lack of a global clock, and independent failure of components. A command and control may be present in the distributed computing but no zombie computer is present in this type of system.[2]

Illegal applications

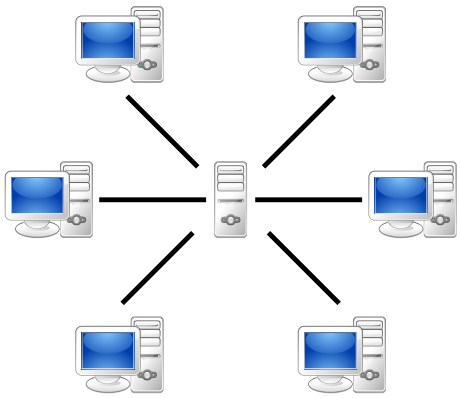
Botnets sometimes compromise computers whose security defenses have been breached and control ceded to a third party. Each such compromised device, known as a "bot," is created when a computer is penetrated by software from a *malware* (malicious software) distribution. The controller of a botnet is able to direct the activities of these compromised computers through communication channels formed by standards-based network protocols such as IRC and Hypertext Transfer Protocol (HTTP).[3]

Botnets are increasingly rented out by cyber criminals as commodities for a variety of purposes.[4]

4.4.2 Architecture of a Botnet

The methods on which a botnets are built for communications. Botnet architecture evolved over time, and not all botnets exhibit the same topology for command and control. Advanced topology is more resilient to shutdown, enumeration or discovery. However, some topologies limit the marketability of the botnet to third parties. Typical botnet topologies are star, multi-server, hierarchical and random.

Client–server model



A network based on the client-server model, where individual clients request services and resources from centralized servers

The Client–server model appeared on the first types of botnets that appeared online and has usually been built on Internet Relay Chat or by using Domains or Websites which will have the commands listed for the botnet to be controlled. In IRC commands tend to be simpler and botnets tend to be smaller if built on an IRC network. Since IRC networks require low bandwidth and use simple methods for communication, they have been used to host botnets that tend to be simple in construction and have been used many times for coordinating DDoS attacks or spam campaigns while switching channels to avoid being taken down. Blocking certain keywords has sometimes proved effective in stopping a botnet based on IRC.

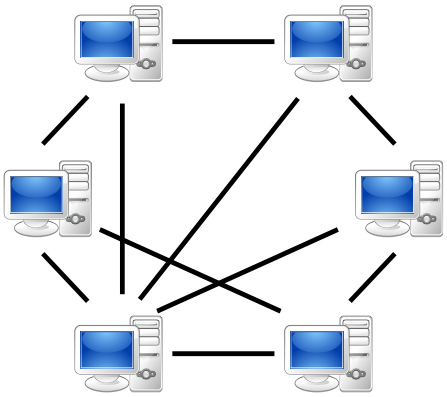
Most of the largest botnets that have been built tended to use domains rather than IRC in their construction.(see

Rustock botnet see also Srizbi botnet.) Almost always they have been hosted with bullet proof hosting services.(See Bulletproof hosting.) Since most of the time botnets based on the Client-server model have been taken down in a matter of time, hackers have moved toward P2P as an alternative to avoid botnet takedowns.

Botnet servers are typically redundant, linked for greater redundancy so as to reduce the threat of a takedown. Actual botnet communities usually consist of one or several controllers that rarely have highly developed command hierarchies; they rely on individual peer-to-peer relationships.[5]

The botnet server structure mentioned above has inherent vulnerabilities and problems. For example, finding one server with one botnet channel can often reveal the other servers, as well as their bots. A botnet server structure that lacks redundancy is vulnerable to at least the temporary disconnection of that server. However, recent IRC server software includes features to mask other connected servers and bots, eliminating that approach.

Peer-to-peer



A peer-to-peer (P2P) network in which interconnected nodes ("peers") share resources amongst each other without the use of a centralized administrative system

Since most of the time IRC networks and Domains can be taken down with time, hackers have moved on to P2P as a way to make it harder to be taken down. Some have even been known to use encryption as a way to secure or lock down the botnet from others, most of the time when they use encryption it is Public-Key encryption and has presented challenges in both implementing it and breaking it.(See Gameover ZeuS See also ZeroAccess botnet.)

Some newer botnets are almost entirely P2P. Command and control is embedded into the botnet rather than relying on external servers, thus avoiding any single point of failure and evading many countermeasures.[6] Commanders can be identified just through secure keys, and all data except the binary itself can be encrypted. For example, a spyware program may encrypt all suspected passwords with a public key that is hard-coded into it, or distributed with the bot software. Only with the private key (known only by the botnet operators) can the data captured by the bot be read.

In the P2P method of command and control the bot only tends to know a list of peers of which it can send commands to and that are passed on to other peers further down the botnet. The list tends to be around 256 peers which allows it to be small enough for it to allow commands to be quickly passed on to other peers and makes it harder to disrupt the operation of the botnet while allowing it to remain online if major numbers of peers are taken down in a takedown effort.

4.4.3 Core components of a botnet

There are several core components in a botnet which have been used. The main ones are listed below

Command and control

In the field of computer security, command and control (C&C) infrastructure consists of servers and other technical infrastructure used to control malware in general, and, in particular, botnets. Command and control servers may be either directly controlled by the malware operators, or themselves run on hardware compromised by malware. Fast-flux DNS can be used as a way to make it difficult to track down the control servers, which may change from day to day. Control servers may also hop from DNS domain to DNS domain, with domain generation algorithms being used to create new DNS names for controller servers.

In some cases, computer security experts have succeeded in destroying or subverting malware command and control networks, by, among other means, seizing servers or getting them cut off from the Internet, denying access to domains that were due to be used by malware to contact its C&C infrastructure, and, in some cases, breaking into the C&C network itself. In response to this, C&C operators have resorted to using techniques such as overlaying their C&C networks on other existing benign infrastructure such as IRC or Tor, using peer-to-peer networking systems that are not dependent on any fixed servers, and using public

key encryption to defeat attempts to break into or spoof the network.

Zombie computer

In computer science, a zombie computer is a computer connected to the Internet that has been compromised by a hacker, computer virus or trojan horse and can be used to perform malicious tasks of one sort or another under remote direction. Botnets of zombie computers are often used to spread e-mail spam and launch denial-of-service attacks. Most owners of zombie computers are unaware that their system is being used in this way. Because the owner tends to be unaware, these computers are metaphorically compared to zombies. A coordinated DDoS attack by multiple botnet machines also resembles a zombie horde attack.

4.4.4 Construction of a botnet

This example illustrates how a botnet is created and used for malicious gain

1. A hacker purchases or builds a Trojan and/or exploit kit and uses it to start infecting users' computers, whose payload is a malicious application—the *bot*.
2. The *bot* on the infected PC logs into a particular C&C server. (This allows the bot master to keep logs of how many bots are active and online.)
3. The bot master may then use the bots to gather keystrokes or use form grabbing to steal online credentials and may rent out the botnet as DDoS and/or spam as a service or sell the credentials online for a profit.
4. Depending on the quality and capability of the bots the value is increased or decreased.

Common features in a botnet

- Most botnets currently feature distributed denial-of-service attacks in which multiple systems submit as many requests as possible to a single Internet computer or service, overloading it and preventing it from servicing legitimate requests. An example is an attack on a victim's server. The victim's server is bombarded with requests by the bots, attempting to connect to the server therefore overloading it.
- Spyware is software which sends information to its creators about a user's activities – typically passwords, credit card numbers and other information that can be sold on the black market. Compromised machines that are located within a corporate network can be worth more to the bot herder, as they can often gain access to confidential corporate information. Several targeted attacks on large corporations aimed to steal sensitive information, such as the Aurora botnet.[7]
- E-mail spam are e-mail messages disguised as messages from people, but are either advertising, annoying, or malicious.
- Click fraud occurs when the user's computer visits websites without the user's awareness to create false web traffic for personal or commercial gain.
- Bitcoin Mining has been added to some of the more recent botnets have which include bitcoin mining[8] as a feature in order to generate profits for the operator of the botnet.

The botnet controller community features a constant and continuous struggle over who has the most bots, the highest overall bandwidth, and the most "high-quality" infected machines, like university, corporate, and even government machines.[9]

Organization

While botnets are often named after the malware that created them, multiple botnets typically use the same malware, but are operated by different entities.[10]

A botnet's originator (known as a "bot herder" or "bot master") can control the group remotely, usually through IRC or Domains, and often for criminal purposes. This is known as the **command-and-control (C&C)**. Though rare, more experienced botnet operators program command protocols from scratch. These protocols include a server program, a client program for operation, and the program that embeds the client on the victim's machine. These communicate over a network, using a unique encryption scheme for stealth and protection against detection or intrusion into the botnet.

A bot typically runs hidden and uses a covert channel (e.g. the RFC 1459 (IRC) standard, Twitter, or IM) to communicate with its C&C server. Generally, the perpetrator has compromised multiple systems using various tools (exploits, buffer overflows, as well as others; see also RPC). Newer bots can automatically scan their environment and propagate themselves using vulnerabilities and weak passwords. Generally, the more vulnerabilities a bot can scan and propagate through, the more valuable it becomes to a botnet controller community. The process of stealing computing resources as a result of a system being joined to a "botnet" is sometimes referred to as "scrumping."

To thwart detection, some botnets are scaling back in size. As of 2006, the average size of a network was estimated at 20,000 computers.[11]

Recruitment

Computers can be co-opted into a botnet when they execute malicious software. This can be accomplished by luring users into making a drive-by download, exploiting web browser vulnerabilities, or by tricking the user into running a Trojan horse program, which may come from an email attachment. This malware will typically install modules that allow the computer to be commanded and controlled by the botnet's operator. After the software is downloaded, it will call home (send a reconnection packet) to the host computer. When the re-connection is made, depending on how it is written, a Trojan may then delete itself, or may remain present to update and maintain the modules. Many computer users are unaware that their computer is infected with bots.[12]

The first botnet was first acknowledged and exposed by Earthlink during a lawsuit with notorious spammer Khan C. Smith[13] in 2001 for the purpose of bulk spam accounting for nearly 25% of all spam at the time.

4.4.5 Countermeasures

The geographic dispersal of botnets means that each recruit must be individually identified/corralled/repaired and limits the benefits of filtering. Some botnets use free DNS hosting services such as DynDns.org, No-IP.com, and Afraid.org to point a subdomain towards an IRC server that harbors the bots. While these free DNS services do not themselves host attacks, they provide reference points (often hard-coded into the botnet executable). Removing such services can cripple an entire botnet. Some botnets implement custom versions of well-known protocols. The implementation differences can be used for detection of botnets. For example, Mega-D features a slightly modified SMTP protocol implementation for testing spam capability. Bringing down the Mega-D's SMTP server disables the entire pool of bots that rely upon the same SMTP server.[14]

Computer and network security companies have released software to counter botnets. Norton AntiBot was aimed at consumers, but most target enterprises and/or ISPs. Host-based techniques use heuristics to identify bot behavior that has bypassed conventional anti-virus software. Network-based approaches tend to use the techniques described above; shutting down C&C servers, nullrouting DNS entries, or completely shutting down IRC servers. BotHunter is software, developed with support from the U.S. Army Research Office, that detects botnet activity within a network by analysing network traffic and comparing it to patterns characteristic of malicious processes.

Some botnets are capable of detecting and reacting to attempts to investigate them, reacting perhaps with a DDoS attack on the IP address of the investigator.

Researchers at Sandia National Laboratories are analyzing botnets' behavior by simultaneously running one million Linux kernels—a similar scale to a botnet—as virtual machines on a 4,480-node high-performance computer cluster to emulate a very large network, allowing them to watch how botnets work and experiment with ways to stop them.[15]

4.4.6 Historical list of botnets

- Researchers at the University of California, Santa Barbara took control of a botnet that was six times smaller than expected. In some countries, it is common that users change their IP address a few times in one day. Estimating the size of the botnet by the number of IP addresses is often used by researchers, possibly leading to inaccurate assessments.[39]

4.4.7 See also

- Anti-spam techniques (e-mail)
- Backdoor:Win32.Hupigon
- Carna botnet
- Command and control (malware)
- Computer worm
- E-mail address harvesting
- E-mail spam
- List poisoning
- Spambot
- Spamtrap
- Timeline of computer viruses and worms
- Xor DDoS
- Zombie (computer science)
- ZeroAccess botnet

4.4.8 References

[1] "botnet". Retrieved 9 June 2016.

[2] "Forensics and Incident Response". www.peerlyst.com. Retrieved 3 April 2016.

[3] Ramneek, Puri (2003-08-08). "Bots &; Botnet: An Overview" (PDF). SANS Institute. Retrieved 12 November 2013.

[4] Danchev, Dancho (11 October 2013). "Novice cybercriminals offer commercial access to five mini botnets". Retrieved 28 June 2015.

[5] "what is a Botnet trojan?". DSL Reports. Retrieved 7 April 2011.

[6] Wang, Ping et al. (2010). "Peer-to-peer botnets". In Stamp, Mark & Stavroulakis, Peter. *Handbook of Information and Communication Security*. Springer. ISBN 9783642041174.

[7] "Operation Aurora — The Command Structure". Damballa.com. Archived from the original on 11 June 2010. Retrieved 30 July 2010.

[8] "Bitcoin Mining". BitcoinMining.com. Archived from the original on 30 April 2016. Retrieved 30 April 2016.

[9] "Trojan horse, and Virus FAQ". DSLReports. Retrieved 7 April 2011.

[10] Many-to-Many Botnet Relationships, *Damballa*, 8 June 2009.

[11] "Hackers Strengthen Malicious Botnets by Shrinking Them" (PDF). *Computer; News Briefs* (IEEE Computer Society). April 2006. doi:10.1109/MC.2006.136. Retrieved 12 November 2013. The size of bot networks peaked in mid-2004, with many using more than 100,000 infected machines, according to Mark Sunner, chief technology officer at MessageLabs.The average botnet size is now about 20,000 computers, he said.

[12] Teresa Dixon Murray. "Banks can't prevent cyber attacks like those hitting PNC, Key, U.S. Bank this week". Cleveland.com. Retrieved 2 September 2014.

[13] Credeur, Mary. "Atlanta Business Chronicle, Staff Writer". bizjournals.com. Retrieved 22 July 2002.

[14] C.Y. Cho, D. Babic, R. Shin, and D. Song. Inference and Analysis of Formal Models of Botnet Command and Control Protocols, 2010 ACM Conference on Computer and Communications Security.

[15] "Researchers Boot Million Linux Kernels to Help Botnet Research". IT Security & Network Security News. 2009-08-12. Retrieved 23 April 2011.

[16] "Symantec.cloud | Email Security, Web Security, Endpoint Protection, Archiving, Continuity, Instant Messaging Security" (PDF). Messagelabs.com. Retrieved 2014-01-30.

[17] Chuck Miller (2009-05-05). "Researchers hijack control of Torpig botnet". SC Magazine US. Retrieved 10 November 2011.

[18] "Storm Worm network shrinks to about one-tenth of its former size". Tech.Blorge.Com. 2007-10-21. Retrieved 30 July 2010.

[19] Chuck Miller (2008-07-25). "The Rustock botnet spams again". SC Magazine US. Retrieved 30 July 2010.

[20] Stewart, Joe. "Spam Botnets to Watch in 2009". *Secureworks.com*. SecureWorks. Retrieved 9 March 2016.

[21] "Pushdo Botnet — New DDOS attacks on major web sites — Harry Waldron — IT Security". Msmvps.com. 2010-02-02. Retrieved 30 July 2010.

[22] "New Zealand teenager accused of controlling botnet of 1.3 million computers". The H security. 2007-11-30. Retrieved 12 November 2011.

[23] "Technology | Spam on rise after brief reprieve". BBC News. 2008-11-26. Retrieved 24 April 2010.

[24] "Sality: Story of a Peer-to-Peer Viral Network" (PDF). Symantec. 2011-08-03. Retrieved 12 January 2012.

[25] "How FBI, police busted massive botnet". theregister.co.uk. Retrieved 3 March 2010.

[26] "Calculating the Size of the Downadup Outbreak — F-Secure Weblog : News from the Lab". F-secure.com. 2009-01-16. Retrieved 24 April 2010.

[27] "Waledac botnet 'decimated' by MS takedown". The Register. 2010-03-16. Retrieved 23 April 2011.

[28] Gregg Keizer (2008-04-09). "Top botnets control 1M hijacked computers". Computerworld. Retrieved 23 April 2011.

[29] "Botnet sics zombie soldiers on gimpy websites". The Register. 2008-05-14. Retrieved 23 April 2011.

[30] "Infosecurity (UK) - BredoLab downed botnet linked with Spamit.com". .canada.com. Retrieved 10 November 2011.

[31] "Research: Small DIY botnets prevalent in enterprise networks". ZDNet. Retrieved 30 July 2010.

[32] Warner, Gary (2010-12-02). "Oleg Nikolaenko, Mega-D Botmaster to Stand Trial". CyberCrime & Doing Time. Retrieved 6 December 2010.

[33] "New Massive Botnet Twice the Size of Storm — Security/Perimeter". DarkReading. Retrieved 30 July 2010.

[34] Kirk, Jeremy (Aug 16, 2012). "Spamhaus Declares Grum Botnet Dead, but Festi Surges". *PC World*.

[35] "Cómo detectar y borrar el rootkit TDL4 (TDSS/Alureon)". kasperskytienda.es. 2011-07-03. Retrieved 11 July 2011.

[36] "America's 10 most wanted botnets". Networkworld.com. 2009-07-22. Retrieved 10 November 2011.

[37] http://phys.org/news/2015-02-eu-police-malicious-network.html

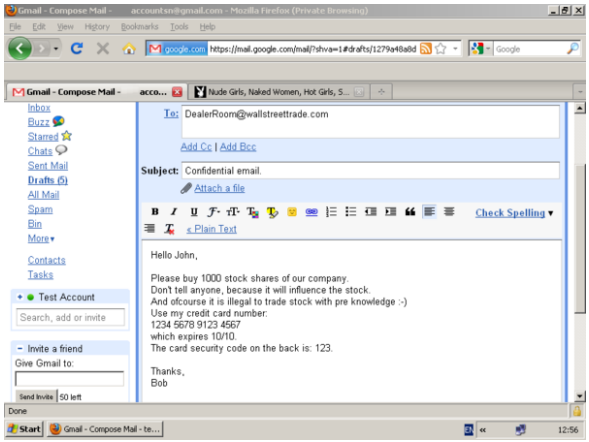
[38] "Discovered: Botnet Costing Display Advertisers over Six Million Dollars per Month". Spider.io. 2013-03-19. Retrieved 21 March 2013.

[39] Espiner, Tom (2011-03-08). "Botnet size may be exaggerated, says Enisa | Security Threats | ZDNet UK". Zdnet.com. Retrieved 10 November 2011.

4.4.9 External links

- Wired.com How-to: Build your own botnet with open source software
- The Honeynet Project & Research Alliance, "Know your Enemy: Tracking Botnets".
- The Shadowserver Foundation - An all volunteer security watchdog group that gathers, tracks, and reports on malware, botnet activity, and electronic fraud.
- NANOG Abstract: Botnets - John Kristoff's NANOG32 Botnets presentation.
- Mobile botnets - An economic and technological assessment of mobile botnets.
- Lowkeysoft - Intrusive analysis of a web-based proxy botnet (including administration screenshots).
- EWeek.com - Is the Botnet Battle Already Lost?.
- Attack of the Bots at *Wired*
- Dark Reading - Botnets Battle Over Turf.
- ATLAS Global Botnets Summary Report - Real-time database of malicious botnet command and control servers.
- FBI LAX Press Release DOJ - FBI April 16, 2008
- Milcord Botnet Defense - DHS-sponsored R&D project that uses machine learning to adaptively detect botnet behavior at the network-level
- A Botnet by Any Other Name - SecurityFocus column by Gunter Ollmann on botnet naming.
- Botnet Bust - SpyEye Malware Mastermind Pleads Guilty, FBI
- LOIC IRC-0 - An Open-Source IRC Botnet for Network Stress Testing
- LOIC SLOW IRC - An Open-Source Botnet With Webpages and IRC as C&C

4.5 Keystroke logging

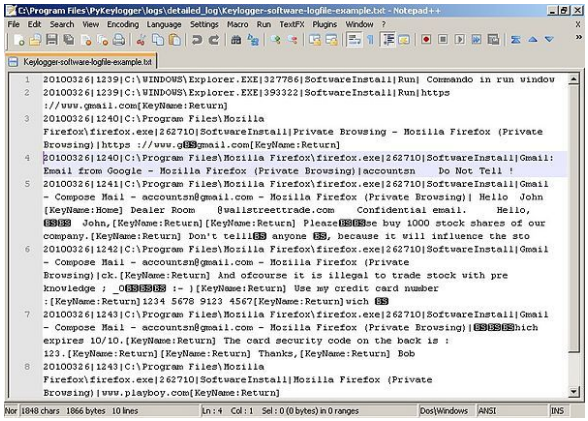


A keylogger example of a screencapture, which holds potentially confidential and private information. The image below holds the corresponding keylogger text result.

Keystroke logging, often referred to as **keylogging** or **keyboard capturing**, is the action of recording (logging) the keys struck on a keyboard, typically covertly, so that the person using the keyboard is unaware that their actions are being monitored.[1] Keylogging can also be used to study human–computer interaction. Numerous keylogging methods exist: they range from hardware and software-based approaches to acoustic analysis.

4.5.1 Application

Software-based keyloggers



A logfile from a software-based keylogger, based on the screencapture above.

These are computer programs designed to work on the target computer's software.[2] Keyloggers are used in IT or-

ganizations to troubleshoot technical problems with computers and business networks. Families and business people use keyloggers legally to monitor network usage without their users' direct knowledge. However, malicious individuals can use keyloggers on public computers to steal passwords or credit card information.

From a technical perspective there are several categories:

- **Hypervisor-based**: The keylogger can theoretically reside in a malware hypervisor running underneath the operating system, which thus remains untouched. It effectively becomes a virtual machine. Blue Pill is a conceptual example.
- **Kernel-based**: A program on the machine obtains root access to hide itself in the OS and intercepts keystrokes that pass through the kernel. This method is difficult both to write and to combat. Such keyloggers reside at the kernel level, which makes them difficult to detect, especially for user-mode applications that don't have root access. They are frequently implemented as rootkits that subvert the operating system kernel to gain unauthorized access to the hardware. This makes them very powerful. A keylogger using this method can act as a keyboard device driver, for example, and thus gain access to any information typed on the keyboard as it goes to the operating system.
- **API-based**: These keyloggers hook keyboard APIs inside a running application. The keylogger registers keystroke events, as if it was a normal piece of the application instead of malware. The keylogger receives an event each time the user presses or releases a key. The keylogger simply records it.
  - Windows APIs such as GetAsyncKeyState(), GetForegroundWindow(), etc. are used to poll the state of the keyboard or to subscribe to keyboard events.[3] A more recent example simply polls the BIOS for pre-boot authentication PINs that have not been cleared from memory.[4]
- **Form grabbing based**: Form grabbing-based keyloggers log web form submissions by recording the web browsing on submit events. This happens when the user completes a form and submits it, usually by clicking a button or hitting enter. This type of keylogger records form data before it is passed over the Internet.
- **Memory injection based**: Memory Injection (MitB)-based keyloggers perform their logging function by altering the memory tables associated with the browser and other system functions. By patching the memory tables or injecting directly into memory, this technique can be used by malware authors to bypass Windows UAC (User Account Control). The Zeus and SpyEye trojans use this method exclusively.[5] Non-Windows systems have analogous protection mechanisms that the keylogger must thwart.
- **Packet analyzers**: This involves capturing network traffic associated with HTTP POST events to retrieve unencrypted passwords. This is made more difficult when connecting via HTTPS, which is one of the reasons HTTPS was invented.
- **Remote access software keyloggers:** These are local software keyloggers with an added feature that allows access to locally recorded data from a remote location. Remote communication may be achieved when one of these methods is used:
  - Data is uploaded to a website, database or an FTP server.
  - Data is periodically emailed to a pre-defined email address.
  - Data is wirelessly transmitted by means of an attached hardware system.
  - The software enables a remote login to the local machine from the Internet or the local network, for data logs stored on the target machine.

Most of these keyloggers aren't stopped by HTTPS encryption because that only protects data in transit between computers. This is a threat in your own computer—the one connected to the keyboard.

Keystroke logging in writing process research
Keystroke logging is now an established research method for the study of writing processes.[6][7] Different programs have been developed to collect online process data of writing activities,[8] including Inputlog, Scriptlog, and Translog.

Keystroke logging is legitimately used as a suitable research instrument in a number of writing contexts. These include studies on cognitive writing processes, which include

- descriptions of writing strategies; the writing development of children (with and without writing difficulties),
- spelling,
- first and second language writing, and
- specialist skill areas such as translation and subtitling.

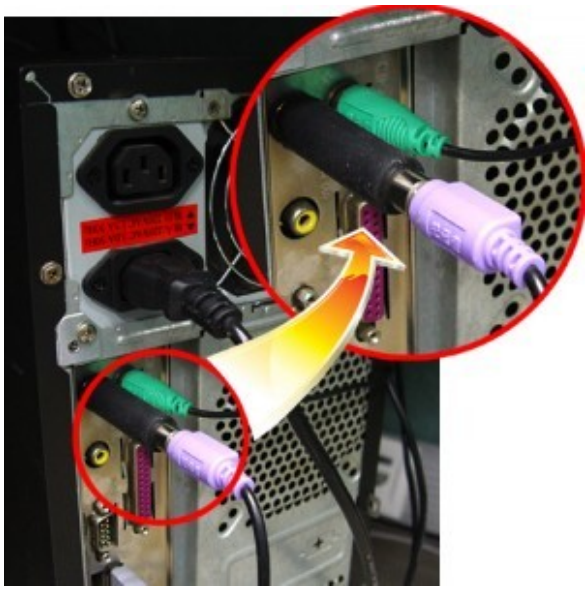
Keystroke logging can be used to research writing, specifically. It can also be integrated in educational domains for second language learning, programming skills, and typing skills.

Related features Software keyloggers may be augmented with features that capture user information without relying on keyboard key presses as the sole input. Some of these features include:

- Clipboard logging. Anything that has been copied to the clipboard can be captured by the program.
- Screen logging. Screenshots are taken to capture graphics-based information. Applications with screen logging abilities may take screenshots of the whole screen, of just one application, or even just around the mouse cursor. They may take these screenshots periodically or in response to user behaviours (for example, when a user clicks the mouse). A practical application that is used by some keyloggers with this screen logging ability, is to take small screenshots around where a mouse has just clicked; thus defeating web-based keyboards (for example, the web-based screen keyboards that are often used by banks), and any web-based on-screen keyboard without screenshot protection.
- Programmatically capturing the text in a control. The Microsoft Windows API allows programs to request the text 'value' in some controls. This means that some passwords may be captured, even if they are hidden behind password masks (usually asterisks).[9]
- The recording of every program/folder/window opened including a screenshot of each and every website visited.
- The recording of search engines queries, instant messenger conversations, FTP downloads and other Internet-based activities (including the bandwidth used).

Hardware-based keyloggers

A hardware-based keylogger.



A connected hardware-based keylogger.

Main article: Hardware keylogger

Hardware-based keyloggers do not depend upon any software being installed as they exist at a hardware level in a computer system.

- Firmware-based: BIOS-level firmware that handles keyboard events can be modified to record these events as they are processed. Physical and/or root-level access is required to the machine, and the software loaded into the BIOS needs to be created for the specific hardware that it will be running on.[10]
- Keyboard hardware: Hardware keyloggers are used for keystroke logging by means of a hardware circuit that is attached somewhere in between the computer keyboard and the computer, typically inline with the keyboard's cable connector. There are also USB connectors based Hardware keyloggers as well as ones for Laptop computers (the Mini-PCI card plugs into the expansion slot of a laptop). More stealthy implementations can be installed or built into standard keyboards, so that no device is visible on the external cable. Both types log all keyboard activity to their internal memory, which can be subsequently accessed, for example, by typing in a secret key sequence.[11] A hardware keylogger has an advantage over a software solution: it is not dependent on being installed on the target computer's operating system and therefore will not interfere with any program running on the target machine or be detected by any software. However its physical presence may be detected if, for example, it is

installed outside the case as an inline device between the computer and the keyboard. Some of these implementations have the ability to be controlled and monitored remotely by means of a wireless communication standard.[12]

- Wireless keyboard and mouse sniffers: These passive sniffers collect packets of data being transferred from a wireless keyboard and its receiver. As encryption may be used to secure the wireless communications between the two devices, this may need to be cracked beforehand if the transmissions are to be read. In some cases this enables an attacker to type arbitrary commands into a victim's computer.[13]

- Keyboard overlays: Criminals have been known to use keyboard overlays on ATMs to capture people's PINs. Each keypress is registered by the keyboard of the ATM as well as the criminal's keypad that is placed over it. The device is designed to look like an integrated part of the machine so that bank customers are unaware of its presence.[14]

- Acoustic keyloggers: Acoustic cryptanalysis can be used to monitor the sound created by someone typing on a computer. Each key on the keyboard makes a subtly different acoustic signature when struck. It is then possible to identify which keystroke signature relates to which keyboard character via statistical methods such as frequency analysis. The repetition frequency of similar acoustic keystroke signatures, the timings between different keyboard strokes and other context information such as the probable language in which the user is writing are used in this analysis to map sounds to letters.[15] A fairly long recording (1000 or more keystrokes) is required so that a big enough sample is collected.[16]

- Electromagnetic emissions: It is possible to capture the electromagnetic emissions of a wired keyboard from up to 20 metres (66 ft) away, without being physically wired to it.[17] In 2009, Swiss researchers tested 11 different USB, PS/2 and laptop keyboards in a semi-anechoic chamber and found them all vulnerable, primarily because of the prohibitive cost of adding shielding during manufacture.[18] The researchers used a wide-band receiver to tune into the specific frequency of the emissions radiated from the keyboards.

- Optical surveillance: Optical surveillance, while not a keylogger in the classical sense, is nonetheless an approach that can be used to capture passwords or PINs. A strategically placed camera, such as a hidden surveillance camera at an ATM, can allow a criminal to watch a PIN or password being entered.[19][20]

- Physical evidence: For a keypad that is used only to enter a security code, the keys which are in actual use will have evidence of use from many fingerprints. A passcode of four digits, if the four digits in question are known, is reduced from 10,000 possibilities to just 24 possibilities (10^4 versus 4! (factorial of 4)). These could then be used on separate occasions for a manual "brute force attack".

- Smartphone sensors: Researchers have demonstrated that it is possible to capture the keystrokes of nearby computer keyboards using only the commodity accelerometer found in smartphones.[21] The attack is made possible by placing a smartphone nearby a keyboard on the same desk. The smartphone's accelerometer can then detect the vibrations created by typing on the keyboard, and then translate this raw accelerometer signal into readable sentences with as much as 80 percent accuracy. The technique involves working through probability by detecting pairs of keystrokes, rather than individual keys. It models "keyboard events" in pairs and then works out whether the pair of keys pressed is on the left or the right side of the keyboard and whether they are close together or far apart on the QWERTY keyboard. Once it has worked this out, it compares the results to a preloaded dictionary where each word has been broken down in the same way.[22] Similar techniques have also been shown to be effective at capturing keystrokes on touchscreen keyboards[23][24][25] while in some cases, in combination with gyroscope.[26][27]

4.5.2 History

An early keylogger was written by Perry Kivolowitz and posted to the Usenet news group net.unix-wizards,net.sources on November 17, 1983.[28] The posting seems to be a motivating factor in restricting access to /dev/kmem on Unix systems. The user-mode program operated by locating and dumping character lists (clists) as they were assembled in the Unix kernel.

In the 1970s, spies installed keystroke loggers in the US Embassy and Consulate buildings in Moscow and St Petersburg.[29][30] They installed the bugs in Selectric II and Selectric III electric typewriters.[31]

Soviet embassies used manual typewriters, rather than electric typewriters, for classified information—apparently because they are immune to such bugs.[31] As of 2013, Russian special services still use typewriters.[30][32][33]

4.5.3 Cracking

Writing simple software applications for keylogging can be trivial, and like any nefarious computer program, can be distributed as a trojan horse or as part of a virus. What is not trivial for an attacker, however, is installing a covert keystroke logger without getting caught and downloading data that has been logged without being traced. An attacker that manually connects to a host machine to download logged keystrokes risks being traced. A trojan that sends keylogged data to a fixed e-mail address or IP address risks exposing the attacker.

Trojans

Researchers devised several methods for solving this problem. They presented a deniable password snatching attack in which the keystroke logging trojan is installed using a virus or worm.[34] [35] An attacker who is caught with the virus or worm can claim to be a victim. The cryptotrojan asymmetrically encrypts the pilfered login/password pairs using the public key of the trojan author and covertly broadcasts the resulting ciphertext. They mentioned that the ciphertext can be steganographically encoded and posted to a public bulletin board such as Usenet.

Use by police

In 2000, the FBI used FlashCrest iSpy to obtain the PGP passphrase of Nicodemo Scarfo, Jr., son of mob boss Nicodemo Scarfo.[36] Also in 2000, the FBI lured two suspected Russian cyber criminals to the US in an elaborate ruse, and captured their usernames and passwords with a keylogger that was covertly installed on a machine that they used to access their computers in Russia. The FBI then used these credentials to hack into the suspects' computers in Russia in order to obtain evidence to prosecute them.[37]

4.5.4 Countermeasures

The effectiveness of countermeasures varies, because keyloggers use a variety of techniques to capture data and the countermeasure needs to be effective against the particular data capture technique. For example, an on-screen keyboard will be effective against hardware keyloggers, transparency will defeat some—but not all—screenloggers and an anti-spyware application that can only disable hook-based keyloggers will be ineffective against kernel-based keyloggers.

Also, keylogger program authors may be able to update the code to adapt to countermeasures that may have proven to be effective against them.

Anti keyloggers

Main article: Anti keylogger

An anti keylogger is a piece of software specifically designed to detect keyloggers on a computer, typically comparing all files in the computer against a database of keyloggers looking for similarities which might signal the presence of a hidden keylogger. As anti keyloggers have been designed specifically to detect keyloggers, they have the potential to be more effective than conventional anti virus software; some anti virus software do not consider certain keyloggers a virus, as under some circumstances a keylogger can be considered a legitimate piece of software.[38]

Live CD/USB

Rebooting the computer using a Live CD or write-protected Live USB is a possible countermeasure against software keyloggers if the CD is clean of malware and the operating system contained on it is secured and fully patched so that it cannot be infected as soon as it is started. Booting a different operating system does not impact the use of a hardware or BIOS based keylogger.

Anti-spyware / Anti-virus programs

Many anti-spyware applications are able to detect some software based keyloggers and quarantine, disable or cleanse them. However, because many keylogging programs are legitimate pieces of software under some circumstances, anti spyware often neglects to label keylogging programs as spyware or a virus. These applications are able to detect software-based keyloggers based on patterns in executable code, heuristics and keylogger behaviours (such as the use of hooks and certain APIs).

No software-based anti-spyware application can be 100% effective against all keyloggers. Also, software-based anti-spyware cannot defeat non-software keyloggers (for example, hardware keyloggers attached to keyboards will always receive keystrokes before any software-based anti-spyware application).

However, the particular technique that the anti-spyware application uses will influence its potential effectiveness against software keyloggers. As a general rule, anti-spyware applications with higher privileges will defeat keyloggers with lower privileges. For example, a hook-based anti-spyware application cannot defeat a kernel-based keylogger (as the keylogger will receive the keystroke messages before the anti-spyware application), but it could potentially defeat hook- and API-based keyloggers.

Network monitors

Network monitors (also known as reverse-firewalls) can be used to alert the user whenever an application attempts to make a network connection. This gives the user the chance to prevent the keylogger from "phoning home" with his or her typed information.

Automatic form filler programs

Main article: Form filler

Automatic form-filling programs may prevent keylogging by removing the requirement for a user to type personal details and passwords using the keyboard. Form fillers are primarily designed for web browsers to fill in checkout pages and log users into their accounts. Once the user's account and credit card information has been entered into the program, it will be automatically entered into forms without ever using the keyboard or clipboard, thereby reducing the possibility that private data is being recorded. However someone with physical access to the machine may still be able to install software that is able to intercept this information elsewhere in the operating system or while in transit on the network. (Transport Layer Security (TLS) reduces the risk that data in transit may be intercepted by network sniffers and proxy tools.)

One-time passwords (OTP)

Using one-time passwords may be keylogger-safe, as each password is invalidated as soon as it is used. This solution may be useful for someone using a public computer. However, an attacker who has remote control over such a computer can simply wait for the victim to enter his/her credentials before performing unauthorised transactions on their behalf while their session is active.

Security tokens

Use of smart cards or other security tokens may improve security against replay attacks in the face of a successful keylogging attack, as accessing protected information would require both the (hardware) security token *as well as* the appropriate password/passphrase. Knowing the keystrokes, mouse actions, display, clipboard etc. used on one computer will not subsequently help an attacker gain access to the protected resource. Some security tokens work as a type of hardware-assisted one-time password system, and others implement a cryptographic challenge-response authentication, which can improve security in a manner conceptually similar to one time passwords. Smartcard readers and their associated keypads for PIN entry may be vulnerable to keystroke logging through a so-called supply chain attack[39] where an attacker substitutes the card reader/PIN entry hardware for one which records the user's PIN.

On-screen keyboards

Most on-screen keyboards (such as the on-screen keyboard that comes with Windows XP) send normal keyboard event messages to the external target program to type text. Software key loggers can log these typed characters sent from one program to another.[40] Additionally, keylogging software can take screenshots of what is displayed on the screen (periodically, and/or upon each mouse click), which means that although certainly a useful security measure, an on-screen keyboard will not protect from all keyloggers.

Keystroke interference softwares

Keystroke interference software is also available.[41] These programs attempt to trick keyloggers by introducing random keystrokes, although this simply results in the keylogger recording more information than it needs to. An attacker has the task of extracting the keystrokes of interest—the security of this mechanism, specifically how well it stands up to cryptanalysis, is unclear.

Speech recognition

Similar to on-screen keyboards, speech-to-text conversion software can also be used against keyloggers, since there are no typing or mouse movements involved. The weakest point of using voice-recognition software may be how the software sends the recognized text to target software after the recognition took place.

Handwriting recognition and mouse gestures

Also, many PDAs and lately tablet PCs can already convert pen (also called stylus) movements on their touchscreens to computer understandable text successfully. Mouse gestures use this principle by using mouse movements instead of a stylus. Mouse gesture programs convert these strokes to user-definable actions, such as typing text. Similarly, graphics tablets and light pens can be used to input these gestures, however these are less common everyday.

The same potential weakness of speech recognition applies to this technique as well.

Macro expanders/recorders

With the help of many programs, a seemingly meaningless text can be expanded to a meaningful text and most of the time context-sensitively, e.g. “en.wikipedia.org” can be expanded when a web browser window has the focus. The biggest weakness of this technique is that these programs send their keystrokes directly to the target program. However, this can be overcome by using the 'alternating' technique described below, i.e. sending mouse clicks to non-responsive areas of the target program, sending meaningless keys, sending another mouse click to target area (e.g. password field) and switching back-and-forth.

Non-technological methods

Alternating between typing the login credentials and typing characters somewhere else in the focus window[42] can cause a keylogger to record more information than they need to, although this could easily be filtered out by an attacker. Similarly, a user can move their cursor using the mouse during typing, causing the logged keystrokes to be in the wrong order e.g., by typing a password beginning with the last letter and then using the mouse to move the cursor for each subsequent letter. Lastly, someone can also use context menus to remove, cut, copy, and paste parts of the typed text without using the keyboard. An attacker who is able to capture only parts of a password will have a smaller key space to attack if he chose to execute a brute-force attack.

Another very similar technique uses the fact that any selected text portion is replaced by the next key typed. e.g., if the password is “secret”, one could type “s”, then some dummy keys “asdfsd”. Then, these dummies could be selected with the mouse, and the next character from the password “e” is typed, which replaces the dummies “asdfsd”.

These techniques assume incorrectly that keystroke logging software cannot directly monitor the clipboard, the selected text in a form, or take a screenshot every time a keystroke or mouse click occurs. They may however be effective against some hardware keyloggers.

4.5.5 See also

- Anti keylogger
- Black-bag cryptanalysis
- Computer surveillance
- Digital footprint
- Hardware keylogger
- Reverse connection
- Spyware
- Trojan horse
- Virtual keyboard

4.5.6 References

[1] “Keylogger”. Oxford dictionaries.

[2] “What is a Keylogger?". PC Tools.

[3] “The Evolution of Malicious IRC Bots” (PDF). Symantec. 2005-11-26: 23–24. Retrieved 2011-03-25.

[4] Jonathan Brossard (2008-09-03). “Bypassing pre-boot authentication passwords by instrumenting the BIOS keyboard buffer (practical low level attaks against x86 pre-boot authentiation software)" (PDF). Iviz Technosolutions. Retrieved 2008-09-23. External link in |publisher= (help)

[5] “SpyEye Targets Opera, Google Chrome Users”. *Krebs on Security*. Retrieved 26 April 2011.

[6] K.P.H. Sullivan & E. Lindgren (Eds., 2006), Studies in Writing: Vol. 18. Computer Key-Stroke Logging and Writing: Methods and Applications. Oxford: Elsevier.

[7] V. W. Berninger (Ed., 2012), Past, present, and future contributions of cognitive writing research to cognitive psychology. New York/Sussex: Taylor & Francis. ISBN 9781848729636

[8] Vincentas (11 July 2013). “Keystroke Logging in SpyWareLoop.com”. Spyware Loop. Retrieved 27 July 2013.

[9] Microsoft. “EM_GETLINE Message()". Microsoft. Retrieved 2009-07-15.

[10] “Apple keyboard hack”. *Apple keyboard hack*. Digital Society. Retrieved 9 June 2011.

[11] “Keyghost”. keyghost.com. Retrieved 2009-04-19. External link in |publisher= (help)

[12] “Keylogger Removal”. *Keylogger Removal*. SpyReveal Anti Keylogger. Retrieved 25 April 2011.

[13] “Keylogger Removal”. *Keylogger Removal*. SpyReveal Anti Keylogger. Retrieved 26 February 2016.

[14] Jeremy Kirk (2008-12-16). “Tampered Credit Card Terminals”. IDG News Service. Retrieved 2009-04-19.

[15] Andrew Kelly (2010-09-10). “Cracking Passwords using Keyboard Acoustics and Language Modeling” (PDF).

[16] Sarah Young (14 September 2005). “Researchers recover typed text using audio recording of keystrokes”. UC Berkeley NewsCenter.

[17] "Remote monitoring uncovered by American techno activists". ZDNet. 2000-10-26. Retrieved 2008-09-23.

[18] Martin Vuagnoux and Sylvain Pasini (2009-06-01). "Compromising Electromagnetic Emanations of Wired and Wireless Keyboards". Lausanne: Security and Cryptography Laboratory (LASEC).

[19] "ATM camera". snopes.com. Retrieved 2009-04-19. External link in |publisher= (help)

[20] Maggi, Federico; Volpatto, Alberto; Gasparini, Simone; Boracchi, Giacomo; Zanero, Stefano (2011). *A fast eavesdropping attack against touchscreens*. 7th International Conference on Information Assurance and Security. IEEE. doi:10.1109/ISIAS.2011.6122840.

[21] Marquardt, Philip; Verma, Arunabh; Carter, Henry; Traynor, Patrick (2011). *(sp)iPhone: decoding vibrations from nearby keyboards using mobile phone accelerometers*. Proceedings of the 18th ACM conference on Computer and communications security. ACM. pp. 561–562. doi:10.1145/2046707.2046771.

[22] "iPhone Accelerometer Could Spy on Computer Keystrokes". Wired. 19 October 2011. Retrieved August 25, 2014. External link in |publisher= (help)

[23] Owusu, Emmanuel; Han, Jun; Das, Sauvik; Perrig, Adrian; Zhang, Joy (2012). *ACCessory: password inference using accelerometers on smartphones*. Proceedings of the Thirteenth Workshop on Mobile Computing Systems and Applications. ACM. doi:10.1145/2162081.2162095.

[24] Aviv, Adam J.; Sapp, Benjamin; Blaze, Matt; Smith, Jonathan M. (2012). *Practicality of accelerometer side channels on smartphones*. Proceedings of the 28th Annual Computer Security Applications Conference. ACM. doi:10.1145/2420950.2420957.

[25] Cai, Liang; Chen, Hao (2011). *TouchLogger: inferring keystrokes on touch screen from smartphone motion* (PDF). Proceedings of the 6th USENIX conference on Hot topics in security. USENIX. Retrieved 25 August 2014.

[26] Xu, Zhi; Bai, Kun; Zhu, Sencun (2012). *TapLogger: inferring user inputs on smartphone touchscreens using on-board motion sensors*. Proceedings of the fifth ACM conference on Security and Privacy in Wireless and Mobile Networks. ACM. pp. 113–124. doi:10.1145/2185448.2185465.

[27] Miluzzo, Emiliano; Varshavsky, Alexander; Balakrishnan, Suhrid; Choudhury, Romit Roy (2012). *Tapprints: your finger taps have fingerprints*. Proceedings of the 10th international conference on Mobile systems, applications, and services. ACM. pp. 323–336. doi:10.1145/2307636.2307666.

[28] "The Security Digest Archives". Retrieved 2009-11-22.

[29] "Soviet Spies Bugged World's First Electronic Typewriters". *qccglobal.com*.

[30] Geoffrey Ingersoll. "Russia Turns To Typewriters To Protect Against Cyber Espionage". 2013.

[31] Sharon A. Maneki. "Learning from the Enemy: The GUNMAN Project". 2012.

[32] Agence France-Presse, Associated Press. "Wanted: 20 electric typewriters for Russia to avoid leaks". *inquirer.net*.

[33] Anna Arutunyan. "Russian security agency to buy typewriters to avoid surveillance".

[34] Young, Adam; Yung, Moti (1997). "Deniable Password Snatching: On the Possibility of Evasive Electronic Espionage". *Proceedings of IEEE Symposium on Security and Privacy* (IEEE): 224–235. doi:10.1109/SECPRI.1997.601339.

[35] Young, Adam; Yung, Moti (1996). "Cryptovirology: extortion-based security threats and countermeasures". *Proceedings of IEEE Symposium on Security and Privacy* (IEEE): 129–140. doi:10.1109/SECPRI.1996.502676.

[36] John Leyden (2000-12-06). "Mafia trial to test FBI spying tactics: Keystroke logging used to spy on mob suspect using PGP". The Register. Retrieved 2009-04-19.

[37] John Leyden (2002-08-16). "Russians accuse FBI Agent of Hacking". The Register.

[38] Theron, kristen (19 February 2016). "What is Anti Keylogger".

[39] Austin Modine (2008-10-10). "Organized crime tampers with European card swipe devices". The Register. Retrieved 2009-04-18.

[40] Scott Dunn (2009-09-10). "Prevent keyloggers from grabbing your passwords". Windows Secrets. Retrieved 2014-05-10.

[41] Christopher Ciabarra (2009-06-10). "Anti Keylogger". Networkintercept.com.

[42] Cormac Herley and Dinei Florencio (2006-02-06). "How To Login From an Internet Cafe Without Worrying About Keyloggers" (PDF). Microsoft Research. Retrieved 2008-09-23.

4.5.7 External links

- Keyloggers at DMOZ

4.6 Form grabbing

Form grabbing is a form of malware that works by retrieving authorization and log-in credentials from a web data form before it is passed over the Internet to a secure server. This allows the malware to avoid HTTPS encryption. This

method is more effective than keylogger software because it will acquire the user's credentials even if they are inputted using virtual keyboard, auto-fill, or copy and paste.[1] It can then sort the information based on its variable names, such as e-mail, account name, and password. Additionally, the form grabber will log the URL and title of the website the data was gathered from.[2]

4.6.1 History

The method was invented in 2003 by the developer of a variant of a Trojan Horse virus called Downloader.Barbew, which attempts to download Backdoor.Barbew from the Internet and bring it over to the local system for execution. However, it was not popularized as a well-known type of malware attack until the emergence of the infamous banking Trojan, Zeus, in 2007.[3] Zeus was used to steal banking information by man-in-the-browser keystroke logging and form grabbing. Like Zeus, the Barbew Trojan was initially spammed to large numbers of individuals through e-mails masquerading as big-name banking companies.[4] Form grabbing as a method first advanced through iterations of Zeus that allowed the module to not only detect the grabbed form data but to also determine how useful the information taken was. In later versions, the form grabber was also privy to the website where the actual data was submitted, leaving sensitive information more vulnerable than before.[5]

4.6.2 Known occurrences

A Trojan known as Tinba (Tiny Banker Trojan) has been built with form grabbing and is able to steal online banking credentials and was first discovered in 2012. Another software called Weyland-Yutani BOT was the first software designed to attack Mac OS platform and can work on Firefox. The web injects templates in Weyland-Yutani BOT are different from existed ones such as Zeus and SpyEye.[6]

4.6.3 Countermeasures

Due to the recent increase in key-logging and form-grabbing, Antivirus companies are adding additional protection to counter the efforts of key-loggers and prevent collecting passwords. These efforts have taken different forms varying from Antivirus companies, such as safepay, password manager, and others.[1] To further counter form grabbing, users' privileges can become limited which would prevent them from installing Browser Helper Objects(BHO) and other form grabbing software. Administrators should create a list of malicious servers to their firewalls.[2]

4.6.4 See also

- Keystroke logging
- Malware
- Trojan horse
- Web security exploits
- Computer insecurity
- Internet privacy
- Tiny Banker Trojan

4.6.5 References

[1] "Capturing Online Passwords and Antivirus." Web log post. Business Information Technology Services, 24 July 2013.

[2] Graham, James, Richard Howard, and Ryan Olson. Cyber Security Essentials. Auerbach Publications, 2011. Print.

[3] • Shevchenko, Sergei. "Downloader.Berbew." Symantec, 13 Feb. 2007.

[4] • Abrams, Lawrence. "CryptoLocker Ransomware Information Guide and FAQ." Bleeding Computers. 20 Dec. 2013.

[5] • "Form Grabbing." Web log post. Rochester Institute of Technology, 10 Sept. 2011.

[6] Kruse, Peter. "Crimekit for MacOSX Launched." Web log post. Canadian Security Intelligence Service, 02 May 2011.

4.7 Web threat

A **web threat** is any threat that uses the World Wide Web to facilitate cybercrime. Web threats use multiple types of malware and fraud, all of which utilize HTTP or HTTPS protocols, but may also employ other protocols and components, such as links in email or IM, or malware attachments or on servers that access the Web. They benefit cybercriminals by stealing information for subsequent sale and help absorb infected PCs into botnets.

Web threats pose a broad range of risks, including financial damages, identity theft, loss of confidential information/data, theft of network resources, damaged brand/personal reputation, and erosion of consumer confidence in e-commerce and online banking.

It is a type of threat related to information technology (IT). The IT risk, i.e. risk affecting has gained and increasing impact on society due to the spread of IT processes. [1] [2] [3]

4.7.1 Delivery methods

Web threats can be divided into two primary categories, based on delivery method – push and pull.[4] Push-based threats use spam, phishing, or other fraudulent means to lure a user to a malicious (often spoofed) website which then collects information and/or injects malware. Push attacks use phishing, DNS poisoning (or pharming), and other means to appear to originate from a trusted source.

Precisely-targeted push-based web threats are often referred to as spear phishing to reflect the focus of their data gathering attack. Spear phishing typically targets specific individuals and groups for financial gain. In other push-based web threats, malware authors use social engineering such as enticing subject lines that reference holidays, popular personalities, sports, pornography, world events and other hot topics to persuade recipients to open the email and follow links to malicious websites or open attachments with malware that accesses the Web.

Pull-based web threats are often referred to as "drive-by" threats by experts (and more commonly as "drive-by downloads" by journalists and the general public), since they can affect any website visitor. Cybercriminals infect legitimate websites, which unknowingly transmit malware to visitors or alter search results to take users to malicious websites. Upon loading the page, the user's browser passively runs a malware downloader in a hidden HTML frame (IFRAME) without any user interaction.

4.7.2 Growth of web threats

"And if today's malware runs mostly runs on Windows because it's the commonest executable platform, tomorrow's will likely run on the Web, for the very same reason. Because, like it or not, the Web is already a huge executable platform, and we should start thinking of it this way, from a security perspective." – Giorgio Maone [5]

The growth of web threats is a result of the popularity of the Web – a relatively unprotected, widely and consistently used medium that is crucial to business productivity, online banking, and e-commerce as well as the everyday lives of people worldwide. The appeal of Web 2.0 applications and websites increases the vulnerability of the Web. Most Web 2.0 applications make use of AJAX, a group of web development programming tools used for creating interactive web applications or rich Internet applications. While users benefit from greater interactivity and more dynamic websites, they are also exposed to the greater security risks inherent in browser client processing.[6]

4.7.3 Examples

In September 2008, malicious hackers broke into several sections of BusinessWeek.com to redirect visitors to malware-hosting websites. Hundreds of pages were compromised with malicious JavaScript pointing to third-party servers.[7]

In August 2008, popular social networking sites were hit by a worm using social engineering techniques to get users to install a piece of malware. The worm installs comments on the sites with links to a fake site. If users follow the link, they are told they need to update their Flash Player. The installer then installs malware rather than the Flash Player. The malware then downloads a rogue anti-spyware application, AntiSpy Spider.[8]

In May 2008, websites worldwide were compromised with a malicious JavaScript. Initially a half million websites worldwide were infected with a SQL injection which leveraged a ZLOB variant which then downloaded additional Trojan onto users' PCs. Then websites in China, Taiwan and Singapore were compromised followed shortly thereafter by humanitarian, government and news sites in the UK, Israel and Asia. In this attack the compromised websites led, through a variety of redirects, to the download of a Trojan.

4.7.4 Prevention and detection

Conventional approaches have failed to fully protect consumers and businesses from web threats. The most viable approach is to implement multi-layered protection—protection in the cloud, at the Internet gateway, across network servers and on the client.

4.7.5 See also

- Asset (computing)
- Attack (computing)
- Botnets
- Browser security
- Countermeasure (computer)
- Cyberwarfare
- Cybercrime
- IT risk
- rich Internet applications
- Internet security

- Internet safety
- Low Orbit Ion Cannon
- High Orbit Ion Cannon
- Threat (computer)
- Vulnerability (computing)
- Web applications
- web development
- Man-in-the-browser
- Denial-of-service attack

4.7.6 References

[1] Cortada, James W. (2003-12-04). *The Digital Hand: How Computers Changed the Work of American Manufacturing, Transportation, and Retail Industries*. USA: Oxford University Press. p. 512. ISBN 0-19-516588-8.

[2] Cortada, James W. (2005-11-03). *The Digital Hand: Volume II: How Computers Changed the Work of American Financial, Telecommunications, Media, and Entertainment Industries*. USA: Oxford University Press. ISBN 978-0-19-516587-6.

[3] Cortada, James W. (2007-11-06). *The Digital Hand, Vol 3: How Computers Changed the Work of American Public Sector Industries*. USA: Oxford University Press. p. 496. ISBN 978-0-19-516586-9.

[4] Trend Micro (2008) Web Threats: Challenges and Solutions from http://us.trendmicro.com/imperia/md/content/us/pdf/webthreats/wp01_webthreats_080303.pdf

[5] Maone, Giorgio (2008) Malware 2.0 is Now! from http://hackademix.net/2008/01/12/malware-20-is-now/

[6] Horwath, Fran (2008) Web 2.0: next-generation web threats from http://www.it-director.com/technology/security/content.php?cid=10162

[7] Naraine, Ryan (2008) Business Week site hacked, serving drive-by exploits from http://blogs.zdnet.com/security/?p=1902#more-1902

[8] Danchev, Dancho (2008) Compromised Web Servers Serving Fake Flash Players from http://ddanchev.blogspot.com/2008/08/compromised-web-servers-serving-fake.html

4.8 Dialer

A **dialer** (American English) or **dialler** (British English) is an electronic device that is connected to a telephone line to monitor the dialed numbers and alter them to seamlessly provide services that otherwise require lengthy National or International access codes to be dialed. A dialer automatically inserts and modifies the numbers depending on the time of day, country or area code dialed, allowing the user to subscribe to the service providers who offer the best rates. For example, a dialer could be programmed to use one service provider for international calls and another for cellular calls. This process is known as prefix insertion or least cost routing. A line powered dialer does not need any external power but instead takes the power it needs from the telephone line.

Another type of dialer is a computer program which creates a connection to the Internet or another computer network over the analog telephone or Integrated Services Digital Network (ISDN) network. Many operating systems already contain such a program for connections through the Point-to-Point Protocol (PPP), such as WvDial.

Many internet service providers offer installation CDs to simplify the process of setting up a proper Internet connection. They either create an entry in the OS's dialer or install a separate dialer (as the AOL software does).

In recent years, the term "dialer" often refers specifically to dialers that connect without the user's full knowledge as to cost, with the creator of the dialer intending to commit fraud.

4.8.1 Dialing modes

In Call Centers there are several dialling modes depending on how the call is placed. 'Manual Dialing' refers to calls that are placed manually by an agent.

There are 4 different dialing modes depending on how software dialers selects the contacts that are going to be called and starts making the calls. Automated dialers such as those sold by Noble Systems and Avaya can place calls using *preview*, *power*, auto dialing, or predictive dialing. The dialing modes are defined according to the campaign and type of business.

Preview

Preview dialing enables agents to first view the available information about the customer and decide when to place the call. In addition to the information about the customer, agents may also view all the history of the customer with

the contact center. After viewing the information about the customer, the agent requests the system to make the call.

For example, preview dialing is useful in debt collection campaigns to allow agents to view information about the customer and define a strategy before starting to talk to the customer. The system delivers preview calls to agents automatically, taking into account the priority of the call and the skills of the agent to handle the call. Preview dialing keeps agents from dialing calls manually.

Power

Also called Progressive dialing, Power dialing places calls only when an agent is available to handle the call. Automated dialers consider the priority and the skills of the agent to automatically place a call to the agent. In power dialing, an agent is always available to talk to the customer.

Power dialing is suitable for all campaigns, from customer care follow-up calls to telemarketing. For example, power dialing is useful to call predictive dialing contacts that resulted in nuisance to ensure that an agent is available to talk to the customer. A 2012 research study conducted by The Bridge Group, Inc. discovered that while inside sales jobs and career demand is up 54 percent, most sales leverage comes with power dialer software.[1]

Predictive

Predictive dialing is a state-of-the-art pacing mode used to call a large number of customers within a short period of time. Predictive dialing optimizes the time of agents by reducing the idle times between connected calls and freeing agents from dialing calls. Predictive dialing gathers statistics concerning the duration of calls, how long it takes for calls to be answered, and how often are calls answered. When an agent is about to become idle, The system places several calls.

Predictive dialing campaigns can achieve agent productivity of 50 minutes per hour and nuisance ratios of 3% or less. The system is continually updating predictive dialing probabilities and monitoring nuisance ratios for performance and compliance with legislation. For example, predictive dialing is useful in sales campaigns to call a large number of contacts and maximizing the working time of agents.

The performance of predictive dialing takes into consideration the accuracy of the contact lists and the policies on nuisance calls. If the contact list is poor, the performance of the predictive dialing campaign is at risk as agents are not connected to live contacts and are not able to do business.

Power IVR

Power IVR is used to deliver a pre-recorded message to a large call list. When a call is answered, Power IVR will play the audio file, and then collect touch tone key responses or speech command at the end of the message, and then transfer the call to an agent or remove the caller from the call list. In other words, IVR is a technology that allows a computer to interact with humans though the use of voice and DTMF tones input via keypad.

Voice Drop

Voice drop is same as Power IVR, except it will not wait for the touch tone key responses or speech command at the end of the message, after playing the message, call is dropped, mostly this is used for payment reminders and similar.

4.8.2 Fraudulent dialer

Dialers are necessary to connect to the internet (at least for non-broadband connections), but some dialers are designed to connect to premium-rate numbers. The providers of such dialers often search for security holes in the operating system installed on the user's computer and use them to set the computer up to dial up through their number, so as to make money from the calls. Alternatively, some dialers inform the user what it is that they are doing, with the promise of special content, accessible only via the special number. Examples of this content include software for download, (usually illegal) trojans posing as MP3s, trojans posing as pornography, or 'underground' programs such as cracks and keygens.

The cost of setting up such a service is relatively low, amounting to a few thousand dollars for telecommunications equipment, whereupon the unscrupulous operator will typically take 90% of the cost of a premium rate call, with very few overheads of their own.

Users with DSLs (or similar broadband connections) are usually not affected. A dialer can be downloaded and installed, but dialing in is not possible as there are no regular phone numbers in the DSL network and users will not typically have their dial-up modem, if any, connected to a phone line. However, if an ISDN adapter or additional analog modem is installed, the dialer might still be able to get a connection.

Malicious dialers can be identified by the following characteristics:

- A download popup opens when opening a website.

- On the website there is only a small hint, if any, about the price.
- The download starts even if the cancel button has been clicked.
- The dialer installs as default connection without any notice.
- The dialer creates unwanted connections by itself and without user interaction.
- The dialer does not show any notice about the price (only few do) before dialing in.
- The high price of the connection is not being shown while connected
- The dialer cannot be uninstalled, or only with serious effort.

Installation routes

Computers running Microsoft Windows without anti-virus software or proper updates could be vulnerable to Visual Basic-scripts which install a trojan horse which changes values in the Windows Registry and sets Internet Explorer security settings in a way that ActiveX controls can be downloaded from the Internet without warning. After this change is made, when a user accesses a malicious page or email message, it can start installing the dialer. The script also disables the modem speaker and messages that normally come up while dialing into a network. Users of Microsoft Office Outlook, Outlook Express and Internet Explorer are especially affected if running ActiveX controls and JavaScript is allowed and the latest security patches from Microsoft have not been installed. In March 2004, there were malicious dialers that could be installed through fake anti-virus software. E-mail spam from a so-called "AntiVirus Team" for example, contained download links to programs named "downloadtool.exe" or "antivirus.exe", which are malicious dialers. Other ways of transmission include electronic greeting cards that link to pages that tricks the user to install ActiveX controls, which in turn install dialers in the background.

Therefore, links in spam emails should never be opened, automatically started downloads should be canceled as soon as discovered, and one should check on each dial-up to the Internet to see whether the displayed phone number is unchanged. Another way to protect oneself is to disable premium numbers through one's phone services, but of course this disables all such services.

One should never run foreign code in a privileged environment unless the source is trustworthy. It is also advisable to protect oneself with anti-malware programs.

4.8.3 German regulatory law

On 15 August 2003, a new law came into effect in Germany called "Gesetz zur Bekämpfung des Missbrauchs von (0)190er/(0)900er Mehrwertdiensterufnummern" ("Law for the combat of misuse of (0)190/(0)900 value added service numbers").

The law contains the following regulations:

- Forced price notices for service providers.
- Maximum price limits, legitimacy checks and automatic disconnects.
- Registration of dialers.
- Blocking of dialers.
- Right of information for consumers from the RegTP.

On 4 March 2004 the German Federal Supreme Court in Karlsruhe decided that fees for the usage of dialers do not have to be paid if it was used without the user's knowledge.

4.8.4 See also

- AvatarDialler A Predictive dialer used in Call Centers
- Predictive dialer System for dialing many numbers, typically used by call centers
- War dialing Automatically scanning a list of telephone numbers to detect computers, usually for nefarious purposes
- Dialer management platform
- XXXDial — historical example of dialer/spyware
- Bettercallcenters.com A website operated by ChaseData Corporation (Plantation, FL USA) offering an overview of their dialer software and extensive blog articles on how to leverage dialers to achieve better solutions for call centers.

4.8.5 References

[1] Krogue, Ken. "Inside Sales Jobs and Career Demand Up 54%: But Most Leverage Comes With Dialer Software and Lead Research". *Forbes*. Forbes.com. Retrieved 2013-04-24.

4.9 Internet bot

For other uses, see Bot (computing).

An **Internet bot**, also known as **web robot**, **WWW robot** or simply **bot**, is a software application that runs automated tasks (scripts) over the Internet.[1] Typically, bots perform tasks that are both simple and structurally repetitive, at a much higher rate than would be possible for a human alone. The largest use of bots is in web spidering (*web crawler*), in which an automated script fetches, analyzes and files information from web servers at many times the speed of a human.

Given the exceptional speed with which bots can perform their relatively simple routines, bots may also be implemented where a response speed faster than that of humans is required. Common examples including gaming bots, whereby a player achieves a significant advantage by implementing some repetitive routine with the use of a bot rather than manually, or auction-site robots, where last-minute bid-placing speed may determine who places the winning bid – using a bot to place counterbids affords a significant advantage over bids placed manually.

Bots are routinely used on the internet where the emulation of human activity is required, for example chat bots. A simple question and answer exchange online may appear to be with another person, when in fact it is simply with a bot.

While bots are often used to simply automate a repetitive online interaction, their ability to mimic actual human conversation and avoid detection has resulted in the use of bots as tools of covert manipulation. On the internet today bots are used to artificially alter, disrupt or even silence legitimate online conversations. Bots are sometimes implemented, for example, to overwhelm the discussion of some topic which the bot's creator wishes to silence. The bot may achieve this by drowning out a legitimate conversation with repetitive bot-placed posts which may in some cases appear to be reasonable and relevant, in others simply unrelated or nonsense chatter, or alternatively by overwhelming the target website's server with constant, repetitive, pointless bot-placed posts. These bots play an important role in modifying, confusing and silencing conversations about, and the dissemination of, real information regarding sensitive events around the world.

The success of bots may be largely due to the very real difficulty in identifying the difference between an online interaction with a bot versus a live human. Given that bots are relatively simple to create and implement, they are a very powerful tool with the potential to influence every segment of the World Wide Web.

Efforts by servers hosting websites to counteract bots vary. Servers may choose to outline rules on the behaviour of internet bots by implementing a robots.txt file: this file is simply text stating the rules governing a bot's behaviour on that server. Any bot interacting with (or 'spidering') any server that does not follow these rules should, in theory, be denied access to, or removed from, the affected website. If the only rule implementation by a server is a posted text file with no associated program/software/app, then adhering to those rules is entirely voluntary – in reality there is no way to enforce those rules, or even to ensure that a bot's creator or implementer acknowledges, or even reads, the robots.txt file contents.

4.9.1 IM and IRC

Some bots communicate with other users of Internet-based services, via instant messaging (IM), Internet Relay Chat (IRC), or another web interface such as Facebook Bots and Twitterbots. These chatterbots may allow people to ask questions in plain English and then formulate a proper response. These bots can often handle many tasks, including reporting weather, zip-code information, sports scores, converting currency or other units, etc. Others are used for entertainment, such as SmarterChild on AOL Instant Messenger and MSN Messenger.

An additional role of IRC bots may be to lurk in the background of a conversation channel, commenting on certain phrases uttered by the participants (based on pattern matching). This is sometimes used as a help service for new users, or for censorship of profanity.

AOL Instant Messenger has now introduced a feature that allows you to make a screen name into a bot. This new feature removes the rate limit on the screen name, however it is now limited in the amount of instant messages that can be sent and received.

4.9.2 Commercial purposes

There has been a great deal of controversy about the use of bots in an automated trading function. Auction website eBay has been to court in an attempt to suppress a third-party company from using bots to traverse their site looking for bargains; this approach backfired on eBay and attracted the attention of further bots. The United Kingdom-based bet exchange Betfair saw such a large amount of traffic coming from bots they launched a WebService API aimed at bot programmers through which Betfair can actively manage bot interactions.

Bot farms are known to be used in online app stores, like the Apple App Store and Google Play, to manipulate positions[2] or to increase positive ratings/reviews.[3]

4.9.3 Malicious purposes

Another, more malicious use of bots is the coordination and operation of an automated attack on networked computers, such as a denial-of-service attack by a botnet. Internet bots can also be used to commit click fraud and more recently have seen usage around MMORPG games as computer game bots. A spambot is an internet bot that attempts to spam large amounts of content on the Internet, usually adding advertising links.

- There are malicious bots (and botnets) of the following types:

1. Spambots that harvest email addresses from contact or guestbook pages
2. Downloader programs that suck bandwidth by downloading entire web sites
3. Website scrapers that grab the content of websites and re-use it without permission on automatically generated doorway pages
4. Viruses and worms
5. DDoS attacks
6. Botnets, zombie computers, etc.

- Bots are also used to buy up good seats for concerts, particularly by ticket brokers who resell the tickets. Bots are employed against entertainment event-ticketing sites. The bots are used by ticket brokers to unfairly obtain the best seats for themselves while depriving the general public from also having a chance to obtain the good seats. The bot runs through the purchase process and obtains better seats by pulling as many seats back as it can.
- Bots are often used in Massively Multiplayer Online Roleplaying Games to farm for resources that would otherwise take significant time or effort to obtain; this is a concern for most online in-game economies.
- Bots are also used to increase views for YouTube videos.
- Bots are used to increase traffic counts on analytics reporting to extract money from advertisers. A study by comScore found that 54 percent of display ads shown in thousands of campaigns between May 2012 and February 2013 never appeared in front of a human being.[4]
- in 2012 reporter Percy Lipinski reported that he discovered millions of bot or botted or pinged views at CNN iReport. CNN iReport quietly removed millions of views from the account of so-called superstar iReporter Chris Morrow. A follow-up investigation led to a story published on the citizen journalist platform, Allvoices: http://www.allvoices.com/contributed-news/14694943-cnn-hit-hard-in-ongoing-pay-per-view-scandal-millions-of- It is not known if the ad revenue received by CNN from the fake views was ever returned to the advertisers.
- Bots may be used on internet forums to automatically post inflammatory or nonsensical posts to disrupt the forum and anger users.

The most widely used anti-bot technique is the use of CAPTCHA, which is a form of Turing test used to distinguish between a human user and a less-sophisticated AI-powered bot, by the use of graphically encoded human-readable text. Examples of providers include Recaptcha, and commercial companies such as Minteye, Solve Media, and NuCaptcha. Captchas, however, are not foolproof in preventing bots as they can often be circumvented by computer character recognition, security holes, and even by outsourcing captcha solving to cheap laborers.

4.9.4 See also

- Software agent
- IRC bot
- Facebook Bots
- Botnet
- Spambot
- UBot Studio
- Feed bot
- Twitterbot

4.9.5 References

[1] Dunham, Ken; Melnick, Jim (2008). *Malicious Bots: An Inside Look into the Cyber-Criminal Underground of the Internet*. CRC Press. ISBN 9781420069068.

[2] "Touch Arcade Forum Discussion on fraud in the Top 25 Free Ranking".

[3] "App Store fake reviews: Here's how they encourage your favourite developers to cheat". *Electricpig*.

[4] Holiday, Ryan. "Fake Traffic Means Real Paydays". BetaBeat.

4.10 Scareware

Not to be confused with careware or shareware.

Scareware is a form of malicious software that uses social engineering to cause shock, anxiety, or the perception of a threat in order to manipulate users into buying unwanted software. Scareware is part of a class of malicious software that includes rogue security software, ransomware and other scam software with malicious payloads, which have limited or no benefit to users, and are pushed by unethical marketing practices. Some forms of spyware and adware also use scareware tactics.

A tactic frequently used by criminals involves convincing users that a virus has infected their computer, then suggesting that they download (and pay for) fake antivirus software to remove it.[1] Usually the virus is entirely fictional and the software is non-functional or malware itself.[2] According to the Anti-Phishing Working Group, the number of scareware packages in circulation rose from 2,850 to 9,287 in the second half of 2008.[3] In the first half of 2009, the APWG identified a 585% increase in scareware programs.[4]

The "scareware" label can also apply to any application or virus (not necessarily sold as above) which pranks users with intent to cause anxiety or panic.

4.10.1 Scam scareware

Internet Security bloggers/writers use the term "scareware" to describe software products that produce frivolous and alarming warnings or threat notices, most typically for fictitious or useless commercial firewall and registry cleaner software. This class of program tries to increase its perceived value by bombarding the user with constant warning messages that do not increase its effectiveness in any way. Software is packaged with a look and feel that mimics legitimate security software in order to deceive consumers.[5]

Some websites display pop-up advertisement windows or banners with text such as: "Your computer may be infected with harmful spyware programs.[6] Immediate removal may be required. To scan, click 'Yes' below." These websites can go as far as saying that a user's job, career, or marriage would be at risk.[7] Products using advertisements such as these are often considered scareware. Serious scareware applications qualify as rogue software.

In recent findings, some scareware is not affiliated with any other installed programs. A user can encounter a pop-up on a website indicating that their PC is infected.[8] In some scenarios, it is possible to become infected with scareware even if the user attempts to cancel the notification. These popups are especially designed to look like they come from the user's operating system when they are actually a webpage.

A 2010 study by Google found 11,000 domains hosting fake anti-virus software, accounting for 50% of all malware delivered via internet advertising.[9]

Starting on March 29, 2011, more than 1.5 million web sites around the world have been infected by the LizaMoon SQL injection attack spread by scareware.[10][11]

Research by Google discovered that scareware was using some of its servers to check for internet connectivity. The data suggested that up to a million machines were infected with scareware.[12] The company has placed a warning in the search results of users whose computers appear to be infected.

Another example of scareware is Smart Fortress. This site scares people into thinking they have lots of viruses on their computer and asks them to buy the professional service.[13]

Spyware

Dialog from SpySheriff, designed to scare users into installing the rogue software

Some forms of spyware also qualify as scareware because they change the user's desktop background, install icons in the computer's notification area (under Microsoft Windows), and generally make a nuisance of themselves, claiming that some kind of spyware has infected the user's computer and that the scareware application will help to remove the infection. In some cases, scareware trojans have replaced the desktop of the victim with large, yellow text reading "Warning! You have spyware!" or a box containing similar text, and have even forced the screensaver to change to "bugs" crawling across the screen. Winwebsec is the term usually used to address the malware that attacks the users of Windows operating system and produces fake

claims similar to that of genuine anti-malware software.[14]

SpySheriff[15] exemplifies spyware/scareware: it purports to remove spyware, but is actually a piece of spyware itself, often accompanying SmitFraud infections. Other antispyware scareware may be promoted using a phishing scam.

4.10.2 Uninstallation of security software

Another approach is to trick users into uninstalling legitimate antivirus software, such as Microsoft Security Essentials, or disabling their firewall.[16] Since antivirus programs typically include protection against being tampered with or disabled by other software, scareware may use social engineering to convince the user to disable programs which would otherwise prevent the malware from working.

4.10.3 Legal action

In 2005, Microsoft and Washington State successfully sued Secure Computer (makers of Spyware Cleaner) for $1 million over charges of using scareware pop-ups.[17] Washington's attorney general has also brought lawsuits against Securelink Networks, High Falls Media, and the makers of Quick Shield.[18]

In October 2008, Microsoft and the Washington attorney general filed a lawsuit against two Texas firms, Branch Software and Alpha Red, producers of the Registry Cleaner XP scareware.[19] The lawsuit alleges that the company sent incessant pop-ups resembling system warnings to consumers' personal computers stating "CRITICAL ERROR MESSAGE! - REGISTRY DAMAGED AND CORRUPTED", before instructing users to visit a web site to download Registry Cleaner XP at a cost of $39.95.

On December 2, 2008, the U.S. Federal Trade Commission ("FTC") filed a Complaint in federal court against Innovative Marketing, Inc., ByteHosting Internet Services, LLC, as well as individuals Sam Jain, Daniel Sundin, James Reno, Marc D'Souza, and Kristy Ross. The Complaint also listed Maurice D'Souza as a Relief Defendant, alleged that he held proceeds of wrongful conduct but not accusing him of violating any law. The FTC alleged that the other Defendants violated the FTC Act by deceptively marketing software, including WinFixer, WinAntivirus, DriveCleaner, ErrorSafe, and XP Antivirus. According to the complaint, the Defendants falsely represented that scans of a consumer's computer showed that it had been compromised or infected and then offered to sell software to fix the alleged problems.[20][21][22]

4.10.4 Prank software

Another type of scareware involves software designed to literally scare the user through the use of unanticipated shocking images, sounds or video.

- An early program of this type is NightMare, a program distributed on the Fish Disks for the Amiga computer (Fish #448) in 1991. When NightMare executes, it lies dormant for an extended (and random) period of time, finally changing the entire screen of the computer to an image of a skull while playing a horrifying shriek on the audio channels.[23]
- Anxiety-based scareware puts users in situations where there are no positive outcomes. For example, a small program can present a dialog box saying "Erase everything on hard drive?" with two buttons, both labeled "OK". Regardless of which button is chosen, nothing is destroyed other than the user's composure.[24]
- This tactic was used in an advertisement campaign by Sir-Tech in 1997 to advertise *Virus: The Game*. When the file is run, a full screen representation of the desktop appears. The software then begins simulating deletion of the Windows folder. When this process is complete, a message is slowly typed on screen saying "Thank God this is only a game." A screen with the purchase information appears on screen and then returns to the desktop. No damage is done to the computer during the advertisement.

4.10.5 See also

- Ransomware
- Rogue security software
- Winwebsec

4.10.6 Notes

[1] "Millions tricked by 'scareware'". BBC News. 2009-10-19. Retrieved 2009-10-20.

[2] 'Scareware' scams trick searchers. BBC News (2009-03-23). Retrieved on 2009-03-23.

[3] "Scareware scammers adopt cold call tactics". The Register. 2009-04-10. Retrieved 2009-04-12.

[4] Phishing Activity Trends Report: 1st Half 2009

[5] John Leydon (2009-10-20). "Scareware Mr Bigs enjoy 'low risk' crime bonanza". The Register. Retrieved 2009-10-21.

[6] Carine Febre (2014-10-20). "Fake Warning Example". Carine Febre. Retrieved 2014-11-21.

[7] "Symantec Security Response: Misleading Applications". Symantec. 2007-08-31. Retrieved 2010-04-15.

[8] JM Hipolito (2009-06-04). "Air France Flight 447 Search Results Lead to Rogue Antivirus". Trend Micro. Retrieved 2009-06-06.

[9] Moheeb Abu Rajab and Luca Ballard (2010-04-13). "The Nocebo Effect on the Web: An Analysis of Fake Anti-Virus Distribution" (PDF). Google. Retrieved 2010-11-18.

[10] content.usatoday.com

[11] reuters.com

[12] "Google to Warn PC Virus Victims via Search Site". BBC News. 2011-07-21. Retrieved 2011-07-22.

[13] "Smart Fortress 2012"

[14] Vincentas (11 July 2013). "Scareware in SpyWareLoop.com". Spyware Loop. Retrieved 27 July 2013.

[15] spywarewarrior.com filed under "Brave Sentry."

[16] theregister.co.uk

[17] Etengoff, Aharon (2008-09-29). "Washington and Microsoft target spammers". The Inquirer. Retrieved 2008-10-04.

[18] Tarun (2008-09-29). "Microsoft to sue scareware security vendors". *Lunarsoft*. Retrieved 2009-09-24. [...] the Washington attorney general (AG) [...] has also brought lawsuits against companies such as Securelink Networks and High Falls Media, and the makers of a product called QuickShield, all of whom were accused of marketing their products using deceptive techniques such as fake alert messages.

[19] "Fighting the scourge of scareware". BBC News. 2008-10-01. Retrieved 2008-10-02.

[20] "Win software". Federal Trade Commission.

[21] "Wanted by the FBI - SHAILESHKUMAR P. JAIN". FBI.

[22] "D'Souza Final Order" (PDF). Federal Trade Commission.

[23] Contents of disk #448. Amiga-stuff.com - see DISK 448.

[24] Dark Drive Prank

4.10.7 Further reading

- O'Dea, Hamish (2009-10-16). "The Modern Rogue – Malware With a Face". Australia: Microsoft.

4.10.8 External links

- Demonstration of scareware on YouTube
- The Case of the Unusable System
- Yes, that PC cleanup app you saw on TV at 3 a.m. is a waste

4.11 Rogue security software

Rogue security software is a form of malicious software and Internet fraud that misleads users into believing there is a virus on their computer, and manipulates them into paying money for a fake malware removal tool (that actually introduces malware to the computer). It is a form of scareware that manipulates users through fear, and a form of ransomware.[1] Rogue security software has become a serious security threat in desktop computing since 2008.[2]

4.11.1 Propagation

Rogue security software mainly relies on social engineering (fraud) to defeat the security built into modern operating system and browser software and install itself onto victims' computers.[2] A website may, for example, display a fictitious warning dialog stating that someone's machine is infected with a computer virus, and encourage them through manipulation to install or purchase scareware in the belief that they are purchasing genuine antivirus software.

Most have a Trojan horse component, which users are misled into installing. The Trojan may be disguised as:

- A browser plug-in or extension (typically toolbar)
- An image, screensaver or archive file attached to an e-mail message
- Multimedia codec required to play a certain video clip
- Software shared on peer-to-peer networks[3]
- A free online malware-scanning service[4]

Some rogue security software, however, propagate onto users' computers as drive-by downloads which exploit security vulnerabilities in web browsers, PDF viewers, or email clients to install themselves without any manual interaction.[3][5]

More recently, malware distributors have been utilizing SEO poisoning techniques by pushing infected URLs to the top of search engine results about recent news events. People looking for articles on such events on a search engine

may encounter results that, upon being clicked, are instead redirected through a series of sites[6] before arriving at a landing page that says that their machine is infected and pushes a download to a "trial" of the rogue program.[7][8] A 2010 study by Google found 11,000 domains hosting fake anti-virus software, accounting for 50% of all malware delivered via internet advertising.[9]

Cold-calling has also become a vector for distribution of this type of malware, with callers often claiming to be from "Microsoft Support" or another legitimate organization.[10]

4.11.2 Common Infection Vectors

Black Hat SEO

Black Hat search engine optimization (SEO) is a technique used to trick search engines into displaying malicious URLs in search results. The malicious webpages are filled with popular keywords in order to achieve a higher ranking in the search results. When the end user searches the web, one of these infected webpages is returned. Usually the most popular keywords from services such as Google Trends are used to generate webpages via PHP scripts placed on the compromised website. These PHP scripts will then monitor for search engine crawlers and feed them with specially crafted webpages that are then listed in the search results. Then, when the user searches for their keyword or images and clicks on the malicious link, they will be redirected to the Rogue security software payload.[11] [12]

Malvertising

Most websites usually employ third-party services for advertising on their webpages. If one of these advertising services is compromised,they may end up inadvertently infecting all of the websites using their service by showing advertising rogue security software. [13]

Spam Campaigns

Spam messages that include malicious attachments, links to binaries and driveby download sites are another common mechanism for distributing Rogue security software. Spam emails are often sent with content associated with typical day-to-day activities such as parcel deliveries, or taxation documents, designed to entice users to click on links or run attachments. When users succumb to these kinds of social engineering tricks they are quickly infected either directly via the attachment, or indirectly via a malicious website. This is known as a driveby download. Usually in drive-by download attacks the malware is installed on the victim's machine without any interaction or awareness and occurs simply by visiting the website. [14]

4.11.3 Operation

Once installed, the rogue security software may then attempt to entice the user into purchasing a service or additional software by:

- Alerting the user with the fake or simulated detection of malware or pornography.[15]
- Displaying an animation simulating a system crash and reboot.[2]
- Selectively disabling parts of the system to prevent the user from uninstalling the malware. Some may also prevent anti-malware programs from running, disable automatic system software updates and block access to websites of anti-malware vendors.
- Installing actual malware onto the computer, then alerting the user after "detecting" them. This method is less common as the malware is likely to be detected by legitimate anti-malware programs.
- Altering system registries and security settings, then "alerting" the user.

Developers of rogue security software may also entice people into purchasing their product by claiming to give a portion of their sales to a charitable cause. The rogue Green antivirus, for example, claims to donate $2 to an environmental care program for each sale made.[16]

Some rogue security software overlaps in function with scareware by also:

- Presenting offers to fix urgent performance problems or perform essential housekeeping on the computer.[15]
- Scaring the user by presenting authentic-looking pop-up warnings and security alerts, which may mimic actual system notices.[17] These are intended to use the trust that the user has in vendors of legitimate security software.[2]

Sanction by the FTC and the increasing effectiveness of anti-malware tools since 2006 have made it difficult for spyware and adware distribution networks—already complex to begin with[18]—to operate profitably.[19] Malware vendors have turned instead to the simpler, more profitable business model of rogue security software, which is targeted directly at users of desktop computers.[20]

Rogue security software is often distributed through highly lucrative affiliate networks, in which affiliates supplied with Trojan kits for the software are paid a fee for every successful installation, and a commission from any resulting purchases. The affiliates then become responsible for setting up infection vectors and distribution infrastructure for the software.[21] An investigation by security researchers into the Antivirus XP 2008 rogue security software found just such an affiliate network, in which members were grossing commissions upwards of $USD150,000 over 10 days, from tens of thousands of successful installations.[22]

4.11.4 Countermeasures

Private efforts

Law enforcement and legislation in all countries were very slow to react to the appearance of rogue security software even though it simply uses new technical means to carry out mainly old and well-established kinds of crimes. In contrast, several private initiatives providing discussion forums and lists of dangerous products were founded soon after the appearance of the first rogue security software. Some reputable vendors also began to provide lists of rogue security software, for example Kaspersky.[23] In 2005, the Anti-Spyware Coalition was founded, a coalition of anti-spyware software companies, academics, and consumer groups.

Many of the private initiatives were at first more or less informal discussions on general Internet forums, but some were started or even entirely carried out by individual people. The perhaps most famous and extensive one is the Spyware Warrior list of rogue/suspect antispyware products and websites by Eric Howes,[24] which has however not been updated since May 2007. The website recommends checking the following websites for new rogue anti-spyware programs, most of which are however not really new and are "simply re-branded clones and knockoffs of the same rogue applications that have been around for years"[25]

In December 2008, the US District Court for Maryland—at the request of the FTC—issued a restraining order against Innovative Marketing Inc, a Kiev-based firm producing and marketing the rogue security software products WinFixer, WinAntivirus, DriveCleaner, ErrorSafe, and XP Antivirus.[26] The company and its US-based web host, ByteHosting Internet Hosting Services LLC, had their assets frozen, were barred from using domain names associated with those products and any further advertisement or false representation.[27]

Law enforcement has also exerted pressure on banks to shut down merchant gateways involved in processing rogue security software purchases. In some cases, the high volume of credit card chargebacks generated by such purchases has also prompted processors to take action against rogue security software vendors.[28]

4.11.5 See also

- Anti-virus
- FraudTool
- List of rogue security software
- Scareware
- Technical support scam
- Winwebsec

4.11.6 References

[1] "Symantec Report on Rogue Security Software" (PDF). Symantec. 2009-10-28. Retrieved 2010-04-15.

[2] "Microsoft Security Intelligence Report volume 6 (July - December 2008)". Microsoft. 2009-04-08. p. 92. Retrieved 2009-05-02.

[3] Doshi, Nishant (2009-01-19), *Misleading Applications – Show Me The Money!*, Symantec, retrieved 2016-03-22

[4] Doshi, Nishant (2009-01-21), *Misleading Applications – Show Me The Money! (Part 2)*, Symantec, retrieved 2016-03-22

[5] "News Adobe Reader and Acrobat Vulnerability". blogs.adobe.com. Retrieved 25 November 2010.

[6] Chu, Kian; Hong, Choon (2009-09-30), *Samoa Earthquake News Leads To Rogue AV*, F-Secure, retrieved 2010-01-16

[7] Hines, Matthew (2009-10-08), *Malware Distributors Mastering News SEO*, eWeek, retrieved 2010-01-16

[8] Raywood, Dan (2010-01-15), *Rogue anti-virus prevalent on links that relate to Haiti earthquake, as donors encouraged to look carefully for genuine sites*, SC Magazine, retrieved 2010-01-16

[9] Moheeb Abu Rajab and Luca Ballard (2010-04-13). "The Nocebo Effect on the Web: An Analysis of Fake Anti-Virus Distribution" (PDF). Google. Retrieved 2010-11-18.

[10] "Warning over anti-virus cold-calls to UK internet users". BBC News. Retrieved 7 March 2012.

[11] "Sophos Technical Papers - Sophos SEO Insights". *sophos.com*.

[12] "Sophos Fake Antivirus Journey from Trojan tpna" (PDF).

[13] "Sophos Fake Antivirus Journey from Trojan tpna" (PDF).

[14] "Sophos Fake Antivirus Journey from Trojan tpna" (PDF).

[15] *"Free Security Scan" Could Cost Time and Money*, Federal Trade Commission, 2008-12-10, retrieved 2009-05-02

[16] "Cantalktech.com". *cantalktech.com*.

[17] "SAP at a crossroads after losing $1.3B verdict". Yahoo! News. 24 November 2010. Retrieved 25 November 2010.

[18] *Testimony of Ari Schwartz on "Spyware"* (PDF), Senate Committee on Commerce, Science, and Transportation, 2005-05-11

[19] Leyden, John (2009-04-11). "Zango goes titsup: End of desktop adware market". The Register. Retrieved 2009-05-05.

[20] Cole, Dave (2006-07-03), *Deceptonomics: A Glance at The Misleading Application Business Model*, Symantec, retrieved 2016-03-22

[21] Doshi, Nishant (2009-01-27), *Misleading Applications – Show Me The Money! (Part 3)*, Symantec, retrieved 2016-03-22

[22] Stewart, Joe. "Rogue Antivirus Dissected - Part 2". *Secureworks.com*. SecureWorks. Retrieved 9 March 2016.

[23] Rogue security software

[24] "Spyware Warrior: Rogue/Suspect Anti-Spyware Products & Web Sites". *spywarewarrior.com*.

[25] "Virus, Spyware, & Malware Removal Guides". *BleepingComputer*.

[26] *Ex Parte Temporary Restraining Order RDB08CV3233* (PDF), United States District Court for the District of Maryland, 2008-12-03, retrieved 2009-05-02

[27] Lordan, Betsy (2008-12-10), *Court Halts Bogus Computer Scans*, Federal Trade Commission, retrieved 2009-05-02

[28] Krebs, Brian (2009-03-20), "Rogue Antivirus Distribution Network Dismantled", *Washington Post*, retrieved 2009-05-02

4.11.7 External links

- Howes, Eric L (2007-05-04), *iSynergy Media Seo Services Guest posting Services*, retrieved 2009-05-02

4.12 Ransomware

Ransomware is a Cryptovirology attack carried out using covertly installed malware that encrypts the victim's files and then requests a ransom payment in return for the decryption key that is needed to recover the encrypted files.[1] Thus, ransomware is an access-denial type of attack that prevents legitimate users from accessing files[2] since it is intractable to decrypt the files without the decryption key. Other attacks superficially lock the system and display messages intended to coax the user into paying. Ransomware typically propagates as a Trojan, whose payload is disguised as a seemingly legitimate file.

While initially popular in Russia, the use of ransomware scams has grown internationally;[3][4][5] in June 2013, security software vendor McAfee released data showing that it had collected over 250,000 unique samples of ransomware in the first quarter of 2013, more than double the number it had obtained in the first quarter of 2012.[6] Wide-ranging attacks involving encryption-based ransomware began to increase through Trojans such as CryptoLocker, which had procured an estimated US$3 million before it was taken down by authorities,[7] and CryptoWall, which was estimated by the US Federal Bureau of Investigation (FBI) to have accrued over $18m by June 2015.[8]

4.12.1 Operation

The first secure data kidnapping attack was invented and implemented by Young and Yung in 1995 at Columbia University. They presented it at the 1996 IEEE Security & Privacy conference. It is called *cryptoviral extortion* and is the following 3-round protocol carried out between the attacker and the victim.[9]

1. [attacker→victim] The attacker generates a key pair and places the corresponding public key in the malware. The malware is released.
2. [victim→attacker] When the malware decides to attack, it generates a random symmetric key and encrypts the victim's data with it. It uses the public key in the malware to encrypt the symmetric key. This is known as hybrid encryption and it results in a small asymmetric ciphertext as well as the symmetric ciphertext of the victim's data. It zeroizes the symmetric key and the original plaintext data to prevent recovery. It puts up a message to the user that includes the asymmetric ciphertext and how to pay the ransom. The victim sends the asymmetric ciphertext and e-money to the attacker.
3. [attacker→victim] The attacker receives the payment, deciphers the asymmetric ciphertext with his private key, and sends the symmetric key to the victim. The victim deciphers the encrypted data with the needed symmetric key.

The symmetric key is randomly generated and will not assist other victims. At no point is the attacker's private key exposed to victims and the victim need only send a very small ciphertext to the attacker (the asymmetric ciphertext).

Ransomware attacks are typically carried out using a Trojan, entering a system through, for example, a downloaded file or a vulnerability in a network service. The program then runs a payload, which locks the system in some fashion, or claims to lock the system but does not (e.g., a scareware program). Payloads may display a fake warning purportedly by an entity such as a law enforcement agency, falsely claiming that the system has been used for illegal activities, contains content such as pornography and "pirated" media, or runs a non-genuine version of Microsoft Windows.[10][11][12]

Some payloads consist simply of an application designed to lock or restrict the system until payment is made, typically by setting the Windows Shell to itself,[13] or even modifying the master boot record and/or partition table to prevent the operating system from booting until it is repaired.[14] The most sophisticated payloads encrypt files, with many using strong encryption to encrypt the victim's files in such a way that only the malware author has the needed decryption key.[9][15][16]

Payment is virtually always the goal, and the victim is coerced into paying for the ransomware to be removed—which may or may not actually occur—either by supplying a program that can decrypt the files, or by sending an unlock code that undoes the payload's changes. A key element in making ransomware work for the attacker is a convenient payment system that is hard to trace. A range of such payment methods have been used, including wire transfers, premium-rate text messages,[17] pre-paid voucher services such as Paysafecard,[3][18][19] and the digital currency Bitcoin.[20][21][22] A 2016 census commissioned by Citrix revealed that larger business are holding bitcoin as contingency plans.[23]

4.12.2 History

Encrypting ransomware

The first known ransomware was "AIDS" (also known as "PC Cyborg"), written in 1989 by Joseph Popp. Its payload hid the files on the hard drive and encrypted their names, and displayed a message claiming that the user's license to use a certain piece of software had expired. The user was asked to pay US$189 to "PC Cyborg Corporation" in order to obtain a repair tool. Popp was declared mentally unfit to stand trial for his actions, but he promised to donate the profits from the malware to fund AIDS research.[24]

The notion of using public key cryptography for such attacks was introduced in 1996 by Adam L. Young and Moti Yung. Young and Yung showed that the AIDS Trojan was ineffective due to its use of symmetric cryptography, since the decryption key can be extracted from its code, and implemented an experimental proof-of-concept cryptovirus on a Macintosh SE/30 that used RSA and the Tiny Encryption Algorithm (TEA) to hybrid encrypt the victim's data. This cryptovirus, implemented in 1995 and described in the 1996 IEEE S&P paper, has the victim send the asymmetric ciphertext to the attacker who deciphers it and returns the symmetric decryption key it contains to the victim for a fee. Young and Yung also proposed that electronic money could be extorted through encryption as well, so that "the virus writer can effectively hold all of the money ransom until half of it is given to him".[9] They referred to these attacks as being "cryptoviral extortion", an overt attack that is part of a larger class of attacks in a field called cryptovirology, which encompasses both overt and covert attacks.[9]

Examples of extortionate ransomware became prominent in May 2005.[25] By mid-2006, Trojans such as Gpcode, TROJ.RANSOM.A, Archiveus, Krotten, Cryzip, and MayArchive began utilizing more sophisticated RSA encryption schemes, with ever-increasing key-sizes. Gpcode.AG, which was detected in June 2006, was encrypted with a 660-bit RSA public key.[26] In June 2008, a variant known as Gpcode.AK was detected. Using a 1024-bit RSA key, it was believed large enough to be computationally infeasible to break without a concerted distributed effort.[27][28][29][30]

Encrypting ransomware returned to prominence in late 2013 with the propagation of CryptoLocker—using the Bitcoin digital currency platform to collect ransom money. In December 2013, *ZDNet* estimated based on Bitcoin transaction information that between 15 October and 18 December, the operators of CryptoLocker had procured about US$27 million from infected users.[31] The CryptoLocker technique was widely copied in the months following, including CryptoLocker 2.0 (though not to be related to CryptoLocker), CryptoDefense (which initially contained a major design flaw that stored the private key on the infected system in a user-retrievable location, due to its use of Windows' built-in encryption APIs),[21][32][33][34] and the August 2014 discovery of a Trojan specifically targeting network-attached storage devices produced by Synology.[35] In January 2015, it was reported that ransomware-styled attacks have occurred against individual websites via hacking, and through ransomware designed to target Linux-based web servers.[36][37][38]

Some ransomware strains have used proxies tied to Tor hidden services to connect to their command and control servers, increasing the difficulty of tracing the exact location of the criminals.[39][40] Furthermore. dark web vendors have increasingly started to offer the technology as a service.[40][41][42]

Non-encrypting ransomware

In August 2010, Russian authorities arrested ten individuals connected to a ransomware Trojan known as WinLock. Unlike the previous Gpcode Trojan, WinLock did not use encryption. Instead, WinLock trivially restricted access to the system by displaying pornographic images, and asked users to send a premium-rate SMS (costing around US$10) to receive a code that could be used to unlock their machines. The scam hit numerous users across Russia and neighboring countries—reportedly earning the group over US$16 million.[12][43]

In 2011, a ransomware Trojan surfaced that imitated the Windows Product Activation notice, and informed users that a system's Windows installation had to be re-activated due to "[being a] victim of fraud". An online activation option was offered (like the actual Windows activation process), but was unavailable, requiring the user to call one of six international numbers to input a 6-digit code. While the malware claimed that this call would be free, it was routed through a rogue operator in a country with high international phone rates, who placed the call on hold, causing the user to incur large international long distance charges.[10]

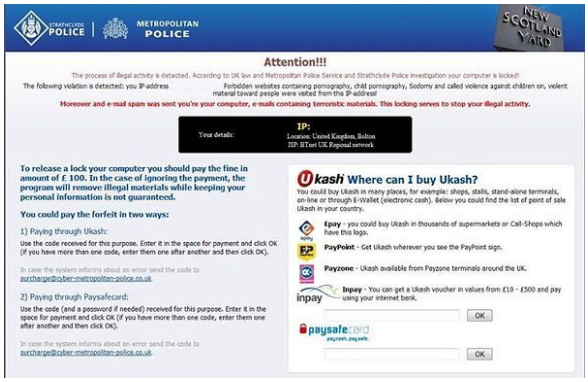
In February 2013, a ransomware Trojan based on the Stamp.EK exploit kit surfaced; the malware was distributed via sites hosted on the project hosting services SourceForge and GitHub that claimed to offer "fake nude pics" of celebrities.[44] In July 2013, an OS X-specific ransomware Trojan surfaced, which displays a web page that accuses the user of downloading pornography. Unlike its Windows-based counterparts, it does not block the entire computer, but simply exploits the behavior of the web browser itself to frustrate attempts to close the page through normal means.[45]

In July 2013, a 21-year-old man from Virginia, whose computer coincidentally did contain pornographic photographs of underaged girls with whom he had conducted sexualized communications, turned himself in to police after receiving and being deceived by ransomware purporting to be an FBI message accusing him of possessing child pornography. An investigation discovered the incriminating files, and the man was charged with child sexual abuse and possession of child pornography.[46]

4.12.3 Notable examples

Reveton

In 2012, a major ransomware Trojan known as Reveton began to spread. Based on the Citadel Trojan (which itself, is based on the Zeus Trojan), its payload displays a warning purportedly from a law enforcement agency claim-



A Reveton payload, fraudulently claiming that the user must pay a fine to the Metropolitan Police Service

ing that the computer has been used for illegal activities, such as downloading unlicensed software or child pornography. Due to this behaviour, it is commonly referred to as the "Police Trojan".[47][48][49] The warning informs the user that to unlock their system, they would have to pay a fine using a voucher from an anonymous prepaid cash service such as Ukash or Paysafecard. To increase the illusion that the computer is being tracked by law enforcement, the screen also displays the computer's IP address, while some versions display footage from a victim's webcam to give the illusion that the user is being recorded.[3][50]

Reveton initially began spreading in various European countries in early 2012.[3] Variants were localized with templates branded with the logos of different law enforcement organizations based on the user's country; for example, variants used in the United Kingdom contained the branding of organizations such as the Metropolitan Police Service and the Police National E-Crime Unit. Another version contained the logo of the royalty collection society PRS for Music, which specifically accused the user of illegally downloading music.[51] In a statement warning the public about the malware, the Metropolitan Police clarified that they would never lock a computer in such a way as part of an investigation.[3][11]

In May 2012, Trend Micro threat researchers discovered templates for variations for the United States and Canada, suggesting that its authors may have been planning to target users in North America.[52] By August 2012, a new variant of Reveton began to spread in the United States, claiming to require the payment of a $200 fine to the FBI using a MoneyPak card.[4][5][50] In February 2013, a Russian citizen was arrested in Dubai by Spanish authorities for his connection to a crime ring that had been using Reveton; ten other individuals were arrested on money laundering charges.[53] In August 2014, Avast Software reported that it had found new variants of Reveton that also distribute password stealing malware as part of its payload.[54]

CryptoLocker

Main article: CryptoLocker

Encrypting ransomware reappeared in September 2013 with a Trojan known as *CryptoLocker*, which generated a 2048-bit RSA key pair and uploaded in turn to a command-and-control server, and used to encrypt files using a whitelist of specific file extensions. The malware threatened to delete the private key if a payment of Bitcoin or a pre-paid cash voucher was not made within 3 days of the infection. Due to the extremely large key size it uses, analysts and those affected by the Trojan considered CryptoLocker extremely difficult to repair.[20][55][56][57] Even after the deadline passed, the private key could still be obtained using an online tool, but the price would increase to 10 BTC—which cost approximately US$2300 as of November 2013.[58][59]

CryptoLocker was isolated by the seizure of the Gameover ZeuS botnet as part of Operation Tovar, as officially announced by the U.S. Department of Justice on 2 June 2014. The Department of Justice also publicly issued an indictment against the Russian hacker Evgeniy Bogachev for his alleged involvement in the botnet.[60][61] It was estimated that at least US$3 million was extorted with the malware before the shutdown.[7]

CryptoLocker.F and TorrentLocker

In September 2014, a wave of ransomware Trojans surfaced that first targeted users in Australia, under the names *CryptoWall* and *CryptoLocker* (which is, as with CryptoLocker 2.0, unrelated to the original CryptoLocker). The Trojans spread via fraudulent e-mails claiming to be failed parcel delivery notices from Australia Post; to evade detection by automatic e-mail scanners that follow all links on a page to scan for malware, this variant was designed to require users to visit a web page and enter a CAPTCHA code before the payload is actually downloaded, preventing such automated processes from being able to scan the payload. Symantec determined that these new variants, which it identified as *CryptoLocker.F*, were again, unrelated to the original CryptoLocker due to differences in their operation.[62][63] A notable victim of the Trojans was the Australian Broadcasting Corporation; live programming on its television news channel ABC News 24 was disrupted for half an hour and shifted to Melbourne studios due to a CryptoWall infection on computers at its Sydney studio.[64][65][66]

Another Trojan in this wave, TorrentLocker, initially contained a design flaw comparable to CryptoDefense; it used the same keystream for every infected computer, making the encryption trivial to overcome. However, this flaw was later fixed.[32] By late-November 2014, it was estimated that over 9,000 users had been infected by TorrentLocker in Australia alone, trailing only Turkey with 11,700 infections.[67]

CryptoWall

Another major ransomware Trojan targeting Windows, CryptoWall, first appeared in 2014. One strain of CryptoWall was distributed as part of a malvertising campaign on the Zedo ad network in late-September 2014 that targeted several major websites; the ads redirected to rogue websites that used browser plugin exploits to download the payload. A Barracuda Networks researcher also noted that the payload was signed with a digital signature in an effort to appear trustworthy to security software.[68] CryptoWall 3.0 used a payload written in JavaScript as part of an email attachment, which downloads executables disguised as JPG images. To further evade detection, the malware creates new instances of explorer.exe and svchost.exe to communicate with its servers. When encrypting files, the malware also deletes volume shadow copies, and installs spyware that steals passwords and Bitcoin wallets.[69]

The FBI reported in June 2015 that nearly 1,000 victims had contacted the bureau's Internet Crime Complaint Center to report CryptoWall infections, and estimated losses of at least $18 million.[8]

The most recent version, CryptoWall 4.0, enhanced its code to avoid antivirus detection, and encrypts not only the data in files but also the file names.[70]

4.12.4 Mitigation

As with other forms of malware, security software might not detect a ransomware payload, or, especially in the case of encrypting payloads, only after encryption is underway or complete, particularly if a new version unknown to the protective software is distributed.[71] If an attack is suspected or detected in its early stages, it takes some time for encryption to take place; immediate removal of the malware (a relatively simple process) before it has completed would stop further damage to data, without salvaging any already lost.[72][73] Security experts have suggested precautionary measures for dealing with ransomware. Using software or other security policies to block known payloads from launching will help to prevent infection, but will not protect against all attacks. Keeping "offline" backups of data stored in locations inaccessible to the infected computer, such as external storage drives, prevents them from being accessed by the ransomware, thus accelerating data restoration.[20][74]

4.12.5 See also

- Phishing

4.12.6 References

[1] Mehmood, Shafqat (3 May 2016). "Enterprise Survival Guide for Ransomware Attacks". *SANS Information Security Training | Cyber Certifications | Research*. www.sans.org. Retrieved 3 May 2016.

[2] Dr. Sam Musa. 5 Steps to Take on Ransomware. Retrieved from http://www.govtech.com/security/5-Steps-Ransomware-Defense-in-Layers-Approach.html

[3] Dunn, John E. "Ransom Trojans spreading beyond Russian heartland". TechWorld. Retrieved 10 March 2012.

[4] "New Internet scam: Ransomware...". FBI. 9 August 2012.

[5] "Citadel malware continues to deliver Reveton ransomware...". Internet Crime Complaint Center (IC3). 30 November 2012.

[6] "Update: McAfee: Cyber criminals using Android malware and ransomware the most". *InfoWorld*. Retrieved 16 September 2013.

[7] "Cryptolocker victims to get files back for free". BBC News. 6 August 2014. Retrieved 18 August 2014.

[8] "FBI says crypto ransomware has raked in >$18 million for cybercriminals". *Ars Technica*. Retrieved 25 June 2015.

[9] Young, A.; M. Yung (1996). *Cryptovirology: extortion-based security threats and countermeasures*. IEEE Symposium on Security and Privacy. pp. 129–140. doi:10.1109/SECPRI.1996.502676. ISBN 0-8186-7417-2.

[10] "Ransomware squeezes users with bogus Windows activation demand". Computerworld. Retrieved 9 March 2012.

[11] "Police warn of extortion messages sent in their name". Helsingin Sanomat. Retrieved 9 March 2012.

[12] McMillian, Robert. "Alleged Ransomware Gang Investigated by Moscow Police". PC World. Retrieved 10 March 2012.

[13] "Ransomware: Fake Federal German Police (BKA) notice". SecureList (Kaspersky Lab). Retrieved 10 March 2012.

[14] "And Now, an MBR Ransomware". SecureList (Kaspersky Lab). Retrieved 10 March 2012.

[15] Adam Young (2005). Zhou, Jianying; Lopez, Javier, eds. "Building a Cryptovirus Using Microsoft's Cryptographic API". *Information Security: 8th International Conference, ISC 2005* (Springer-Verlag). pp. 389–401.

[16] Young, Adam (2006). "Cryptoviral Extortion Using Microsoft's Crypto API: Can Crypto APIs Help the Enemy?". *International Journal of Information Security* (Springer-Verlag) **5** (2): 67–76. doi:10.1007/s10207-006-0082-7.

[17] Danchev, Dancho (22 April 2009). "New ransomware locks PCs, demands premium SMS for removal". ZDNet. Retrieved 2 May 2009.

[18] "Ransomware plays pirated Windows card, demands $143". Computerworld. Retrieved 9 March 2012.

[19] Cheng, Jacqui (18 July 2007). "New Trojans: give us $300, or the data gets it!". Ars Technica. Retrieved 16 April 2009.

[20] "You're infected—if you want to see your data again, pay us $300 in Bitcoins". *Ars Technica*. Retrieved 23 October 2013.

[21] "CryptoDefense ransomware leaves decryption key accessible". *Computerworld*. IDG. Retrieved 7 April 2014.

[22] "What to do if Ransomware Attacks on your Windows Computer?". *Techie Motto*. Retrieved 25 April 2016.

[23] Parker, Luke (9 June 2016). "Large UK businesses are holding bitcoin to pay ransoms". Retrieved 9 June 2016.

[24] Kassner, Michael. "Ransomware: Extortion via the Internet". TechRepublic. Retrieved 10 March 2012.

[25] Schaibly, Susan (26 September 2005). "Files for ransom". Network World. Retrieved 17 April 2009.

[26] Leyden, John (24 July 2006). "Ransomware getting harder to break". The Register. Retrieved 18 April 2009.

[27] Naraine, Ryan (6 June 2008). "Blackmail ransomware returns with 1024-bit encryption key". ZDNet. Retrieved 3 May 2009.

[28] Lemos, Robert (13 June 2008). "Ransomware resisting crypto cracking efforts". SecurityFocus. Retrieved 18 April 2009.

[29] Krebs, Brian (9 June 2008). "Ransomware Encrypts Victim Files with 1,024-Bit Key". *The Washington Post*. Retrieved 16 April 2009.

[30] "Kaspersky Lab reports a new and dangerous blackmailing virus". Kaspersky Lab. 5 June 2008. Retrieved 11 June 2008.

[31] Violet Blue (22 December 2013). "CryptoLocker's crimewave: A trail of millions in laundered Bitcoin". ZDNet. Retrieved 23 December 2013.

[32] "Encryption goof fixed in TorrentLocker file-locking malware". *PC World*. Retrieved 15 October 2014.

[33] "Cryptolocker 2.0 – new version, or copycat?". *WeLiveSecurity*. ESET. Retrieved 18 January 2014.

[34] "New CryptoLocker Spreads via Removable Drives". Trend Micro. Retrieved 18 January 2014.

[35] "Synology NAS devices targeted by hackers, demand Bitcoin ransom to decrypt files". *ExtremeTech* (Ziff Davis Media). Retrieved 18 August 2014.

[36] "File-encrypting ransomware starts targeting Linux web servers". *PC World*. IDG. Retrieved 31 May 2016.

[37] "Cybercriminals Encrypt Website Databases in "RansomWeb" Attacks". *Security Week*. Retrieved 31 May 2016.

[38] "Hackers holding websites to ransom by switching their encryption keys". *The Guardian*. Retrieved 31 May 2016.

[39] "New ransomware employs Tor to stay hidden from security". *The Guardian*. Retrieved 31 May 2016.

[40] "The current state of ransomware: CTB-Locker". *Sophos Blog*. Sophos. Retrieved 31 May 2016.

[41] Brook, Chris (4 June 2015). "Author Behind Ransomware Tox Calls it Quits, Sells Platform". Retrieved 6 August 2015.

[42] Dela Paz, Roland (29 July 2015). "Encryptor RaaS: Yet another new Ransomware-as-a-Service on the Block". Retrieved 6 August 2015.

[43] Leyden, John. "Russian cops cuff 10 ransomware Trojan suspects". The Register. Retrieved 10 March 2012.

[44] "Criminals push ransomware hosted on GitHub and SourceForge pages by spamming 'fake nude pics' of celebrities". *TheNextWeb*. Retrieved 17 July 2013.

[45] "New OS X malware holds Macs for ransom, demands $300 fine to the FBI for 'viewing or distributing' porn". *TheNextWeb*. Retrieved 17 July 2013.

[46] "Man gets ransomware porn pop-up, goes to cops, gets arrested on child porn charges". *Ars Technica*. Retrieved 31 July 2013.

[47] "Gardaí warn of 'Police Trojan' computer locking virus". *TheJournal.ie*. Retrieved 31 May 2016.

[48] "Barrie computer expert seeing an increase in the effects of the new ransomware". *Barrie Examiner*. Postmedia Network. Retrieved 31 May 2016.

[49] "Fake cop Trojan 'detects offensive materials' on PCs, demands money". The Register. Retrieved 15 August 2012.

[50] "Reveton Malware Freezes PCs, Demands Payment". InformationWeek. Retrieved 16 August 2012.

[51] Dunn, John E. "Police alert after ransom Trojan locks up 1,100 PCs". TechWorld. Retrieved 16 August 2012.

[52] Constantian, Lucian. "Police-themed Ransomware Starts Targeting US and Canadian Users". PC World. Retrieved 11 May 2012.

[53] "Reveton 'police ransom' malware gang head arrested in Dubai". *TechWorld*. Retrieved 18 October 2014.

[54] "'Reveton' ransomware upgraded with powerful password stealer". *PC World*. Retrieved 18 October 2014.

[55] "Disk encrypting Cryptolocker malware demands $300 to decrypt your files". *Geek.com*. Retrieved 12 September 2013.

[56] "CryptoLocker attacks that hold your computer to ransom". *The Guardian*. Retrieved 23 October 2013.

[57] "Destructive malware "CryptoLocker" on the loose - here's what to do". *Naked Security*. Sophos. Retrieved 23 October 2013.

[58] "CryptoLocker crooks charge 10 Bitcoins for second-chance decryption service". *NetworkWorld*. Retrieved 5 November 2013.

[59] "CryptoLocker creators try to extort even more money from victims with new service". *PC World*. Retrieved 5 November 2013.

[60] "Wham bam: Global Operation Tovar whacks CryptoLocker ransomware & GameOver Zeus botnet". *Computerworld*. IDG. Retrieved 18 August 2014.

[61] "U.S. Leads Multi-National Action Against "Gameover Zeus" Botnet and "Cryptolocker" Ransomware, Charges Botnet Administrator". *Justice.gov*. U.S. Department of Justice. Retrieved 18 August 2014.

[62] "Australians increasingly hit by global tide of cryptomalware". Symantec. Retrieved 15 October 2014.

[63] Grubb, Ben (17 September 2014). "Hackers lock up thousands of Australian computers, demand ransom". *Sydney Morning Herald*. Retrieved 15 October 2014.

[64] "Australia specifically targeted by Cryptolocker: Symantec". *ARNnet*. 3 October 2014. Retrieved 15 October 2014.

[65] "Scammers use Australia Post to mask email attacks". *Sydney Morning Herald*. 15 October 2014. Retrieved 15 October 2014.

[66] "Ransomware attack knocks TV station off air". *CSO*. Retrieved 15 October 2014.

[67] "Over 9,000 (Vegeta - "OVER 9000!!!!!!") PCs in Australia infected by TorrentLocker ransomware". *CSO.com.au*. Retrieved 18 December 2014.

[68] "Malvertising campaign delivers digitally signed CryptoWall ransomware". *PC World*. Retrieved 25 June 2015.

[69] "CryptoWall 3.0 Ransomware Partners With FAREIT Spyware". Trend Micro. Retrieved 25 June 2015.

[70] Andra Zaharia (5 November 2015). "Security Alert: CryptoWall 4.0 – new, enhanced and more difficult to detect". *HEIMDAL*. Retrieved 5 January 2016.

[71] "Yuma Sun weathers malware attack". *Yuma Sun*. Retrieved 18 August 2014.

[72] Cannell, Joshua. "Cryptolocker Ransomware: What You Need To Know, last updated 06/02/2014". *Malwarebytes Unpacked*. Retrieved 19 October 2013.

[73] Leyden, Josh. "Fiendish CryptoLocker ransomware: Whatever you do, don't PAY". *The Register*. Retrieved 18 October 2013.

[74] "Cryptolocker Infections on the Rise; US-CERT Issues Warning". *Security Week*. 19 November 2013. Retrieved 18 January 2014.

4.12.7 Further reading

- Russinovich, Mark (7 January 2013). "Hunting Down and Killing Ransomware (Scareware)". *Microsoft TechNet blog*.
- Simonite, Tom (4 February 2015). "Holding Data Hostage: The Perfect Internet Crime? Ransomware (Scareware)". *MIT Technology Review*.
- Brad, Duncan (2 March 2015). "Exploit Kits and CryptoWall 3.0". The Rackspace Blog! & NewsRoom.
- "Ransomware on the Rise". The Federal Bureau of Investigation JANUARY 2015.
- Yang, T.; Yang, Y.; Qian, K.; Lo, D.C.T.; Qian, L. & Tao, L. "Automated Detection and Analysis for Android Ransomware". IEEE Internet of Things Journal, CONFERENCE, AUGUST 2015.
- Richet, Jean-Loup. "Extortion on the Internet: the Rise of Crypto-Ransomware" (PDF). Harvard.

4.12.8 External links

- Incidents of Ransomware on the Rise – Federal Bureau of Investigation
- Geeknights 20160418: Ransomware

Chapter 5

Malware for Different Operating System

5.1 Linux malware

Linux malware includes viruses, trojans, worms and other types of malware that affect the Linux operating system. Linux, Unix and other Unix-like computer operating systems are generally regarded as very well-protected against, but not immune to, computer viruses.[1][2]

There has not yet been a single widespread Linux virus or malware infection of the type that is common on Microsoft Windows; this is attributable generally to the malware's lack of root access and fast updates to most Linux vulnerabilities.[2]

5.1.1 Linux vulnerability

Like Unix systems, Linux implements a multi-user environment where users are granted specific privileges and there is some form of access control implemented. To gain control over a Linux system or to cause any serious consequences to the system itself, the malware would have to gain root access to the system.[2]

In the past, it has been suggested that Linux had so little malware because its low market share made it a less profitable target. Rick Moen, an experienced Linux system administrator, counters that:

> [That argument] ignores Unix's dominance in a number of non-desktop specialties, including Web servers and scientific workstations. A virus/trojan/worm author who successfully targeted specifically Apache httpd Linux/x86 Web servers would both have an extremely target-rich environment and instantly earn lasting fame, and yet it doesn't happen.[3]

In 2008 the quantity of malware targeting Linux was noted as increasing. Shane Coursen, a senior technical consultant with Kaspersky Lab, said at the time, "The growth in Linux malware is simply due to its increasing popularity, particularly as a desktop operating system ... The use of an operating system is directly correlated to the interest by the malware writers to develop malware for that OS."[4]

Tom Ferris, a researcher with Security Protocols, commented on one of Kaspersky's reports, stating, "In people's minds, if it's non-Windows, it's secure, and that's not the case. They think nobody writes malware for Linux or Mac OS X. But that's not necessarily true,"[4]

Some Linux users do run Linux-based anti-virus software to scan insecure documents and email which comes from or is going to Windows users. SecurityFocus's Scott Granneman stated:

> ...some Linux machines definitely need anti-virus software. Samba or NFS servers, for instance, may store documents in undocumented, vulnerable Microsoft formats, such as Word and Excel, that contain and propagate viruses. Linux mail servers should run AV software in order to neutralize viruses before they show up in the mailboxes of Outlook and Outlook Express users.[1]

Because they are predominantly used on mail servers which may send mail to computers running other operating systems, Linux virus scanners generally use definitions for, and scan for, all known viruses for all computer platforms. For example, the open source ClamAV "Detects ... viruses, worms and trojans, including Microsoft Office macro viruses, mobile malware, and other threats."[5]

Viruses and trojan horses

The viruses listed below pose a potential, although minimal, threat to Linux systems. If an infected binary containing one of the viruses were run, the system would be temporarily infected; Linux kernel is memory resident and read-only. Any infection level would depend on which user with what privileges ran the binary. A binary run under the root account would be able to infect the entire system. Privilege

escalation vulnerabilities may permit malware running under a limited account to infect the entire system.

It is worth noting that this is true for any malicious program that is run without special steps taken to limit its privileges. It is trivial to add a code snippet to any program that a user may download and let this additional code download a modified login server, an open mail relay, or similar program, and make this additional component run any time the user logs in. No special malware writing skills are needed for this. Special skill may be needed for tricking the user to run the (trojan) program in the first place.

The use of software repositories significantly reduces any threat of installation of malware, as the software repositories are checked by maintainers, who try to ensure that their repository is malware-free. Subsequently, to ensure safe distribution of the software, checksums are made available. These make it possible to reveal modified versions that may have been introduced by e.g. hijacking of communications using a man-in-the-middle attack or via a redirection attack such as ARP or DNS poisoning. Careful use of these digital signatures provides an additional line of defense, which limits the scope of attacks to include only the original authors, package and release maintainers and possibly others with suitable administrative access, depending on how the keys and checksums are handled.

Worms and targeted attacks

The classical threat to Unix-like systems are vulnerabilities in network daemons, such as SSH and web servers. These can be used by worms or for attacks against specific targets. As servers are patched quite quickly when a vulnerability is found, there have been only a few widespread worms of this kind. As specific targets can be attacked through a vulnerability that is not publicly known there is no guarantee that a certain installation is secure. Also servers without such vulnerabilities can be successfully attacked through weak passwords.

Web scripts

Linux servers may also be used by malware without any attack against the system itself, where e.g. web content and scripts are insufficiently restricted or checked and used by malware to attack visitors. Some attacks use complicated malware to attack Linux servers, but when most get full root access then hackers are able to attack by[6] modifying anything like replacing binaries or injecting modules. This may allow the redirection of users to different content on the web.[7] Typically, a CGI script meant for leaving comments, could, by mistake, allow inclusion of code exploiting vulnerabilities in the web browser.

Buffer overruns

Older Linux distributions were relatively sensitive to buffer overrun attacks: if the program did not care about the size of the buffer itself, the kernel provided only limited protection, allowing an attacker to execute arbitrary code under the rights of the vulnerable application under attack. Programs that gain root access even when launched by a non-root user (via the setuid bit) were particularly attractive to attack. However, as of 2009 most of the kernels include address space layout randomization (ASLR), enhanced memory protection and other extensions making such attacks much more difficult to arrange.

Cross-platform viruses

An area of concern identified in 2007 is that of cross-platform viruses, driven by the popularity of cross-platform applications. This was brought to the forefront of malware awareness by the distribution of an OpenOffice.org virus called Badbunny.

Stuart Smith of Symantec wrote the following:

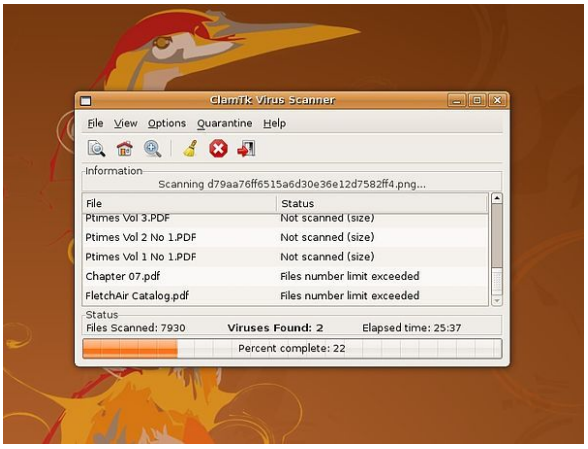
> What makes this virus worth mentioning is that it illustrates how easily scripting platforms, extensibility, plug-ins, ActiveX, etc, can be abused. All too often, this is forgotten in the pursuit to match features with another vendor... The ability for malware to survive in a cross-platform, cross-application environment has particular relevance as more and more malware is pushed out via Web sites. How long until someone uses something like this to drop a JavaScript infecter on a Web server, regardless of platform?[8]

Social engineering

As is the case with any operating system, Linux is vulnerable to malware that tricks the user into installing it through social engineering. In December 2009 a malicious waterfall screensaver was discovered that contained a script that used the infected Linux PC in denial-of-service attacks.[9]

5.1.2 Anti-virus applications

There are a number of anti-virus applications available which will run under the Linux operating system. Most of these applications are looking for exploits which could affect users of Microsoft Windows.



The ClamTk GUI for ClamAV running a scan on Ubuntu 8.04 Hardy Heron

For Microsoft Windows-specific threats

These applications are useful for computers (typically, servers) which will pass on files to MS Windows users. They do not look for Linux-specific threats.

- Avast! (proprietary; freeware version available)
- AVG (proprietary; freeware version available)
- Avira (proprietary; freeware version was available, discontinued due to lack of demand)[10]
- BitDefender (proprietary; freeware version available)
- Comodo (proprietary; freeware version available) [11]
- ClamAV (free and open source software)[12]
- Dr.Web (proprietary) [13]
- EScan for Linux (proprietary)
- F-Prot (proprietary; freeware version available)[14]
- F-Secure Linux (proprietary)
- Kaspersky Linux Security (proprietary)[15]
- McAfee VirusScan Enterprise for Linux (proprietary)[16]
- Panda Security for Linux (proprietary)[17]
- Symantec AntiVirus for Linux (proprietary)[18]
- Trend Micro ServerProtect for Linux (proprietary)

For Linux-specific threats

These applications look for actual threats to the Linux computers on which they are running.

- chkrootkit (free and open source software)[19]
- Comodo (proprietary) [20]
- ESET (proprietary) (detects OS X, Windows malware as well)[21][22][23][24]
- rkhunter (free and open source software)[25]
- Sophos (proprietary) (detects Windows malware, too)[26][27]

Linux malware can also be detected (and analyzed) using memory forensics tools, such as the following.

- Second Look (proprietary)[28]
- Volatility[29] (free and open source software)[30]

5.1.3 Threats

The following is a partial list of known Linux malware. However, few if any are in the wild, and most have been rendered obsolete by Linux updates or were never a threat. Known malware is not the only or even the most important threat: new malware or attacks directed to specific sites can use vulnerabilities previously unknown to the community or unused by malware.

Botnets

- Mayhem - 32/64-bit Linux/FreeBSD multifunctional botnet[31]
- Linux/Remaiten - A threat targeting the Internet of Things.[32][33][34]

Rootkits

- Snakso-A - 64-bit Linux webserver rootkit[35]

Trojans

- Effusion - 32/64-bit injector for Apache/Nginx webservers, (7 Jan 2014)[36]
- Hand of Thief - Banking trojan, 2013,[37][38]
- Kaiten - Linux.Backdoor.Kaiten trojan horse[39]

- Rexob - Linux.Backdoor.Rexob trojan[40]
- Waterfall screensaver backdoor - on gnome-look.org[41]
- Tsunami.gen — Backdoor.Linux.Tsunami.gen[42]
- Turla — HEUR:Backdoor.Linux.Turla.gen[43][44]
- Xor DDoS A Trojan malware that hijacks Linux systems and uses them to launch DDoS attacks which have reached loads of 150+ Gbps.[45]
- HummingBad - has infected over 10 million Android operating systems. User details are sold and adverts are tapped on without the user's knowledge thereby generating fraudulent advertising revenue.[46]

Viruses

- 42[47][48]
- Arches[49]
- Alaeda - Virus.Linux.Alaeda[50]
- Binom - Linux/Binom[51]
- Bliss - requires root privileges
- Brundle[52]
- Bukowski[53]
- Caveat [54][55]
- Coin [56][57]
- Diesel - Virus.Linux.Diesel.962[58]
- Hasher [59][60]
- Kagob a - Virus.Linux.Kagob.a[61]
- Kagob b - Virus.Linux.Kagob.b[62]
- Lacrimae (aka Crimea) [63][64]
- Linux.Encoder.1[65][66]
- MetaPHOR (also known as Simile)[67]
- Nuxbee - Virus.Linux.Nuxbee.1403[68]
- OSF.8759
- PiLoT[69][70]
- Podloso - Linux.Podloso (The iPod virus)[71][72]
- RELx [73]
- Rike - Virus.Linux.Rike.1627[74]
- RST - Virus.Linux.RST.a[75] (known for infecting Korean release of Mozilla Suite 1.7.6 and Thunderbird 1.0.2 in September 2005[76])
- Satyr - Virus.Linux.Satyr.a[77]
- Staog
- Vit - Virus.Linux.Vit.4096[78]
- Winter - Virus.Linux.Winter.341[79]
- Winux (also known as Lindose and PEElf)[80]
- Wit virus[81]
- Zariche - Linux.Zariche.A (and variants)[82]
- ZipWorm - Virus.Linux.ZipWorm[83]

Worms

- Adm - Net-Worm.Linux.Adm[84]
- Adore[85]
- Bad Bunny - Perl.Badbunny[8][86]
- Cheese - Net-Worm.Linux.Cheese[87]
- Devnull
- Kork[88]
- Linux/Lion
- Linux.Darlloz - Targets home routers, set-top boxes, security cameras and industrial control systems.[89][90]
- Linux/Lupper.worm[91]
- Mighty - Net-Worm.Linux.Mighty[92]
- Millen - Linux.Millen.Worm[93]
- Ramen worm - targeted only Red Hat Linux distributions versions 6.2 and 7.0
- Slapper[94]
- SSH Bruteforce[95]

5.1.4 See also

- Comparison of computer viruses
- Timeline of computer viruses and worms
- Trojan horse (computing)
- Computer worm
- Computer virus
- Botnet
- Spyware

5.1.5 References

[1] Granneman, Scott (October 2003). "Linux vs. Windows Viruses". Retrieved 2008-03-06.

[2] Yeargin, Ray (July 2005). "The short life and hard times of a linux virus". Archived from the original on 1 May 2008. Retrieved 2015-12-06.

[3] "Virus Department". Retrieved 2015-12-24.

[4] Patrizio, Andy (April 2006). "Linux Malware On The Rise". Retrieved 2008-03-08.

[5] ClamAV (2010). "Clam AntiVirus 0.96 User Manual" (PDF). Retrieved 2011-02-22.

[6] Prince, Brian (5 January 2013). "Stealthy Apache Exploit Redirects Victims to Blackhole Malware.".

[7] Prince, Brian (May 1, 2013). "Stealthy Apache Exploit Redirects Victims to Blackhole Malware". *eWeek*. Retrieved Nov 19, 2014.

[8] Smith, Stuart (June 2007). "Bad Bunny". Retrieved 2008-02-20.

[9] Kissling, Kristian (December 2009). "Malicious Screensaver: Malware on Gnome-Look.org". Retrieved 2009-12-12.

[10] "Discontinuation of Antivirus solutions for Linux systems on June 30th 2016".

[11] Comodo Group (2015). "Comodo Antivirus for Linux". Retrieved 17 October 2012.

[12] "ClamAV". Retrieved 2011-02-22.

[13] "Dr.Web anti-virus for Linux". Dashke. Retrieved 2010-05-25.

[14] FRISK Software International (2011). "F-PROT Antivirus for Linux x86 / BSD x86". Retrieved 13 December 2011.

[15] "Kaspersky Linux Security - Gateway, mail and file server, workstation protection for Linux/FreeBSD". Kaspersky Lab. Retrieved 2009-02-11.

[16] "McAfee VirusScan Enterprise for Linux". McAfee. Retrieved 2012-12-27.

[17] "Panda Security Antivirus Protection for Linux". Panda Security. Retrieved 2009-01-13.

[18] Symantec (January 2009). "System requirements for Symantec AntiVirus for Linux 1.0". Retrieved 2009-03-07.

[19] "Chkrootkit".

[20] "COMODO Antivirus for Linux (CAVL) v1.1.268025.1 is released!". comodo.com. 2013-02-28. Retrieved 2014-06-12.

[21] "ESET File Security - Antivirus Protection for Linux, BSD, and Solaris". Eset. Retrieved 2008-10-26.

[22] "ESET Mail Security - Linux, BSD, and Solaris mail server protection". Eset. Retrieved 2008-10-26.

[23] "ESET NOD32 Antivirus for Linux Gateway Devices". Eset. Retrieved 2008-10-26.

[24] "ESET NOD32 Antivirus 4 for Linux Desktop". Eset. Retrieved 2014-06-12.

[25] "Root Kit Hunter".

[26] "Botnets, a free tool and 6 years of Linux/Rst-B | Naked Security". Nakedsecurity.sophos.com. 2008-02-13. Retrieved 2013-08-11.

[27] "Sophos AV for Linux".

[28] "Second Look".

[29] volatilesystems.com

[30] "Volatility".

[31] Kovalev et al (17 July 2014), Mayhem – a hidden threat for *nix web servers, Virus Bulletin

[32] "Meet Remaiten - a Linux bot on steroids targeting routers and potentially other IoT devices". *We Live Security*. Retrieved 3 April 2016.

[33] "Threat Detail - ESET Virusradar". *virusradar.com*. Retrieved 3 April 2016.

[34] Mohit Kumar (31 March 2016). "Advanced Malware targeting Internet of the Things and Routers". *The Hacker News*. Retrieved 3 April 2016.

[35] Leyden, John (21 November 2012), Evildoers can now turn all sites on a Linux server into silent hell-pits, The Register, retrieved 21 November 2012

[36] Kovalev et al Effusion – a new sophisticated injector for Nginx web servers, Virus Bulletin

[37] rsa.com. "Thieves Reaching for Linux—"Hand of Thief" Trojan Targets Linux #INTH3WILD » Speaking of Security - The RSA Blog and Podcast". Blogs.rsa.com. Retrieved 2013-08-11.

[38] Vaughan, Steven J. "Linux desktop Trojan 'Hand of Thief' steals in". ZDNet. Retrieved 2013-08-11.

[39] Florio, Elia (February 2006). "Linux.Backdoor.Kaiten". Retrieved 2008-03-08.

[40] Florio, Elia (December 2007). "Linux.Backdoor.Rexob". Retrieved 2008-03-08.

[41] Vervloesem, Koen (December 2009). "Linux malware: an incident and some solutions". Retrieved 2010-09-16.

[42] "Backdoor.Linux.Tsunami.gen". Securelist. Retrieved 2014-05-09.

[43] "The 'Penquin' Turla - Securelist". *securelist.com*. Retrieved 10 November 2015.

[44] Joey-Elijah Sneddon. "Yes, This Trojan Infects Linux. No, It's Not The Tuxpocalypse - OMG! Ubuntu!". *OMG! Ubuntu!*. Retrieved 10 November 2015.

[45] Akamai Technologies (29 September 2015). "OR DDoS Botnet Launching 20 Attacks a Day From Compromised Linux Machines, Says Akamai". Retrieved 18 March 2016.

[46] Samuel Gibbs. "HummingBad malware infects 10m Android devices". Retrieved 2016-07-06.

[47] herm1t (August 2008). "Linux.42: Using CRC32B (SSE4.2) instruction in polymorphic decryptor".

[48] Ferrie, Peter (September 2008). "Life, the Universe, and Everything".

[49] herm1t (August 2006). "Infecting ELF-files using function padding for Linux". Archived from the original on 22 January 2012.

[50] Kaspersky Lab (May 2007). "Virus.Linux.Alaeda". Archived from the original on 13 July 2009. Retrieved 2008-03-08.

[51] McAfee (December 2004). "Linux/Binom". Retrieved 2008-03-08.

[52] Rieck, Konrad and Konrad Kretschmer (August 2001). "Brundle Fly 0.0.1 - A Good-Natured Linux ELF Virus". Archived from the original on May 14, 2008. Retrieved 2008-03-08.

[53] de Almeida Lopes, Anthony (July 2007). "Project Bukowski". Retrieved 2008-03-08.

[54] herm1t (February 2008). "Caveat virus".

[55] Ferrie, Peter (July 2009). "Can you spare a seg?".

[56] herm1t (October 2007). "Reverse of a coin: A short note on segment alignment".

[57] Ferrie, Peter (September 2009). "Heads or tails?".

[58] Kaspersky Lab (February 2002). "Virus.Linux.Diesel.962". Archived from the original on October 28, 2007. Retrieved 2008-03-08.

[59] herm1t (October 2007). "Hashin' the elves".

[60] Ferrie, Peter (August 2009). "Making a hash of things".

[61] Kaspersky Lab (April 2001). "Virus.Linux.Kagob.a". Archived from the original on 13 July 2009. Retrieved 2008-03-08.

[62] Kaspersky Lab (n.d.). "Virus.Linux.Kagob.b". Archived from the original on 13 July 2009. Retrieved 2008-03-08.

[63] herm1t (June 2008). "README".

[64] Ferrie, Peter (February 2008). "Crimea river".

[65] "Linux.Encoder.1". *drweb.com*. Retrieved 10 November 2015.

[66] Lucian Constantin (10 November 2015). "First Linux ransomware program cracked, for now". *Computerworld*. Retrieved 10 November 2015.

[67] The Mental Driller (February 2002). "Metamorphism in practice or "How I made MetaPHOR and what I've learnt"". Retrieved 2008-03-08.

[68] Kaspersky Lab (December 2001). "Virus.Linux.Nuxbee.1403". Archived from the original on 2 March 2012. Retrieved 2008-03-08.

[69] herm1t (November 2007). "INT 0x80? No, thank you!".

[70] Ferrie, Peter (September 2009). "Flying solo".

[71] Ferrie, Peter (April 2007). "Linux.Podloso". Retrieved 2008-03-08.

[72] Ferrie, Peter (April 2007). "The iPod virus". Retrieved 2008-03-08.

[73] herm1t (December 2009). "From position-independent to self-relocatable viral code".

[74] Kaspersky Lab (August 2003). "Virus.Linux.Rike.1627". Archived from the original on 2 March 2012. Retrieved 2008-03-08.

[75] Kaspersky Lab (January 2002). "Virus.Linux.RST.a". Retrieved 2008-03-08.

[76] "The ways of viruses in Linux HOW SAFE?" (PDF). Archived from the original (PDF) on 2014-05-17. Retrieved 2009-08-21.

[77] Kaspersky Lab (March 2001). "Virus.Linux.Satyr.a". Archived from the original on 2 March 2012. Retrieved 2008-03-08.

[78] Kaspersky Lab (March 2000). "Virus.Linux.Vit.4096". Archived from the original on November 7, 2007. Retrieved 2008-03-08.

[79] Kaspersky Lab (October 2000). "Virus.Linux.Winter.341". Retrieved 2008-03-08.

[80] Rautiainen, Sami; et al. (March 2001). "F-Secure Virus Descriptions: Lindose". Archived from the original on June 21, 2008. Retrieved 2008-03-08.

[81] "The Wit Virus: A virus built on the ViT ELF virus" (PDF). Retrieved 2008-12-31.

[82] TMZ (January 2015). "Linux.Zariche - ESET Virusradar".

[83] Kaspersky Lab (January 2001). "Virus.Linux.ZipWorm". Archived from the original on 13 July 2009. Retrieved 2008-03-08.

[84] Kaspersky Lab (May 2001). "Net-Worm.Linux.Adm". Retrieved 2008-03-08.

[85] Rautiainen, Sami (April 2001). "F-Secure Virus Descriptions: Adore". Retrieved 2008-03-08.

[86] Smith, Stuart (May 2007). "Perl.Badbunny". Retrieved 2008-03-08.

[87] Kaspersky Lab (May 2001). "Net-Worm.Linux.Cheese". Retrieved 2008-03-08.

[88] Rautiainen, Sami (April 2001). "F-Secure Virus Descriptions: Kork". Retrieved 2008-03-08.

[89] Mohit Kumar (2013-11-30). "Linux worm targeting Routers, Set-top boxes and Security Cameras with PHP-CGI Vulnerability". The Hacker News. Retrieved 2013-12-04.

[90] Joe Casad (3 December 2013). "New Worm Attacks Linux Devices". Linux Magazine. Retrieved 4 December 2013.

[91] McAfee (June 2005). "Linux/Lupper.worm Description". Retrieved 2010-10-10.

[92] Kaspersky Lab (October 2002). "Net-Worm.Linux.Mighty". Retrieved 2008-03-08.

[93] Perriot, Frederic (February 2007). "Linux.Millen.Worm". Retrieved 2008-03-08.

[94] Rautiainen, Sami; et al. (September 2002). "F-Secure Virus Descriptions: Slapper". Retrieved 2008-03-08.

[95] Voss, Joel (December 2007). "SSH Bruteforce Virus by AltSci Concepts". Retrieved 2008-03-13.

5.1.6 External links

- Linuxvirus on the Official Ubuntu Documentation

5.2 Palm OS viruses

While some viruses do exist for Palm OS based devices, very few have ever been designed. Typically, mobile devices are difficult for virus writers to target, since their simplicity provides fewer security holes to target compared to a desktop.

5.2.1 Viruses for Palm OS

5.2.2 References

[1] Distributed as a Trojan in the guise of a crack for shareware program Liberty by Liberty's author Aaron Ardiri. For more information, see http://web.archive.org/web/20011221221530/http://www.inquirer.net:80/infotech/sep2000wk1/info_main.htm. Archived from the original on December 21, 2001. Retrieved July 7, 2007. Missing or empty |title= (help)

[2] Virus that infects handheld devices that run Palm OS. It was discovered on September 22, 2000. For more information, see http://www.symantec.com/security_response/writeup.jsp?docid=2000-121918-4538-99

5.3 Mobile Malware

Mobile malware is malicious software that targets mobile phones or wireless-enabled Personal digital assistants (PDA), by causing the collapse of the system and loss or leakage of confidential information. As wireless phones and PDA networks have become more and more common and have grown in complexity, it has become increasingly difficult to ensure their safety and security against electronic attacks in the form of viruses or other malware.

5.3.1 History

Cell phone malware were initially demonstrated by Brazilian software engineer Marcos Velasco. He created a virus that could be used by anyone in order to educate the public of the threat.[1]

The first known mobile virus, "Timofonica", originated in Spain and was identified by antivirus labs in Russia and Finland in June 2000. "Timofonica" sent SMS messages to GSM mobile phones that read (in Spanish) "Information for you: Telefónica is fooling you." These messages were sent through the Internet SMS gate of the MoviStar mobile operator.[2]

In June 2004, it was discovered that a company called Ojam had engineered an anti-piracy Trojan virus in older versions

of its mobile phone game, *Mosquito.* This virus sent SMS text messages to the company without the user's knowledge. Although this malware was removed from the game's more recent versions, it still exists in older, unlicensed versions, and these may still be distributed on file-sharing networks and free software download web sites.

In July 2004, computer hobbyists released a proof-of-concept mobile virus *Cabir,* that replicates and spreads itself on Bluetooth wireless networks and infects mobile phones running the Symbian OS.[3][4]

In March 2005, it was reported that a computer worm called Commwarrior-A had been infecting Symbian series 60 mobile phones.[5] This specific worm replicated itself through the phone's Multimedia Messaging Service (MMS), sending copies of itself to other phone owners listed in the phone user's address book. Although the worm is not considered harmful, experts agree that it heralded a new age of electronic attacks on mobile phones.

In August 2010, Kaspersky Lab reported a trojan designated Trojan-SMS.AndroidOS.FakePlayer.a.[6] This was the first malicious program classified as a Trojan SMS that affects smartphones running on Google's Android operating system, and which had already infected a number of mobile devices,[7][8] sending SMS messages to premium rate numbers without the owner's knowledge or consent, and accumulating huge bills.

Currently, various antivirus software companies like Trend Micro, AVG, avast!, Comodo, Kaspersky Lab, PSafe, and Softwin are working to adapt their programs to the mobile operating systems that are most at risk. Meanwhile, operating system developers try to curb the spread of infections with quality control checks on software and content offered through their digital application distribution platforms, such as Google Play or Apple's App Store. Recent studies however show that mobile antivirus programs are ineffective due to the rapid evolution of mobile malware.[9]

5.3.2 Taxonomy

Four types of the most common malicious programs are known to affect mobile devices:

- **Expander**:

Expanders target mobile meters for additional phone billing and profit

- **Worm**: The main objective of this stand-alone type of malware is to endlessly reproduce itself and spread to other devices. Worms may also contain harmful and misleading instructions. Mobile worms may be transmitted via text messages SMS or MMS and typically do not require user interaction for execution.

- **Trojan**: Unlike worms, a Trojan horse always requires user interaction to be activated. This kind of virus is usually inserted into seemingly attractive and non-malicious executable files or applications that are downloaded to the device and executed by the user. Once activated, the malware can cause serious damage by infecting and deactivating other applications or the phone itself, rendering it paralyzed after a certain period of time or a certain number of operations. Usurpation data (spyware) synchronizes with calendars, email accounts, notes, and any other source of information before it is sent to a remote server.

- **Spyware**: This malware poses a threat to mobile devices by collecting, using, and spreading a user's personal or sensitive information without the user's consent or knowledge. It is mostly classified into four categories: system monitors, trojans, adware, and tracking cookies.

- **Ghost Push**: This is kind of malware which infects the Android OS by automatically gaining root access, download malicious software, convert to system app and then losing root access which virtually impossible to remove the infection by factory reset unless the firmware is reflashed. The malware hogs all system resources making it unresponsive and drains the battery. The advertisements always appeared anytime either full screen, part of a display, or in status bar. The unnecessary apps is automatically activate and sometimes downloads another malicious software when connected to the internet. It is harder to detect and remove. It steals personal data of the user by the phone.

5.3.3 Notable mobile malicious programs

- **Cabir:** This malware infects mobile phones running on Symbian OS and was first identified in June 2004. When a phone is infected, the message 'Caribe' is displayed on the phone's screen and is displayed every time the phone is turned on. The worm then attempts to spread to other phones in the area using wireless Bluetooth signals, although the recipient has to confirm this manually.

- **Duts:** This parasitic file infector virus is the first known virus for the Pocket PC platform. It attempts to infect all EXE files that are larger than 4096 bytes in the current directory.

- **Skulls:** A trojan horse piece of code that targets mainly Symbian OS. Once downloaded, the virus replaces all phone desktop icons with images of a skull. It also renders all phone applications useless. This malware also tends to mass text messages containing malicious links to all contacts accessible through the device in order to spread the damage. This mass texting can also give rise to high expenses.
- **Commwarrior:** This malware was identified in 2005. It was the first worm to use MMS messages in order to spread to other devices. It can spread through Bluetooth as well. It infects devices running under OS Symbian Series 60. The executable worm file, once launched, hunts for accessible Bluetooth devices and sends the infected files under a random name to various devices.
- **Gingermaster**: A trojan developed for an Android platform that propagates by installing applications that incorporate a hidden malware for installation in the background. It exploits the frailty in the version Gingerbread (2.3) of the operating system to use superuser permissions by privileged escalation. Then it creates a service that steals information from infected terminals (user ID, number SIM, phone number, IMEI, IMSI, screen resolution and local time) by sending it to a remote server through petitions HTTP.
- **DroidKungFu**: A trojan content in Android applications, which when executed, obtains root privileges and installs the file com.google. ssearch.apk, which contains a back door that allows files to be removed, open home pages to be supplied, and 'open web and download and install' application packages. This virus collects and sends to a remote server all available data on the terminal.
- **Ikee**: The first worm known for iOS platforms. It only works on terminals that were previously made a process of jailbreak, and spreads by trying to access other devices using the SSH protocol, first through the subnet that is connected to the device. Then, it repeats the process generating a random range and finally uses some preset ranges corresponding to the IP address of certain telephone companies. Once the computer is infected, the wallpaper is replaced by a photograph of the singer Rick Astley, a reference to the Rickroll phenomenon.
- **Gunpoder :** This worm file infector virus is the first known virus that officially infected the Google Play Store in few countries, including Brazil.[10]
- Shedun: adware serving malware able to root Android devices.

5.3.4 See also

- Computer virus
- File binder
- Individual mobility
- Malware
- Dendroid (Malware)
- Trojan horse (computing)
- Worm (computing)
- Mobile operating system

5.3.5 References

[1] Gralla, Preston (2005). *PC Pest Control: Protect Your Computers from Malicious Internet Invaders*. *Google Books* (1005 Gravenstein Highway North, Sebastopol, CA 95472, US: O'Reilly Media, Inc.). p. 237. ISBN 0-596-00926-7. Retrieved 18 January 2014.

[2] "Mobile Phones Swamped by E-Mail Virus". *ecommercetimes.com*. 7 June 2000.

[3] Malware Goes Mobile, Mikko Hypponen, Scientific American, November 2006, pp. 70-77.

[4] Hantula, Richard (2010). *How Do Cell Phones Work?*. *Google Books* (132 West 31st Street, New York NY 10001, US: Infobase Publishing). p. 27. ISBN 978-1-43812-805-4. Retrieved 18 January 2014.

[5] Computer Virus Timeline (infoplease.com)

[6] Android Virus Security Lab

[7] "First SMS Trojan detected for smartphones running Android". Kaspersky Lab. Retrieved 2010-10-18.

[8] "Information about Smartphone Virus and Prevention tips". MyPhoneFactor.in. Retrieved 2013-01-12.

[9] Suarez-Tangil, Guillermo; Juan E. Tapiador; Pedro Peris-Lopez; Arturo Ribagorda (2014). "Evolution, Detection and Analysis of Malware in Smart Devices" (PDF). *IEEE Communications Surveys & Tutorials*.

[10] Mobile virus hack Google Play user on Brazil

5.3.6 External links

- Mobile Malware Evolution: An Overview
- JavaMites: Next Generation Mobile Security Threats
- primer virus para teléfonos móviles.
- El País: Los virus se enganchan a los móviles
- INTECO (Instituto Internacional de Tecnologias de la Comunicación)
- Informes Panda Security
- Trend Micro Threat Encyclopedia
- Kioskea.net: Antivirus para telefonos móviles

5.4 Macro virus

In computing terminology, a **macro virus** is a virus that is written in a macro language: a programming language which is embedded inside a software application (e.g., word processors and spreadsheet applications). Some applications, such as Microsoft Office, allow macro programs to be embedded in documents such that the macros are run automatically when the document is opened, and this provides a distinct mechanism by which malicious computer instructions can spread. This is one reason it can be dangerous to open unexpected attachments in e-mails. Many antivirus programs can detect macro viruses, however they are still difficult to detect.

5.4.1 Fundamentals

A macro is a series of commands and actions that helps automating some tasks - usually a quite short and simple program. However they are created, they need to be executed by some system which interprets the stored commands. Some macro systems are self-contained programs, but others are built into complex applications (for example word processors) to allow users to repeat sequences of commands easily, or to allow developers to tailor the application to local needs.

5.4.2 Operation

A macro virus can be spread through e-mail attachments, removable media, networks and the Internet, and is notoriously difficult to detect.[1] A common way for a macro virus to infect a computer is by replacing normal macros with a virus. The macro virus replaces regular commands with the same name and runs when the command is selected. These malicious macros may start automatically when a document is opened or closed, without the user's knowledge.[2]

Once a file containing a macro virus is opened, the virus can infect the system. When triggered, it will begin to embed itself in other documents and templates. It may corrupt other parts of the system, depending on what resources a macro in this application can access. When the infected documents are shared with other users and systems, the virus spreads. Macro viruses have been used as a method of installing software on a system without the user's consent, as they can be used to download and install software from the internet through the use of automated key-presses. However, this is uncommon as it is usually not fruitful for the virus coder since the installed software is usually noticed and uninstalled by the user.

Since a macro virus depends on the application rather than the operating system, it can infect a computer running any operating system to which the targeted application has been ported. In particular, since Microsoft Word is available on Macintosh computers, word macro viruses can attack some Macs in addition to Windows platforms.[1]

An example of a macro virus is the Melissa virus which appeared in March of 1999. When a user opens a Microsoft Word document containing the Melissa virus, their computer becomes infected. The virus then sends itself by email to the first 50 people in the person's address book. This made the virus replicate at a fast rate.[3]

Not all macro viruses are detected by antivirus softwares. Caution when opening email attachments and other documents decreases the chance of becoming infected.

5.4.3 See also

- Malware
- Computer virus
- Computer worm
- Trojan horse (computing)
- Ransomware (malware)
- Spyware

5.4.4 References

[1] "Frequently Asked Questions: Word Macro Viruses". Microsoft. Retrieved 2006-06-18.

[2] "Information Bulletin: Macro Virus Update". Computer Incident Advisory Capability. Retrieved 2006-06-18.

[3] "How Computer Viruses Work". How Stuff Works inc. Retrieved 2006-06-18.

5.4.5 Further reading

- Microsoft Corporation. (2006). Introduction to Security. Retrieved June 18, 2006
- The Trustees of Indiana University. (2006). What are computer Viruses, Worms, and Trojan Horses? Retrieved June 18, 2006
- Macro Viruses from Security News & Information

5.5 ANTI (computer virus)

ANTI is an obsolete computer virus affecting Apple Macintosh computers running early System versions up to Mac OS 8.1. It is particularly notable for being the first Macintosh virus not to create additional resources within infected files; instead, it patches any existing resource that has an ID of 1 and begins with a JSR instruction (generally the Main resource in a given application).[1]

The virus carries no payload, and thus can exist and spread indefinitely without being noticed until an anti-virus program is run. It can not spread if MultiFinder is running. Based on the virus's design, Danny Schwendener of ETH Zurich hypothesised that it had been intended to form part of a copy protection scheme,[1] but the author of the virus is unknown.

5.5.1 References

[1] List of known Macintosh viruses

5.6 INIT 1984

INIT 1984 is a computer virus that was set up to trigger on Macintosh computers running the classic Mac OS on any given Friday the 13th. The virus was first discovered and isolated in March, 1992. It functions by infecting startup (INIT) files, then modifying or deleting those files upon startup on Friday the 13th. The virus has a low threat assessment.

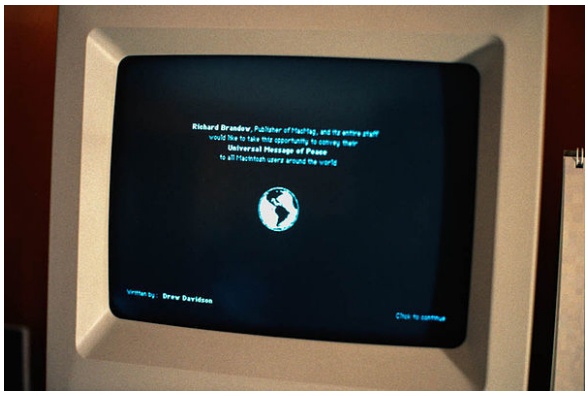
5.6.1 References

- "INIT 1984". *Technical Details*. Symantec Corporation. February 13, 2007. Retrieved 12 September 2013.

5.6.2 Further reading

- Ray, John; Ray, William C. (May 24, 2003). *Maximum Mac OS X Security*. Sams. ISBN 978-0672323812. Retrieved Sep 12, 2013.

5.7 MacMag



This is the MacMag virus 'Universal Peace' message, as displayed on a Mac SE in March of 1988.

The **MacMag** virus, also known by various other names, was a computer virus introduced in 1988 by Richard Brandow, who at the time was editor and publisher of MacMag computer magazine in Montréal.[1][2][3]

5.7.1 Operation of the virus

The virus infected Macintosh computers, and the intention was that on 2 March 1988 all infected computers would show the message "RICHARD BRANDOW, publisher of MacMag, and its entire staff would like to take this opportunity to convey their UNIVERSAL MESSAGE OF PEACE to all Macintosh users around the world", and the virus would then delete itself. According to the virus itself, it was written by Drew Davidson. The virus was a boot sector virus, which was spread in the form of a HyperCard stack called "New Apple Products," which contained very poor pictures of the then-new Apple scanner. It copied a resource into the System folder on a Mac, as an "initial" program, which would run automatically every time the system started up. The program then copied itself onto any bootable disk which was opened.[4]

5.7.2 Damage caused

Brandow intended the virus to be benign, giving a friendly message and causing no harm. However a bug in the

virus caused infected Mac II computers to undergo system crashes before this date. Another bug, which affected very few users, caused files other than the original virus to be deleted during the termination stage. It also caused a great deal of anxiety among users who found that their computers were infected with an unwanted program the nature of which was unknown. The virus infected Aldus software's FreeHand, and Aldus had to recall thousands of copies of FreeHand, leading them to threaten legal action.

5.7.3 References

- Norstad, John. The Viruses. Disinfectant 3.7.1 ©1988-1997 Northwestern University.

[1] Bocij, Paul *The Dark Side of the Internet: Protecting Yourself and Your Family from Online Criminals*, Praeger Publishers Inc 2006, ISBN 0-275-98575-X, ISBN 978-0-275-98575-2

[2] "Chapter 8 MacMag". Cknow.com. 2009-05-09. Retrieved 2010-06-14.

[3] "The Risks Digest Volume 6: Issue 44". Catless.ncl.ac.uk. Retrieved 2010-06-14.

[4] "Virus History". Thermopyle.tripod.com. Retrieved 2010-06-14.

5.8 MDEF

MDEF was a computer virus affecting Macintosh machines. There are four known strains. The first, MDEF A (aka Garfield), was discovered in May 1990. Strains B (aka Top Cat), C, and D were discovered in August 1990, October 1990, and January 1991, respectively.

MDEF A, B, and C can infect application files and system files, and sometimes document files as well. The D strain will infect only applications. None of the viruses were designed to do damage, but they often do. MDEF D can sometimes damage applications beyond repair.

Quick action by computer security personnel and the New York State Police resulted in identification of the author, a juvenile. This was the same person responsible for writing the CDEF virus.

5.8.1 References

- Norstad, John. The Viruses. Disinfectant 3.7.1 ©1988-1997 Northwestern University.

5.9 nVIR

nVIR is an obsolete computer virus which can replicate on Macintosh computers running any System version from 4.1 to OS 8. The source code to the original nVIR has been made widely available, and so numerous variants have arisen. Each variant causes somewhat different symptoms, such as: application crashes, printing errors on laser printers, slow system response time, or unpredictable system crashes. nVIR spreads through any nVIR-infected program, but due to the long period of time nVIR lies basically dormant in a host system, nVIR generally finds its way into system backups and is not detected until the first overt symptoms appear. For example, if a disk used in an infected Macintosh is removed and inserted in a second Macintosh, the other machine will become infected if any application on that disk is executed in the second machine. Further, any method used to transfer programs between Macintoshes will spread nVIR, including file transfer over a network. However, nVIR cannot spread via a print network's hardware.

nVIR carries an additional code resource, CODE 256 (though some variants carry CODE 255), and patches the jump table to point to it. The original application's entry point is saved in the nVIR 2 resource. nVIR introduces to the System file the INIT 32 resource which is executed at startup, at which time nVIR patches the TEInit trap. Any application subsequently calling this trap will be infected. The nVIR 3 (or nVIR 5) resource is a copy of INIT 32. An nVIR 10 resource in the System file will prevent nVIR infection. If an application calls OpenResFile prior to TEInit, that application will be damaged.

nVIR 0 resource holds a counter that is set to 1000 on the first infection of the system. Each reboot decrements the counter by 1. Each application launch decrements it by 2. When the counter reaches 0, nVIR will beep 1 out of 8 reboots and 1 of 4 infected application launches. If MacinTalk is installed in the machine's System folder, the machine may occasionally say "Don't Panic". Otherwise, it may beep unexpectedly.

nVIR has been known to 'hybridize' with different variants of nVIR on the same machine.

5.9.1 External links

- A Vaccine for the 'nVIR' Virus, by Mike Scanlin, MacTech
- Mac OS/nVIR virus, by McAfee
- nVIR A, by Virus-Test-Center, University of Hamburg

- nVIR B, by Virus-Test-Center, University of Hamburg
- nVIR C, by Virus-Test-Center, University of Hamburg
- nVIR B countermeasures, UC Berkeley

5.10 Scores (computer virus)

Scores was a computer virus affecting Macintosh machines. It was first discovered in Spring 1988. It was written by a disgruntled programmer and specifically attacks two applications that were under development at his former company. These programs were never released to the public.[1][2][3]

5.10.1 Overview

Scores infects the System, Notepad, and Scrapbook files under System 6 and System 7. There is a simple way to identify infection. Normal Notepad and Scrapbook icons will have specific icons under System 7, or little Macintosh icons under System 6. If the icons are blank document icons, it is a good indication the system is infected.

Scores begins to spread to other applications two days after infection. The Finder and DA Handler often become infected as well.

Scores was not designed to do anything besides spread itself and attack the two specific applications. However, there is a serious conflict between the virus and System 6.0.4 or above, where Apple began using resources of the same type that Scores uses. In these cases the system files will be damaged.

The alleged author of the virus was questioned by the Federal Bureau of Investigation (FBI) soon after the virus was discovered. There were no federal laws with which to charge the author, so they remain free to this day. This loophole resulted in the "Computer Virus Eradication Act of 1988".

5.10.2 References

- Norstad, John. The Viruses. Disinfectant 3.7.1, 1988-1997 Northwestern University
- John Norstad. INFO-MAC Digest (through Virus-L), Volume 6, Issue 40, The Scores Virus 1988-04-18
- Symantec Antivirus, Scores
- The New York Times, Technology, Sabotage Aimed at Computer Company Destroys Government Data. 1988-07-04
- Keith Petersen. VIRUS-L Virus Discussion List,FBI to investigate rogue computer program at NASA. 1988-07-06
- Joshua Yeidel. VIRUS-L Virus Discussion List, SCORES Virus (Mac) Sighted At Washington State U. 1988-11-22
- Joe Simpson, Virus-L Digest, A description of computer virus epidemic at Miami U. 1988-04-28

[1] "Virus:MacOS/Scores.A". *www.microsoft.com*. Retrieved 2016-03-23.

[2] "Scores - The Virus Encyclopedia". *virus.wikidot.com*. Retrieved 2016-03-23.

[3] "Scores Virus". *agn-www.informatik.uni-hamburg.de*. Retrieved 2016-03-23.

5.11 SevenDust (computer virus)

SevenDust was a computer virus that infected computers running certain versions of Mac OS. It was discovered in 1998. It was originally referred to as 666, by McAfee.

5.11.1 See also

- Computer virus
- Comparison of computer viruses

5.11.2 External links

- MacOS.Sevendust, by Symantec
- MacOS/SevenDust, by McAfee
- 666, by McAfee

5.12 KeyRaider

KeyRaider is computer malware that affects Apple iOS devices, specifically iPhones, and allows criminals to steal users' login and password information, as well as to lock the devices and demand a ransom to unlock them. It was discovered by researchers from Palo Alto Networks and WeiPhone in August of 2015, and is believed to have led

to more than 225,000 people having their login and password information being stolen, making it, according to cybersecurity columnist, Joseph Steinberg, "one of the most damaging pieces of malware ever discovered in the Apple universe."[1] The malware was originally found on a Chinese website,[2] but has spread to 18 countries including the United States.[3] KeyRaider affects only iPhones that have been jailbroken.[1]

5.12.1 References

[1] Joseph Steinberg (August 31, 2015). "Massive iPhone User Data Breach: What You Need to Know". *Inc.* Retrieved September 2, 2015.

[2] "Chinese iPhone users hit by 'KeyRaider' malware". *BBC*. September 1, 2015. Retrieved September 2, 2015.

[3] David Goldman (September 1, 2015). "More than 225,000 Apple iPhone Account Hacked". *CNN*. Retrieved September 2, 2015.

5.13 Wirelurker

WireLurker is a family of malware targeting both Mac OS and iOS systems.[1] The malware was designed to target users in China that use Apple mobile and desktop devices.[2] The malware was suspected of infecting thousands of Chinese mobile devices.[3] The security firm Palo Alto Networks is credited with uncovering the malware.[1]

5.13.1 How it works

WireLurker monitors any iOS device connected via USB with an infected OS X computer and installs downloaded third-party applications or automatically generated malicious applications onto the device. WireLurker can infect a device regardless of whether it is jailbroken or not. WireLurker is a complex form of malware that utilizes techniques such as file hiding, code obfuscation and encryption. WireLurker is capable of stealing a variety of information from the mobile devices it infects and regularly requests updates from the attackers command and control server.[1]

5.13.2 Arrests

Three individuals in China were arrested for the suspicion of creating and distributing the WireLurker malware. The suspects, identified only by their surnames as Wang, Lee and Chen were taken into custody on Thursday November 13, 2014. Chinese authorities believe the suspects created the malware for financial gains.[4]

5.13.3 Protection

Several steps can be taken in order to protect yourself from WireLurker and other malware.

- Do not install software or applications from unknown or unreliable sources.
- Make sure that System Preferences on you Mac are set to: 'Allow apps downloaded from: Mac App Store and identified developers'.
- Keep your security software up to date on your Mac or desktop.
- Keep your iOS software up to date on your mobile device.
- Do not connect your mobile device to unknown computers.[5]

5.13.4 References

[1] Xiao, Claud. "WireLurker: A New Era in OS X and iOS Malware". *http://researchcenter.paloaltonetworks.com*. External link in |website= (help)

[2] Perlroth, Nicole. "Malicious Software Campaign Targets Apple Users in China". *http://bits.blogs.nytimes.com*. External link in |website= (help)

[3] Clover, Juli. "Chinese Authorities Shut Down WireLurker Distribution Site, Arrest Suspects Involved". *http://www.macrumors.com*. External link in |website= (help)

[4] Kovacs, Eduard. "Alleged Creators of WireLurker Malware Arrested in China". *http://www.securityweek.com/*. External link in |website= (help)

[5] "Norton - WireLurker". *https://community.norton.com*. External link in |website= (help)

5.13.5 External links

- Palo Alto Networks Research Center

5.14 XcodeGhost

XcodeGhost (and variant XcodeGhost S) are a modified versions of Apple's Xcode development environment that are considered malware.[1] The software first gained widespread attention in September 2015, when a number of apps originating from China harbored the malicious code.[2] It was thought to be the "first large-scale attack on Apple's App Store," according to the BBC. The problems were first

identified by researchers at Alibaba, a leading e-commerce firm in China.[3] Over 4000 apps are infected, according to FireEye, far more than the 25 initially acknowledged by Apple,[4] including apps from authors outside China.

Security firm Palo Alto Networks surmised that because network speeds were slower in China, developers in the country looked for local copies for Apple Xcode development environment, and encountered altered versions that had been posted on domestic web sites. This opened the door for the malware to be inserted into high profile apps used on iOS devices.[5][6]

Even two months after the initial reports, security firm FireEye reported that hundreds of enterprises were still using infected apps and that XcodeGhost remained "a persistent security risk".[7][8] The firm also identified a new variant of the malware and dubbed it XcodeGhost S, and among the apps that were infected were the popular messaging app WeChat and a Netease app Music 163.[9]

5.14.1 Discovery

On September 16, 2015, a Chinese iOS developer mentioned[10] on the social network Sina Weibo that a malware in Xcode injects third party code into apps compiled with it.

Alibaba researchers then published[11] detailed information on the malware and called it XcodeGhost.

On September 17, 2015, Palo Alto Networks published several reports on the malware.[12][13][14][15]

5.14.2 Operation

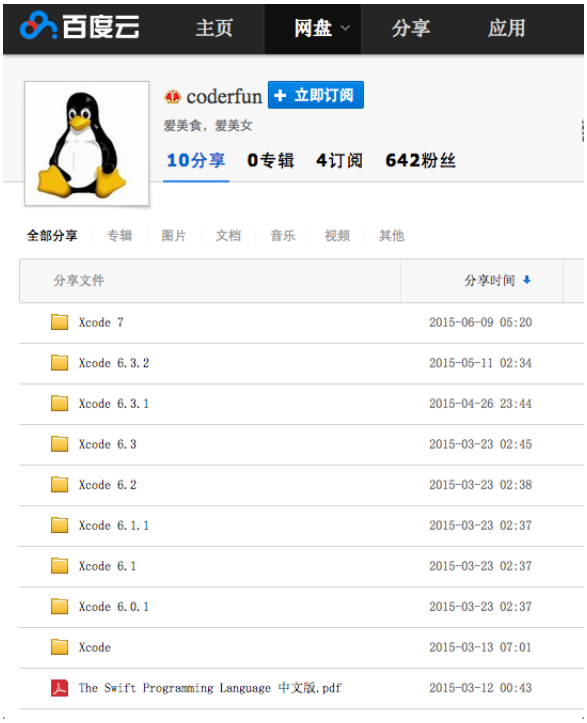
Propagation

Because of the slow download speed from Apple Servers, Chinese iOS developers used to download Xcode from third party website such as Baidu Yun, a cloud storage service hosted by Baidu, or get copies from co-workers. Attackers took advantage of this situation by distributing compromised version on such file hosting websites.[16]

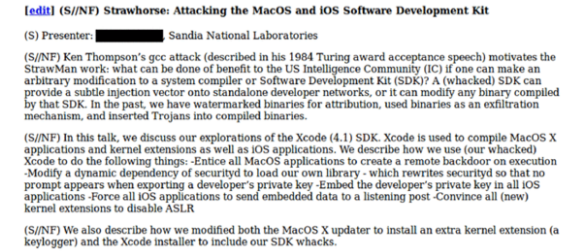
Palo Alto Networks suspects that the malware was available in March 2015.[15]

Attack vector

Origins The attacker used a compiler backdoor attack. The novelty of this attack is the modification of the Xcode compiler. However, according to Edward Snowden leaked documents, CIA security researcher from Sandia National Laboratories claimed that they "they had created a modified



Different versions of infected Xcode on Baidu Yun.



[edit] (S//NF) Strawhorse: Attacking the MacOS and iOS Software Development Kit

(S) Presenter: [redacted], Sandia National Laboratories

(S//NF) Ken Thompson's gcc attack (described in his 1984 Turing award acceptance speech) motivates the StrawMan work: what can be done of benefit to the US Intelligence Community (IC) if one can make an arbitrary modification to a system compiler or Software Development Kit (SDK)? A (whacked) SDK can provide a subtle injection vector onto standalone developer networks, or it can modify any binary compiled by that SDK. In the past, we have watermarked binaries for attribution, used binaries as an exfiltration mechanism, and inserted Trojans into compiled binaries.

(S//NF) In this talk, we discuss our explorations of the Xcode (4.1) SDK. Xcode is used to compile MacOS X applications and kernel extensions as well as iOS applications. We describe how we use (our whacked) Xcode to do the following things: -Entice all MacOS applications to create a remote backdoor on execution -Modify a dynamic dependency of securityd to load our own library - which rewrites securityd so that no prompt appears when exporting a developer's private key -Embed the developer's private key in all iOS applications -Force all iOS applications to send embedded data to a listening post -Convince all (new) kernel extensions to disable ASLR

(S//NF) We also describe how we modified both the MacOS X updater to install an extra kernel extension (a keylogger) and the Xcode installer to include our SDK whacks.

Leaked document from Edward Snowden. "Strawhorse: Attacking the MacOS and iOS Software Development Kit".

version of Apple's proprietary software development tool, Xcode, which could sneak surveillance backdoors into any apps or programs created using the tool.".[17]

Modified files Known version of XcodeGhost adds extra files[12] to the original Xcode application :

- Core service framework on iOS, iOS simulator and OS X platforms
- IDEBundleInjection framework added on IOS, IOS simulator and OS X platforms

XcodeGhost also modified the linker to link the malicious files[15] into the compiled app. This step is reported on the compiling log but not on the Xcode IDE.

Both iOS and OS X apps are vulnerable to XcodeGhost.

Deployment XcodeGhost compromised the CoreServices layer which contains highly used features and frameworks used by the app.[18] When a developer compiled his application with a compromised version of Xcode, the malicious CoreServices are automatically integrated into the app without the developer knowledge.

Then the malicious files will add extra code in UIWindow class and UIDevice class. The UIWindow class is "an object that manages and coordinates the views an app displays on a device screen".[19]

The UIDevice class provides a singleton instance representing the current device. From this instance the attacker can obtain information about the device such as assigned name, device model, and operating-system name and version.[20]

Behavior on infected devices

Remote control security risks XcodeGhost can be remotely controlled via commands sent by an attacker from a Command and control server through HTTP. This data is encrypted using the DES algorithm in ECB mode. Not only is this encryption mode known to be weak, the encryption keys can also be found using reverse engineering. An attacker could perform a man in the middle attack and transmit fake HTTP traffic to the device (to open a dialog box or open specific app for example).

Stealing user device information When the infected app is launched, either by using an iPhone or the simulator inside Xcode, XcodeGhost will automatically collect device informations such as :

- Current time
- Current infected app's name
- The app's bundle identifier
- Current device's name and type
- Current system's language and country
- Current device's UUID
- Network type

Then the malware will encrypt those data and send it to a command and control server. The server differs from version to version of XcodeGhost; Palo Alto Networks was able to find three server URLs :

- http://init.crash-analytics.com
- http://init.icloud-diagnostics.com
- http://init.icloud-analysis.com

The last domain was also used in the iOS malware KeyRaider.[12]

Read and write from clipboard XcodeGhost is also able, each time an infected app is launched, to store the data written in the iOS clipboard. The malware is also capable to modify this data. This can be particularly dangerous if the user uses a password management app.

Hijack opening specific URLs XcodeGhost is also able to open specific URLs when the infected app is launched. Since Apple iOS and OS X work with Inter-App Communication URL mechanism[21] (e.g. : 'whatsapp://', 'Facebook://', 'iTunes://'?), the attacker can open any apps installed on the compromised phone or computer (in case of an infected OS X app). Such mechanism could be harmful with password management apps or even on phishing website.

Prompting alert dialog In its current known version XcodeGhost cannot prompt alert dialog on the user device.[15] However, since it only requires to minor changes.

By using a UIAlertView class with the UIAlertViewStyleLoginAndPasswordInput property, the infected app can display a fake alert dialog box that looks like a normal Apple ID user credential check and send the input to the Command and control server.

Infected apps

Among all the Chinese apps, IMs app, banking apps, mobile carrier's app, maps, stock trading apps, SNS apps and games were infected. Popular apps used all over the world were also infected such as WeChat, a popular instant messaging app, CamScanner, an app to scan document using the smartphone camera or WinZip.

Pangu Team claimed that they counted 3,418 infected apps.[22]

Fox-it, a Netherland-based security company reports that they found thousand of malicious traffic outside China.[23]

5.14.3 Removal

Neutralizing command and control servers and compromised versions of Xcode

Since the article of Alibaba and Palo Alto Networks, Amazon took down all the servers that were used by XcodeGhost. Baidu also removed all malicious Xcode installers from its cloud storage service.

Removing malicious apps from AppStore

On September 18, 2015 Apple admitted the existence of the malware and began asking all developers with compromised apps to compile their apps with a clean version of Xcode before submitting them for review again.

Pangu Team released a tool[24] to detect infected apps on a device, but like other antivirus apps it won't run on a device that hasn't been jailbroken. Apple does not allow antivirus apps into the iOS App Store.[25]

Checking Xcode version

Apple advises Xcode developers to verify[26][27] their version of Xcode and to always have GateKeeper activated on their machine.

5.14.4 References

[1] Dan Goodin (September 21, 2015). "Apple scrambles after 40 malicious "XcodeGhost" apps haunt App Store". *Ars Technica*. Retrieved 2015-11-05.

[2] Joe Rossignol (September 20, 2015). "What You Need to Know About iOS Malware XcodeGhost". *macrumors.com*. Retrieved 2015-11-05.

[3] "Apple's App Store infected with XcodeGhost malware in China". *BBC News*. 21 September 2015. Retrieved 2015-11-05.

[4] https://www.fireeye.com/blog/executive-perspective/2015/09/protecting_our_custo.html

[5] Byford, Sam (September 20, 2015). "Apple removes malware-infected App Store apps after major security breach". *The Verge*. Retrieved 2015-11-05.

[6] James Temperton (September 21, 2015). "Apple App Store hack: XcodeGhost attack strikes China (Wired UK)". *Wired UK*. Retrieved 2015-11-05.

[7] Kirk, Jeremy (November 4, 2015). "Many US enterprises still running XcodeGhost-infected Apple apps, FireEye says". *InfoWorld*. Retrieved 2015-11-05.

[8] Ben Lovejoy (November 4, 2015). "A modified version of XcodeGhost remains a threat as compromised apps found in 210 enterprises". *9to5Mac*. Retrieved 2015-11-05.

[9] Yong Kang; Zhaofeng Chen; Raymond Wei (3 November 2015). "XcodeGhost S: A New Breed Hits the US". *FireEye*. Retrieved 2015-11-05. XcodeGhost S: A New Breed Hits the US

[10] "First mention of XcodeGhost on SinaWeibo". *Sina Weibo*. September 17, 2015. Retrieved 2015-11-11.

[11] "Xcode□□□□□□ – XcodeGhost□□□□-□□□□-□□□□-□□□□□". *jaq.alibaba.com*. Retrieved 2015-11-11.

[12] Claud Xiao (September 17, 2015). "Novel Malware XcodeGhost Modifies Xcode, Infects Apple iOS Apps and Hits App Store - Palo Alto Networks Blog". *Palo Alto Networks Blog*. Retrieved 2015-11-11.

[13] Claud Xiao (September 18, 2015). "Malware XcodeGhost Infects 39 iOS Apps, Including WeChat, Affecting Hundreds of Millions of Users - Palo Alto Networks Blog". *Palo Alto Networks Blog*. Retrieved 2015-11-11.

[14] Claud Xiao (September 18, 2015). "Update: XcodeGhost Attacker Can Phish Passwords and Open URLs through Infected Apps - Palo Alto Networks Blog". *Palo Alto Networks Blog*. Retrieved 2015-11-11.

[15] Claud Xiao (September 21, 2015). "More Details on the XcodeGhost Malware and Affected iOS Apps - Palo Alto Networks Blog". *Palo Alto Networks Blog*. Retrieved 2015-11-11.

[16] Thomas Fox-Brewster (September 18, 2015). "Hackers Sneak Malware Into Apple App Store 'To Steal iCloud Passwords'". *Forbes*. Retrieved 2015-11-11.

[17] Jeremy Scahill; Josh Begley (March 10, 2015). "The CIA Campaign to Steal Apple's Secrets". *The Intercept*. Retrieved 2015-11-11.

[18] "Core Services Layer". *developer.apple.com*. Retrieved 2015-11-11.

[19] "UIWindow Class Reference". *developer.apple.com*. Retrieved 2015-11-11.

[20] "UIDevice Class Reference". *developer.apple.com*. Retrieved 2015-11-11.

[21] "Inter-App Communication". *developer.apple.com*. Retrieved 2015-11-11.

[22] "Pangu Team on Weibo". September 21, 2015. Retrieved 2015-11-11.

[23] "Combined research Fox-IT and Palo Alto Networks revealed popular apps infected with malware". *Fox-it*. September 18, 2015. Retrieved 2015-11-11.

[24] "Xcode□□□□,XcodeGhost□□□□ - □□□□". *x.pangu.io*. Retrieved 2015-11-11.

[25] http://www.macworld.co.uk/feature/iosapps/is-ipad-iphone-ios-safe-xcodeghost-what-security-software-need-3453938/#antivirus

[26] "□□ XcodeGhost □□□□□□". *Apple*. Archived from the original on November 14, 2015. Retrieved June 17, 2016.

[27] "Validating Your Version of Xcode - News and Updates - Apple Developer". *developer.apple.com*. Retrieved 2015-11-11.

5.15 Brain Test

Brain Test was a piece of malware masquerading as an Android app that tested the users IQ.[1][2] Brain Test was discovered by security firm Check Point and was available in the Google Play app store until 15 September 2015.[1] Check Point described Brain Test as "A new level of sophistication in malware".[1]

Brain Test was uploaded on two separate occasions (com.zmhitlte.brain and com.mile.brain), starting in August 2015, both times Google's "Bouncer" failed to detect the malware. After the first removal on 24 August 2015 the software was reintroduced using an obfuscation technique. Tim Erin of Tripwire said the "Bypassing the vetting processes of Apple and Google is the keystone in a mobile malware campaign."

The malware turned out to include a rootkit, the revelation being described as "more cunning than first thought".[3]

The malware is thought to have been written by Chinese actor, according to Shaulov of Check Point, based on the use of a packing/obfuscation tool from Baidu. Eleven Paths, a Telefonica-owned company, found links to may other pieces of malware, based on the id used to access Umeng, Internet domains accessed by the apps and shared jpg and png images.[4]

It appears the app was first detected on a Nexus 5 using Check Point's Mobile Threat Prevention System. The fact that the system was unable to remove the malware alerted the software company's researchers that it was an unusual threat.

According to Check Point, it may be necessary to re-flash the ROM on a device if Brain Test has successfully installed a reinstaller in the system directory.

5.15.1 Features

The malware was uploaded in two forms. The packing feature was only present in the second.

- Evades detection by Google Bouncer by avoiding malicious behavior on Google servers with IP addresses 209.85.128.0–209.85.255.255, 216.58.192.0–216.58.223.255, 173.194.0.0–173.194.255.255, or 74.125.0.0–74.125.255.255, or domain names "google", "android" or "1e100".
- Root exploits. Four exploits to gain root access to the system were included, to account for variations in the kernel and drivers of different manufacturers and Android versions,[5] which provide alternative paths to root.
- External payloads - via command and control system. The system used up to five external servers to provide variable payload, believed to be primarily advertising related.
- Packing and time delay. The main downloaded malware portion sits in a sound file, the bootstrap code unpacks this after a time delay.
- Dual install and re-install. Two copies of the malware are installed. If one is removed the other re-installs it.

5.15.2 See also

- Shedun
- Xcode Ghost

5.15.3 References

[1] Polkovnichenko, Andrey; Boxiner, Alon (21 September 2015). "BrainTest – A New Level of Sophistication in Mobile Malware". Retrieved 27 November 2015.

[2] Graham Cluley (23 September 2015). "Malware hits the Google Play Android app store again (and again)".

[3] Cett, Hans (2 November 2015). "Brain Test malware more cunning than 1st thought". *GoMo News*. Retrieved 27 November 2015.

[4] Detailed coverage at Forbes Chinese Cybercriminals Breached Google Play To Infect 'Up To 1 Million' Androids

[5] Kerner, Sean Michael (21 September 2015). "Malicious Brain Test App Thwarts Google Play Android Security". *eweek.com*. Retrieved 27 November 2015.

5.15.4 External links

- Detailed coverage at Forbes
- Video from Graham Cluley on Brain Test
- Washington Post.

5.16 Dendroid (Malware)

Dendroid is a Malware that affects Android OS and targets the mobile platform.[1]

It was first discovered in early of 2014 by Symantec and appeared on the underground for sale for $300.[2] Some things were noted in Dendroid, such as being able to hide from emulators at the time.[3] When first discovered in 2014 it was one of the most sophisticated Android RATs known during that time[4] It was one of the first Trojan applications to get past Google's Bouncer and caused researchers to warn about it being easier to create Android malware due to it[5] It also seemed to follow in the footsteps of Zeus and SpyEye by having simple to use Command and control panels[6] The code appeared to be leaked somewhere around 2014[7] It was noted that an apk binder was included in the leak which provided a simple way to bind dendroid to legitimate applications.

It's capable of

- deleting call logs
- Opening web pages
- Dialing any number
- Recording calls
- SMS intercepting
- Upload images, video
- Opening an application
- Able to perform DoS attack
- Can change the command and control server[8]

5.16.1 See also

- Botnet
- Command and control (malware)
- Denial-of-service attack
- File binder
- Shedun
- Trojan horse
- Zombie (computer science)
- Zeus (malware)

5.16.2 References

[1] http://www.symantec.com/connect/blogs/android-rats-branch-out-dendroid

[2] http://securityaffairs.co/wordpress/22848/cyber-crime/dendroid-new-android-rat.html

[3] https://www.bluecoat.com/security-blog/2014-05-27/dendroid-under-hood-%E2%80%93-look-inside-android-rat-kit

[4] https://www.helpnetsecurity.com/2014/03/07/dendroid-spying-rat-malware-found-on-google-play/

[5] http://www.pcworld.com/article/2105500/new-crimeware-tool-dendroid-makes-it-easier-to-create-android-malware-res
html

[6] https://www.mysonicwall.com/sonicalert/searchresults.aspx?ev=article&id=718

[7] http://www.securityweek.com/source-code-android-rat-dendroid-leaked-online

[8] http://thehackernews.com/2014/03/symantec-discovered-android-malware.html

5.17 DroidKungFu

DroidKungFu is a Malware that affects Android OS and it targets the mobile platform in china. First piece of malware found in the Android Market is in March 2011.[1]

5.17.1 History

It was discovered by two US based researcher named Yajin Zhou and Xuxian Jiang from the North Carolina State University.[2] It targets the Android 2.2 platform and allows the hackers to access and control devices. DroidKungFu malware can still collect some user data through the back door.[3]

5.17.2 Process of DroidKungFu Malware

DroidkungFu malware encrypts two different root exploits, exploid and regeagainsthecage,to break android security.[4] Once executed, it decrypts the exploits and communicate with a remote server without user knowledge.[5]

Function:

- Silent mobile device rooting
- Unlocks all system files and functions

- Install itself without any user interaction

It collect data such as

- IMEI number
- Phone model
- Android OS version
- Network operator
- Network type
- Information stored in the Phone & SD Card memory [6]

5.17.3 See also

- Botnet
- Command and control (malware)
- Denial-of-service attack
- File binder
- Shedun
- Trojan horse
- Zombie (computer science)
- Zeus (malware)

5.17.4 References

[1] "DroidkungFu Malware targets china". Retrieved 2011-07-23.

[2] "Researcher who involved in finding DroidKungFu Malware". Retrieved 2011-06-20.

[3] "Droidkungfu malware targets android users in China". Retrieved 2011-07-06.

[4] "DroidKungFu - Complete overview". Retrieved 2011-07-06.

[5] "Android malware discovery(DroidKungFu)". Retrieved 2011-07-06.

[6] "Droidkungfu malware function detailed". Retrieved 2011-07-06.

5.18 Shedun

Shedun is a family of malware software (also known as Kemoge, Shiftybug and Shuanet[1][2][3]) targeting the Android (operating system) first identified in late 2015 by mobile security company Lookout (company), affecting roughly 20,000[4] popular Android applications.[3][5][6][7][8][9]

Avira Protection Labs stated that Shedun family malware is detected to cause approximately 1500-2000 infections per day.[10] All three variants of the virus are known to share roughly ~80% of the same source code.[11][12]

The malware's primary attack vector is repackaging legitimate Android applications (e.g. Facebook, Twitter, WhatsApp, Candy Crush, Google Now, Snapchat[13])[4][14][15] with adware included, the app which remains functional is then released to a third party app store;[16] once downloaded, the application generates revenue by serving ads (estimated to amount to $2 US per installation[15]), most users cannot get rid of the virus without getting a new device, as the only other way to get rid of the malware is to root affected devices and re-flash a custom ROM.[5][17][18]

In addition, Shedun-type malware has been detected preinstalled on 26 different types[19] of Chinese Android-based hardware such as Smartphones and Tablet computers.[20][21][22][23] [24][25][26][27][28][29][30][31][32][33][34]

Shedun-family malware is known for auto-rooting the Android OS [14][35] using well-known exploits like Exynos-Abuse, Memexploit und Framaroot [36] (causing a potential privilege escalation[15][37][38])[39] and for serving trojanized adware and install themselves within the system partition of the operating system, so that not even a factory reset can remove the malware from infected devices.[40][41]

Shedun malware is known for targeting the Android Accessibility Service,[2][40][42][43][44][45][46] as well as for downloading and installing arbitrary applications[47] (usually adware) without permission,[3] it is classified as "aggressive adware" for installing potentially unwanted program [48][49][50] applications and serving ads.[51]

As of April 2016, Shedun malware is, by most security researchers, considered to be next to impossible to remove entirely.[52][53][54][55][56][57]

Avira Security researcher Pavel Ponomariov, specialized in Android malware detection tools, mobile threats detection and mobile malware detection automation research,[58] has published an in-depth analysis of the computer virus.[10]

5.18.1 See also

- Brain Test

- Dendroid (Malware)
- Computer virus
- File binder
- Individual mobility
- Malware
- Trojan horse (computing)
- Worm (computing)
- Mobile operating system

5.18.2 References

[1] https://hackbails.wordpress.com/2015/11/05/trojanized-adware-already-infected-more-than-20000-android-apps/

[2] "Android Adware Abuses Accessibility Service to Install Apps". *SecurityWeek.com*. Retrieved 2016-04-20.

[3] Manish Singh. "New Android Adware Can Download, Install Apps Without Permission: Report". *NDTV Gadgets360.com*.

[4] "Three new malware strains infect 20k apps, impossible to wipe, only affect Android". *AppleInsider Forums*.

[5] "Hackers reveal Android trojan malware that is IMPOSSIBLE to remove". *Mail Online*. 5 November 2015.

[6] http://appleinsider.com/articles/15/11/05/three-new-malware-strains-infect-20k-apps-impossible-to-wipe-only-affect-android

[7] "Android Malware On The Loose: Shuanet, ShiftyBug And Shedun Signatures Found On 20,000 Apps Outside Google Play Store". *Droid Report*.

[8] "Shedun Trojan goes solo". *Darkmatters*.

[9] http://lavasoft.com/mylavasoft/company/blog/popular-mobile-apps-repackaged-with-trojans

[10] "Shedun: adware/malware family threatening your Android device". *Avira Blog*.

[11] "Neue Welle von Android-Malware lässt sich kaum mehr entfernen". *Elektronikpraxis.vogel.de*. Retrieved 2016-04-20.

[12] PMK Presse, Messe & Kongresse Verlags GmbH. "Gemeinsamkeiten: Shuanet, Shedun & ShiftyBug". *Itseccity.de*. Retrieved 2016-04-20.

[13] "Android-Malware: Adware war gestern. Android-Trojaner auf dem Vormarsch.". *botfrei Blog*.

[14] "New type of auto-rooting Android adware is nearly impossible to remove". *Ars Technica*.

[15] Michael Mimoso. "Shuanet Adware Roots Android Devices - Threatpost - The first stop for security news". *Threatpost - The first stop for security news*.

[16] "Adware Shedun nistet sich gegen den Willen der Nutzer in Android ein". *ITespresso.de*.

[17] "Android Trojan Software Morphs Into Real Apps, Nearly Impossible To Remove From Device's System: Report". *Yibada*.

[18] "Android-Malware: Neue Schadsoftware rootet Geräte und ist kaum zu entfernen - Golem.de".

[19] Swati Khandelwal (3 September 2015). "26 Android Phone Models Shipped with Pre-Installed Spyware". *The Hacker News*.

[20] "G Data : Mobile Malware Report" (PDF). *Public.gdatasoftware.com*. Retrieved 2016-04-20.

[21] Catalin Cimpanu (4 September 2015). "24 Chinese Android Smartphone Models Come with Pre-Installed Malware". *softpedia*.

[22] David Gilbert. "Amazon Selling $40 Android Tablets That Come With Pre-Installed Malware". *International Business Times*.

[23] "Chinese smartphones infected with pre-installed malwareSecurity Affairs". *Security Affairs*.

[24] "Chinese Android smartphones now shipping with pre-installed malware". *SC Magazine*.

[25] Diane Samson. "Malware Found Pre-Installed on Xiaomi, Huawei, Lenovo Phones". *iDigitalTimes.com*.

[26] "Amazon's $40 Chinese Android Tablets Infected With Pre-Installed Malware". *Design & Trend*.

[27] Jeremy Kirk (5 March 2014). "Pre-installed malware found on new Android phones". *Computerworld*.

[28] "G Data : Mobile Malware Report" (PDF). *Public.gdatasoftware.com*. Retrieved 2016-04-20.

[29] Waqas. "Amazon Store, a safe haven for Android Tablets with pre-installed malware". *HackRead*.

[30] "Pre-Installed Android Malware Raises Security Risks in Supply Chain".

[31] "Some Android Phones Come With Malware Pre-Installed: Report". *The Huffington Post*.

[32] "Brand New Android Smartphones Coming with Spyware and Malware". *WCCFtech*.

[33] "Chinese Android smartphone comes with malware pre-installed". *Graham Cluley*.

[34] Martin Brinkmann (8 September 2015). "Beware, your Android phone might come with preloaded spyware". *gHacks Technology News*.

[35] "Trojan adware on Android can give itself root access". *The Tech Report*.

[36] "Shedun, Shuanet und Shiftybug: Android-Smartphone vor Malware schützen".

[37] "Android-Nutzer: Achtung vor Trojaner-Adware Shedun - Check & Secure -". - *Check & Secure* -.

[38] "New Android adware tries to root your phone so you can't remove it". *ExtremeTech*.

[39] "More than 20,000 apps auto-root Android devices". *SC Magazine UK*.

[40] "Android's accessibility service grants god-mode p0wn power".

[41] "Trojanized adware family abuses accessibility service to install whatever apps it wants | Lookout Blog". *Blog.lookout.com*. 2015-11-19. Retrieved 2016-04-10.

[42] "Shedun trojan adware is hitting the Android Accessibility Service". *Theinquirer.net*. Retrieved 2016-04-20.

[43] "Shedun adware can install any malicious mobile appSecurity Affairs". *Security Affairs*.

[44] *Shedun gaining accessibility service privileges*. 18 November 2015 – via YouTube.

[45] Dennis Schirrmacher (20 November 2015). "Android-Malware: Werbeterror wie von Geisterhand". *Security*.

[46] "Der Adware – Trojaner Shedun". *trojaner-info.de*. 6 December 2015.

[47] Swati Khandelwal (20 November 2015). "This Malware Can Secretly Auto-Install any Android App to Your Phone". *The Hacker News*.

[48] "Trojaner-Adware installiert selbstständig ungewollte Android-Apps". *Areamobile.de*. Retrieved 2016-04-20.

[49] "Shedun: Neue Android-Adware installiert Apps ohne deine Einwilligung". *Androidmag*.

[50] John Woll. "Installation auch nach Ablehnung: Neue dreiste Android-Adware".

[51] "Android Shedun Malware: New Malware That Can Grant Access to Your Phone; Malware Impossible To Be Removed?". *Yibada*.

[52] "Gefährliche Android-Schadsoftware: Oft hilft nur neues Gerät". *Noz.de*. Retrieved 2016-04-20.

[53] "Shedun trojan adware is hitting the Android Accessibility Service". *The Inquirer*. 2015-11-20. Retrieved 2016-04-10.

[54] "Lookout discovers new trojanized adware; 20K popular apps caught in the crossfire | Lookout Blog". *Blog.lookout.com*. 2015-11-04. Retrieved 2016-04-10.

[55] "Shuanet, ShiftyBug and Shedun malware could auto-root your Android". *Betanews.com*. Retrieved 2016-04-10.

[56] "New Family Of Android Malware Virtually Impossible To Remove: Say Hello To Shedun, Shuanet And ShiftyBug : PERSONAL TECH". Tech Times. Retrieved 2016-04-10.

[57] Goodin, Dan (2015-11-19). "Android adware can install itself even when users explicitly reject it". Ars Technica. Retrieved 2016-04-10.

[58] "Pavel Ponomariov - Avira Blog". *Avira Blog*.

Chapter 6

Protection Against Malware

6.1 Anti-keylogger

An **anti-keylogger** (or **anti–keystroke logger**) is a type of software specifically designed for the detection of keystroke logger software; often, such software will also incorporate the ability to delete or at least immobilize hidden keystroke logger software on your computer. In comparison to most anti-virus or anti-spyware software, the primary difference is that an anti-keylogger does not make a distinction between a *legitimate* keystroke-logging program and an *illegitimate* keystroke-logging program (such as malware); all keystroke-logging programs are flagged and optionally removed, whether they appear to be legitimate keystroke-logging software or not.

6.1.1 Use of anti-keyloggers

Keyloggers are sometimes part of malware packages downloaded onto computers without the owners' knowledge. Detecting the presence of a keylogger on a computer can be difficult. So-called anti- keylogging programs have been developed to thwart keylogging systems, and these are often effective when used properly.

Anti-keyloggers are used both by large organizations as well as individuals in order to scan for and remove (or in some cases simply immobilize) keystroke logging software on your computer. It is generally advised the software developers that anti-keylogging scans be run on a regular basis in order to reduce the amount of time during which a keylogger may record your keystrokes; for example, if you scan your system once every three days, there is a maximum of only three days during which a keylogger could be hidden on your computer and recording your keystrokes.

Public computers

Public computers are extremely susceptible to the installation of keystroke logging software and hardware, and there are documented instances of this occurring.[1] Public computers are particularly susceptible to keyloggers because any number of people can gain access to the machine and install both a hardware keylogger and a software keylogger, either or both of which can be secretly installed in a matter of minutes.[2] Anti-keyloggers are often used on a daily basis to ensure that public computers are not infected with keyloggers, and are safe for public use.

Gaming usage

Keyloggers have been prevalent in the online gaming industry, being used to secretly record a gamer's access credentials, user name and password, when logging into an account, this information is sent back to the hacker. The hacker can sign on later to the account and change the password to the account, thus stealing it.

World of Warcraft has been of particular importance to game hackers and has been the target of numerous keylogging viruses. Anti-keyloggers are used by many World of Warcraft and other gaming community members in order to try to keep their gaming accounts secure.

Financial institutions

Financial institutions have become the target of keyloggers,[3] particularly those institutions which do not use advanced security features such as PIN pads or screen keyboards.[4] Anti-keyloggers are used to run regular scans of any computer on which banking or client information is accessed, protecting passwords, banking information, and credit card numbers from identity thieves.

Personal use

The most common use of an anti-keylogger is by individuals wishing to protect their privacy while using their computer; uses range from protecting financial information used in online banking, any passwords, personal communication, and

virtually any other information which may be typed into your computer. Keyloggers are often installed by people you know, and many times have been installed by an ex-partner hoping to spy on their ex-partner's activities, particularly chat.[5]

6.1.2 Types

Signature-based

This type of software has a signature base, that is strategic information that helps to uniquely identify a keylogger, and the list contains as many known keyloggers as possible. Some vendors make some effort or availability of an up-to-date listing for download by customers. Each time you run a 'System Scan' this software compares the contents of your hard disk drive, item by item, against the list, looking for any matches.

This type of software is a rather widespread one, but it has its own drawbacks The biggest drawback of signature-based anti-keyloggers is that, while using them you can only be sure that you are protected from keyloggers found on your signature-base list, thus staying absolutely vulnerable to unknown or unrecognized keyloggers. A criminal can download one of many famous keyloggers, change it just enough and your anti-keylogger won't recognize it.

Heuristic analysis

This software doesn't use signature bases, it uses a checklist of known features, attributes, and methods that keyloggers are knwon use.

It analyzes the methods of work of all the modules in your PC, thus blocking the activity of any module that is similar to the work of keyloggers. Though this method gives better keylogging protection than signature-based anti-keyloggers, it has its own drawbacks. One of them is that this type of software blocks non-keyloggers also. Several 'non-harmful' software modules, either part of the operating system or part of legitimate apps, use processes which keyloggers also use, which can trigger a false positive. Usually all the non signature-based keyloggers have the option to allow the user to unblock selected modules, but this can cause difficulties for inexperienced users who are unable to discern good modules from bad modules when maually choosing to block or unblock.

6.1.3 See also

- Keystroke logger
- Hardware keylogger
- Software anti-keylogger

6.1.4 References

[1] "Keyloggers found plugged into library computers". *SC Magazine*. Retrieved 25 April 2011.

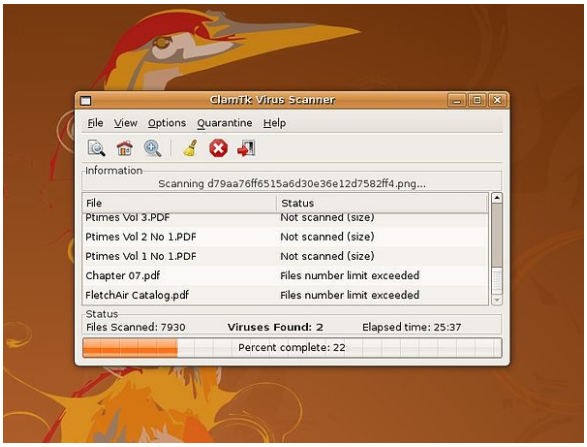
[2] "Anti Keylogging & Public Computers". *Anti Keylogging & Public Computers*. Archived from the original on 22 May 2011. Retrieved 10 May 2011.

[3] "Cyber threat landscape faced by financial and insurance industry". Dr Kim-Kwang Raymond Choo. Retrieved 21 February 2011.

[4] "Privacy Watch: More Criminals Use Keystroke Loggers". *Privacy Watch: More Criminals Use Keystroke Loggers*. PC World About.

[5] "Is someone you know spying on you?". *USA Today*. 4 March 2010. Retrieved 25 April 2011.

6.2 Antivirus software



ClamTk, an open source antivirus based on the ClamAV antivirus engine, originally developed by Tomasz Kojm in 2001.

Antivirus or **anti-virus** software (often abbreviated as **AV**), sometimes known as **anti-malware** software, is computer software used to prevent, detect and remove malicious software.[1]

Antivirus software was originally developed to detect and remove computer viruses, hence the name. However, with the proliferation of other kinds of malware, antivirus software started to provide protection from other computer threats. In particular, modern antivirus software can protect from: malicious browser helper objects (BHOs), browser hijackers, ransomware, keyloggers, backdoors, rootkits, trojan horses, worms, malicious LSPs, dialers, fraudtools,

adware and spyware.[2] Some products also include protection from other computer threats, such as infected and malicious URLs, spam, scam and phishing attacks, online identity (privacy), online banking attacks, social engineering techniques, advanced persistent threat (APT) and botnet DDoS attacks.[3]

6.2.1 History

See also: Timeline of notable computer viruses and worms

1949–1980 period (pre-antivirus days)

Although the roots of the computer virus date back as early as 1949, when the Hungarian scientist John von Neumann published the *"Theory of self-reproducing automata"*,[4] the first known computer virus appeared in 1971 and was dubbed the "Creeper virus".[5] This computer virus infected Digital Equipment Corporation's (DEC) PDP-10 mainframe computers running the TENEX operating system.[6][7]

The Creeper virus was eventually deleted by a program created by Ray Tomlinson and known as "The Reaper".[8] Some people consider "The Reaper" the first antivirus software ever written – it may be the case, but it is important to note that the Reaper was actually a virus itself specifically designed to remove the Creeper virus.[8][9][10]

The Creeper virus was followed by several other viruses. The first known that appeared "in the wild" was "Elk Cloner", in 1981, which infected Apple II computers.[11][12][13]

In 1983, the term *"computer virus"* was coined by Fred Cohen in one of the first ever published academic papers on computer viruses.[14] Cohen used the term *"computer virus"* to describe a program that: *"affect other computer programs by modifying them in such a way as to include a (possibly evolved) copy of itself."*[15] (note that a more recent, and precise, definition of computer virus has been given by the Hungarian security researcher Péter Ször: *"a code that recursively replicates a possibly evolved copy of itself"*[16][17])

The first IBM PC compatible "in the wild" computer virus, and one of the first real widespread infections, was "Brain" in 1986. From then, the number of viruses has grown exponentially.[18][19] Most of the computer viruses written in the early and mid-1980s were limited to self-reproduction and had no specific damage routine built into the code. That changed when more and more programmers became acquainted with computer virus programming and created viruses that manipulated or even destroyed data on infected computers.[20]

Before internet connectivity was widespread, computer viruses were typically spread by infected floppy disks. Antivirus software came into use, but was updated relatively infrequently. During this time, virus checkers essentially had to check executable files and the boot sectors of floppy disks and hard disks. However, as internet usage became common, viruses began to spread online.[21]

1980–1990 period (early days)

There are competing claims for the innovator of the first antivirus product. Possibly, the first publicly documented removal of an "in the wild" computer virus (i.e. the "Vienna virus") was performed by Bernd Fix in 1987.[22][23]

In 1987, Andreas Lüning and Kai Figge founded G Data Software and released their first antivirus product for the Atari ST platform.[24] Dubiously, they later also produced Virus Construction Kits.[25] In 1987, the *Ultimate Virus Killer (UVK)* was also released.[26] This was the de facto industry standard virus killer for the Atari ST and Atari Falcon, the last version of which (version 9.0) was released in April 2004. In 1987, in USA, John McAfee founded the McAfee company (now part of Intel Security[27]) and, at the end of that year, he released the first version of VirusScan.[28] In the meanwhile, in Slovakia, Peter Paško and Miroslav Trnka created the first version of NOD32 antivirus (albeit they established ESET only in 1992).[29][30]

In 1987, Fred Cohen wrote that **there is no algorithm that can perfectly detect all possible computer viruses**.[31]

Finally, in the end of 1987, the first two heuristic antivirus utilities were released: *FluShot Plus* by Ross Greenberg[32][33][34] and *Anti4us* by Erwin Lanting.[35][36] However, the kind of heuristic they were using was totally different from the one used today by many antivirus products. The first antivirus product with a heuristic engine which resembles the ones used nowadays was F-PROT in 1991.[37] The early heuristic engines were based on dividing the binary in different sections: data section, code section (in legitimate binary it usually starts always from the same location). Indeed, the initial viruses re-organise the layout of the sections, or override the initial portion of section in order to jump to the very end of the file where malicious code was located and then, later on, go back to resume the execution of the original code. This was a very specific pattern, not used at the time by any legitimate software, that initially represented a very nice heuristic to catch where something was suspicious or not. Later, in time, other kind of more advanced heuristics have been added, such as: suspicious sections name, incorrect header size, wildcards and regular expressions and partial pattern in-memory matching.

In 1988, the growth of antivirus companies continued. In

Germany, Tjark Auerbach founded Avira (*H+BEDV* at the time) and released the first version of *AntiVir* (named *"Luke Filewalker"* at the time). In Bulgaria, Dr. Vesselin Bontchev released his first freeware antivirus program (he later joined FRISK Software). Also Frans Veldman released the first version of ThunderByte Antivirus, also known as *TBAV* (he sold his company to Norman Safeground in 1998). In Czech Republic, Pavel Baudiš and Eduard Kučera started avast! (at the time *ALWIL Software*) and released their first version of avast! antivirus. In June 1988, in South Korea, Dr. Ahn Cheol-Soo released its first antivirus software, called *V1* (he founded AhnLab later in 1995). Finally, in the Autumn 1988, in United Kingdom, Alan Solomon founded S&S International and created his *Dr. Solomon's Anti-Virus Toolkit* (although he launched it commercially only in 1991 – in 1998 Dr. Solomon's company was acquired by McAfee). In November 1988 a professor at the Panamerican University in Mexico City named Alejandro E. Carriles copyrighted the first antivirus software in Mexico under the name "Byte Matabichos" (Byte Bugkiller) to help solve the rampant virus infestation among students.[38]

Also in 1988, a mailing list named VIRUS-L[39] was started on the BITNET/EARN network where new viruses and the possibilities of detecting and eliminating viruses were discussed. Some members of this mailing list were: Alan Solomon, Eugene Kaspersky (Kaspersky Lab), Friðrik Skúlason (FRISK Software), John McAfee (McAfee), Luis Corrons (Panda Security), Mikko Hyppönen (F-Secure), Péter Szőr, Tjark Auerbach (Avira) and Dr. Vesselin Bontchev (FRISK Software).[39]

In 1989, in Iceland, Friðrik Skúlason created the first version of F-PROT Anti-Virus back in 1989 (he founded FRISK Software only in 1993). In the meanwhile, in United States, Symantec (founded by Gary Hendrix in 1982) launched its first *Symantec antivirus for Macintosh* (SAM).[40][41] SAM 2.0, released March 1990, incorporated technology allowing users to easily update SAM to intercept and eliminate new viruses, including many that didn't exist at the time of the program's release.[42]

In the end of the 1980s, in United Kingdom, Jan Hruska and Peter Lammer founded the security firm Sophos and began producing their first antivirus and encryption products. In the same period, in Hungary, also VirusBuster was founded (which has recently being incorporated by Sophos).

1990–2000 period (emergence of the antivirus industry)

In 1990, in Spain, Mikel Urizarbarrena founded Panda Security (*Panda Software* at the time).[43] In Hungary, the security researcher Péter Szőr released the first version of *Pasteur* antivirus. In Italy, Gianfranco Tonello created the first version of VirIT eXplorer antivirus (he founded TG Soft one year later).[44]

In 1990, the Computer Antivirus Research Organization (CARO) was founded. In 1991, CARO released the *"Virus Naming Scheme"*, originally written by Friðrik Skúlason and Vesselin Bontchev.[45] Although this naming scheme is now outdated, it remains the only existing standard that most computer security companies and researchers ever attempted to adopt. CARO members includes: Alan Solomon, Costin Raiu, Dmitry Gryaznov, Eugene Kaspersky, Friðrik Skúlason, Igor Muttik, Mikko Hyppönen, Morton Swimmer, Nick FitzGerald, Padgett Peterson, Peter Ferrie, Righard Zwienenberg and Dr. Vesselin Bontchev.[46][47]

In 1991, in the USA, Symantec released the first version of Norton Anti-Virus. In the same year, in Czechoslovakia, Jan Gritzbach and Tomáš Hofer founded AVG Technologies (*Grisoft* at the time), although they released the first version of their *Anti-Virus Guard* (AVG) only in 1992. On the other hand, in Finland, F-Secure (founded in 1988 by Petri Allas and Risto Siilasmaa – with the name of Data Fellows) released the first version of their antivirus product. F-Secure claims to be the first antivirus firm to establish a presence on the World Wide Web.[48]

In 1991, the European Institute for Computer Antivirus Research (EICAR) was founded to further antivirus research and improve development of antivirus software.[49][50]

In 1992, in Russia, Igor Danilov released the first version of *SpiderWeb*, which later became Dr. Web.[51]

In 1994, AV-TEST reported that there were 28,613 unique malware samples (based on MD5) in their database.[52]

Over time other companies were been founded. In 1996, in Romania, Bitdefender was founded and released the first version of *Anti-Virus eXpert* (AVX).[53] In 1997, in Russia, Eugene Kaspersky and Natalia Kaspersky co-founded security firm Kaspersky Lab.[54]

In 1996, there was also the first "in the wild" Linux virus, known as *"Staog"*.[55]

In 1999, AV-TEST reported that there were 98,428 unique malware samples (based on MD5) in their database.[52]

2000–2005 period

In 2000, Rainer Link and Howard Fuhs started the first open source antivirus engine, called *OpenAntivirus Project*.[56]

In 2001, Tomasz Kojm released the first version of ClamAV, the first ever open source antivirus engine to be commercialised. In 2007, ClamAV was bought by Sourcefire,[57] which in turn was acquired by Cisco Systems in 2013.[58]

In 2002, in United Kingdom, Morten Lund and Theis Søndergaard co-founded the antivirus firm BullGuard.[59]

In 2005, AV-TEST reported that there were 333,425 unique malware samples (based on MD5) in their database.[52]

2005 to present

In 2007, AV-TEST reported a number of 5,490,960 new unique malware samples (based on MD5) only for that year.[52] In 2012 and 2013, antivirus firms reported a new malware samples range from 300,000 to over 500,000 per day.[60][61]

Over the years it has become necessary for antivirus software to use several different strategies (e.g. specific email and network protection or low level modules) and detection algorithms, as well as to check an increasing variety of files, rather than just executables, for several reasons:

- Powerful macros used in word processor applications, such as Microsoft Word, presented a risk. Virus writers could use the macros to write viruses embedded within documents. This meant that computers could now also be at risk from infection by opening documents with hidden attached macros.[62]
- The possibility of embedding executable objects inside otherwise non-executable file formats can make opening those files a risk.[63]
- Later email programs, in particular Microsoft's Outlook Express and Outlook, were vulnerable to viruses embedded in the email body itself. A user's computer could be infected by just opening or previewing a message.[64]

In 2005, F-Secure was the first security firm that developed an Anti-Rootkit technology, called *BlackLight*.[65]

Given the consideration that most of the people is nowadays connected to the Internet round-the-clock, in 2008, Jon Oberheide first proposed a Cloud-based antivirus design.[66]

In February 2008 McAfee Labs added the industry-first cloud-based anti-malware functionality to VirusScan under Artemis name. It was tested by AV-Comparatives in February 2008[67] and officially unveiled in August 2008 in McAfee VirusScan.[68]

Cloud AV created problems for comparative testing of security software – part of the AV definitions was out of testers control (on constantly updated AV company servers) thus making results non-repeatable. As a result, Anti-Malware Testing Standards Organisation (AMTSO) started working on methodology of testing cloud products which was adopted on 7 May 2009.[69]

In 2011, AVG introduced a similar cloud service, called Protective Cloud Technology.[70]

Most recently, the industry has seen approaches to the problem of detecting and mitigating Zero-day attacks. One method from Bromium involves micro-virtualization to protect desktops from malicious code execution initiated by the end user. Another approach from SentinelOne focuses on behavioral detection by building a full context around every process execution path in real time.[71][72]

6.2.2 Identification methods

One of the few solid theoretical results in the study of computer viruses is Frederick B. Cohen's 1987 demonstration that there is no algorithm that can perfectly detect all possible viruses.[31] However, using different layers of defense, a good detection rate may be achieved.

There are several methods which antivirus engine can use to identify malware:

- **Sandbox detection**: is a particular behavioural-based detection technique that, instead of detecting the behavioural fingerprint at run time, it executes the programs in a virtual environment, logging what actions the program performs. Depending on the actions logged, the antivirus engine can determine if the program is malicious or not.[73] If not, then, the program is executed in the real environment. Albeit this technique has shown to be quite effective, given its heaviness and slowness, it is rarely used in end-user antivirus solutions.[74]
- **Data mining techniques**: are one of the latest approach applied in malware detection. Data mining and machine learning algorithms are used to try to classify the behaviour of a file (as either malicious or benign) given a series of file features, that are extracted from the file itself.[75][76][77][78][79][80][81][82][83][84][85][86][87][88]

Signature-based detection

Traditional antivirus software relies heavily upon signatures to identify malware.[89]

Substantially, when a malware arrives in the hands of an antivirus firm, it is analysed by malware researchers or by dynamic analysis systems. Then, once it is determined to be a malware, a proper signature of the file is extracted and added to the signatures database of the antivirus software.[90]

Although the signature-based approach can effectively contain malware outbreaks, malware authors have tried to stay a step ahead of such software by writing "oligomorphic", "polymorphic" and, more recently, "metamorphic" viruses, which encrypt parts of themselves or otherwise modify themselves as a method of disguise, so as to not match virus signatures in the dictionary.[91]

Heuristics

Many viruses start as a single infection and through either mutation or refinements by other attackers, can grow into dozens of slightly different strains, called variants. Generic detection refers to the detection and removal of multiple threats using a single virus definition.[92]

For example, the Vundo trojan has several family members, depending on the antivirus vendor's classification. Symantec classifies members of the Vundo family into two distinct categories, *Trojan.Vundo* and *Trojan.Vundo.B*.[93][94]

While it may be advantageous to identify a specific virus, it can be quicker to detect a virus family through a generic signature or through an inexact match to an existing signature. Virus researchers find common areas that all viruses in a family share uniquely and can thus create a single generic signature. These signatures often contain non-contiguous code, using wildcard characters where differences lie. These wildcards allow the scanner to detect viruses even if they are padded with extra, meaningless code.[95] A detection that uses this method is said to be "heuristic detection."

Rootkit detection

Main article: Rootkit

Anti-virus software can attempt to scan for rootkits. A rootkit is a type of malware designed to gain administrative-level control over a computer system without being detected. Rootkits can change how the operating system functions and in some cases can tamper with the anti-virus program and render it ineffective. Rootkits are also difficult to remove, in some cases requiring a complete re-installation of the operating system.[96]

Real-time protection

Real-time protection, on-access scanning, background guard, resident shield, autoprotect, and other synonyms refer to the automatic protection provided by most antivirus, anti-spyware, and other anti-malware programs. This monitors computer systems for suspicious activity such as computer viruses, spyware, adware, and other malicious objects in 'real-time', in other words while data loaded into the computer's active memory: when inserting a CD, opening an email, or browsing the web, or when a file already on the computer is opened or executed.[97]

6.2.3 Issues of concern

Unexpected renewal costs

Some commercial antivirus software end-user license agreements include a clause that the subscription will be automatically renewed, and the purchaser's credit card automatically billed, at the renewal time without explicit approval. For example, McAfee requires users to unsubscribe at least 60 days before the expiration of the present subscription[98] while BitDefender sends notifications to unsubscribe 30 days before the renewal.[99] Norton AntiVirus also renews subscriptions automatically by default.[100]

Rogue security applications

Main article: Rogue security software

Some apparent antivirus programs are actually malware masquerading as legitimate software, such as WinFixer, MS Antivirus, and Mac Defender.[101]

Problems caused by false positives

A "false positive" or "false alarm" is when antivirus software identifies a non-malicious file as malware. When this happens, it can cause serious problems. For example, if an antivirus program is configured to immediately delete or quarantine infected files, as is common on Microsoft Windows antivirus applications, a false positive in an essential file can render the Windows operating system or some applications unusable.[102] Recovering from such damage to critical software infrastructure incurs technical support costs and businesses can be forced to close whilst remedial action is undertaken.[103][104] For example, in May 2007 a faulty virus signature issued by Symantec mistakenly removed essential operating system files, leaving thousands of PCs unable to boot.[105]

Also in May 2007, the executable file required by Pegasus Mail on Windows was falsely detected by Norton AntiVirus as being a Trojan and it was automatically removed, preventing Pegasus Mail from running. Norton AntiVirus had falsely identified three releases of Pegasus Mail as malware,

and would delete the Pegasus Mail installer file when that happened.[106] In response to this Pegasus Mail stated:

In April 2010, McAfee VirusScan detected svchost.exe, a normal Windows binary, as a virus on machines running Windows XP with Service Pack 3, causing a reboot loop and loss of all network access.[107][108]

In December 2010, a faulty update on the AVG anti-virus suite damaged 64-bit versions of Windows 7, rendering it unable to boot, due to an endless boot loop created.[109]

In October 2011, Microsoft Security Essentials (MSE) removed the Google Chrome web browser, rival to Microsoft's own Internet Explorer. MSE flagged Chrome as a Zbot banking trojan.[110]

In September 2012, Sophos' anti-virus suite identified various update-mechanisms, including its own, as malware. If it was configured to automatically delete detected files, Sophos Antivirus could render itself unable to update, required manual intervention to fix the problem.[111][112]

System and interoperability related issues

Running (the real-time protection of) multiple antivirus programs concurrently can degrade performance and create conflicts.[113] However, using a concept called multiscanning, several companies (including G Data[114] and Microsoft[115]) have created applications which can run multiple engines concurrently.

It is sometimes necessary to temporarily disable virus protection when installing major updates such as Windows Service Packs or updating graphics card drivers.[116] Active antivirus protection may partially or completely prevent the installation of a major update. Anti-virus software can cause problems during the installation of an operating system upgrade, e.g. when upgrading to a newer version of Windows "in place" — without erasing the previous version of Windows. Microsoft recommends that anti-virus software be disabled to avoid conflicts with the upgrade installation process.[117][118][119]

The functionality of a few computer programs can be hampered by active anti-virus software. For example, TrueCrypt, a disk encryption program, states on its troubleshooting page that anti-virus programs can conflict with TrueCrypt and cause it to malfunction or operate very slowly.[120] Anti-virus software can impair the performance and stability of games running in the Steam platform.[121]

Support issues also exist around antivirus application interoperability with common solutions like SSL VPN remote access and network access control products.[122] These technology solutions often have policy assessment applications which require that an up to date antivirus is installed and running. If the antivirus application is not recognized by the policy assessment, whether because the antivirus application has been updated or because it is not part of the policy assessment library, the user will be unable to connect.

Effectiveness

Studies in December 2007 showed that the effectiveness of antivirus software had decreased in the previous year, particularly against unknown or zero day attacks. The computer magazine *c't* found that detection rates for these threats had dropped from 40-50% in 2006 to 20-30% in 2007. At that time, the only exception was the NOD32 antivirus, which managed a detection rate of 68 percent.[123] According to the *ZeuS tracker* website the average detection rate for all variants of the well-known ZeuS trojan is as low as 40%.[124]

The problem is magnified by the changing intent of virus authors. Some years ago it was obvious when a virus infection was present. The viruses of the day, written by amateurs, exhibited destructive behavior or pop-ups. Modern viruses are often written by professionals, financed by criminal organizations.[125]

In 2008, Eva Chen, CEO of Trend Micro, stated that the anti-virus industry has over-hyped how effective its products are — and so has been misleading customers — for years.[126]

Independent testing on all the major virus scanners consistently shows that none provide 100% virus detection. The best ones provided as high as 99.9% detection for simulated real-world situations, while the lowest provided 91.1% in tests conducted in August 2013. Many virus scanners produce false positive results as well, identifying benign files as malware.[127]

Although methodologies may differ, some notable independent quality testing agencies include AV-Comparatives, ICSA Labs, West Coast Labs, Virus Bulletin, AV-TEST and other members of the Anti-Malware Testing Standards Organization.[128][129]

New viruses

Anti-virus programs are not always effective against new viruses, even those that use non-signature-based methods that should detect new viruses. The reason for this is that the virus designers test their new viruses on the major antivirus applications to make sure that they are not detected before releasing them into the wild.[130]

Some new viruses, particularly ransomware, use polymorphic code to avoid detection by virus scanners. Jerome Segura, a security analyst with ParetoLogic,

explained:[131]

A proof of concept virus has used the Graphics Processing Unit (GPU) to avoid detection from anti-virus software. The potential success of this involves bypassing the CPU in order to make it much harder for security researchers to analyse the inner workings of such malware.[132]

Rootkits

Detecting rootkits is a major challenge for anti-virus programs. Rootkits have full administrative access to the computer and are invisible to users and hidden from the list of running processes in the task manager. Rootkits can modify the inner workings of the operating system[133] and tamper with antivirus programs.

Damaged files

If a file has been infected by a computer virus, anti-virus software will attempt to remove the virus code from the file during disinfection, but it is not always able to restore the file to its undamaged state.[134][135] In such circumstances, damaged files can only be restored from existing backups or shadow copies (this is also true for ransomware[136]); installed software that is damaged requires re-installation[137] (however, see System File Checker).

Firmware issues

Active anti-virus software can interfere with a firmware update process.[138] Any writeable firmware in the computer can be infected by malicious code.[139] This is a major concern, as an infected BIOS could require the actual BIOS chip to be replaced to ensure the malicious code is completely removed.[140] Anti-virus software is not effective at protecting firmware and the motherboard BIOS from infection.[141] In 2014, security researchers discovered that USB devices contain writeable firmware which can be modified with malicious code (dubbed "BadUSB"), which anti-virus software cannot detect or prevent. The malicious code can run undetected on the computer and could even infect the operating system prior to it booting up.[142][143]

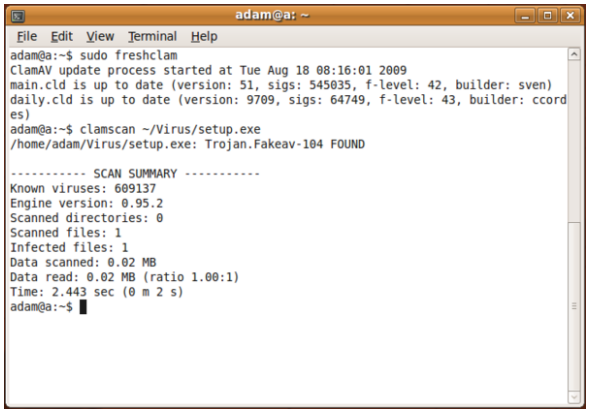
6.2.4 Performance and other drawbacks

Antivirus software has some drawbacks, first of which that it can impact a computer's performance.[144]

Furthermore, inexperienced users can be lulled into a false sense of security when using the computer, considering themselves to be invulnerable, and may have problems understanding the prompts and decisions that antivirus software presents them with. An incorrect decision may lead to a security breach. If the antivirus software employs heuristic detection, it must be fine-tuned to minimize misidentifying harmless software as malicious (false positive).[145]

Antivirus software itself usually runs at the highly trusted kernel level of the operating system to allow it access to all the potential malicious process and files, creating a potential avenue of attack.[146]

6.2.5 Alternative solutions



The command-line virus scanner of Clam AV 0.95.2, an open source antivirus originally developed by Tomasz Kojm in 2001. Here running a virus signature definition update, scanning a file and identifying a Trojan.

Installed antivirus solutions, running on individual computers, although the most used, is only one method of guarding against malware. Other alternative solutions are also used, including: Unified Threat Management (UTM), hardware and network firewalls, Cloud-based antivirus and on-line scanners.

Hardware and network firewall

Network firewalls prevent unknown programs and processes from accessing the system. However, they are not antivirus systems and make no attempt to identify or remove anything. They may protect against infection from outside the protected computer or network, and limit the activity of any malicious software which is present by blocking incoming or outgoing requests on certain TCP/IP ports. A firewall is designed to deal with broader system threats that come from network connections into the system and is not an alternative to a virus protection system.

Cloud antivirus

Cloud antivirus is a technology that uses lightweight agent software on the protected computer, while offloading the majority of data analysis to the provider's infrastructure.[147]

One approach to implementing cloud antivirus involves scanning suspicious files using multiple antivirus engines. This approach was proposed by an early implementation of the cloud antivirus concept called CloudAV. CloudAV was designed to send programs or documents to a network cloud where multiple antivirus and behavioral detection programs are used simultaneously in order to improve detection rates. Parallel scanning of files using potentially incompatible antivirus scanners is achieved by spawning a virtual machine per detection engine and therefore eliminating any possible issues. CloudAV can also perform "retrospective detection," whereby the cloud detection engine rescans all files in its file access history when a new threat is identified thus improving new threat detection speed. Finally, CloudAV is a solution for effective virus scanning on devices that lack the computing power to perform the scans themselves.[148]

Some examples of cloud anti-virus products are Panda Cloud Antivirus and Immunet. Now Comodo Cloud Antivirus Beta3 Version 1.0.376043.87 is Released![149][150]

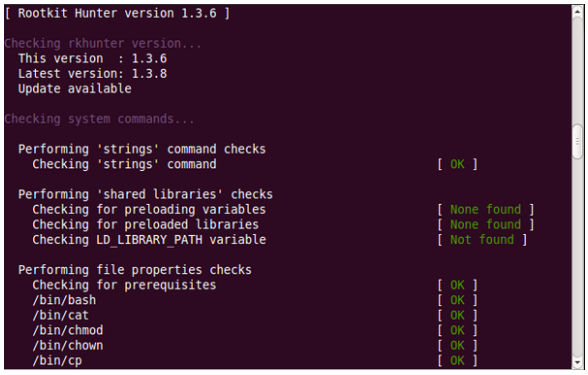
Online scanning

Some antivirus vendors maintain websites with free online scanning capability of the entire computer, critical areas only, local disks, folders or files. Periodic online scanning is a good idea for those that run antivirus applications on their computers because those applications are frequently slow to catch threats. One of the first things that malicious software does in an attack is disable any existing antivirus software and sometimes the only way to know of an attack is by turning to an online resource that is not installed on the infected computer.[151]

Specialist tools

Virus removal tools are available to help remove stubborn infections or certain types of infection. Examples include Trend Micro's *Rootkit Buster*,[152] and rkhunter for the detection of rootkits, Avira's *AntiVir Removal Tool*,[153] *PCTools Threat Removal Tool*,[154] and AVG's Anti-Virus Free 2011.[155]

A rescue disk that is bootable, such as a CD or USB storage device, can be used to run antivirus software outside of the installed operating system, in order to remove infections while they are dormant. A bootable antivirus disk can be useful when, for example, the installed operating system is no longer bootable or has malware that is resisting all attempts to be removed by the installed antivirus software. Examples of some of these bootable disks include the *Avira AntiVir Rescue System*,[153] *PCTools Alternate Operating System Scanner*,[156] and *AVG Rescue CD*.[157] The AVG Rescue CD software can also be installed onto a USB storage device, that is bootable on newer computers.[157]



The command-line rkhunter scanner, an engine to scan for Linux rootkits. Here running the tool on Ubuntu.

6.2.6 Usage and risks

According to an FBI survey, major businesses lose $12 million annually dealing with virus incidents.[158] A survey by Symantec in 2009 found that a third of small to medium-sized business did not use antivirus protection at that time, whereas more than 80% of home users had some kind of antivirus installed.[159] According to a sociological survey conducted by G Data Software in 2010 49% of women did not use any antivirus program at all.[160]

6.2.7 See also

- Anti-virus and anti-malware software
- CARO, the Computer Antivirus Research Organization
- Comparison of antivirus software
- Comparison of computer viruses
- EICAR, the European Institute for Computer Antivirus Research
- Firewall software
- Internet security
- Linux malware
- Quarantine (computing)

- Sandbox (computer security)
- Timeline of computer viruses and worms
- Virus hoax

6.2.8 References

[1] Naveen, Sharanya. "Antivirus software". Retrieved 31 May 2016.

[2] lifehacker: The Difference Between Antivirus and Anti-Malware (and Which to Use)

[3] "What is antivirus software?". Microsoft.

[4] John von Neumann: "Theory of self-reproducing automata" (1949)

[5] Thomas Chen, Jean-Marc Robert (2004). "The Evolution of Viruses and Worms". Retrieved 2009-02-16.

[6] From the first email to the first YouTube video: a definitive internet history. Tom Meltzer and Sarah Phillips. *The Guardian*. 23 October 2009

[7] *IEEE Annals of the History of Computing, Volumes 27-28*. IEEE Computer Society, 2005. 74. Retrieved from Google Books on 13 May 2011. "[...]from one machine to another led to experimentation with the *Creeper* program, which became the world's first computer worm: a computation that used the network to recreate itself on another node, and spread from node to node."

[8] John Metcalf (2014). "Core War: Creeper & Reaper". Retrieved 2014-05-01.

[9] Creeper – The Virus Encyclopedia

[10] What was the First Antivirus Software?

[11] "Elk Cloner". Retrieved 2010-12-10.

[12] "Top 10 Computer Viruses: No. 10 – Elk Cloner". Retrieved 2010-12-10.

[13] "List of Computer Viruses Developed in 1980s". Retrieved 2010-12-10.

[14] Fred Cohen: "Computer Viruses – Theory and Experiments" (1983)

[15] Fred Cohen 1988 "On the implications of Computer Viruses and Methods of Defense"

[16] Péter Ször: "The Art of Computer Virus Research and Defense" (2005)

[17] VirusBulletin: "In memoriam: Péter Ször 1970-2013" (2013)

[18] History of viruses

[19] Leyden, John (January 19, 2006). "PC virus celebrates 20th birthday". *The Register*. Retrieved March 21, 2011.

[20] "About computer viruses of 1980's" (PDF). Retrieved 2016-02-17.

[21] Panda Security (April 2004). "(II) Evolution of computer viruses". Archived from the original on 2 August 2009. Retrieved 2009-06-20.

[22] Kaspersky Lab Virus list

[23] Wells, Joe (1996-08-30). "Virus timeline". IBM. Archived from the original on 4 June 2008. Retrieved 2008-06-06.

[24] G Data Software AG (2011). "G Data presents security firsts at CeBIT 2010". Retrieved 22 August 2011.

[25] G Data Software AG (2016). "Virus Construction Set II". Retrieved 3 July 2016.

[26] Karsmakers, Richard (January 2010). "The ultimate Virus Killer Book and Software". Retrieved 6 July 2016.

[27] "McAfee Becomes Intel Security". McAfee Inc. Retrieved 15 January 2014.

[28] Cavendish, Marshall (2007). *Inventors and Inventions, Volume 4*. Paul Bernabeo. p. 1033. ISBN 0761477675.

[29] "About ESET Company". 16 February 2016.

[30] "ESET NOD32 Antivirus". Vision Square. 16 February 2016.

[31] Cohen, Fred, An Undetectable Computer Virus (Archived), 1987, IBM

[32] Patricia A. Yevics:"Flu Shot for Computer Viruses"

[33] How friends help friends on the Internet: The Ross Greenberg Story

[34] Anti-virus is 30 years old

[35] A Brief History of Antivirus Software

[36] Antivirus software history

[37] http://www.frisk.is/fyrirtaeki.html

[38] Direccion General del Derecho de Autor, SEP, Mexico D.F. Registry 20709/88 Book 8, page 40, dated November 24, 1988.

[39] VIRUS-L mailing list archive

[40] Symantec Softwares and Internet Security at PCM

[41] SAM Identifies Virus-Infected Files, Repairs Applications, *InfoWorld*, May 22, 1989

[42] SAM Update Lets Users Program for New Viruses, *InfoWorld*, Feb 19, 1990

[43] Naveen, Sharanya. "Panda Security". Retrieved 31 May 2016.

[44] TG Soft History

[45] Skúlason and Bontchev: "Virus Naming Scheme" (1991)

[46] "CARO Members". CARO. Retrieved 6 June 2011.

[47] CAROids, Hamburg 2003

[48] "F-Secure Weblog : News from the Lab". F-secure.com. Retrieved 2012-09-23.

[49] "About EICAR". *EICAR official website*. Retrieved 28 October 2013.

[50] David Harley, Lysa Myers & Eddy Willems. "Test Files and Product Evaluation: the Case for and against Malware Simulation" (PDF). *AVAR2010 13th Association of anti Virus Asia Researchers International Conference*. Archived from the original (PDF) on 29 September 2011. Retrieved June 30, 2011.

[51] "Dr. Web LTD Doctor Web / Dr. Web Reviews, Best AntiVirus Software Reviews, Review Centre". Reviewcentre.com. Retrieved 2014-02-17.

[52] [In 1994, AV-Test.org reported 28,613 unique malware samples (based on MD5). "A Brief History of Malware; The First 25 Years"]

[53] "BitDefender Product History".

[54] "InfoWatch Management". InfoWatch. Retrieved 12 August 2013.

[55] Linuxvirus

[56]

[57] "Sourcefire acquires ClamAV". ClamAV. 2007-09-17. Retrieved 2008-02-12.

[58] "Cisco Completes Acquisition of Sourcefire". *cisco.com*. 2013-10-07. Retrieved 2014-06-18.

[59] "(german) Interview with Morten Lund in Brandeins".

[60] "The digital detective: Mikko Hypponen's war on malware is escalating." (March 2012, Wired)

[61] James Lyne: "Everyday cybercrime – and what you can do about it" (February 2013, TED)

[62] Szor 2005, pp. 66–67

[63] "New virus travels in PDF files". 7 August 2001. Retrieved 2011-10-29.

[64] Slipstick Systems (February 2009). "Protecting Microsoft Outlook against Viruses". Archived from the original on 2 June 2009. Retrieved 2009-06-18.

[65] "Www f secure com blacklight". F-Secure Corporation. Retrieved 16 February 2016.

[66] Jon Oberheide: "CloudAV: N-Version Antivirus in the Network Cloud" (2008, Usenix)

[67] McAfee Artemis Preview Report

[68] McAfee Third Quarter 2008

[69] AMTSO Best Practices for Testing In-the-Cloud Security Products

[70] "TECHNOLOGY OVERVIEW". *AVG Security*. Retrieved 16 February 2015.

[71] NetworkWorld, Ellen Messmer, August 19, 2014:"Start-up offers up endpoint detection and response for behavior-based malware detection"

[72] HSToday.US, Kylie Bull, June 19, 2014:"Bromium Research Reveals Insecurity In Existing Endpoint Malware Protection Deployments"

[73] "Sandboxing against unknown zero day threats". Retrieved 2015-01-30.

[74] Szor 2005, pp. 474–481

[75] A Machine Learning Approach to Anti-virus System

[76] Data Mining Methods for Malware Detection

[77] Data mining and Machine Learning in Cybersecurity

[78] Analysis of Machine learning Techniques Used in Behavior-Based Malware Detection

[79] A survey of data mining techniques for malware detection using file features

[80] Intelligent automatic malicious code signatures extraction

[81] Malware Detection by Data Mining Techniques Based on Positionally Dependent Features

[82] Data mining methods for detection of new malicious executables

[83] IMDS: Intelligent Malware Detection System

[84] Learning to Detect and Classify Malicious Executables in the Wild

[85] Malware detection using statistical analysis of byte-level file content

[86] An intelligent PE-malware detection system based on association mining

[87] Malware detection based on mining API calls

[88] "Andromaly": a behavioral malware detection framework for android devices

[89] Fox-Brewster, Thomas. "Netflix Is Dumping Anti-Virus, Presages Death Of An Industry". Forbes. Retrieved September 4, 2015.

[90] Automatic Malware Signature Generation

[91] Szor 2005, pp. 252–288

[92] "Generic detection". Kaspersky. Retrieved 2013-07-11.

[93] Symantec Corporation (February 2009). "Trojan.Vundo". Archived from the original on 9 April 2009. Retrieved 2009-04-14.

[94] Symantec Corporation (February 2007). "Trojan.Vundo.B". Archived from the original on 27 April 2009. Retrieved 2009-04-14.

[95] "Antivirus Research and Detection Techniques". Extreme-Tech. Archived from the original on 27 February 2009. Retrieved 2009-02-24.

[96] Rootkit

[97] Kaspersky Lab Technical Support Portal Archived 13 February 2011 at WebCite

[98] Kelly, Michael (October 2006). "Buying Dangerously". Retrieved 2009-11-29.

[99] Bitdefender (2009). "Automatic Renewal". Retrieved 2009-11-29.

[100] Symantec (2014). "Norton Automatic Renewal Service FAQ". Retrieved 2014-04-09.

[101] SpywareWarrior (2007). "Rogue/Suspect Anti-Spyware Products & Web Sites". Retrieved 2009-11-29.

[102] Emil Protalinski (November 11, 2008). "AVG incorrectly flags user32.dll in Windows XP SP2/SP3". *Ars Technica*. Retrieved 2011-02-24.

[103] *McAfee to compensate businesses for buggy update*, retrieved 2 December 2010

[104] *Buggy McAfee update whacks Windows XP PCs*, archived from the original on 13 January 2011, retrieved 2 December 2010

[105] Aaron Tan (May 24, 2007). "Flawed Symantec update cripples Chinese PCs". *CNET Networks*. Retrieved 2009-04-05.

[106] David Harris (June 29, 2009). "January 2010 - Pegasus Mail v4.52 Release". *Pegasus Mail*. Archived from the original on 28 May 2010. Retrieved 2010-05-21.

[107] "McAfee DAT 5958 Update Issues". 21 April 2010. Archived from the original on 24 April 2010. Retrieved 22 April 2010.

[108] "Botched McAfee update shutting down corporate XP machines worldwide". 21 April 2010. Archived from the original on 22 April 2010. Retrieved 22 April 2010.

[109] John Leyden (December 2, 2010). "Horror AVG update ballsup bricks Windows 7". *The Register*. Retrieved 2010-12-02.

[110] *MSE false positive detection forces Google to update Chrome*, retrieved 3 October 2011

[111] *Sophos Antivirus Detects Itself as Malware, Deletes Key Binaries*, The Next Web, retrieved 5 March 2014

[112] *Shh/Updater-B false positive by Sophos anti-virus products*, Sophos, retrieved 5 March 2014

[113] Microsoft (January 2007). "Plus! 98: How to Remove McAfee VirusScan". Archived from the original on 27 September 2014. Retrieved 2014-09-27.

[114] Robert Vamosi (May 28, 2009). "G-Data Internet Security 2010". *PC World. Retrieved 2011-02-24.*

[115] Kelly Jackson Higgins (May 5, 2010). "New Microsoft Forefront Software Runs Five Antivirus Vendors' Engines". *Darkreading*. Retrieved 2011-02-24.

[116] Microsoft (April 2009). "Steps to take before you install Windows XP Service Pack 3". Archived from the original on 8 December 2009. Retrieved 2009-11-29.

[117] "Upgrading from Windows Vista to Windows 7". Retrieved 24 March 2012. Mentioned within "Before you begin".

[118] "Upgrading to Microsoft Windows Vista recommended steps.". Retrieved 24 March 2012.

[119] "How to troubleshoot problems during installation when you upgrade from Windows 98 or Windows Millennium Edition to Windows XP". 7 May 2007. Retrieved 24 March 2012. Mentioned within "General troubleshooting".

[120] "Troubleshooting". Retrieved 2011-02-17.

[121] "Spyware, Adware, and Viruses Interfering with Steam". Retrieved 11 April 2013. Steam support page.

[122] Field Notice: FN - 63204 - Cisco Clean Access has Interoperability issue with Symantec Anti-virus - delays Agent start-up

[123] Dan Goodin (December 21, 2007). "Anti-virus protection gets worse". *Channel Register*. Retrieved 2011-02-24.

[124]

[125] Dan Illett (July 13, 2007). "Hacking poses threats to business". *Computer Weekly*. Retrieved 2009-11-15.

[126] Tom Espiner (June 30, 2008). "Trend Micro: Antivirus industry lied for 20 years". *ZDNet*. Retrieved 2014-09-27.

[127] AV Comparatives (December 2013). "Whole Product Dynamic "Real World" Production Test" (PDF). Archived (PDF) from the original on 2 January 2013. Retrieved 2 January 2014.

[128] Guidelines released for antivirus software tests

[129] Harley, David (2011). *AVIEN Malware Defense Guide for the Enterprise*. Elsevier. p. 487. ISBN 9780080558660. Retrieved 2013-06-10.

[130] Kotadia, Munir (July 2006). "Why popular antivirus apps 'do not work'". Retrieved 14 April 2010.

[131] The Canadian Press (April 2010). "Internet scam uses adult game to extort cash". *CBC News*. Archived from the original on 18 April 2010. Retrieved 17 April 2010.

[132] Researchers up evilness ante with GPU-assisted malware - Coming to a PC near you, by Dan Goodin

[133] GIBSON RESEARCH CORPORATION SERIES: Security Now!

[134] "Why F-PROT Antivirus fails to disinfect the virus on my computer?". Retrieved 2015-08-20.

[135] "Actions to be performed on infected objects". Retrieved 2015-08-20.

[136] "Cryptolocker Ransomware: What You Need To Know". Retrieved 2014-03-28.

[137] "How Anti-Virus Software Works". Retrieved 2011-02-16.

[138] "BT Home Hub Firmware Upgrade Procedure". Retrieved 2011-03-06.

[139] "The 10 faces of computer malware". July 17, 2009. Retrieved 2011-03-06.

[140] "New BIOS Virus Withstands HDD Wipes". 27 March 2009. Retrieved 2011-03-06.

[141] "Phrack Inc. Persistent BIOS Infection". June 1, 2009. Archived from the original on 30 April 2011. Retrieved 2011-03-06.

[142] "Turning USB peripherals into BadUSB". Retrieved 2014-10-11.

[143] "Why the Security of USB Is Fundamentally Broken". 2014-07-31. Retrieved 2014-10-11.

[144] "How Antivirus Software Can Slow Down Your Computer". Support.com Blog. Retrieved 2010-07-26.

[145] "Softpedia Exclusive Interview: Avira 10". *Ionut Ilascu*. Softpedia. 14 April 2010. Retrieved 2011-09-11.

[146] "Norton AntiVirus ignores malicious WMI instructions". *Munir Kotadia*. CBS Interactive. 21 October 2004. Retrieved 2009-04-05.

[147] Zeltser, Lenny (October 2010). "What Is Cloud Anti-Virus and How Does It Work?". Archived from the original on 10 October 2010. Retrieved 2010-10-26.

[148] Jon Erickson (August 6, 2008). "Antivirus Software Heads for the Clouds". *Information Week*. Retrieved 2010-02-24.

[149] "Comodo Cloud Antivirus released". wikipost.org. Retrieved 2016-05-30.

[150] "Comodo Cloud Antivirus User Guideline PDF" (pdf). help.comodo.com. Retrieved 2016-05-30.

[151] Brian Krebs (March 9, 2007). "Online Anti-Virus Scans: A Free Second Opinion". *Washington Post*. Retrieved 2011-02-24.

[152] Ryan Naraine (February 2, 2007). "Trend Micro ships free 'rootkit buster'". *ZDNet*. Retrieved 2011-02-24.

[153] Neil J. Rubenking (March 26, 2010). "Avira AntiVir Personal 10". *PC Magazine*. Retrieved 2011-02-24.

[154] Neil J. Rubenking (September 16, 2010). "PC Tools Spyware Doctor with AntiVirus 2011". *PC Magazine*. Retrieved 2011-02-24.

[155] Neil J. Rubenking (October 4, 2010). "AVG Anti-Virus Free 2011". *PC Magazine*. Retrieved 2011-02-24.

[156] Neil J. Rubenking (November 19, 2009). "PC Tools Internet Security 2010". *PC Magazine*. Retrieved 2011-02-24.

[157] Carrie-Ann Skinner (March 25, 2010). "AVG Offers Free Emergency Boot CD". *PC World. Retrieved 2011-02-24.*

[158] "FBI estimates major companies lose $12m annually from viruses". 30 January 2007. Retrieved 20 February 2011.

[159] Michael Kaiser (April 17, 2009). "Small and Medium Size Businesses are Vulnerable". *National Cyber Security Alliance*. Retrieved 2011-02-24.

[160] "Nearly 50% of women don't use antivirus". SPAMfighter.

6.2.9 Bibliography

- Szor, Peter (2005), *The Art of Computer Virus Research and Defense*, Addison-Wesley, ISBN 0-321-30454-3

6.2.10 External links

- Antivirus software at DMOZ

6.3 Browser security

Browser security is the application of Internet security [1] to web browsers in order to protect networked data and computer systems from breaches of privacy or malware. Security exploits of browsers often use JavaScript - sometimes with cross-site scripting (XSS)[2] - sometimes with a secondary payload using Adobe Flash.[3] Security exploits can also take advantage of vulnerabilities (security holes) that are commonly exploited in all browsers (including Mozilla Firefox,[4] Google Chrome,[5] Opera,[6] Microsoft Internet Explorer,[7] and Safari[8]).

6.3.1 Security

Web browsers can be breached in one or more of the following ways:

- Operating system is breached and malware is reading/modifying the browser memory space in privilege mode [9]
- Operating system has a malware running as a background process, which is reading/modifying the browser memory space in privileged mode
- Main browser executable can be hacked
- Browser components may be hacked
- Browser plugins can be hacked
- Browser network communications could be intercepted outside the machine [10]

The browser may not be aware of any of the breaches above and may show user a safe connection is made.

Whenever a browser communicates with a website, the website, as part of that communication, collects some information about the browser (in order to process the formatting of the page to be delivered, if nothing else).[11] If malicious code has been inserted into the website's content, or in a worst-case scenario, if that website that has been specifically designed to host malicious code, then vulnerabilities specific to a particular browser can allow this malicious code to run processes within the browser application in unintended ways (and remember, one of the bits of information that a website collects from a browser communication is the browser's identity- allowing specific vulnerabilities to be exploited).[12] Once an attacker is able to run processes on the visitor's machine, then exploiting known security vulnerabilities can allow the attacker to gain privileged access (if the browser isn't already running with privileged access) to the "infected" system in order to perform an even greater variety of malicious processes and activities, on the machine or even the victim's whole network.[13]

Breaches of web browser security are usually for the purpose of bypassing protections to display pop-up advertising[14] collecting personally identifiable information (PII) for either Internet marketing or identity theft, website tracking or web analytics about a user against their will using tools such as web bugs, Clickjacking, Likejacking (where Facebook's like button is targeted),[15][16][17][18] HTTP cookies, zombie cookies or Flash cookies (Local Shared Objects or LSOs);[3] installing adware, viruses, spyware such as Trojan horses (to gain access to users' personal computers via cracking) or other malware including online banking theft using man-in-the-browser attacks.

Vulnerabilities in the web browser software itself can be minimized by keeping browser software updated,[19] but will not be sufficient if the underlying operating system is compromised, for example, by a rootkit.[20] Some subcomponents of browsers such as scripting, add-ons, and cookies[21][22][23] are particularly vulnerable ("the confused deputy problem") and also need to be addressed.

Following the principle of defence in depth, a fully patched and correctly configured browser may not be sufficient to ensure that browser-related security issues cannot occur. For example, a rootkit can capture keystrokes while someone logs into a banking website, or carry out a man-in-the-middle attack by modifying network traffic to and from a web browser. DNS hijacking or DNS spoofing may be used to return false positives for mistyped website names, or to subvert search results for popular search engines. Malware such as RSPlug simply modifies a system's configuration to point at rogue DNS servers.

Browsers can use more secure methods of network communication to help prevent some of these attacks:

- DNS: DNSSec and DNSCrypt, for example with non-default DNS servers such as Google Public DNS or OpenDNS.
- HTTP: HTTP Secure and SPDY with digitally signed public key certificates or Extended Validation Certificates.

Perimeter defenses, typically through firewalls and the use of filtering proxy servers that block malicious websites and perform antivirus scans of any file downloads, are commonly implemented as a best practice in large organizations to block malicious network traffic before it reaches a browser.

The topic of browser security has grown to the point of spawning the creation of entire organizations, such as The Browser Exploitation Framework Project,[24] creating platforms to collect tools to breach browser security, ostensibly in order to test browsers and network systems for vulnerabilities.

Plugins and extensions

Although not part of the browser per se, browser plugins and extensions extend the attack surface, exposing vulnerabilities in Adobe Flash Player, Adobe (Acrobat) Reader, Java plugin, and ActiveX that are commonly exploited. Malware may also be implemented as a browser extension, such as a browser helper object in the case of Internet Explorer.[25] Browsers like Google Chrome and Mozilla Firefox can block—or warn users of—insecure plugins.

Conversely, extensions may be used to harden the security configuration. US-CERT recommends to block Flash using NoScript.[26] Charlie Miller recommended "not to install Flash"[27] at the computer security conference CanSecWest. Several other security experts also recommend to either not install Adobe Flash Player or to block it.[28]

6.3.2 Password security model

The contents of a web page is arbitrary, but controlled by the entity owning the domain named displayed in the address bar. If HTTPS is used, then encryption is used to secure against attackers with access to the network from changing the page contents. For normal password usage on the WWW, when the user is confronted by a dialog asking for his password, the user is supposed to look at the address bar to determine whether the domain name in the address bar is the correct place to send the password.[29] For example, for Google's single sign-on system (used on e.g. youtube.com), the user should always check that the address bar says "https://accounts.google.com" before inputting his password.

The browser guarantees that the address bar is correct. Which is also a reason why browsers will generally display a warning when entering fullscreen mode, on top of where the address bar would normally be, so that a fullscreen website cannot make a fake browser user interface with a fake address bar.[30]

6.3.3 Privacy

Main articles: Internet privacy and Privacy mode

Flash

Main article: Local shared object § Privacy concerns

An August 2009 study by the Social Science Research Network found that 50% of websites using Flash were also employing flash cookies, yet privacy policies rarely disclosed them, and user controls for privacy preferences were lacking.[31] Most browsers' cache and history delete functions do not affect Flash Player's writing Local Shared Objects to its own cache, and the user community is much less aware of the existence and function of Flash cookies than HTTP cookies.[32] Thus, users having deleted HTTP cookies and purged browser history files and caches may believe that they have purged all tracking data from their computers when in fact Flash browsing history remains. As well as manual removal, the BetterPrivacy addon for Firefox can remove Flash cookies.[3] Adblock Plus can be used to filter out specific threats[14] and Flashblock can be used to give an option before allowing content on otherwise trusted sites.[33]

6.3.4 Hardware browser

A hardware-based solution which runs a non-writable, read-only file system and web browser based on the LiveCD approach. Brian Krebs recommends the use of a LiveCD to be protected from organized cybercrime. The first such hardware browser was the ZeusGard Secure Hardware Browser which was released in late 2013. Each time the bootable media is started the browser starts in a known clean and secure operating environment. Data is never stored on the device and the media cannot be overwritten, so it's clean each time it boots.

6.3.5 Browser hardening

Browsing the Internet as a least-privilege user account (i.e. without administrator privileges) limits the ability of a security exploit in a web browser from compromising the whole operating system.[34]

Internet Explorer 4 and later allows the blacklisting[35][36][37] and whitelisting[38][39] of ActiveX controls, add-ons and browser extensions in various ways.

Internet Explorer 7 added "protected mode", a technology that hardens the browser through the application of a security sandboxing feature of Windows Vista called Mandatory Integrity Control.[40] Google Chrome provides a sandbox to limit web page access to the operating system.[41]

Suspected malware sites reported to Google,[42] and confirmed by Google, are flagged as hosting malware in certain browsers.[43]

There are third-party extensions and plugins available to harden even the latest browsers,[44] and some for older browsers and operating systems. Whitelist-based software such as NoScript can block JavaScript and Adobe Flash which is used for most attacks on privacy, allowing users to choose only sites they know are safe - AdBlock Plus also uses whitelist ad filtering rules subscriptions, though both the software itself and the filtering list maintainers have come under controversy for by-default allowing some sites to pass the pre-set filters.[45]

6.3.6 See also

- Filter bubble
- Frame injection

- Identity driven networking
- Internet safety
- Network security policy
- Web application security

6.3.7 References

[1] *Internet security overview*, retrieved 2015-07-06

[2] Maone, Giorgio. "NoScript :: Add-ons for Firefox". *Mozilla Add-ons*. Mozilla Foundation.

[3] NC (Social Science Research Network). "BetterPrivacy :: Add-ons for Firefox". *Mozilla Add-ons*. Mozilla Foundation.

[4] Keizer, Greg. Firefox 3.5 Vulnerability Confirmed. Retrieved 19 November 2010.

[5] Messmer, Ellen and NetworkWorld. "Google Chrome Tops 'Dirty Dozen' Vulnerable Apps List". Retrieved 19 November 2010.

[6] Skinner, Carrie-Ann. Opera Plugs "Severe" Browser Hole. Retrieved 19 November 2010.

[7] Bradly, Tony. "It's Time to Finally Drop Internet Explorer 6" . Retrieved 19 November 2010.

[8] "Browser". Mashable. Retrieved 2 September 2011.

[9] Smith, Dave. "The Yontoo Trojan: New Mac OS X Malware Infects Google Chrome, Firefox And Safari Browsers Via Adware". IBT Media Inc. Retrieved 21 March 2013.

[10] Goodin, Dan. "MySQL.com breach leaves visitors exposed to malware". Retrieved 26 September 2011.

[11] Clinton Wong. "HTTP Transactions". O'Reilly.

[12] "9 Ways to Know Your PC is Infected with Malware".

[13] "Symantec Security Response Whitepapers".

[14] Palant, Wladimir. "Adblock Plus :: Add-ons for Firefox". *Mozilla Add-ons*. Mozilla Foundation.

[15] "Facebook privacy probed over 'like,' invitations". *CBC News*. 23 September 2010. Retrieved 24 August 2011.

[16] Albanesius, Chloe (19 August 2011). "German Agencies Banned From Using Facebook, 'Like' Button". *PC Magazine*. Retrieved 24 August 2011.

[17] McCullagh, Declan (2 June 2010). "Facebook 'Like' button draws privacy scrutiny". CNET News. Retrieved 19 December 2011.

[18] Roosendaal, Arnold (30 November 2010). "Facebook Tracks and Traces Everyone: Like This!". Retrieved 27 September 2011.

[19] State of Vermont. "Web Browser Attacks". Retrieved 11 April 2012.

[20] "Windows Rootkit Overview" (PDF). Symantec. Retrieved 2013-04-20.

[21] "Cross Site Scripting Attack". Retrieved 20 May 2013.

[22] Lenny Zeltser. "Mitigating Attacks on the Web Browser and Add-Ons". Retrieved 20 May 2013.

[23] Dan Goodin. "Two new attacks on SSL decrypt authentication cookies". Retrieved 20 May 2013.

[24] "beefproject.com".

[25] "How to Create a Rule That Will Block or Log Browser Helper Objects in Symantec Endpoint Protection". Symantec.com. Retrieved 12 April 2012.

[26] "Securing Your Web Browser". Archived from the original on 26 March 2010. Retrieved 2010-03-27.

[27] "Pwn2Own 2010: interview with Charlie Miller". 2010-03-01. Retrieved 2010-03-27.

[28] "Expert says Adobe Flash policy is risky". 2009-11-12. Retrieved 2010-03-27.

[29] John C. Mitchell. "Browser Security Model" (PDF).

[30] http://feross.org/html5-fullscreen-api-attack/

[31] "Soltani, Ashkan, Canty, Shannon, Mayo, Quentin, Thomas, Lauren and Hoofnagle, Chris Jay: Flash Cookies and Privacy". 2009-08-10. Retrieved 2009-08-18.

[32] "Local Shared Objects -- "Flash Cookies"". Electronic Privacy Information Center. 2005-07-21. Archived from the original on 16 April 2010. Retrieved 2010-03-08.

[33] Chee, Philip. "Flashblock :: Add-ons for Firefox". *Mozilla Add-ons*. Mozilla Foundation.

[34] "Using a Least-Privileged User Account". Microsoft. Retrieved 2013-04-20.

[35] "How to Stop an ActiveX control from running in Internet Explorer". Microsoft. Retrieved 2014-11-22.

[36] "Internet Explorer security zones registry entries for advanced users". Microsoft. Retrieved 2014-11-22.

[37] "Out-of-date ActiveX control blocking". Microsoft. Retrieved 2014-11-22.

[38] "Internet Explorer Add-on Management and Crash Detection". Microsoft. Retrieved 2014-11-22.

[39] "How to Manage Internet Explorer Add-ons in Windows XP Service Pack 2". Microsoft. Retrieved 2014-11-22.

[40] Matthew Conover. "Analysis of the Windows Vista Security Model" (PDF). Symantec Corporation. Retrieved 2007-10-08.

[41] "Browser Security: Lessons from Google Chrome".

[42] "Report malicious software (URL) to Google".

[43] "Google Safe Browsing".

[44] "5 Ways to Secure Your Web Browser". ZoneAlarm.

[45] "Adblock Plus Will Soon Block Fewer Ads — SiliconFilter". Siliconfilter.com. Retrieved 2013-04-20.

6.4 Internet security

Internet security is a branch of computer security specifically related to the Internet, often involving browser security but also network security on a more general level as it applies to other applications or operating systems on a whole. Its objective is to establish rules and measures to use against attacks over the Internet.[1] The Internet represents an insecure channel for exchanging information leading to a high risk of intrusion or fraud, such as phishing.[2] Different methods have been used to protect the transfer of data, including encryption and from-the-ground-up engineering.[3]

6.4.1 Threats

Malicious software

A computer user can be tricked or forced into downloading software onto a computer that is of malicious intent. Such software comes in many forms, such as viruses, Trojan horses, spyware, and worms.

- Malware, short for malicious software, is any software used to disrupt computer operation, gather sensitive information, or gain access to private computer systems. Malware is defined by its malicious intent, acting against the requirements of the computer user, and does not include software that causes unintentional harm due to some deficiency. The term badware is sometimes used, and applied to both true (malicious) malware and unintentionally harmful software.
- A botnet is a network of zombie computers that have been taken over by a robot or bot that performs large-scale malicious acts for the creator of the botnet.
- Computer Viruses are programs that can replicate their structures or effects by infecting other files or structures on a computer. The common use of a virus is to take over a computer to steal data.
- Computer worms are programs that can replicate themselves throughout a computer network, performing malicious tasks throughout.
- Ransomware is a type of malware which restricts access to the computer system that it infects, and demands a ransom paid to the creator(s) of the malware in order for the restriction to be removed.
- Scareware is scam software with malicious payloads, usually of limited or no benefit, that are sold to consumers via certain unethical marketing practices. The selling approach uses social engineering to cause shock, anxiety, or the perception of a threat, generally directed at an unsuspecting user.
- Spyware refers to programs that surreptitiously monitor activity on a computer system and report that information to others without the user's consent.
- A Trojan horse, commonly known as a *Trojan*, is a general term for malicious software that pretends to be harmless, so that a user willingly allows it to be downloaded onto the computer.

Denial-of-service attacks

A denial-of-service attack (DoS attack) or distributed denial-of-service attack (DDoS attack) is an attempt to make a computer resource unavailable to its intended users. Although the means to carry out, motives for, and targets of a DoS attack may vary, it generally consists of the concerted efforts to prevent an Internet site or service from functioning efficiently or at all, temporarily or indefinitely. According to businesses who participated in an international business security survey, 25% of respondents experienced a DoS attack in 2007 and 16.8% experienced one in 2010.[4]

Phishing

Main article: Phishing

Phishing occurs when the attacker pretends to be a trustworthy entity, either via email or web page. Victims are directed to fake web pages, which are dressed to look legitimate, via spoof emails, instant messenger/social media or other avenues. Often tactics such as email spoofing are used to make emails appear to be from legitimate senders, or long complex subdomains hide the real website host.[5][6] Insurance group RSA said that phishing accounted for worldwide losses of $1.5 Billion in 2012.[7]

Application vulnerabilities

Main article: Application security

Applications used to access Internet resources may contain security vulnerabilities such as memory safety bugs or flawed authentication checks. The most severe of these bugs can give network attackers full control over the computer. Most security applications and suites are incapable of adequate defense against these kinds of attacks.[8]

6.4.2 Remedies

Network layer security

TCP/IP protocols may be secured with cryptographic methods and security protocols. These protocols include Secure Sockets Layer (SSL), succeeded by Transport Layer Security (TLS) for web traffic, Pretty Good Privacy (PGP) for email, and IPsec for the network layer security.

Internet Protocol Security (IPsec)

IPsec is designed to protect TCP/IP communication in a secure manner. It is a set of security extensions developed by the Internet Task Force (IETF). It provides security and authentication at the IP layer by transforming data using encryption. Two main types of transformation that form the basis of IPsec: the Authentication Header (AH) and ESP. These two protocols provide data integrity, data origin authentication, and anti-replay service. These protocols can be used alone or in combination to provide the desired set of security services for the Internet Protocol (IP) layer.

The basic components of the IPsec security architecture are described in terms of the following functionalities:

- Security protocols for AH and ESP
- Security association for policy management and traffic processing
- Manual and automatic key management for the Internet key exchange (IKE)
- Algorithms for authentication and encryption

The set of security services provided at the IP layer includes access control, data origin integrity, protection against replays, and confidentiality. The algorithm allows these sets to work independently without affecting other parts of the implementation. The IPsec implementation is operated in a host or security gateway environment giving protection to IP traffic.

Security token

Some online sites offer customers the ability to use a six-digit code which randomly changes every 30–60 seconds on a security token. The keys on the security token have built in mathematical computations and manipulate numbers based on the current time built into the device. This means that every thirty seconds there is only a certain array of numbers possible which would be correct to validate access to the online account. The website that the user is logging into would be made aware of that devices' serial number and would know the computation and correct time built into the device to verify that the number given is indeed one of the handful of six-digit numbers that works in that given 30-60 second cycle. After 30–60 seconds the device will present a new random six-digit number which can log into the website.[9]

Electronic mail security

Background Email messages are composed, delivered, and stored in a multiple step process, which starts with the message's composition. When the user finishes composing the message and sends it, the message is transformed into a standard format: an RFC 2822 formatted message. Afterwards, the message can be transmitted. Using a network connection, the mail client, referred to as a mail user agent (MUA), connects to a mail transfer agent (MTA) operating on the mail server. The mail client then provides the sender's identity to the server. Next, using the mail server commands, the client sends the recipient list to the mail server. The client then supplies the message. Once the mail server receives and processes the message, several events occur: recipient server identification, connection establishment, and message transmission. Using Domain Name System (DNS) services, the sender's mail server determines the mail server(s) for the recipient(s). Then, the server opens up a connection(s) to the recipient mail server(s) and sends the message employing a process similar to that used by the originating client, delivering the message to the recipient(s).

Pretty Good Privacy (PGP) Pretty Good Privacy provides confidentiality by encrypting messages to be transmitted or data files to be stored using an encryption algorithm such as Triple DES or CAST-128. Email messages can be protected by using cryptography in various ways, such as the following:

- Signing an email message to ensure its integrity and confirm the identity of its sender.
- Encrypting the body of an email message to ensure its confidentiality.
- Encrypting the communications between mail servers to protect the confidentiality of both message body and message header.

The first two methods, message signing and message body encryption, are often used together; however, encrypting the transmissions between mail servers is typically used only when two organizations want to protect emails regularly sent between each other. For example, the organizations could establish a virtual private network (VPN) to encrypt the communications between their mail servers over the Internet.[10] Unlike methods that can only encrypt a message body, a VPN can encrypt entire messages, including email header information such as senders, recipients, and subjects. In some cases, organizations may need to protect header information. However, a VPN solution alone cannot provide a message signing mechanism, nor can it provide protection for email messages along the entire route from sender to recipient.

Multipurpose Internet Mail Extensions (MIME) MIME transforms non-ASCII data at the sender's site to Network Virtual Terminal (NVT) ASCII data and delivers it to client's Simple Mail Transfer Protocol (SMTP) to be sent through the Internet.[11] The server SMTP at the receiver's side receives the NVT ASCII data and delivers it to MIME to be transformed back to the original non-ASCII data.

Message Authentication Code A Message authentication code (MAC) is a cryptography method that uses a secret key to encrypt a message. This method outputs a MAC value that can be decrypted by the receiver, using the same secret key used by the sender. The Message Authentication Code protects both a message's data integrity as well as its authenticity.[12]

Firewalls

A computer firewall controls access between networks. It generally consists of gateways and filters which vary from one firewall to another. Firewalls also screen network traffic and are able to block traffic that is dangerous. Firewalls act as the intermediate server between SMTP and Hypertext Transfer Protocol (HTTP) connections.[13]

Role of firewalls in web security Firewalls impose restrictions on incoming and outgoing Network packets to and from private networks. Incoming or outgoing traffic must pass through the firewall; only authorized traffic is allowed to pass through it. Firewalls create checkpoints between an internal private network and the public Internet, also known as *choke points*(borrowed from the identical military term of a combat limiting geographical feature). Firewalls can create choke points based on IP source and TCP port number. They can also serve as the platform for IPsec. Using tunnel mode capability, firewall can be used to implement VPNs. Firewalls can also limit network exposure by hiding the internal network system and information from the public Internet.

Types of firewall

Packet filter A packet filter is a first generation firewall that processes network traffic on a packet-by-packet basis. Its main job is to filter traffic from a remote IP host, so a router is needed to connect the internal network to the Internet. The router is known as a screening router, which screens packets leaving and entering the network.

Stateful packet inspection In a stateful firewall the circuit-level gateway is a proxy server that operates at the network level of an Open Systems Interconnection (OSI) model and statically defines what traffic will be allowed. Circuit proxies will forward Network packets (formatted unit of data) containing a given port number, if the port is permitted by the algorithm. The main advantage of a proxy server is its ability to provide Network Address Translation (NAT), which can hide the user's IP address from the Internet, effectively protecting all internal information from the Internet.

Application-level gateway An application-level firewall is a third generation firewall where a proxy server operates at the very top of the OSI model, the IP suite application level. A network packet is forwarded only if a connection is established using a known protocol. Application-level gateways are notable for analyzing entire messages rather than individual packets of data when the data are being sent or received.

Browser choice

Main article: Browser security

Web browser statistics tend to affect the amount a Web browser is exploited. For example, Internet Explorer 6, which used to own a majority of the Web browser market share,[14] is considered extremely insecure[15] because vulnerabilities were exploited due to its former popularity. Since browser choice is more evenly distributed (Internet Explorer at 28.5%, Firefox at 18.4%, Google Chrome at 40.8%, and so on)[14] and vulnerabilities are exploited in many different browsers.[16][17][18]

6.4.3 Internet security products

Antivirus

Antivirus software and Internet security programs can protect a programmable device from attack by detecting and eliminating viruses; Antivirus software was mainly shareware in the early years of the Internet, but there are now several free security applications on the Internet to choose from for all platforms.[19]

Password managers

A password manager is a software application that helps a user store and organize passwords. Password managers usually store passwords encrypted, requiring the user to create a master password; a single, ideally very strong password which grants the user access to their entire password database.[20]

Security suites

So called *security suites* were first offered for sale in 2003 (McAfee) and contain a suite of firewalls, anti-virus, anti-spyware and more.[21] They may now offer theft protection, portable storage device safety check, private Internet browsing, cloud anti-spam, a file shredder or make security-related decisions (answering popup windows) and several were free of charge[22] as of at least 2012.

6.4.4 See also

- Comparison of antivirus software
- Comparison of firewalls
- Cyberspace Electronic Security Act (in the US)
- *Firewalls and Internet Security* (book)
- Goatse Security
- Identity Driven Networking
- Internet Crime Complaint Center
- Internet safety
- Network security policy
- Outpost Security Suite
- Web literacy (Security)
- Usability of web authentication systems

6.4.5 References

[1] Gralla, Preston (2007). *How the Internet Works*. Indianapolis: Que Pub. ISBN 0-7897-2132-5.

[2] Rhee, M. Y. (2003). *Internet Security: Cryptographic Principles,Algorithms and Protocols*. Chichester: Wiley. ISBN 0-470-85285-2.

[3] An example of a completely re-engineered computer is the Librem laptop which uses components certified by web-security experts. It was launched after a crowd funding campaign in 2015.

[4] "Information Security: A Growing Need of Businesses and Industries Worldwide". *University of Alabama at Birmingham Business Program*. Retrieved 20 November 2014.

[5] Ramzan, Zulfikar (2010). "Phishing attacks and countermeasures". In Stamp, Mark & Stavroulakis, Peter. *Handbook of Information and Communication Security*. Springer. ISBN 9783642041174.

[6] Van der Merwe, A J, Loock, M, Dabrowski, M. (2005), Characteristics and Responsibilities involved in a Phishing Attack, Winter International Symposium on Information and Communication Technologies, Cape Town, January 2005.

[7] "2012 Global Losses From Phishing Estimated At $1.5 Bn". FirstPost. February 20, 2013. Retrieved December 21, 2014.

[8] "Improving Web Application Security: Threats and Countermeasures". *msdn.microsoft.com*. Retrieved 2016-04-05.

[9] Margaret Rouse (September 2005). "What is a security token?". SearchSecurity.com. Retrieved 2014-02-14.

[10] "Virtual Private Network". NASA. Retrieved 2014-02-14.

[11] Asgaut Eng (1996-04-10). "Network Virtual Terminal". The Norwegian Institute of Technology ppv.org. Retrieved 2014-02-14.

[12] "What Is a Message Authentication Code?". Wisegeek.com. Retrieved 2013-04-20.

[13] "Firewalls - Internet Security". sites.google.com. Retrieved 2016-06-30.

[14] "Browser Statistics". W3Schools.com. Retrieved 2011-08-10.

[15] Bradly, Tony. "It's Time to Finally Drop Internet Explorer 6". PCWorld.com. Retrieved 2010-11-09.

[16] Messmer, Ellen and NetworkWorld (2010-11-16). "Google Chrome Tops 'Dirty Dozen' Vulnerable Apps List". PCWorld.com. Retrieved 2010-11-09.

[17] Keizer, Greg (2009-07-15). "Firefox 3.5 Vulnerability Confirmed". PCWorld.com. Retrieved 2010-11-09.

[18] Skinner, Carrie-Ann. "Opera Plugs "Severe" Browser Hole". PC World.com. Archived from the original on May 20, 2009. Retrieved 2010-11-09.

[19] Larkin, Eric (2008-08-26). "Build Your Own Free Security Suite". Retrieved 2010-11-09.

[20] "USE A FREE PASSWORD MANAGER" (PDF). scsc-cbkk.org.

[21] Rebbapragada, Narasu. "All-in-one Security". PC World.com. Archived from the original on October 27, 2010. Retrieved 2010-11-09.

[22] "Free products for PC security". 2015-10-08.

6.4.6 External links

- National Institute of Standards and Technology (NIST.gov) - Information Technology portal with links to computer- and cyber security
- National Institute of Standards and Technology (NIST.gov) -Computer Security Resource Center - Guidelines on Electronic Mail Security, version 2
- The Internet Engineering Task Force.org - UK organization -IP Authentication Header 1998
- The Internet Engineering Task Force.org - UK organization -Encapsulating Security Payload
- Wireless Safety.org - Up to date info on security threats, news stories, and step by step tutorials
- PwdHash Stanford University - Firefox & IE browser extensions that transparently convert a user's password into a domain-specific password.
- Internet security.net - by JC Montejo & Goio Miranda (free security programs), est 2007.
- Internet and Data Security Guide UK anonymous membership site
- Cybertelecom.org Security - surveying federal Internet security work
- DSL Reports.com- Broadband Reports, FAQs and forums on Internet security, est 1999
- FBI Safe Online Surfing Internet Challenge - Cyber Safety for Young Americans (FBI)

6.5 Mobile security

This article is about security threats to mobile devices. For using mobile devices for secure system access, see Computer security § Hardware protection mechanisms.

Mobile security or **mobile phone security** has become increasingly important in mobile computing. Of particular concern is the security of personal and business information now stored on smartphones.

More and more users and businesses employ smartphones as communication tools, but also as a means of planning and organizing their work and private life. Within companies, these technologies are causing profound changes in the organization of information systems and therefore they have become the source of new risks. Indeed, smartphones collect and compile an increasing amount of sensitive information to which access must be controlled to protect the privacy of the user and the intellectual property of the company.

All smartphones, as computers, are preferred targets of attacks. These attacks exploit weaknesses related to smartphones that can come from means of communication like Short Message Service (SMS, aka text messaging), Multimedia Messaging Service (MMS), Wi-Fi networks, Bluetooth and GSM, the de facto global standard for mobile communications. There are also attacks that exploit software vulnerabilities from both the web browser and operating system. Finally, there are forms of malicious software that rely on the weak knowledge of average users.

Different security counter-measures are being developed and applied to smartphones, from security in different layers of software to the dissemination of information to end users. There are good practices to be observed at all levels, from design to use, through the development of operating systems, software layers, and downloadable apps.

6.5.1 Challenges of mobile security

Threats

A smartphone user is exposed to various threats when they use their phone. In just the last two quarters of 2012, the number of unique mobile threats grew by 261%, according to ABI Research.[1] These threats can disrupt the operation of the smartphone, and transmit or modify user data. For these reasons, the applications deployed there must guarantee privacy and integrity of the information they handle. In addition, since some apps could themselves be malware, their functionality and activities should be limited (for example, restricting the apps from accessing location infor-

mation via GPS, blocking access to the user's address book, preventing the transmission of data on the network, sending SMS messages that are billed to the user, etc.).

There are three prime targets for attackers:[2]

- **Data**: smartphones are devices for data management, therefore they may contain sensitive data like credit card numbers, authentication information, private information, activity logs (calendar, call logs);
- **Identity**: smartphones are highly customizable, so the device or its contents are associated with a specific person. For example, every mobile device can transmit information related to the owner of the mobile phone contract, and an attacker may want to steal the identity of the owner of a smartphone to commit other offenses;
- **Availability**: by attacking a smartphone one can limit access to it and deprive the owner of the service.

The source of these attacks are the same actors found in the non-mobile computing space:[2]

- **Professionals**, whether commercial or military, who focus on the three targets mentioned above. They steal sensitive data from the general public, as well as undertake industrial espionage. They will also use the identity of those attacked to achieve other attacks;
- **Thieves** who want to gain income through data or identities they have stolen. The thieves will attack many people to increase their potential income;
- **Black hat hackers** who specifically attack availability.[3] Their goal is to develop viruses, and cause damage to the device.[4] In some cases, hackers have an interest in stealing data on devices.
- **Grey hat hackers** who reveal vulnerabilities.[5] Their goal is to expose vulnerabilities of the device.[6] Grey hat hackers do not intend on damaging the device or stealing data.[7]

Consequences

When a smartphone is infected by an attacker, the attacker can attempt several things:

- The attacker can manipulate the smartphone as a zombie machine, that is to say, a machine with which the attacker can communicate and send commands which will be used to send unsolicited messages (spam) via sms or email;[8]
- The attacker can easily force the smartphone to make phone calls. For example, one can use the API (library that contains the basic functions not present in the smartphone) PhoneMakeCall by Microsoft, which collects telephone numbers from any source such as yellow pages, and then call them.[8] But the attacker can also use this method to call paid services, resulting in a charge to the owner of the smartphone. It is also very dangerous because the smartphone could call emergency services and thus disrupt those services;[8]
- A compromised smartphone can record conversations between the user and others and send them to a third party.[8] This can cause user privacy and industrial security problems;
- An attacker can also steal a user's identity, usurp their identity (with a copy of the user's sim card or even the telephone itself), and thus impersonate the owner. This raises security concerns in countries where smartphones can be used to place orders, view bank accounts or are used as an identity card;[8]
- The attacker can reduce the utility of the smartphone, by discharging the battery.[9] For example, they can launch an application that will run continuously on the smartphone processor, requiring a lot of energy and draining the battery. One factor that distinguishes mobile computing from traditional desktop PCs is their limited performance. Frank Stajano and Ross Anderson first described this form of attack, calling it an attack of "battery exhaustion" or "sleep deprivation torture";[10]
- The attacker can prevent the operation and/or starting of the smartphone by making it unusable.[11] This attack can either delete the boot scripts, resulting in a phone without a functioning OS, or modify certain files to make it unusable (e.g. a script that launches at startup that forces the smartphone to restart) or even embed a startup application that would empty the battery;[10]
- The attacker can remove the personal (photos, music, videos, etc.) or professional data (contacts, calendars, notes) of the user.[11]

6.5.2 Attacks based on communication

Attack based on SMS and MMS

Some attacks derive from flaws in the management of SMS and MMS.

Some mobile phone models have problems in managing binary SMS messages. It is possible, by sending an ill-formed

block, to cause the phone to restart, leading to denial of service attacks. If a user with a Siemens S55 received a text message containing a Chinese character, it would lead to a denial of service.[12] In another case, while the standard requires that the maximum size of a Nokia Mail address is 32 characters, some Nokia phones did not verify this standard, so if a user enters an email address over 32 characters, that leads to complete dysfunction of the e-mail handler and puts it out of commission. This attack is called “curse of silence”. A study on the safety of the SMS infrastructure revealed that SMS messages sent from the Internet can be used to perform a distributed denial of service (DDoS) attack against the mobile telecommunications infrastructure of a big city. The attack exploits the delays in the delivery of messages to overload the network.

Another potential attack could begin with a phone that sends an MMS to other phones, with an attachment. This attachment is infected with a virus. Upon receipt of the MMS, the user can choose to open the attachment. If it is opened, the phone is infected, and the virus sends an MMS with an infected attachment to all the contacts in the address book. There is a real world example of this attack: the virus Commwarrior[11] uses the address book and sends MMS messages including an infected file to recipients. A user installs the software, as received via MMS message. Then, the virus began to send messages to recipients taken from the address book.

Attacks based on communication networks

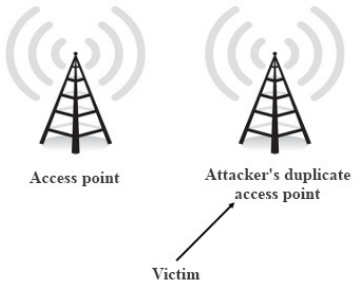
Attacks based on the GSM networks The attacker may try to break the encryption of the mobile network. The GSM network encryption algorithms belong to the family of algorithms called A5. Due to the policy of security through obscurity it has not been possible to openly test the robustness of these algorithms. There were originally two variants of the algorithm: A5/1 and A5/2 (stream ciphers), where the former was designed to be relatively strong, and the latter was designed to be weak on purpose to allow easy cryptanalysis and eavesdropping. ETSI forced some countries (typically outside Europe) to use A5/2. Since the encryption algorithm was made public, it was proved it was possible to break the encryption: A5/2 could be broken on the fly, and A5/1 in about 6 hours .[13] In July 2007, the 3GPP approved a change request to prohibit the implementation of A5/2 in any new mobile phones, which means that is has been decommissioned and is no longer implemented in mobile phones. Stronger public algorithms have been added to the GSM standard, the A5/3 and A5/4 (Block ciphers), otherwise known as KASUMI or UEA1[14] published by the ETSI. If the network does not support A5/1, or any other A5 algorithm implemented by the phone, then the base station can specify A5/0 which is the null-algorithm, whereby the radio traffic is sent unencrypted. Even in case mobile phones are able to use 3G or 4G which have much stronger encryption than 2G GSM, the base station can downgrade the radio communication to 2G GSM and specify A5/0 (no encryption) .[15] This is the basis for eavesdropping attacks on mobile radio networks using a fake base station commonly called an IMSI catcher.

In addition, tracing of mobile terminals is difficult since each time the mobile terminal is accessing or being accessed by the network, a new temporary identity (TMSI) is allocated to the mobile terminal. The TSMI is used as identity of the mobile terminal the next time it accesses the network. The TMSI is sent to the mobile terminal in encrypted messages.

Once the encryption algorithm of GSM is broken, the attacker can intercept all unencrypted communications made by the victim’s smartphone.

Attacks based on Wi-Fi See also: Wi-Fi § Network_security

An attacker can try to eavesdrop on Wi-Fi communications



Access Point spoofing

to derive information (e.g. username, password). This type of attack is not unique to smartphones, but they are very vulnerable to these attacks because very often the Wi-Fi is the only means of communication they have to access the internet. The security of wireless networks (WLAN) is thus an important subject. Initially wireless networks were secured by WEP keys. The weakness of WEP is a short encryption key which is the same for all connected clients. In addition, several reductions in the search space of the keys have been found by researchers. Now, most wireless networks are protected by the WPA security protocol. WPA is based on the "Temporal Key Integrity Protocol (TKIP)" which was designed to allow migration from WEP to WPA on the equipment already deployed. The major improvements in security are the dynamic encryption keys. For

small networks, the WPA is a "pre-shared key" which is based on a shared key. Encryption can be vulnerable if the length of the shared key is short. With limited opportunities for input (i.e. only the numeric keypad) mobile phone users might define short encryption keys that contain only numbers. This increases the likelihood that an attacker succeeds with a brute-force attack. The successor to WPA, called WPA2, is supposed to be safe enough to withstand a brute force attack.

As with GSM, if the attacker succeeds in breaking the identification key, it will be possible to attack not only the phone but also the entire network it is connected to.

Many smartphones for wireless LANs remember they are already connected, and this mechanism prevents the user from having to re-identify with each connection. However, an attacker could create a WIFI access point twin with the same parameters and characteristics as the real network. Using the fact that some smartphones remember the networks, they could confuse the two networks and connect to the network of the attacker who can intercept data if it does not transmit its data in encrypted form.[16][17][18]

Lasco is a worm that initially infects a remote device using the SIS file format.[19] SIS file format (Software Installation Script) is a script file that can be executed by the system without user interaction. The smartphone thus believes the file to come from a trusted source and downloads it, infecting the machine.[19]

Principle of Bluetooth-based attacks Main article: Bluetooth § Security
See also: Bluesnarfing and Bluebugging

Security issues related to Bluetooth on mobile devices have been studied and have shown numerous problems on different phones. One easy to exploit vulnerability: unregistered services do not require authentication, and vulnerable applications have a virtual serial port used to control the phone. An attacker only needed to connect to the port to take full control of the device.[20] Another example: a phone must be within reach and Bluetooth in discovery mode. The attacker sends a file via Bluetooth. If the recipient accepts, a virus is transmitted. For example: Cabir is a worm that spreads via Bluetooth connection.[11] The worm searches for nearby phones with Bluetooth in discoverable mode and sends itself to the target device. The user must accept the incoming file and install the program. After installing, the worm infects the machine.

6.5.3 Attacks based on vulnerabilities in software applications

Other attacks are based on flaws in the OS or applications on the phone.

Web browser

See also: Browser security

The mobile web browser is an emerging attack vector for mobile devices. Just as common Web browsers, mobile web browsers are extended from pure web navigation with widgets and plug-ins, or are completely native mobile browsers.

Jailbreaking the iPhone with firmware 1.1.1 was based entirely on vulnerabilities on the web browser.[21] As a result, the exploitation of the vulnerability described here underlines the importance of the Web browser as an attack vector for mobile devices. In this case, there was a vulnerability based on a stack-based buffer overflow in a library used by the web browser (Libtiff).

A vulnerability in the web browser for Android was discovered in October 2008. As the iPhone vulnerability above, it was due to an obsolete and vulnerable library. A significant difference with the iPhone vulnerability was Android's sandboxing architecture which limited the effects of this vulnerability to the Web browser process.

Smartphones are also victims of classic piracy related to the web: phishing, malicious websites, etc. The big difference is that smartphones do not yet have strong antivirus software available.

Operating system

See also: Operating_system § Security

Sometimes it is possible to overcome the security safeguards by modifying the operating system itself. As real-world examples, this section covers the manipulation of firmware and malicious signature certificates. These attacks are difficult.

In 2004, vulnerabilities in virtual machines running on certain devices were revealed. It was possible to bypass the bytecode verifier and access the native underlying operating system. The results of this research were not published in detail. The firmware security of Nokia's Symbian Platform Security Architecture (PSA) is based on a central configuration file called SWIPolicy. In 2008 it was possible to manipulate the Nokia firmware before it is installed, and in fact in some downloadable versions of it, this file was human

readable, so it was possible to modify and change the image of the firmware.[22] This vulnerability has been solved by an update from Nokia.

In theory smartphones have an advantage over hard drives since the OS files are in ROM, and cannot be changed by malware. However, in some systems it was possible to circumvent this: in the Symbian OS it was possible to overwrite a file with a file of the same name.[22] On the Windows OS, it was possible to change a pointer from a general configuration file to an editable file.

When an application is installed, the signing of this application is verified by a series of certificates. One can create a valid signature without using a valid certificate and add it to the list.[23] In the Symbian OS all certificates are in the directory: c:\resource\swicertstore\dat. With firmware changes explained above it is very easy to insert a seemingly valid but malicious certificate.

6.5.4 Attacks based on hardware vulnerabilities

Electromagnetic Waveforms

In 2015, researchers at the French government agency ANSSI demonstrated the capability to trigger the voice interface of certain smartphones remotely by using "specific electromagnetic waveforms".[24] The exploit took advantage of antenna-properties of headphone wires while plugged into the audio-output jacks of the vulnerable smartphones and effectively spoofed audio input to inject commands via the audio interface.[24]

Juice Jacking

See also: Juice_jacking

Juice Jacking is a method of physical or a hardware vulnerability specific to mobile platforms. Utilizing the dual purpose of the USB charge port, many devices have been susceptible to having data ex-filtrated from, or malware installed on to a mobile device by utilizing malicious charging kiosks set up in public places, or hidden in normal charge adapters.

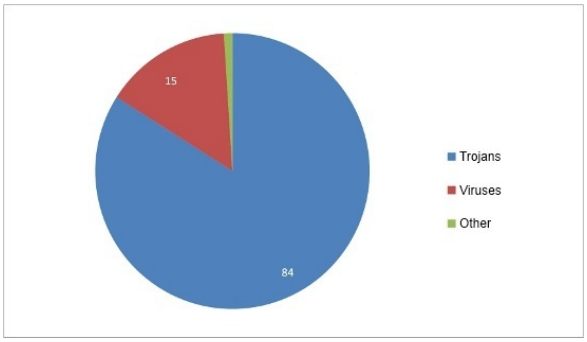
6.5.5 Password cracking

In 2010, researcher from the University of Pennsylvania investigated the possibility of cracking a device's password through a smudge attack (literally imaging the finger smudges on the screen to discern the user's password).[25] The researchers were able to discern the device password up to 68% of the time under certain conditions.[25] Outsiders may perform over-the-shoulder on victims, such as watching specific keystrokes or pattern gestures, to unlock device password or passcode.

6.5.6 Malicious software (malware)

See also: Malware

As smartphones are a permanent point of access to the internet (mostly on), they can be compromised as easily as computers with malware. A malware is a computer program that aims to harm the system in which it resides. Trojans, worms and viruses are all considered malware. A Trojan is a program that is on the smartphone and allows external users to connect discreetly. A worm is a program that reproduces on multiple computers across a network. A virus is malicious software designed to spread to other computers by inserting itself into legitimate programs and running programs in parallel. However, it must be said that the malware are far less numerous and important to smartphones as they are to computers.



Types of malware based on their number of smartphones in 2009

[26]

Nonetheless, recent studies show that the evolution of malware in smartphones have rocketed in the last few years posing a threat to analysis and detection.[27]

The three phases of malware attacks

Typically an attack on a smartphone made by malware takes place in 3 phases: the infection of a host, the accomplishment of its goal, and the spread of the malware to other systems. Malware often use the resources offered by the infected smartphones. It will use the output devices such as Bluetooth or infrared, but it may also use the address book or email address of the person to infect the user's acquain-

tances. The malware exploits the trust that is given to data sent by an acquaintance.

Infection Infection is the means used by the malware to get into the smartphone, it can either use one of the faults previously presented or may use the gullibility of the user. Infections are classified into four classes according to their degree of user interaction:[28]

Explicit permission the most benign interaction is to ask the user if it is allowed to infect the machine, clearly indicating its potential malicious behavior. This is typical behavior of a proof of concept malware.

Implied permission this infection is based on the fact that the user has a habit of installing software. Most trojans try to seduce the user into installing attractive applications (games, useful applications etc.) that actually contain malware.

Common interaction this infection is related to a common behavior, such as opening an MMS or email.

No interaction the last class of infection is the most dangerous. Indeed, a worm that could infect a smartphone and could infect other smartphones without any interaction would be catastrophic.

Accomplishment of its goal Once the malware has infected a phone it will also seek to accomplish its goal, which is usually one of the following: monetary damage, damage data and/or device, and concealed damage:[29]

Monetary damages the attacker can steal user data and either sell them to the same user, or sell to a third party.

Damage malware can partially damage the device, or delete or modify data on the device.

Concealed damage the two aforementioned types of damage are detectable, but the malware can also leave a backdoor for future attacks or even conduct wiretaps.

Spread to other systems Once the malware has infected a smartphone, it always aims to spread one way or another:[30]

- It can spread through proximate devices using Wi-Fi, Bluetooth and infrared;
- It can also spread using remote networks such as telephone calls or SMS or emails.

Examples of malware

Here are various malware that exist in the world of smartphones with a short description of each.

Viruses and trojans Main article: Mobile virus

- **Cabir** (also known as **Caribe**, **SybmOS/Cabir**, **Symbian/Cabir** and **EPOC.cabir**) is the name of a computer worm developed in 2004 that is designed to infect mobile phones running Symbian OS. It is believed to be the first computer worm that can infect mobile phones
- **Commwarrior**, found March 7, 2005, is the first worm that can infect many machines from MMS.[11] It is sent in the form of an archive file COMMWARRIOR.ZIP that contains a file COMMWARRIOR.SIS. When this file is executed, Commwarrior attempts to connect to nearby devices by Bluetooth or infrared under a random name. It then attempts to send MMS message to the contacts in the smartphone with different header messages for each person, who receive the MMS and often open them without further verification.
- **Phage** is the first Palm OS virus that was discovered.[11] It transfers to the Palm from a PC via synchronization. It infects all applications that are in the smartphone and it embeds its own code to function without the user and the system detecting it. All that the system will detect is that its usual applications are functioning.
- **RedBrowser** is a Trojan which is based on java.[11] The Trojan masquerades as a program called "RedBrowser" which allows the user to visit WAP sites without a WAP connection. During application installation, the user sees a request on their phone that the application needs permission to send messages. Therefore, if the user accepts, RedBrowser can send sms to paid call centers. This program uses the smartphone's connection to social networks (Facebook, Twitter, etc.) to get the contact information for the user's acquaintances (provided the required permissions have been given) and will send them messages.
- **WinCE.PmCryptic.A** is a malicious software on Windows Mobile which aims to earn money for its authors. It uses the infestation of memory cards that are inserted in the smartphone to spread more effectively.[31]

- **CardTrap** is a virus that is available on different types of smartphone, which aims to deactivate the system and third party applications. It works by replacing the files used to start the smartphone and applications to prevent them from executing.[32] There are different variants of this virus such as **Cardtrap.A** for SymbOS devices. It also infects the memory card with malware capable of infecting Windows.

- **Ghost Push** is a malicious software on Android OS which automatically root the android device and installs malicious applications directly to system partition then unroots the device to prevent users from removing the threat by master reset (The threat can be removed only by reflashing). It cripples the system resources, executes quickly, and harder to detect.

Ransomware Mobile ransomware is a type of malware that locks users out of their mobile devices in a pay-to-unlock-your-device ploy, it has grown by leaps and bounds as a threat category since 2014.[33] Specific to mobile computing platforms, users are often less security-conscious, particularly as it pertains to scrutinizing applications and web links trusting the native protection capability of the mobile device operating system. Mobile ransomware poses a significant threat to businesses reliant on instant access and availability of their proprietary information and contacts. The likelihood of a traveling businessman paying a ransom to unlock their device is significantly higher since they are at a disadvantage given inconveniences such as timeliness and less likely direct access to IT staff.

Spyware Main article: Spyware

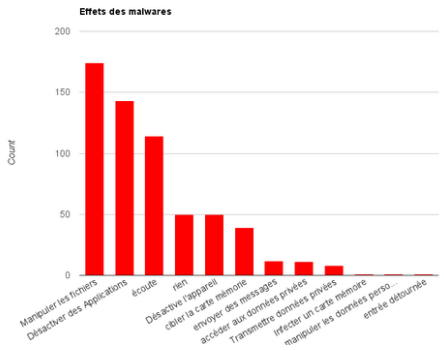
- **Flexispy** is an application that can be considered as a trojan, based on Symbian. The program sends all information received and sent from the smartphone to a Flexispy server. It was originally created to protect children and spy on adulterous spouses.[11]

Number of malware Below is a diagram which loads the different behaviors of smartphone malware in terms of their effects on smartphones:[26]

We can see from the graph that at least 50 malwares exhibit no negative behavior, except their ability to spread.[26]

Portability of malware across platforms

There is a multitude of malware. This is partly due to the variety of operating systems on smartphones. However attackers can also choose to make their malware target multiple platforms, and malware can be found which attacks an OS but is able to spread to different systems.



Effects of Malware

To begin with, malware can use runtime environments like Java virtual machine or the .NET Framework. They can also use other libraries present in many operating systems.[34] Other malware carry several executable files in order to run in multiple environments and they utilize these during the propagation process. In practice, this type of malware requires a connection between the two operating systems to use as an attack vector. Memory cards can be used for this purpose, or synchronization software can be used to propagate the virus.

6.5.7 Countermeasures

The security mechanisms in place to counter the threats described above are presented in this section. They are divided into different categories, as all do not act at the same level, and they range from the management of security by the operating system to the behavioral education of the user. The threats prevented by the various measures are not the same depending on the case. Considering the two cases mentioned above, in the first case one would protect the system from corruption by an application, and in the second case the installation of a suspicious software would be prevented.

Security in operating systems

The first layer of security within a smartphone is at the level of the operating system (OS). Beyond the usual roles of an operating system (e.g. resource management, scheduling processes) on a smartphone, it must also establish the protocols for introducing external applications and data without introducing risk.

A central idea found in the mobile operating systems is the idea of a sandbox. Since smartphones are currently being designed to accommodate many applications, they must put in place mechanisms to ensure these facilities are safe for themselves, for other applications and data on the system, and the user. If a malicious program manages to reach a device, it is necessary that the vulnerable area presented by the system be as small as possible. Sandboxing extends this idea to compartmentalize different processes, preventing them from interacting and damaging each other. Based on the history of operating systems, sandboxing has different implementations. For example, where iOS will focus on limiting access to its public API for applications from the App Store by default, Managed Open In allows you to restrict which apps can access which types of data. Android bases its sandboxing on its legacy of Linux and TrustedBSD.

The following points highlight mechanisms implemented in operating systems, especially Android.

Rootkit Detectors The intrusion of a rootkit in the system is a great danger in the same way as on a computer. It is important to prevent such intrusions, and to be able to detect them as often as possible. Indeed, there is concern that with this type of malicious program, the result could be a partial or complete bypass of the device security, and the acquisition of administrator rights by the attacker. If this happens, then nothing prevents the attacker from studying or disabling the safety features that were circumvented, deploying the applications they want, or disseminating a method of intrusion by a rootkit to a wider audience.[35][36] We can cite, as a defense mechanism, the Chain of trust in iOS. This mechanism relies on the signature of the different applications required to start the operating system, and a certificate signed by Apple. In the event that the signature checks are inconclusive, the device detects this and stops the boot-up.[37] If the Operating System is compromised due to Jailbreaking, root kit detection may not work if it is disabled by the Jailbreak method or software is loaded after Jailbreak disables Rootkit Detection.

Process isolation Android uses mechanisms of user process isolation inherited from Linux. Each application has a user associated with it, and a tuple (UID, GID). This approach serves as a sandbox: while applications can be malicious, they can not get out of the sandbox reserved for them by their identifiers, and thus cannot interfere with the proper functioning of the system. For example, since it is impossible for a process to end the process of another user, an application can thus not stop the execution of another.[35][38][39][40][41]

File permissions From the legacy of Linux, there are also filesystem permissions mechanisms. They help with sandboxing: a process can not edit any files it wants. It is therefore not possible to freely corrupt files necessary for the operation of another application or system. Furthermore, in Android there is the method of locking memory permissions. It is not possible to change the permissions of files installed on the SD card from the phone, and consequently it is impossible to install applications.[42][43][44]

Memory Protection In the same way as on a computer, memory protection prevents privilege escalation. Indeed, if a process managed to reach the area allocated to other processes, it could write in the memory of a process with rights superior to their own, with root in the worst case, and perform actions which are beyond its permissions on the system. It would suffice to insert function calls are authorized by the privileges of the malicious application.[41]

Development through runtime environments Software is often developed in high-level languages, which can control what is being done by a running program. For example, Java Virtual Machines continuously monitor the actions of the execution threads they manage, monitor and assign resources, and prevent malicious actions. Buffer overflows can be prevented by these controls.[45][46][41]

Security software

Above the operating system security, there is a layer of security software. This layer is composed of individual components to strengthen various vulnerabilities: prevent malware, intrusions, the identification of a user as a human, and user authentication. It contains software components that have learned from their experience with computer security; however, on smartphones, this software must deal with greater constraints (see limitations).

Antivirus and firewall An antivirus software can be deployed on a device to verify that it is not infected by a known threat, usually by signature detection software that detects malicious executable files. A firewall, meanwhile, can watch over the existing traffic on the network and ensure that a malicious application does not seek to communicate through it. It may equally verify that an installed application does not seek to establish suspicious communication, which may prevent an intrusion attempt.[47][48][49][36]

Visual Notifications In order to make the user aware of any abnormal actions, such as a call they did not initiate, one can link some functions to a visual notification

that is impossible to circumvent. For example, when a call is triggered, the called number should always be displayed. Thus, if a call is triggered by a malicious application, the user can see, and take appropriate action.

Turing test In the same vein as above, it is important to confirm certain actions by a user decision. The Turing test is used to distinguish between a human and a virtual user, and it often comes as a captcha.

Biometric identification Another method to use is biometrics.[50] Biometrics is a technique of identifying a person by means of their morphology(by recognition of the eye or face, for example) or their behavior (their signature or way of writing for example). One advantage of using biometric security is that users can avoid having to remember a password or other secret combination to authenticate and prevent malicious users from accessing their device. In a system with strong biometric security, only the primary user can access the smartphone.

Resource monitoring in the smartphone

When an application passes the various security barriers, it can take the actions for which it was designed. When such actions are triggered, the activity of a malicious application can be sometimes detected if one monitors the various resources used on the phone. Depending on the goals of the malware, the consequences of infection are not always the same; all malicious applications are not intended to harm the devices on which they are deployed. The following sections describe different ways to detect suspicious activity.[51]

Battery Some malware is aimed at exhausting the energy resources of the phone. Monitoring the energy consumption of the phone can be a way to detect certain malware applications.[35]

Memory usage Memory usage is inherent in any application. However, if one finds that a substantial proportion of memory is used by an application, it may be flagged as suspicious.

Network traffic On a smartphone, many applications are bound to connect via the network, as part of their normal operation. However, an application using a lot of bandwidth can be strongly suspected of attempting to communicate a lot of information, and disseminate data to many other devices. This observation only allows a suspicion, because some legitimate applications can be very resource-intensive in terms of network communications, the best example being streaming video.

Services One can monitor the activity of various services of a smartphone. During certain moments, some services should not be active, and if one is detected, the application should be suspected. For example, the sending of an SMS when the user is filming video: this communication does not make sense and is suspicious; malware may attempt to send SMS while its activity is masked.[52]

The various points mentioned above are only indications and do not provide certainty about the legitimacy of the activity of an application. However, these criteria can help target suspicious applications, especially if several criteria are combined.

Network surveillance

Network traffic exchanged by phones can be monitored. One can place safeguards in network routing points in order to detect abnormal behavior. As the mobile's use of network protocols is much more constrained than that of a computer, expected network data streams can be predicted (e.g. the protocol for sending an SMS), which permits detection of anomalies in mobile networks.

Spam filters As is the case with email exchanges, we can detect a spam campaign through means of mobile communications (SMS, MMS). It is therefore possible to detect and minimize this kind of attempt by filters deployed on network infrastructure that is relaying these messages.

Encryption of stored or transmitted information Because it is always possible that data exchanged can be intercepted, communications, or even information storage, can rely on encryption to prevent a malicious entity from using any data obtained during communications. However, this poses the problem of key exchange for encryption algorithms, which requires a secure channel.

Telecom network monitoring The networks for SMS and MMS exhibit predictable behavior, and there is not as much liberty compared with what one can do with protocols such as TCP or UDP. This implies that one cannot predict the use made of the common protocols of the web; one might generate very little traffic by consulting simple pages, rarely, or generate heavy traffic

by using video streaming. On the other hand, messages exchanged via mobile phone have a framework and a specific model, and the user does not, in a normal case, have the freedom to intervene in the details of these communications. Therefore, if an abnormality is found in the flux of network data in the mobile networks, the potential threat can be quickly detected.

Manufacturer surveillance

In the production and distribution chain for mobile devices, it is the responsibility of manufacturers to ensure that devices are delivered in a basic configuration without vulnerabilities. Most users are not experts and many of them are not aware of the existence of security vulnerabilities, so the device configuration as provided by manufacturers will be retained by many users. Below are listed several points which manufacturers should consider.

Remove debug mode Phones are sometimes set in a debug mode during manufacturing, but this mode must be disabled before the phone is sold. This mode allows access to different features, not intended for routine use by a user. Due to the speed of development and production, distractions occur and some devices are sold in debug mode. This kind of deployment exposes mobile devices to exploits that utilize this oversight.[53][54]

Default settings When a smartphone is sold, its default settings must be correct, and not leave security gaps. The default configuration is not always changed, so a good initial setup is essential for users. There are, for example, default configurations that are vulnerable to denial of service attacks.[35][55]

Security audit of apps Along with smart phones, appstores have emerged. A user finds themselves facing a huge range of applications. This is especially true for providers who manage appstores because they are tasked with examining the apps provided, from different points of view (e.g. security, content). The security audit should be particularly cautious, because if a fault is not detected, the application can spread very quickly within a few days, and infect a significant number of devices.[35]

Detect suspicious applications demanding rights When installing applications, it is good to warn the user against sets of permissions that, grouped together, seem potentially dangerous, or at least suspicious. Frameworks like such as Kirin, on Android, attempt to detect and prohibit certain sets of permissions.[56]

Revocation procedures Along with appstores appeared a new feature for mobile apps: remote revocation. First developed by Android, this procedure can remotely and globally uninstall an application, on any device that has it. This means the spread of a malicious application that managed to evade security checks can be immediately stopped when the threat is discovered.[57][58]

Avoid heavily customized systems Manufacturers are tempted to overlay custom layers on existing operating systems, with the dual purpose of offering customized options and disabling or charging for certain features. This has the dual effect of risking the introduction of new bugs in the system, coupled with an incentive for users to modify the systems to circumvent the manufacturer's restrictions. These systems are rarely as stable and reliable as the original, and may suffer from phishing attempts or other exploits.

Improve software patch processes New versions of various software components of a smartphone, including operating systems, are regularly published. They correct many flaws over time. Nevertheless, manufacturers often do not deploy these updates to their devices in a timely fashion, and sometimes not at all. Thus, vulnerabilities persist when they could be corrected, and if they are not, since they are known, they are easily exploitable.[56]

User awareness

Much malicious behavior is allowed by the carelessness of the user. From simply not leaving the device without a password, to precise control of permissions granted to applications added to the smartphone, the user has a large responsibility in the cycle of security: to not be the vector of intrusion. This precaution is especially important if the user is an employee of a company that stores business data on the device. Detailed below are some precautions that a user can take to manage security on a smartphone.

A recent survey by internet security experts BullGuard showed a lack of insight into the rising number of malicious threats affecting mobile phones, with 53% of users claiming that they are unaware of security software for Smartphones. A further 21% argued that such protection was unnecessary, and 42% admitted it hadn't crossed their mind ("Using APA," 2011). These statistics show consumers are not concerned about security risks because they believe it is not a serious problem. The key here is to always remember smartphones are effectively handheld computers and are just as vulnerable.

Being skeptical A user should not believe everything that may be presented, as some information may be phishing or attempting to distribute a malicious application. It is therefore advisable to check the reputation of the application that they want to buy before actually installing it.[59]

Permissions given to applications The mass distribution of applications is accompanied by the establishment of different permissions mechanisms for each operating system. It is necessary to clarify these permissions mechanisms to users, as they differ from one system to another, and are not always easy to understand. In addition, it is rarely possible to modify a set of permissions requested by an application if the number of permissions is too great. But this last point is a source of risk because a user can grant rights to an application, far beyond the rights it needs. For example, a note taking application does not require access to the geolocation service. The user must ensure the privileges required by an application during installation and should not accept the installation if requested rights are inconsistent.[60][55][61]

Be careful Protection of a user's phone through simple gestures and precautions, such as locking the smartphone when it is not in use, not leaving their device unattended, not trusting applications, not storing sensitive data, or encrypting sensitive data that cannot be separated from the device.[62][63]

Ensure data Smartphones have a significant memory and can carry several gigabytes of data. The user must be careful about what data it carries and whether they should be protected. While it is usually not dramatic if a song is copied, a file containing bank information or business data can be more risky. The user must have the prudence to avoid the transmission of sensitive data on a smartphone, which can be easily stolen. Furthermore, when a user gets rid of a device, they must be sure to remove all personal data first.[64]

These precautions are measures that leave no easy solution to the intrusion of people or malicious applications in a smartphone. If users are careful, many attacks can be defeated, especially phishing and applications seeking only to obtain rights on a device.

Centralized storage of text messages

One form of mobile protection allows companies to control the delivery and storage of text messages, by hosting the messages on a company server, rather than on the sender or receiver's phone. When certain conditions are met, such as an expiration date, the messages are deleted.[65]

Limitations of certain security measures

The security mechanisms mentioned in this article are to a large extent inherited from knowledge and experience with computer security. The elements composing the two device types are similar, and there are common measures that can be used, such as antivirus and firewall. However, the implementation of these solutions is not necessarily possible or at least highly constrained within a mobile device. The reason for this difference is the technical resources offered by computers and mobile devices: even though the computing power of smartphones is becoming faster, they have other limitations than their computing power.

- **Single-task system:** Some operating systems, including some still commonly used, are single-tasking. Only the foreground task is executed. It is difficult to introduce applications such as antivirus and firewall on such systems, because they could not perform their monitoring while the user is operating the device, when there would be most need of such monitoring.
- **Energy autonomy:** A critical one for the use of a smartphone is energy autonomy. It is important that the security mechanisms not consume battery resources, without which the autonomy of devices will be affected dramatically, undermining the effective use of the smartphone.
- **Network** Directly related to battery life, network utilization should not be too high. It is indeed one of the most expensive resources, from the point of view of energy consumption. Nonetheless, some calculations may need to be relocated to remote servers in order to preserve the battery. This balance can make implementation of certain intensive computation mechanisms a delicate proposition.[66]

Furthermore, it should be noted that it is common to find that updates exist, or can be developed or deployed, but this is not always done. One can, for example, find a user who does not know that there is a newer version of the operating system compatible with the smartphone, or a user may discover known vulnerabilities that are not corrected until the end of a long development cycle, which allows time to exploit the loopholes.[54]

Next Generation of mobile security

There is expected to be four mobile environments that will make up the security framework:[67]

Rich operating system In this category will fall tradicional Mobile OS like Android, iOS, Symbian OS or Windows Phone. They will provide the traditional functionaity and security of an OS to the applications.

Secure Operating System (Secure OS) A secure kernel which will run in parallel with a fully featured Rich OS, on the same processor core. It will include drivers for the Rich OS ("normal world") to communicate with the secure kernel ("secure world"). The trusted infrastructure could include interfaces like the display or keypad to regions of PCI-E address space and memories.

Trusted Execution Environment (TEE) Made up of hardware and software. It helps in the control of access rights and houses sensitive applications, which need to be isolated from the Rich OS. It effectively acts as a firewall between the "normal world" and "secure world".

Secure Element (SE) The SE consists of tamper resistant hardware and associated software. It can provide high levels of security and work in tandem with the TEE. The SE will be mandatory for hosting proximity payment applications or official electronic signatures.

6.5.8 See also

- Browser security
- Computer security
- Information security
- Mobile Malware
- Mobile secure gateway
- Phone hacking
- Telephone tapping
- Wireless Public Key Infrastructure (WPKI)
- Wireless security

6.5.9 Notes

[1] BYOD and Increased Malware Threats Help Driving Billion Dollar Mobile Security Services Market in 2013, ABI Research

[2] Bishop 2004.

[3] Olson, Parmy. "Your smartphone is hackers' next big target". CNN. Retrieved August 26, 2013.

[4] (PDF) http://www.gov.mu/portal/sites/cert/files/Guide%20on%20Protection%20Against%20Hacking.pdf. Missing or empty |title= (help)

[5] Lemos, Robert. "New laws make hacking a black-and-white choice". CNET News.com. Retrieved September 23, 2002.

[6] McCaney, Kevin. "'Unknowns' hack NASA, Air Force, saying 'We're here to help'". Retrieved May 7, 2012.

[7] Bilton 2010.

[8] Guo, Wang & Zhu 2004, p. 3.

[9] Dagon, Martin & Starder 2004, p. 12.

[10] Dixon & Mishra 2010, p. 3.

[11] Töyssy & Helenius 2006, p. 113.

[12] Siemens 2010, p. 1.

[13] Gendrullis 2008, p. 266.

[14] European Telecommunications Standards Institute 2011, p. 1.

[15] Jøsang, Miralabé & Dallot 2015.

[16] Roth, Polak & Rieffel 2008, p. 220.

[17] Gittleson, Kim (28 March 2014) Data-stealing Snoopy drone unveiled at Black Hat BBC News, Technology, Retrieved 29 March 2014

[18] Wilkinson, Glenn (25 September 2012) Snoopy: A distributed tracking and profiling framework Sensepost, Retrieved 29 March 2014

[19] Töyssy & Helenius 2006, p. 27.

[20] Mulliner 2006, p. 113.

[21] Dunham, Abu Nimeh & Becher 2008, p. 225.

[22] Becher 2009, p. 65.

[23] Becher 2009, p. 66.

[24] Kasmi C, Lopes Esteves J (13 August 2015). "IEMI Threats for Information Security: Remote Command Injection on Modern Smartphones". *IEEE Transactions on Electromagnetic Compatibility*. doi:10.1109/TEMC.2015.2463089. Lay summary – *WIRED* (14 October 2015).

[25] Aviv, Adam J.; Gibson, Katherine; Mossop, Evan; Blaze, Matt; Smith, Jonathan M. *Smudge Attacks on Smartphone Touch Screens* (PDF). 4th USENIX Workshop on Offensive Technologies.

[26] Schmidt et al. 2009a, p. 3.

[27] Suarez-Tangil, Guillermo; Juan E. Tapiador; Pedro Peris-Lopez; Arturo Ribagorda (2014). "Evolution, Detection and Analysis of Malware in Smart Devices" (PDF). *IEEE Communications Surveys & Tutorials*.

[28] Becher 2009, p. 87.

[29] Becher 2009, p. 88.

[30] Mickens & Noble 2005, p. 1.

[31] Raboin 2009, p. 272.

[32] Töyssy & Helenius 2006, p. 114.

[33] Haas, Peter D. (2015-01-01). "Ransomware goes mobile: An analysis of the threats posed by emerging methods". UTICA COLLEGE.

[34] Becher 2009, p. 91-94.

[35] Becher 2009, p. 12.

[36] Schmidt, Schmidt & Clausen 2008, p. 5-6.

[37] Halbronn & Sigwald 2010, p. 5-6.

[38] Ruff 2011, p. 127.

[39] Hogben & Dekker 2010, p. 50.

[40] Schmidt, Schmidt & Clausen 2008, p. 50.

[41] Shabtai et al. 2009, p. 10.

[42] Becher 2009, p. 31.

[43] Schmidt, Schmidt & Clausen 2008, p. 3.

[44] Shabtai et al. 2009, p. 7-8.

[45] Pandya 2008, p. 15.

[46] Becher 2009, p. 22.

[47] Becher et al. 2011, p. 96.

[48] Becher 2009, p. 128.

[49] Becher 2009, p. 140.

[50] Thirumathyam & Derawi 2010, p. 1.

[51] Schmidt, Schmidt & Clausen 2008, p. 7-12.

[52] Becher 2009, p. 126.

[53] Becher et al. 2011, p. 101.

[54] Ruff 2011, p. 11.

[55] Hogben & Dekker 2010, p. 45.

[56] Becher 2009, p. 13.

[57] Becher 2009, p. 34.

[58] Ruff 2011, p. 7.

[59] Hogben & Dekker 2010, p. 46-48.

[60] Ruff 2011, p. 7-8.

[61] Shabtai et al. 2009, p. 8-9.

[62] Hogben & Dekker 2010, p. 43.

[63] Hogben & Dekker 2010, p. 47.

[64] Hogben & Dekker 2010, p. 43-45.

[65] Charlie Sorrel (2010-03-01). "TigerText Deletes Text Messages From Receiver's Phone". *Wired*. Archived from the original on 2010-10-17. Retrieved 2010-03-02.

[66] Becher 2009, p. 40.

[67] http://www.insidesecure.com/Markets-solutions/Payment-and-Mobile-Banking/Mobile-Security

6.5.10 References

Books

- Bishop, Matt (2004). *Introduction to Computer Security*. Addison Wesley Professional. ISBN 978-0-321-24744-5.
- Dunham, Ken; Abu Nimeh, Saeed; Becher, Michael (2008). *Mobile Malware Attack and Defense*. Syngress Media. ISBN 978-1-59749-298-0.
- Rogers, David (2013). *Mobile Security: A Guide for Users*. Copper Horse Solutions Limited. ISBN 978-1-291-53309-5.

Articles

- Becher, Michael (2009). *Security of Smartphones at the Dawn of Their Ubiquitousness* (PDF) (Dissertation). Mannheim University.
- Becher, Michael; Freiling, Felix C.; Hoffmann, Johannes; Holz, Thorsten; Uellenbeck, Sebastian; Wolf, Christopher (May 2011). *Mobile Security Catching Up? Revealing the Nuts and Bolts of the Security of Mobile Devices* (PDF). 2011 IEEE Symposium on Security and Privacy. pp. 96–111. doi:10.1109/SP.2011.29. ISBN 978-1-4577-0147-4.
- Bilton, Nick (26 July 2010). "Hackers With Enigmatic Motives Vex Companies". *The New York Times*. p. 5.
- Cai, Fangda; Chen, Hao; Wu, Yuanyi; Zhang, Yuan (2015). *AppCracker: Widespread Vulnerabilities in Userand Session Authentication in Mobile Apps* (PDF) (Dissertation). University of California, Davis.
- Crussell, Johnathan; Gibler, Clint; Chen, Hao (2012). *Attack of the Clones: Detecting Cloned Applications on Android Markets* (PDF) (Dissertation). University of California, Davis.

- Dagon, David; Martin, Tom; Starder, Thad (October–December 2004). "Mobile Phones as Computing Devices: The Viruses are Coming!". *IEEE Pervasive Computing* **3** (4): 11. doi:10.1109/MPRV.2004.21.
- Dixon, Bryan; Mishra, Shivakant (June–July 2010). *On and Rootkit and Malware Detection in Smartphones* (PDF). 2010 International Conference on Dependable Systems and Networks Workshops (DSN-W). ISBN 978-1-4244-7728-9.
- Gendrullis, Timo (November 2008). *A real-world attack breaking A5/1 within hours*. Proceedings of CHES '08. Springer. pp. 266–282. doi:10.1007/978-3-540-85053-3_17.
- Guo, Chuanxiong; Wang, Helen; Zhu, Wenwu (November 2004). *Smart-Phone Attacks and Defenses* (PDF). ACM SIGCOMM HotNets. Association for Computing Machinery, Inc. Retrieved March 31, 2012.
- Halbronn, Cedric; Sigwald, John (2010). *Vulnerabilities & iPhone Security Model* (PDF). HITB SecConf 2010.
- Hogben, Giles; Dekker, Marnix (December 2010). "Smartphones: Information security Risks, Opportunities and Recommendations for users". ENISA.
- Jøsang, Audun; Miralabé, Laurent; Dallot, Léonard (2015). *It's not a bug, it's a feature: 25 years of mobile network insecurity* (PDF). European Conference on Cyber Warfare and Security (ECCWS 2015).
- Mickens, James W.; Noble, Brian D. (2005). *Modeling epidemic spreading in mobile environments*. WiSe '05 Proceedings of the 4th ACM workshop on Wireless security. Association for Computing Machinery, Inc. pp. 77–86. doi:10.1145/1080793.1080806.
- Mulliner, Collin Richard (2006). *Security of Smart Phones* (PDF) (M.Sc. thesis). University of California, Santa Barbara.
- Pandya, Vaibhav Ranchhoddas (2008). *Iphone Security Analysis* (PDF) (Thesis). San Jose State University.
- Raboin, Romain (December 2009). *La sécurité des smartphones* (PDF). Symposium sur la sécurité des technologies de l'information et des communications 2009. SSTIC09 (in French).
- Racic, Radmilo; Ma, Denys; Chen, Hao (2006). *Exploiting MMS Vulnerabilities to Stealthily Exhaust Mobile Phone's Battery* (PDF) (Dissertation). University of California, Davis.
- Roth, Volker; Polak, Wolfgang; Rieffel, Eleanor (2008). *Simple and Effective Defense Against Evil Twin Access Points*. ACM SIGCOMM HotNets. doi:10.1145/1352533.1352569. ISBN 978-1-59593-814-5.
- Ruff, Nicolas (2011). *Sécurité du système Android* (PDF). Symposium sur la sécurité des technologies de l'information et des communications 2011. SSTIC11 (in French).
- Ruggiero, Paul; Foote, Jon. *Cyber Threats to Mobile Phones* (PDF) (thesis). US-CERT.
- Schmidt, Aubrey-Derrick; Schmidt, Hans-Gunther; Clausen, Jan; Yüksel, Kamer Ali; Kiraz, Osman; Camtepe, Ahmet; Albayrak, Sahin (October 2008). *Enhancing Security of Linux-based Android Devices* (PDF). Proceedings of 15th International Linux Kongress.
- Schmidt, Aubrey-Derrick; Schmidt, Hans-Gunther; Batyuk, Leonid; Clausen, Jan Hendrik; Camtepe, Seyit Ahmet; Albayrak, Sahin (April 2009a). *Smartphone Malware Evolution Revisited: Android Next Target?* (PDF). 4th International Conference on Malicious and Unwanted Software (MALWARE). ISBN 978-1-4244-5786-1. Retrieved 2010-11-30.
- Shabtai, Asaf; Fledel, Yuval; Kanonov, Uri; Elovici, Yuval; Dolev, Shlomi (2009). "Google Android: A State-of-the-Art Review of Security Mechanisms". *CoRR*. arXiv:0912.5101v1.
- Thirumathyam, Rubathas; Derawi, Mohammad O. (2010). *Biometric Template Data Protection in Mobile Device Using Environment XML-database*. 2010 2nd International Workshop on Security and Communication Networks (IWSCN). ISBN 978-1-4244-6938-3.
- Töyssy, Sampo; Helenius, Marko (2006). "About malicious software in smartphones". *Journal in Computer Virology* (Springer Paris) **2** (2): 109–119. doi:10.1007/s11416-006-0022-0. Retrieved 2010-11-30.

Websites

- European Telecommunications Standards Institute (2011). "3GPP Confidentiality and Integrity Algorithms & UEA1 UIA1". Archived from the original on 12 May 2012.
- Siemens (2010). "Series M Siemens SMS DoS Vulnerability".

6.5.11 Further reading

- CIGREF (October 2010). "Sécurisation de la mobilité" (PDF) (in French).
- Chong, Wei Hoo (November 2007). *iDEN Smartphone Embedded Software Testing* (PDF). Fourth International Conference on Information Technology, 2007. ITNG '07. doi:10.1109/ITNG.2007.103. ISBN 0-7695-2776-0.
- Jansen, Wayne; Scarfone, Karen (October 2008). "Guidelines on Cell Phone and PDA Security: Recommendations of the National Institute of Standards and Technology" (PDF). National Institute of Standards and Technology. Retrieved April 21, 2012.
- Lee, Sung-Min; Suh, Sang-bum; Jeong, Bokdeuk; Mo, Sangdok (January 2008). *A Multi-Layer Mandatory Access Control Mechanism for Mobile Devices Based on Virtualization*. 5th IEEE Consumer Communications and Networking Conference, 2008. CCNC 2008. doi:10.1109/ccnc08.2007.63. ISBN 978-1-4244-1456-7. Archived from the original on May 16, 2013.
- Li, Feng; Yang, Yinying; Wu, Jie (March 2010). *CPMC: An Efficient Proximity Malware Coping Scheme in Smartphone-based Mobile Networks* (PDF). INFOCOM, 2010 Proceedings IEEE. doi:10.1109/INFCOM.2010.5462113.
- Ni, Xudong; Yang, Zhimin; Bai, Xiaole; Champion, Adam C.; Xuan, Dong (October 2009). *Distribute: Differentiated User Access Control on Smartphones* (PDF). 6th IEEE International Conference on Mobile Adhoc and Periodic Sensor Systems, 2009. MASS '09. ISBN 978-1-4244-5113-5. Archived from the original (PDF) on July 9, 2014.
- Ongtang, Machigar; McLaughlin, Stephen; Enck, William; Mcdaniel, Patrick (December 2009). *Semantically Rich Application-Centric Security in Android* (PDF). Annual Computer Security Applications Conference, 2009. ACSAC '09. ISSN 1063-9527.
- Schmidt, Aubrey-Derrick; Bye, Rainer; Schmidt, Hans-Gunther; Clausen, Jan; Kiraz, Osman; Yüksel, Kamer A.; Camtepe, Seyit A.; Albayrak, Sahin (2009b). *Static Analysis of Executables for Collaborative Malware Detection on Android* (PDF). IEEE International Conference Communications, 2009. ICC '09. ISSN 1938-1883.
- Yang, Feng; Zhou, Xuehai; Jia, Gangyong; Zhang, Qiyuan (2010). *A Non-cooperative Game Approach for Intrusion Detection Systems in Smartphone systems*. 8th Annual Communication Networks and Services Research Conference. doi:10.1109/CNSR.2010.24. ISBN 978-1-4244-6248-3. Archived from the original on May 16, 2013.

6.5.12 External links

- How Applications Lead your mobile to be Hacked - Ujjwal Sahay
- Android iOS Mobile Security Review
- Best Antivirus for Android

6.6 Network security

Network security consists of the policies and practices adopted to prevent and monitor unauthorized access, misuse, modification, or denial of a computer network and network-accessible resources. Network security involves the authorization of access to data in a network, which is controlled by the network administrator. Users choose or are assigned an ID and password or other authenticating information that allows them access to information and programs within their authority. Network security covers a variety of computer networks, both public and private, that are used in everyday jobs; conducting transactions and communications among businesses, government agencies and individuals. Networks can be private, such as within a company, and others which might be open to public access. Network security is involved in organizations, enterprises, and other types of institutions. It does as its title explains: It secures the network, as well as protecting and overseeing operations being done. The most common and simple way of protecting a network resource is by assigning it a unique name and a corresponding password.

6.6.1 Network security concepts

Network security starts with authenticating, commonly with a username and a password. Since this requires just one detail authenticating the user name—i.e., the password—this is sometimes termed one-factor authentication. With two-factor authentication, something the user 'has' is also used (e.g., a security token or 'dongle', an ATM card, or a mobile phone); and with three-factor authentication, something the user 'is' also used (e.g., a fingerprint or retinal scan).

Once authenticated, a firewall enforces access policies such as what services are allowed to be accessed by the network users.[1] Though effective to prevent unauthorized access, this component may fail to check potentially harmful content such as computer worms or Trojans being transmitted

over the network. Anti-virus software or an intrusion prevention system (IPS)[2] help detect and inhibit the action of such malware. An anomaly-based intrusion detection system may also monitor the network like wireshark traffic and may be logged for audit purposes and for later high-level analysis.

Communication between two hosts using a network may be encrypted to maintain privacy.

Honeypots, essentially decoy network-accessible resources, may be deployed in a network as surveillance and early-warning tools, as the honeypots are not normally accessed for legitimate purposes. Techniques used by the attackers that attempt to compromise these decoy resources are studied during and after an attack to keep an eye on new exploitation techniques. Such analysis may be used to further tighten security of the actual network being protected by the honeypot. A honeypot can also direct an attacker's attention away from legitimate servers. A honeypot encourages attackers to spend their time and energy on the decoy server while distracting their attention from the data on the real server. Similar to a honeypot, a honeynet is a network set up with intentional vulnerabilities. Its purpose is also to invite attacks so that the attacker's methods can be studied and that information can be used to increase network security. A honeynet typically contains one or more honeypots.[3]

6.6.2 Security management

Security management for networks is different for all kinds of situations. A home or small office may only require basic security while large businesses may require high-maintenance and advanced software and hardware to prevent malicious attacks from hacking and spamming.

Types of Attacks

Networks are subject to attacks from malicious sources. Attacks can be from two categories: "Passive" when a network intruder intercepts data traveling through the network, and "Active" in which an intruder initiates commands to disrupt the network's normal operation.[4]

Types of attacks include:[5]

- Passive
  - Network
    - Wiretapping
    - Port scanner
    - Idle scan
- Active
  - Denial-of-service attack
  - DNS spoofing
  - Man in the middle
  - ARP poisoning
  - VLAN hopping
  - Smurf attack
  - Buffer overflow
  - Heap overflow
  - Format string attack
  - SQL injection
  - Phishing
  - Cross-site scripting
  - CSRF
  - Cyber-attack

6.6.3 See also

- Cloud computing security
- Crimeware
- Cyber security standards
- Data Loss Prevention
- Greynet
- Identity Based Security
- Information Leak Prevention
- Metasploit Project
- Mobile security
- Netsentron
- Network Security Toolkit
- TCP Gender Changer
- TCP sequence prediction attack
- Timeline of hacker history
- Wireless LAN Security
- Dynamic secrets
- Low Orbit Ion Cannon
- High Orbit Ion Cannon

6.6.4 References

[1] A Role-Based Trusted Network Provides Pervasive Security and Compliance - interview with Jayshree Ullal, senior VP of Cisco

[2] Dave Dittrich, *Network monitoring/Intrusion Detection Systems (IDS)*, University of Washington.

[3] ""Honeypots, Honeynets"". Honeypots.net. 2007-05-26. Retrieved 2011-12-09.

[4] Wright, Joe; Jim Harmening (2009) "15" Computer and Information Security Handbook Morgan Kaufmann Publications Elsevier Inc p. 257

[5] http://www.cnss.gov/Assets/pdf/cnssi_4009.pdf

6.6.5 Further reading

- *Case Study: Network Clarity*, SC Magazine 2014
- Cisco. (2011). What is network security?. Retrieved from cisco.com
- pcmag.com
- Security of the Internet (*The Froehlich/Kent Encyclopedia of Telecommunications vol. 15*. Marcel Dekker, New York, 1997, pp. 231–255.)
- *Introduction to Network Security*, Matt Curtin.
- *Security Monitoring with Cisco Security MARS*, Gary Halleen/Greg Kellogg, Cisco Press, Jul. 6, 2007.
- *Self-Defending Networks: The Next Generation of Network Security*, Duane DeCapite, Cisco Press, Sep. 8, 2006.
- *Security Threat Mitigation and Response: Understanding CS-MARS*, Dale Tesch/Greg Abelar, Cisco Press, Sep. 26, 2006.
- *Securing Your Business with Cisco ASA and PIX Firewalls*, Greg Abelar, Cisco Press, May 27, 2005.
- *Deploying Zone-Based Firewalls*, Ivan Pepelnjak, Cisco Press, Oct. 5, 2006.
- *Network Security: PRIVATE Communication in a PUBLIC World*, Charlie Kaufman | Radia Perlman | Mike Speciner, Prentice-Hall, 2002. ISBN .
- *Network Infrastructure Security*, Angus Wong and Alan Yeung, Springer, 2009.

6.7 Defensive computing

Defensive computing is a form of practice for computer users to help reduce the risk of computing problems, by avoiding dangerous computing practices. The primary goal of this method of computing is to be able to anticipate and prepare for potentially problematic situations prior to their occurrence, despite any adverse conditions of a computer system or any mistakes made by other users. This can be achieved through adherence to a variety of general guidelines, as well as the practice of specific computing techniques.

Strategies for defensive computing could be divided into two categories, network security and the backup and restoration of data.

6.7.1 Network security

Users put their computers at risk when accessing the Internet and other networks. The use of either of these allows others to gain access to a user's system and important information. By implementing certain strategies, defensive users aim to reduce the risk associated with network access.

Firewall

A firewall is a collection of security measures that protects a computer from harmful inbound and outbound traffic on the Internet and prevents the unauthorized access of computer systems.[1] These security measures are integrated into the form of special software that runs autonomously either on individual computer systems, or externally through built in software within routers and modems.

Not all firewall software will protect computers from sending unauthorized or harmful outbound traffic. An important defensive computing strategy is to seek and implement quality firewall software that filters both inbound and outbound traffic.[2]

Anti-malware software

A basic strategy for all defensive computer users is to install and use anti-malware software. Firewalls may not completely protect a computer. Malicious software may be able to get through a firewall and onto a system. Anti-Malware such as anti-virus, anti-phishing and email filtering software offer some protection against harmful software that reside within a computer. The amount of malicious software available over the Internet is steadily increasing.[3] It is important for defensive users to use to anti-malware that is both

effective and easily updated in order to combat new strains of malicious software that are developed.[2]

Skepticism

An important aspect of defensive computing is for users to be skeptical of the data to which they have access via the Internet.[4] Malicious software can exist in a multitude of different forms and many are misleading to general computer users and even some anti-malware software. Defensive users think critically about the information they can access, to reduce their chances of downloading and spreading malicious software. Strategies include scanning email attachments prior to opening them and manually filtering suspicious emails from inboxes. Users should be aware of persuasive subject lines and headings in emails from any address, as they may actually contain malicious software or spam, which can mislead users into false advertisement resulting in identity theft.[2] Defensive users can scan files they download prior to opening them and can also configure their computers to show file extensions, revealing potentially dangerous files that appear harmless.[4] Skepticism can also be applied to the websites visited by users. As with emails, users can be lead to false advertisements. Also, malicious software can unknowingly be downloaded and infect a computer, just by visiting a certain website.

6.7.2 Backup and recovery procedures

Despite the efforts of a defensive computer user, the loss of important data can occur due to malware, power outages, equipment failure and general misuse. Although the loss of data cannot be completely prevented, defensive users can take steps to minimize the amount of data lost and restore systems to their previous state.

Backup of files

A defensive strategy against unintentional data loss is the regular backup of important files. Users can make multiple copies of important data and store them either on the same computer or on another device such as a compact disc or an external hard drive.[5] Users can also upload important files to the Internet, provided they have access to Internet storage services.

Restoration

Some operating systems give users the option of performing a procedure that restores a computer to a predetermined state. If no option is available, a user can obtain the appropriate restoration software for their system. In the event of a system failure or a serious case of data loss, a user can restore any lost or changed files and remove any malicious files that did not previously exist.[5]

6.7.3 Good practices for protecting data

- Regularly backup important files, documents and emails.
- Do not use the administrator account for day-to-day activities.
- Keep software up-to-date with the latest versions.
- Keep antivirus and antispyware up-to-date with latest versions.
- Use different passwords
- Disable auto run feature from USB flash drives. Some viruses, specially worms, spread automatically through USB flash drives [6]
- Always connect to the Internet behind a firewall
- When in doubt, throw it out

6.7.4 See also

- Stopping e-mail abuse
- Phishing
- Computer insecurity
- End-user computing
- Network security
- Computer worm
- Computer security

6.7.5 References

[1] http://www.cs.unm.edu/~{}treport/tr/02-12/firewall.pdf, A History and Survey of Network Firewalls

[2] http://news.cnet.com/8301-13554_3-9923976-33.html, The Pillars of Defensive Computing

[3] http://www.washingtonpost.com/wp-dyn/content/article/2008/03/19/AR2008031901439.html, Antivirus Firms Scrambling to Keep Up

[4] http://www.melbpc.org.au/pcupdate/2206/2206article6.htm, How To Protect Yourself From Virus Infection

[5] http://www.microsoft.com/protect/yourself/data/what.mspx, How to Decide what Data to Back Up

[6] http://news.cnet.com/8301-13554_3-10027754-33.html, Be safer than NASA: Disable autorun

6.7.6 External links

- Defensive Computing Blog by Michael Horowitz at ComputerWorld.com
- Defensive computing priorities by Michael Horowitz December 2009

6.8 Firewall (computing)

In computing, a **firewall** is a network security system that monitors and controls the incoming and outgoing network traffic based on predetermined security rules.[1] A firewall typically establishes a barrier between a trusted, secure internal network and another outside network, such as the Internet, that is assumed not to be secure or trusted.[2] Firewalls are often categorized as either *network firewalls* or *host-based firewalls*. Network firewalls are a software appliance running on general purpose hardware or hardware-based firewall computer appliances that filter traffic between two or more networks. Host-based firewalls provide a layer of software on one host that controls network traffic in and out of that single machine.[3][4] Firewall appliances may also offer other functionality to the internal network they protect such as acting as a DHCP[5][6] or VPN[7][8][9][10] server for that network.[11][12]

6.8.1 History

The term *firewall* originally referred to a wall intended to confine a fire or potential fire within a building.[13] Later uses refer to similar structures, such as the metal sheet separating the engine compartment of a vehicle or aircraft from the passenger compartment.

Firewall technology emerged in the late 1980s when the Internet was a fairly new technology in terms of its global use and connectivity.[14] The predecessors to firewalls for network security were the routers used in the late 1980s:[15]

- Clifford Stoll's discovery of German spies tampering with his system[15]
- Bill Cheswick's "Evening with Berferd" 1992, in which he set up a simple electronic "jail" to observe an attacker[15]
- In 1988, an employee at the NASA Ames Research Center in California sent a memo by email to his colleagues[16] that read, "We are currently under attack from an Internet VIRUS! It has hit Berkeley, UC San Diego, Lawrence Livermore, Stanford, and NASA Ames."
- The Morris Worm spread itself through multiple vulnerabilities in the machines of the time. Although it was not malicious in intent, the Morris Worm was the first large scale attack on Internet security; the online community was neither expecting an attack nor prepared to deal with one.[17]

First generation: packet filters

Screenshot of Gufw: The firewall shows its settings for incoming and outgoing traffic.

The first type of firewall was the packet filter which looks at network addresses and ports of the packet and determines if that packet should be allowed or blocked.[18] The first paper published on firewall technology was in 1988, when engineers from Digital Equipment Corporation (DEC) developed filter systems known as packet filter firewalls. This fairly basic system was the first generation of what is now a highly involved and technical internet security feature. At AT&T Bell Labs, Bill Cheswick and Steve Bellovin were continuing their research in packet filtering and developed a working model for their own company based on their original first generation architecture.[19]

Packet filters act by inspecting the "packets" which are transferred between computers on the Internet. If a packet

does not match the packet filter's set of filtering rules, the packet filter will drop (silently discard) the packet or reject it (discard it, and send "error responses" to the source). Conversely, if the packet matches one or more of the programmed filters, the packet is allowed to pass. This type of packet filtering pays no attention to whether a packet is part of an existing stream of traffic (i.e. it stores no information on connection "state"). Instead, it filters each packet based only on information contained in the packet itself (most commonly using a combination of the packet's source and destination address, its protocol, and, for TCP and UDP traffic, the port number). TCP and UDP protocols constitute most communication over the Internet, and because TCP and UDP traffic by convention uses well known ports for particular types of traffic, a "stateless" packet filter can distinguish between, and thus control, those types of traffic (such as web browsing, remote printing, email transmission, file transfer), unless the machines on each side of the packet filter are both using the same non-standard ports.[20]

Packet filtering firewalls work mainly on the first three layers of the OSI reference model, which means most of the work is done between the network and physical layers, with a little bit of peeking into the transport layer to figure out source and destination port numbers.[21] When a packet originates from the sender and filters through a firewall, the device checks for matches to any of the packet filtering rules that are configured in the firewall and drops or rejects the packet accordingly. When the packet passes through the firewall, it filters the packet on a protocol/port number basis (GSS). For example, if a rule in the firewall exists to block telnet access, then the firewall will block the TCP protocol for port number 23.[22]

Second generation: "stateful" filters

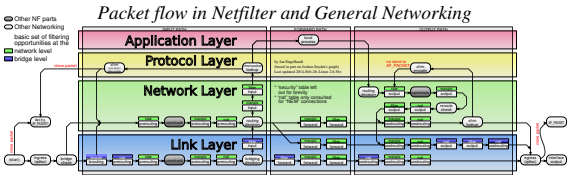
Main article: Stateful firewall

From 1989–1990 three colleagues from AT&T Bell Laboratories, Dave Presotto, Janardan Sharma, and Kshitij Nigam, developed the second generation of firewalls, calling them Circuit-level gateways.[23]

Second-generation firewalls perform the work of their first-generation predecessors but operate up to layer 4 (transport layer) of the OSI model. This is achieved by retaining packets until enough information is available to make a judgement about its state.[24] Known as stateful packet inspection, it records all connections passing through it and determines whether a packet is the start of a new connection, a part of an existing connection, or not part of any connection.[25] Though static rules are still used, these rules can now contain *connection state* as one of their test criteria.

Certain denial-of-service attacks bombard the firewall with thousands of fake connection packets in an attempt to overwhelm it by filling its connection state memory.[26]

Third generation: application layer



Flow of network packets through Netfilter, a Linux kernel module

Main article: Application level firewall

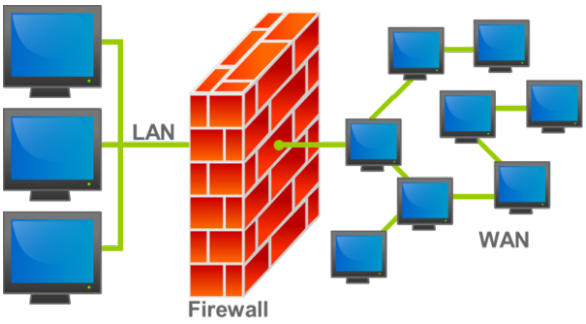
Marcus Ranum, Wei Xu, and Peter Churchyard developed an Application Firewall known as Firewall Toolkit (FWTK). In June 1994, Wei Xu extended the FWTK with the Kernel enhancement of IP filter and socket transparent. This was known as the first transparent Application firewall, released as a commercial product of Gauntlet firewall at Trusted Information Systems. Gauntlet firewall was rated one of the number one firewalls during 1995–1998.

The key benefit of application layer filtering is that it can "understand" certain applications and protocols (such as File Transfer Protocol (FTP), Domain Name System (DNS), or Hypertext Transfer Protocol (HTTP)). This is useful as it is able to detect if an unwanted protocol is attempting to bypass the firewall on an allowed port, or detect if a protocol is being abused in any harmful way. As of 2012, the so-called next-generation firewall (NGFW) is nothing more than the "widen" or "deepen" inspection at application-stack. For example, the existing deep packet inspection functionality of modern firewalls can be extended to include

- Intrusion prevention systems (IPS)
- User identity integration (by binding user IDs to IP or MAC addresses for "reputation"); and/or
- Web application firewall (WAF). WAF attacks may be implemented in the tool "WAF Fingerprinting utilizing timing side channels" (WAFFle)[27]

6.8.2 Types

Firewalls vary in type depending on where communication originates, where it is intercepted, and the state of communication being traced.[28]



An illustration of where a firewall would be located in a network

Network layer or packet filters

Network layer firewalls, also called packet filters, operate at a relatively low level of the TCP/IP protocol stack, not allowing packets to pass through the firewall unless they match the established rule set. The firewall administrator may define the rules; or default rules may apply. The term "packet filter" originated in the context of BSD operating systems.

Network layer firewalls generally fall into two subcategories, stateful and stateless. Stateful firewalls maintain context about active sessions, and use that "state information" to speed packet processing. Any existing network connection can be described by several properties, including source and destination IP address, UDP or TCP ports, and the current stage of the connection's lifetime (including session initiation, handshaking, data transfer, or completion connection). If a packet does not match an existing connection, it will be evaluated according to the ruleset for new connections. If a packet matches an existing connection based on comparison with the firewall's state table, it will be allowed to pass without further processing.

Stateless firewalls require less memory, and can be faster for simple filters that require less time to filter than to look up a session. They may also be necessary for filtering stateless network protocols that have no concept of a session. However, they cannot make more complex decisions based on what stage communications between hosts have reached.

Newer firewalls can filter traffic based on many packet attributes like source IP address, source port, destination IP address or port, destination service like WWW or FTP. They can filter based on protocols, TTL values, netblock of originator, of the source, and many other attributes.

Commonly used packet filters on various versions of Unix are *IPFilter* (various), *ipfw* (FreeBSD/Mac OS X), *NPF* (NetBSD), *PF* (OpenBSD, and some other BSDs), *iptables*/*ipchains* (Linux).

Application-layer

Main article: Application layer firewall

Application-layer firewalls work on the application level of the TCP/IP stack (i.e., all browser traffic, or all telnet or ftp traffic), and may intercept all packets traveling to or from an application. They block other packets (usually dropping them without acknowledgment to the sender).

On inspecting all packets for improper content, firewalls can restrict or prevent outright the spread of networked computer worms and trojans. The additional inspection criteria can add extra latency to the forwarding of packets to their destination.

Application firewalls function by determining whether a process should accept any given connection. Application firewalls accomplish their function by hooking into socket calls to filter the connections between the application layer and the lower layers of the OSI model. Application firewalls that hook into socket calls are also referred to as socket filters. Application firewalls work much like a packet filter but application filters apply filtering rules (allow/block) on a per process basis instead of filtering connections on a per port basis. Generally, prompts are used to define rules for processes that have not yet received a connection. It is rare to find application firewalls not combined or used in conjunction with a packet filter.[29]

Also, application firewalls further filter connections by examining the process ID of data packets against a ruleset for the local process involved in the data transmission. The extent of the filtering that occurs is defined by the provided ruleset. Given the variety of software that exists, application firewalls only have more complex rulesets for the standard services, such as sharing services. These per process rulesets have limited efficacy in filtering every possible association that may occur with other processes. Also, these per process rulesets cannot defend against modification of the process via exploitation, such as memory corruption exploits. Because of these limitations, application firewalls are beginning to be supplanted by a new generation of application firewalls that rely on mandatory access control (MAC), also referred to as sandboxing, to protect vulnerable services.[30]

Proxies

Main article: Proxy server

A proxy server (running either on dedicated hardware or as software on a general-purpose machine) may act as a firewall by responding to input packets (connection requests,

for example) in the manner of an application, while blocking other packets. A proxy server is a gateway from one network to another for a specific network application, in the sense that it functions as a proxy on behalf of the network user.[2]

Proxies make tampering with an internal system from the external network more difficult and misuse of one internal system would not necessarily cause a security breach exploitable from outside the firewall (as long as the application proxy remains intact and properly configured). Conversely, intruders may hijack a publicly reachable system and use it as a proxy for their own purposes; the proxy then masquerades as that system to other internal machines. While use of internal address spaces enhances security, crackers may still employ methods such as IP spoofing to attempt to pass packets to a target network.

Network address translation

Main article: Network address translation

Firewalls often have network address translation (NAT) functionality, and the hosts protected behind a firewall commonly have addresses in the "private address range", as defined in RFC 1918. Firewalls often have such functionality to hide the true address of protected hosts. Originally, the NAT function was developed to address the limited number of IPv4 routable addresses that could be used or assigned to companies or individuals as well as reduce both the amount and therefore cost of obtaining enough public addresses for every computer in an organization. Although NAT on its own is not considered a security feature, hiding the addresses of protected devices has become an often used percepted defense against network reconnaissance.[31]

6.8.3 See also

- Access control list
- Bastion host
- Comparison of firewalls
- Computer security
- Distributed firewall
- Egress filtering
- End-to-end principle
- Firewall pinhole
- Firewalls and Internet Security
- Golden Shield Project
- Guard (information security)
- Identity-based security
- IP fragmentation attacks
- List of Unix-like router or firewall distributions
- Mangled packet
- Next-Generation Firewall
- Personal firewall
- Screened-subnet firewall
- Unidirectional network
- Unified threat management
- Virtual firewall
- Vulnerability scanner
- Windows Firewall

6.8.4 References

[1] Boudriga, Noureddine (2010). *Security of mobile communications*. Boca Raton: CRC Press. pp. 32–33. ISBN 0849379423.

[2] Oppliger, Rolf (May 1997). "Internet Security: FIREWALLS and BEYOND". *Communications of the ACM* **40** (5): 94. doi:10.1145/253769.253802.

[3] Vacca, John R. (2009). *Computer and information security handbook*. Amsterdam: Elsevier. p. 355. ISBN 9780080921945.

[4] "What is Firewall?". Retrieved 2015-02-12.

[5] "Firewall as a DHCP Server and Client". *Palo Alto Networks*. Retrieved 2016-02-08.

[6] "DHCP". *www.shorewall.net*. Retrieved 2016-02-08.

[7] "What is a VPN Firewall? - Definition from Techopedia". *Techopedia.com*. Retrieved 2016-02-08.

[8] "VPNs and Firewalls". *technet.microsoft.com*. Retrieved 2016-02-08.

[9] "VPN and Firewalls (Windows Server)". *Resources and Tools for IT Professionals | TechNet*.

[10] "Configuring VPN connections with firewalls".

[11] Andrés, Steven; Kenyon, Brian; Cohen, Jody Marc; Johnson, Nate; Dolly, Justin (2004). Birkholz, Erik Pack, ed. *Security Sage's Guide to Hardening the Network Infrastructure*. Rockland, MA: Syngress. pp. 94–95. ISBN 9780080480831.

[12] Naveen, Sharanya. "Firewall". Retrieved 7 June 2016.

[13] Canavan, John E. (2001). *Fundamentals of Network Security* (1st ed.). Boston, MA: Artech House. p. 212. ISBN 9781580531764.

[14] Liska, Allan (Dec 10, 2014). *Building an Intelligence-Led Security Program*. Syngress. p. 3. ISBN 0128023708.

[15] Ingham, Kenneth; Forrest, Stephanie (2002). "A History and Survey of Network Firewalls" (PDF). Retrieved 2011-11-25.

[16] Firewalls by Dr.Talal Alkharobi

[17] RFC 1135 The Helminthiasis of the Internet

[18] Peltier, Justin; Peltier, Thomas R. (2007). *Complete Guide to CISM Certification*. Hoboken: CRC Press. p. 210. ISBN 9781420013252.

[19] Ingham, Kenneth; Forrest, Stephanie (2002). "A History and Survey of Network Firewalls" (PDF). p. 4. Retrieved 2011-11-25.

[20] TCP vs. UDP By Erik Rodriguez

[21] William R. Cheswick, Steven M. Bellovin, Aviel D. Rubin (2003). "Google Books Link". *Firewalls and Internet Security: repelling the wily hacker*

[22] Aug 29, 2003 Virus may elude computer defenses by Charles Duhigg, Washington Post

[23] *Proceedings of National Conference on Recent Developments in Computing and Its Applications, August 12–13, 2009*. I.K. International Pvt. Ltd. 2009-01-01. Retrieved 2014-04-22.

[24] Conway, Richard (204). *Code Hacking: A Developer's Guide to Network Security*. Hingham, Massachusetts: Charles River Media. p. 281. ISBN 1-58450-314-9.

[25] Andress, Jason (May 20, 2014). *The Basics of Information Security: Understanding the Fundamentals of InfoSec in Theory and Practice* (2nd ed.). Elsevier Science. ISBN 9780128008126.

[26] Chang, Rocky (October 2002). "Defending Against Flooding-Based Distributed Denial-of-Service Attacks: A Tutorial". *IEEE Communications Magazine* **40** (10): 42–43. doi:10.1109/mcom.2002.1039856.

[27] "WAFFle: Fingerprinting Filter Rules of Web Application Firewalls". 2012.

[28] "Firewalls". MemeBridge. Retrieved 13 June 2014.

[29] "Software Firewalls: Made of Straw? Part 1 of 2". Symantec Connect Community. 2010-06-29. Retrieved 2014-03-28.

[30] "Auto Sandboxing". Comodo Inc. Retrieved 2014-08-28.

[31] "Advanced Security: Firewall". Microsoft. Retrieved 2014-08-28.

6.8.5 External links

- Internet Firewalls: Frequently Asked Questions, compiled by Matt Curtin, Marcus Ranum and Paul Robertson.
- Firewalls Aren't Just About Security - Cyberoam Whitepaper focusing on Cloud Applications Forcing Firewalls to Enable Productivity.
- Evolution of the Firewall Industry - Discusses different architectures and their differences, how packets are processed, and provides a timeline of the evolution.
- A History and Survey of Network Firewalls - provides an overview of firewalls at the various ISO levels, with references to the original papers where first firewall work was reported.
- Software Firewalls: Made of Straw? Part 1 and Software Firewalls: Made of Straw? Part 2 - a technical view on software firewall design and potential weaknesses

6.9 Intrusion detection system

An **intrusion detection system** (**IDS**) is a device or software application that monitors network or system activities for malicious activities or policy violations and produces electronic reports to a management station. IDS come in a variety of "flavors" and approach the goal of detecting suspicious traffic in different ways. There are network based (NIDS) and host based (HIDS) intrusion detection systems. NIDS is a network security system focusing on the attacks that come from the inside of the network (authorized users). Some systems may attempt to stop an intrusion attempt but this is neither required nor expected of a monitoring system. Intrusion detection and prevention systems (IDPS) are primarily focused on identifying possible incidents, logging information about them, and reporting attempts. In addition, organizations use IDPSes for other purposes, such as identifying problems with security policies, documenting existing threats and deterring individuals from violating security policies. IDPSes have become a necessary addition to the security infrastructure of nearly every organization.[1]

IDPSes typically record information related to observed events, notify security administrators of important observed events and produce reports. Many IDPSes can also respond to a detected threat by attempting to prevent it from succeeding. They use several response techniques, which involve the IDPS stopping the attack itself, changing the secu-

rity environment (e.g. reconfiguring a firewall) or changing the attack's content.[1]

Intrusion prevention systems (**IPS**), also known as **intrusion detection and prevention systems** (**IDPS**), are network security appliances that monitor network or system activities for malicious activity. The main functions of intrusion prevention systems are to identify malicious activity, log information about this activity, report it and attempt to block or stop it.[2].

Intrusion prevention systems are considered extensions of intrusion detection systems because they both monitor network traffic and/or system activities for malicious activity. The main differences are, unlike intrusion detection systems, intrusion prevention systems are placed in-line and are able to actively prevent or block intrusions that are detected.[3]:273[4]:289 IPS can take such actions as sending an alarm, dropping detected malicious packets, resetting a connection or blocking traffic from the offending IP address.[5] An IPS also can correct cyclic redundancy check (CRC) errors, defragment packet streams, mitigate TCP sequencing issues, and clean up unwanted transport and network layer options.[3]:278[6].

6.9.1 Terminology

- **Alarm filtering:** The process of categorizing attack alerts produced from an IDS in order to distinguish false positives from actual attacks.[7]
- **Attacker** or **Intruder:** An entity which tries to find a way to gain unauthorized access to information, inflict harm or engage in other malicious activities.
- **Burglar Alarm:** A signal suggesting that a system has been or is being attacked.[7]
- **Clandestine user:** A person who acts as a supervisor and tries to use his privileges so as to avoid being captured.
- **Confidence value:** A value an organization places on an IDS based on past performance and analysis to help determine its ability to effectively identify an attack.[7]
- **Detection Rate:** The detection rate is defined as the number of intrusion instances detected by the system (True Positive) divided by the total number of intrusion instances present in the test set.[8]
- **False Alarm Rate:** defined as the number of 'normal' patterns classified as attacks (False Positive) divided by the total number of 'normal' patterns.[8]
- **False Negative:** When no alarm is raised when an attack has taken place.[7]
- **False Positive:** An event signaling an IDS to produce an alarm when no attack has taken place.[7]
- **Masquerader:** A person who attempts to gain unauthorized access to a system by pretending to be an authorized user. They are generally outside users.
- **Misfeasor:** They are commonly internal users and can be of two types:
  1. An authorized user with limited permissions.
  2. A user with full permissions and who misuses their powers.
- **Noise:** Data or interference that can trigger a false positive or obscure a true positive.[7]
- **Site policy:** Guidelines within an organization that control the rules and configurations of an IDS.[7]
- **Site policy awareness:** An IDS's ability to dynamically change its rules and configurations in response to changing environmental activity.[7]
- **True Negative:** An event when no attack has taken place and no detection is made.
- **True Positive:** A legitimate attack which triggers an IDS to produce an alarm.[7]

6.9.2 Classifications

Intrusion prevention systems can be classified into four different types:[2][9]

1. **Network-based intrusion prevention system (NIPS)**: monitors the entire network for suspicious traffic by analyzing protocol activity.
2. **Wireless intrusion prevention systems (WIPS)**: monitor a wireless network for suspicious traffic by analyzing wireless networking protocols.
3. **Network behavior analysis (NBA)**: examines network traffic to identify threats that generate unusual traffic flows, such as distributed denial of service (DDoS) attacks, certain forms of malware and policy violations.
4. **Host-based intrusion prevention system (HIPS)**: an installed software package which monitors a single host for suspicious activity by analyzing events occurring within that host.

Network intrusion detection systems

Network intrusion detection systems (NIDS) are placed at a strategic point or points within the network to monitor traffic to and from all devices on the network. It performs an analysis of passing traffic on the entire subnet, and matches the traffic that is passed on the subnets to the library of known attacks. Once an attack is identified, or abnormal behavior is sensed, the alert can be sent to the administrator. An example of an NIDS would be installing it on the subnet where firewalls are located in order to see if someone is trying to break into the firewall. Ideally one would scan all inbound and outbound traffic, however doing so might create a bottleneck that would impair the overall speed of the network. OPNET and NetSim are commonly used tools for simulation network intrusion detection systems. NID Systems are also capable of comparing signatures for similar packets to link and drop harmful detected packets which have a signature matching the records in the NIDS. When we classify the designing of the NIDS according to the system interactivity property, there are two types: on-line and off-line NIDS. On-line NIDS deals with the network in real time. It analyses the Ethernet packets and applies some rules, to decide if it is an attack or not. Off-line NIDS deals with stored data and passes it through some processes to decide if it is an attack or not.[10]

Host intrusion detection systems

Main article: Host-based intrusion detection system

Host intrusion detection systems (HIDS) run on individual hosts or devices on the network. A HIDS monitors the inbound and outbound packets from the device only and will alert the user or administrator if suspicious activity is detected. It takes a snapshot of existing system files and matches it to the previous snapshot. If the critical system files were modified or deleted, an alert is sent to the administrator to investigate. An example of HIDS usage can be seen on mission critical machines, which are not expected to change their configurations.

Intrusion detection systems can also be system-specific using custom tools and honeypots.

6.9.3 Passive and reactive systems

In a passive system, the intrusion detection system (IDS) sensor detects a potential security breach, logs the information and signals an alert on the console or owner. In a reactive system, also known as an intrusion prevention system (IPS), the IPS auto-responds to the suspicious activity by resetting the connection or by reprogramming the firewall to block network traffic from the suspected malicious source. The term **intrusion detection and prevention systems** (IDPS) is commonly used where this can happen automatically or at the command of an operator; systems that both “detect (alert)" and “prevent”.

6.9.4 Comparison with firewalls

Though they both relate to network security, an intrusion detection system (IDS) differs from a firewall in that a firewall looks outwardly for intrusions in order to stop them from happening. Firewalls limit access between networks to prevent intrusion and do not signal an attack from inside the network. An IDS evaluates a suspected intrusion once it has taken place and signals an alarm. An IDS also watches for attacks that originate from within a system. This is traditionally achieved by examining network communications, identifying heuristics and patterns (often known as signatures) of common computer attacks, and taking action to alert operators. A system that terminates connections is called an intrusion prevention system, and is another form of an application layer firewall.

6.9.5 Detection methods

The majority of intrusion prevention systems utilize one of three detection methods: signature-based, statistical anomaly-based, and stateful protocol analysis.[4]:301[11]

1. **Signature-Based Detection**: Signature based IDS monitors packets in the Network and compares with pre-configured and pre-determined attack patterns known as signatures.
2. **Statistical anomaly-based detection**: An IDS which is anomaly based will monitor network traffic and compare it against an established baseline. The baseline will identify what is “normal” for that network – what sort of bandwidth is generally used, what protocols are used that it may raise a False Positive alarm for a legitimate use of bandwidth if the baselines are not intelligently configured.[7]
3. **Stateful Protocol Analysis Detection**: This method identifies deviations of protocol states by comparing observed events with “predetermined profiles of generally accepted definitions of benign activity.”[4]

6.9.6 Limitations

- Noise can severely limit an intrusion detection system’s effectiveness. Bad packets generated from software bugs, corrupt DNS data, and local packets that

escaped can create a significantly high false-alarm rate.[12]

- It is not uncommon for the number of real attacks to be far below the number of false-alarms. Number of real attacks is often so far below the number of false-alarms that the real attacks are often missed and ignored.[12]
- Many attacks are geared for specific versions of software that are usually outdated. A constantly changing library of signatures is needed to mitigate threats. Outdated signature databases can leave the IDS vulnerable to newer strategies.[12]
- For signature-based IDSes there will be lag between a new threat discovery and its signature being applied to the IDS. During this lag time the IDS will be unable to identify the threat.[7]
- It cannot compensate for a weak identification and authentication mechanisms or for weaknesses in network protocols. When an attacker gains access due to weak authentication mechanism then IDS cannot prevent the adversary from any malpractise.
- Encrypted packets are not processed by the intrusion detection software. Therefore, the encrypted packet can allow an intrusion to the network that is undiscovered until more significant network intrusions have occurred.
- Intrusion detection software provides information based on the network address that is associated with the IP packet that is sent into the network. This is beneficial if the network address contained in the IP packet is accurate. However, the address that is contained in the IP packet could be faked or scrambled.
- Due to the nature of NIDS systems, and the need for them to analyse protocols as they are captured, NIDS systems can be susceptible to same protocol based attacks that network hosts may be vulnerable. Invalid data and TCP/IP stack attacks may cause an NIDS to crash.[13]

6.9.7 Evasion techniques

Main article: Intrusion detection system evasion techniques

There are a number of techniques which attackers are using, the following are considered 'simple' measures which can be taken to evade IDS:

- Fragmentation: by sending fragmented packets, the attacker will be under the radar and can easily bypass the detection system's ability to detect the attack signature.
- Avoiding defaults: The TCP port utilised by a protocol does not always provide an indication to the protocol which is being transported. For example, an IDS may expect to detect a trojan on port 12345. If an attacker had reconfigured it to use a different port the IDS may not be able to detect the presence of the trojan.
- Coordinated, low-bandwidth attacks: coordinating a scan among numerous attackers (or agents) and allocating different ports or hosts to different attackers makes it difficult for the IDS to correlate the captured packets and deduce that a network scan is in progress.
- Address spoofing/proxying: attackers can increase the difficulty of the ability of Security Administrators to determine the source of the attack by using poorly secured or incorrectly configured proxy servers to bounce an attack. If the source is spoofed and bounced by a server then it makes it very difficult for IDS to detect the origin of the attack.
- Pattern change evasion: IDSs generally rely on 'pattern matching' to detect an attack. By changing the data used in the attack slightly, it may be possible to evade detection. For example, an IMAP server may be vulnerable to a buffer overflow, and an IDS is able to detect the attack signature of 10 common attack tools. By modifying the payload sent by the tool, so that it does not resemble the data that the IDS expects, it may be possible to evade detection.

6.9.8 Development

One preliminary IDS concept consisted of a set of tools intended to help administrators review audit trails.[14] User access logs, file access logs, and system event logs are examples of audit trails.

Fred Cohen noted in 1984 that it is impossible to detect an intrusion in every case, and that the resources needed to detect intrusions grow with the amount of usage.

Dorothy E. Denning, assisted by Peter G. Neumann, published a model of an IDS in 1986 that formed the basis for many systems today.[15] Her model used statistics for anomaly detection, and resulted in an early IDS at SRI International named the Intrusion Detection Expert System (IDES), which ran on Sun workstations and could consider both user and network level data.[16] IDES had a dual approach with a rule-based Expert System to detect known types of intrusions plus a statistical anomaly detection component based on profiles of users, host systems, and target systems. Lunt proposed adding an Artificial neural network as a third component. She said all three components could then report to a resolver. SRI followed IDES in 1993

with the Next-generation Intrusion Detection Expert System (NIDES).[17]

The Multics intrusion detection and alerting system (MIDAS), an expert system using P-BEST and Lisp, was developed in 1988 based on the work of Denning and Neumann.[18] Haystack was also developed in that year using statistics to reduce audit trails.[19]

Wisdom & Sense (W&S) was a statistics-based anomaly detector developed in 1989 at the Los Alamos National Laboratory.[20] W&S created rules based on statistical analysis, and then used those rules for anomaly detection.

In 1990, the Time-based Inductive Machine (TIM) did anomaly detection using inductive learning of sequential user patterns in Common Lisp on a VAX 3500 computer.[21] The Network Security Monitor (NSM) performed masking on access matrices for anomaly detection on a Sun-3/50 workstation.[22] The Information Security Officer's Assistant (ISOA) was a 1990 prototype that considered a variety of strategies including statistics, a profile checker, and an expert system.[23] ComputerWatch at AT&T Bell Labs used statistics and rules for audit data reduction and intrusion detection.[24]

Then, in 1991, researchers at the University of California, Davis created a prototype Distributed Intrusion Detection System (DIDS), which was also an expert system.[25] The Network Anomaly Detection and Intrusion Reporter (NADIR), also in 1991, was a prototype IDS developed at the Los Alamos National Laboratory's Integrated Computing Network (ICN), and was heavily influenced by the work of Denning and Lunt.[26] NADIR used a statistics-based anomaly detector and an expert system.

The Lawrence Berkeley National Laboratory announced Bro in 1998, which used its own rule language for packet analysis from libpcap data.[27] Network Flight Recorder (NFR) in 1999 also used libpcap.[28] APE was developed as a packet sniffer, also using libpcap, in November, 1998, and was renamed Snort one month later. APE has since become the world's largest used IDS/IPS system with over 300,000 active users.[29]

The Audit Data Analysis and Mining (ADAM) IDS in 2001 used tcpdump to build profiles of rules for classifications.[30]

In 2003, Yongguang Zhang and Wenke Lee argue for the importance of IDS in networks with mobile nodes.[31]

6.9.9 Free intrusion detection systems

- ACARM-ng
- AIDE
- Bro NIDS
- Fail2ban
- OSSEC HIDS
- Prelude Hybrid IDS
- Samhain
- Snort
- Suricata

6.9.10 See also

- Anomaly-based intrusion detection system
- Application protocol-based intrusion detection system (APIDS)
- Artificial immune system
- Bypass switch
- Denial-of-service attack
- DNS analytics
- IDMEF : Intrusion Detection Message Exchange Format
- Protocol-based intrusion detection system (PIDS)
- Real-time adaptive security
- Security management
- Software-defined protection

6.9.11 References

[1] Scarfone, Karen; Mell, Peter (February 2007). "Guide to Intrusion Detection and Prevention Systems (IDPS)" (PDF). *Computer Security Resource Center* (National Institute of Standards and Technology) (800–94). Retrieved 1 January 2010.

[2] "NIST – Guide to Intrusion Detection and Prevention Systems (IDPS)" (PDF). February 2007. Retrieved 2010-06-25.

[3] Robert C. Newman (19 February 2009). *Computer Security: Protecting Digital Resources*. Jones & Bartlett Learning. ISBN 978-0-7637-5994-0. Retrieved 25 June 2010.

[4] Michael E. Whitman; Herbert J. Mattord (2009). *Principles of Information Security*. Cengage Learning EMEA. ISBN 978-1-4239-0177-8. Retrieved 25 June 2010.

[5] Tim Boyles (2010). *CCNA Security Study Guide: Exam 640-553*. John Wiley and Sons. p. 249. ISBN 978-0-470-52767-2. Retrieved 29 June 2010.

[6] Harold F. Tipton; Micki Krause (2007). *Information Security Management Handbook*. CRC Press. p. 1000. ISBN 978-1-4200-1358-0. Retrieved 29 June 2010.

[7] nitin.; Mattord, verma (2008). *Principles of Information Security*. Course Technology. pp. 290–301. ISBN 978-1-4239-0177-8.

[8] http://www.users.cs.york.ac.uk/~{}jac/PublishedPapers/AdhocNetsFinal.pdf

[9] John R. Vacca (2010). *Managing Information Security*. Syngress. p. 137. ISBN 978-1-59749-533-2. Retrieved 29 June 2010.

[10] Abdullah A. Mohamed, "Design Intrusion Detection System Based On Image Block Matching", International Journal of Computer and Communication Engineering, IACSIT Press, Vol. 2, No. 5, September 2013.

[11] Engin Kirda; Somesh Jha; Davide Balzarotti (2009). *Recent Advances in Intrusion Detection: 12th International Symposium, RAID 2009, Saint-Malo, France, September 23–25, 2009, Proceedings*. Springer. p. 162. ISBN 978-3-642-04341-3. Retrieved 29 June 2010.

[12] Anderson, Ross (2001). *Security Engineering: A Guide to Building Dependable Distributed Systems*. New York: John Wiley & Sons. pp. 387–388. ISBN 978-0-471-38922-4.

[13] http://www.giac.org/paper/gsec/235/limitations-network-intrusion-detection/100739

[14] Anderson, James P., "Computer Security Threat Monitoring and Surveillance," Washing, PA, James P. Anderson Co., 1980.

[15] Denning, Dorothy E., "An Intrusion Detection Model," Proceedings of the Seventh IEEE Symposium on Security and Privacy, May 1986, pages 119–131

[16] Lunt, Teresa F., "IDES: An Intelligent System for Detecting Intruders," Proceedings of the Symposium on Computer Security; Threats, and Countermeasures; Rome, Italy, November 22–23, 1990, pages 110–121.

[17] Lunt, Teresa F., "Detecting Intruders in Computer Systems," 1993 Conference on Auditing and Computer Technology, SRI International

[18] Sebring, Michael M., and Whitehurst, R. Alan., "Expert Systems in Intrusion Detection: A Case Study," The 11th National Computer Security Conference, October, 1988

[19] Smaha, Stephen E., "Haystack: An Intrusion Detection System," The Fourth Aerospace Computer Security Applications Conference, Orlando, FL, December, 1988

[20] Vaccaro, H.S., and Liepins, G.E., "Detection of Anomalous Computer Session Activity," The 1989 IEEE Symposium on Security and Privacy, May, 1989

[21] Teng, Henry S., Chen, Kaihu, and Lu, Stephen C-Y, "Adaptive Real-time Anomaly Detection Using Inductively Generated Sequential Patterns," 1990 IEEE Symposium on Security and Privacy

[22] Heberlein, L. Todd, Dias, Gihan V., Levitt, Karl N., Mukherjee, Biswanath, Wood, Jeff, and Wolber, David, "A Network Security Monitor," 1990 Symposium on Research in Security and Privacy, Oakland, CA, pages 296–304

[23] Winkeler, J.R., "A UNIX Prototype for Intrusion and Anomaly Detection in Secure Networks," The Thirteenth National Computer Security Conference, Washington, DC., pages 115–124, 1990

[24] Dowell, Cheri, and Ramstedt, Paul, "The ComputerWatch Data Reduction Tool," Proceedings of the 13th National Computer Security Conference, Washington, D.C., 1990

[25] Snapp, Steven R, Brentano, James, Dias, Gihan V., Goan, Terrance L., Heberlein, L. Todd, Ho, Che-Lin, Levitt, Karl N., Mukherjee, Biswanath, Smaha, Stephen E., Grance, Tim, Teal, Daniel M. and Mansur, Doug, "DIDS (Distributed Intrusion Detection System) -- Motivation, Architecture, and An Early Prototype," The 14th National Computer Security Conference, October, 1991, pages 167–176.

[26] Jackson, Kathleen, DuBois, David H., and Stallings, Cathy A., "A Phased Approach to Network Intrusion Detection," 14th National Computing Security Conference, 1991

[27] Paxson, Vern, "Bro: A System for Detecting Network Intruders in Real-Time," Proceedings of The 7th USENIX Security Symposium, San Antonio, TX, 1998

[28] Amoroso, Edward, "Intrusion Detection: An Introduction to Internet Surveillance, Correlation, Trace Back, Traps, and Response," Intrusion.Net Books, Sparta, New Jersey, 1999, ISBN 0-9666700-7-8

[29] Kohlenberg, Toby (Ed.), Alder, Raven, Carter, Dr. Everett F. (Skip), Jr., Esler, Joel., Foster, James C., Jonkman Marty, Raffael, and Poor, Mike, "Snort IDS and IPS Toolkit," Syngress, 2007, ISBN 978-1-59749-099-3

[30] Barbara, Daniel, Couto, Julia, Jajodia, Sushil, Popyack, Leonard, and Wu, Ningning, "ADAM: Detecting Intrusions by Data Mining," Proceedings of the IEEE Workshop on Information Assurance and Security, West Point, NY, June 5–6, 2001

[31] Intrusion Detection Techniques for Mobile Wireless Networks, ACM WINET 2003 <http://www.cc.gatech.edu/~{}wenke/papers/winet03.pdf>

This article incorporates public domain material from the National Institute of Standards and Technology document "Guide to Intrusion Detection and Prevention Systems, SP800-94" by Karen Scarfone, Peter Mell (retrieved on 1 January 2010).

6.9.12 Further reading

- Hansen, James V.; Benjamin Lowry, Paul; Meservy, Rayman; McDonald, Dan (2007). "Genetic programming for prevention of cyberterrorism through dynamic and evolving intrusion detection". *Decision Support Systems (DSS)* **43** (4): 1362–1374. doi:10.1016/j.dss.2006.04.004.
- Scarfone, Karen; Mell, Peter (February 2007). "Guide to Intrusion Detection and Prevention Systems (IDPS)" (PDF). *Computer Security Resource Center* (National Institute of Standards and Technology) (800-94). Retrieved 1 January 2010.
- Saranya, J.; Padmavathi, G. (2015). "A Brief Study on Different Intrusions and Machine Learning-based Anomaly Detection Methods in Wireless Sensor Networks" (PDF). *Avinashilingam Institute for Home Science and Higher Education for Women* (6(4)). Retrieved 4 April 2015.
- Singh, Abhishek. "Evasions In Intrusion Prevention Detection Systems". Virus Bulletin. Retrieved April 2010.
- Bezroukov, Nikolai (11 December 2008). "Architectural Issues of Intrusion Detection Infrastructure in Large Enterprises (Revision 0.82)". Softpanorama. Retrieved 30 July 2010.
- P.M. Mafra and J.S. Fraga and A.O. Santin (2014). "Algorithms for a distributed IDS in MANETs". *Journal of Computer and System Sciences* **80** (3): 554–570. doi:10.1016/j.jcss.2013.06.011.

6.9.13 External links

- Intrusion Detection Systems at DMOZ
- Common vulnerabilities and exposures (CVE) by product
- NIST SP 800-83, Guide to Malware Incident Prevention and Handling
- NIST SP 800-94, Guide to Intrusion Detection and Prevention Systems (IDPS)
- Study by Gartner "Magic Quadrant for Network Intrusion Prevention System Appliances"

6.10 Data loss prevention software

Data loss prevention software that are designed to detect potential data breaches / data ex-filtration transmissions and prevent them by monitoring, detecting and blocking sensitive data while **in-use** (endpoint actions), **in-motion** (network traffic), and **at-rest** (data storage). In data leakage incidents, sensitive data is disclosed to unauthorized personnel either by malicious intent or inadvertent mistake. Such sensitive data can come in the form of private or company information, intellectual property (IP), financial or patient information, credit-card data, and other information depending on the business and the industry.

The terms "data loss" and "data leak" are closely related and are often used interchangeably, though they are somewhat different.[1] Data loss incidents turn into data leak incidents in cases where media containing sensitive information is lost and subsequently acquired by unauthorized party. However, a data leak is possible without the data being lost in the originating side. Some other terms associated with **data leakage prevention** are **information leak detection and prevention** (**ILDP**), **information leak prevention** (**ILP**), **content monitoring and filtering** (**CMF**), **information protection and control** (**IPC**), and **extrusion prevention system** (**EPS**), as opposed to intrusion prevention system.

6.10.1 DLP Categories

The technological means employed for dealing with data leakage incidents can be divided into the following categories: standard security measures, advanced/intelligent security measures, access control and encryption, and designated DLP systems .[2]

Standard security measures

Standard security measures, such as firewalls, intrusion detection systems (IDSs), and antivirus software, are commonly available mechanisms that guard computers against outsider as well as insider attacks. The use of firewall, for example, limits the access of outsiders to the internal network, and an intrusion detection system detects intrusion attempts by outsiders. Inside attacks can be averted through antivirus scans that detect Trojan horses installed on PCs which send confidential information, and by the use of thin clients, which operate in a client-server architecture with no personal or sensitive data stored on a client's computer.

Advanced security measures

Advanced security measures employ machine learning and temporal reasoning algorithms for detecting abnormal access to data (e.g., databases or information retrieval systems) or abnormal email exchange, honeypots for detecting authorized personnel with malicious intentions, and activity-based verification (e.g., recognition of keystrokes

dynamics), and user activity monitoring for detecting abnormal access to data.

Designated DLP solutions

Designated DLP solutions detect and prevent unauthorized attempts to copy or send sensitive data, intentionally or unintentionally, without authorization, mainly by personnel who are authorized to access the sensitive information. In order to classify certain information as sensitive, these solutions use mechanisms, such as exact data matching, structured data fingerprinting, statistical methods, rule and regular expression matching, published lexicons, conceptual definitions, and keywords .[3]

6.10.2 Types of DLP systems

Network DLP (a.k.a. data in motion <DiM>)

Typically a software or hardware solution that is installed at network egress points near the perimeter. It analyzes network traffic to detect sensitive data that is being sent in violation of information security policies. Network DLP solutions have multiple security control points which all trace back to be analyzed by a central management server.[1]

Endpoint DLP (a.k.a. data in use <DiU>)

Such systems run on end-user workstations or servers in the organization. Like network-based systems, endpoint-based can address internal as well as external communications, and can therefore be used to control information flow between groups or types of users (e.g. 'Chinese walls'). They can also control email and Instant Messaging communications before they are stored in the corporate archive, such that a blocked communication (i.e., one that was never sent, and therefore not subject to retention rules) will not be identified in a subsequent legal discovery situation. Endpoint systems have the advantage that they can monitor and control access to physical devices (such as mobile devices with data storage capabilities) and in some cases can access information before it has been encrypted. Some endpoint-based systems can also provide application controls to block attempted transmissions of confidential information, and provide immediate feedback to the user. They have the disadvantage that they need to be installed on every workstation in the network, cannot be used on mobile devices (e.g., cell phones and PDAs) or where they cannot be practically installed (for example on a workstation in an internet café).

Data identification

DLP solutions include a number of techniques for identifying confidential or sensitive information. Sometimes confused with discovery, data identification is a process by which organizations use a DLP technology to determine what to look for (in motion, at rest, or in use).

Data is classified as **structured** or **unstructured**. Structured data resides in fixed fields within a file such as a spreadsheet, while unstructured data refers to free-form text as in text documents or PDF files.[4] An estimated 80% of all data is unstructured and 20% structured.[5] Data classification is divided into **content analysis**, focused on structured data, and **contextual analysis** which looks at the place of origin or the application or system that generated the data.[6]

Methods for describing sensitive content are abundant. They can be divided into two categories: **precise methods** and **imprecise methods**.

Precise methods are, by definition, those that involve Content Registration and trigger almost zero false positive incidents.

All other methods are **imprecise** and can include: keywords, lexicons, regular expressions, extended regular expressions, meta data tags, bayesian analysis, statistical analysis such as Machine Learning, etc.

The strength of the analysis engine directly correlates to its accuracy. **The accuracy of DLP identification is important to lowering/avoiding false positives and negatives**. Accuracy can depend on many variables, some of which may be situational or technological. **Testing for accuracy is recommended to ensure a solution has virtually zero false positives/negatives**. High False Positive Rates will cause the system to be DLD not DLP.[7]

Data leakage detection

Sometimes a data distributor gives sensitive data to a set of third parties. Some time later, some of the data is found in an unauthorized place (e.g., on the web or on a user's laptop). The distributor must then investigate if data leaked from one or more of the third parties, or if it was independently gathered by other means.[8]

Data at-rest

"Data at rest" specifically refers to old archived information that is stored on either a client PC hard drive, on a network storage drive or remote file server, or even data stored on a backup system, such as a tape or CD media. This information is of great concern to businesses and government

institutions simply because the longer data is left unused in storage, the more likely it might be retrieved by unauthorized individuals outside the Network.[9] In order to protect this phase of data, systems use methods such as access control and data encryption.[1]

Data in-use

"Data in use" refers to active data stored in databases that the user is currently interacting with. DLP systems that protect data in-use may monitor and flag certain unauthorized activities.[1] These activities include screen-capture, copy/paste, print and fax operations involving sensitive data. It can also be intentional or unintentional attempts to transmit sensitive data over communication channels such as IM or a website.[10]

Data in-motion

"Data in motion" is data that is currently traversing through a network to an endpoint destination. These networks can be internal or external. DLP systems that protect data in-motion monitor sensitive data that is being sent over a network through various communication channels such as email or IM.[1]

6.10.3 See also

- Metadata removal tool

6.10.4 References

[1] Asaf Shabtai, Yuval Elovici, Lior Rokach, A Survey of Data Leakage Detection and Prevention Solutions, Springer-Verlag New York Incorporated, 2012

[2] Phua, C., Protecting organisations from personal data breaches, Computer Fraud and Security, 1:13-18, 2009

[3] Ouellet, E., Magic Quadrant for Content-Aware Data Loss Prevention, Technical Report, RA4 06242010, Gartner RAS Core Research, 2012

[4] http://www.pcmag.com/encyclopedia/term/53486/unstructured-data

[5] Brian E. Burke, "Information Protection and Control survey: Data Loss Prevention and Encryption trends," IDC, May 2008

[6] https://securosis.com/assets/library/reports/DLP-Whitepaper.pdf

[7] "Core DLP Technology - © GTB Technologies, Inc. 2006".

[8] Panagiotis Papadimitriou, Hector Garcia-Molina (January 2011), "Data Leakage Detection" (PDF), *IEEE Transactions on Knowledge and Data Engineering* **23** (1): 51–63, doi:10.1109/TKDE.2010.100

[9] Costante, E., Vavilis, S., Etalle, S., Petkovic, M., & Zannone, N. Database Anomalous Activities: Detection and Quantification .SECRYPT 2013

[10] Gugelmann, D.; Studerus, P.; Lenders, V.; Ager, B. (2015-07-01). "Can Content-Based Data Loss Prevention Solutions Prevent Data Leakage in Web Traffic?". *IEEE Security Privacy* **13** (4): 52–59. doi:10.1109/MSP.2015.88. ISSN 1540-7993.

6.10.5 External links

- Cost of a Data Breach

Chapter 7

Countermeasures

7.1 Computer and network surveillance

This article is about monitoring of computer and network activity. For information on methods of preventing unauthorized access to computer data, see computer security.
Main article: Surveillance

Computer and network surveillance is the monitoring of computer activity and data stored on a hard drive, or data being transferred over computer networks such as the Internet. The monitoring is often carried out covertly and may be completed by governments, corporations, criminal organizations, or individuals. It may or may not be legal and may or may not require authorization from a court or other independent government agency.

Computer and network surveillance programs are widespread today and almost all Internet traffic can be monitored for illegal activity.[1]

Surveillance allows governments and other agencies to maintain social control, recognize and monitor threats, and prevent and investigate criminal activity. With the advent of programs such as the Total Information Awareness program, technologies such as high speed surveillance computers and biometrics software, and laws such as the Communications Assistance For Law Enforcement Act, governments now possess an unprecedented ability to monitor the activities of citizens.[2]

However, many civil rights and privacy groups, such as Reporters Without Borders, the Electronic Frontier Foundation, and the American Civil Liberties Union, have expressed concern that with increasing surveillance of citizens we will end up in or are even already in a mass surveillance society, with limited political and/or personal freedoms. Such fear has led to numerous lawsuits such as *Hepting v. AT&T*.[2][3] The hacktivist group Anonymous has hacked into government websites in protest of what it considers "draconian surveillance".[4][5]

7.1.1 Network surveillance

See also: Signals intelligence

The vast majority of computer surveillance involves the monitoring of data and traffic on the Internet.[6] For example, in the United States, the Communications Assistance For Law Enforcement Act, mandates that all phone calls and broadband internet traffic (emails, web traffic, instant messaging, etc.) be available for unimpeded, real-time monitoring by Federal law enforcement agencies.[7][8][9]

Packet capture (also known as "packet sniffing") is the monitoring of data traffic on a computer network.[10] Data sent between computers over the Internet or between any networks takes the form of small chunks called packets, which are routed to their destination and assembled back into a complete message. A Packet Capture Appliance intercepts these packets, so that they may be examined and analyzed. Computer technology is needed to perform traffic analysis and sift through intercepted data to look for important/useful information. Under the Communications Assistance For Law Enforcement Act, all U.S. telecommunications providers are required to install such packet capture technology so that Federal law enforcement and intelligence agencies are able to intercept all of their customers' broadband Internet and voice over Internet protocol (VoIP) traffic.[11]

There is far too much data gathered by these packet sniffers for human investigators to manually search through. Thus, automated Internet surveillance computers sift through the vast amount of intercepted Internet traffic, filtering out, and reporting to investigators those bits of information which are "interesting", for example, the use of certain words or phrases, visiting certain types of web sites, or communicating via email or chat with a certain individual or group.[12] Billions of dollars per year are spent by agencies such as the Information Awareness Office, NSA, and the FBI, for the development, purchase, implementation, and operation of systems which intercept and analyze this data, extracting only the information that is useful to law enforcement and

intelligence agencies.[13]

Similar systems are now used by Iranian secret police to identify and suppress dissidents. All of the technology has been allegedly installed by German Siemens AG and Finnish Nokia.[14]

The Internet's rapid development has become a primary form of communication. More people are potentially subject to Internet surveillance. There are advantages and disadvantages to network monitoring. For instance, systems described as "Web 2.0"[15] have greatly impacted modern society. An advantage to online surveillance is that large social media platforms, such as YouTube, Twitter and Facebook, enable people to contact friends, family, and strangers daily. Tim O' Reilly, who first explained the concept of "Web 2.0",[15] stated that Web 2.0 provides communication platforms that are "user generated", with self-produced content, motivating more people to communicate with friends online.[16] However, Internet surveillance also has a disadvantage. One researcher from Uppsala University said "Web 2.0 surveillance is directed at large user groups who help to hegemonically produce and reproduce surveillance by providing user-generated (self-produced) content. We can characterize Web 2.0 surveillance as mass self-surveillance".[17] Surveillance companies monitor people while they are focused on work or entertainment. This can emotionally affect people; this is because it can cause emotions like jealousy. A research group states "...we set out to test the prediction that feelings of jealousy lead to 'creeping' on a partner through Facebook, and that women are particularly likely to engage in partner monitoring in response to jealousy".[18] The study shows that women can become jealous of other people when they are in an online group.

7.1.2 Corporate surveillance

See also: Computer surveillance in the workplace

Corporate surveillance of computer activity is very common. The data collected is most often used for marketing purposes or sold to other corporations, but is also regularly shared with government agencies. It can be used as a form of business intelligence, which enables the corporation to better tailor their products and/or services to be desirable by their customers. Or the data can be sold to other corporations, so that they can use it for the aforementioned purpose. Or it can be used for direct marketing purposes, such as targeted advertisements, where ads are targeted to the user of the search engine by analyzing their search history and emails[19] (if they use free webmail services), which is kept in a database.[20]

One important component of prevention is establishing the business purposes of monitoring, which may include the following:

- Preventing misuse of resources. Companies can discourage unproductive personal activities such as online shopping or web surfing on company time. Monitoring employee performance is one way to reduce unnecessary network traffic and reduce the consumption of network bandwidth.
- Promoting adherence to policies. Online surveillance is one means of verifying employee observance of company networking policies.
- Preventing lawsuits. Firms can be held liable for discrimination or employee harassment in the workplace. Organizations can also be involved in infringement suits through employees that distribute copyrighted material over corporate networks.
- Safeguarding records. Federal legislation requires organizations to protect personal information. Monitoring can determine the extent of compliance with company policies and programs overseeing information security. Monitoring may also deter unlawful appropriation of personal information, and potential spam or viruses.
- Safeguarding company assets. The protection of intellectual property, trade secrets, and business strategies is a major concern. The ease of information transmission and storage makes it imperative to monitor employee actions as part of a broader policy.

A second component of prevention is determining the ownership of technology resources. The ownership of the firm's networks, servers, computers, files, and e-mail should be explicitly stated. There should be a distinction between an employee's personal electronic devices, which should be limited and proscribed, and those owned by the firm.

For instance, Google, the world's most popular search engine, stores identifying information for each web search. An IP address and the search phrase used are stored in a database for up to 18 months.[21] Google also scans the content of emails of users of its Gmail webmail service, in order to create targeted advertising based on what people are talking about in their personal email correspondences.[22] Google is, by far, the largest Internet advertising agency—millions of sites place Google's advertising banners and links on their websites, in order to earn money from visitors who click on the ads. Each page containing Google advertisements adds, reads, and modifies "cookies" on each visitor's computer.[23] These cookies track the user across all of these sites, and gather information about their web

surfing habits, keeping track of which sites they visit, and what they do when they are on these sites. This information, along with the information from their email accounts, and search engine histories, is stored by Google to use to build a profile of the user to deliver better-targeted advertising.[22]

The United States government often gains access to these databases, either by producing a warrant for it, or by simply asking. The Department of Homeland Security has openly stated that it uses data collected from consumer credit and direct marketing agencies for augmenting the profiles of individuals that it is monitoring.[20]

7.1.3 Malicious software

Further information: Spyware, Computer virus, Trojan (computer security), Keylogger, and Backdoor (computing)

In addition to monitoring information sent over a computer network, there is also a way to examine data stored on a computer's hard drive, and to monitor the activities of a person using the computer. A surveillance program installed on a computer can search the contents of the hard drive for suspicious data, can monitor computer use, collect passwords, and/or report back activities in real-time to its operator through the Internet connection.[24] Keylogger is an example of this type of program. Normal keylogging programs store their data on the local hard drive, but some are programmed to automatically transmit data over the network to a remote computer or Web server.

There are multiple ways of installing such software. The most common is remote installation, using a backdoor created by a computer virus or trojan. This tactic has the advantage of potentially subjecting multiple computers to surveillance. Viruses often spread to thousands or millions of computers, and leave "backdoors" which are accessible over a network connection, and enable an intruder to remotely install software and execute commands. These viruses and trojans are sometimes developed by government agencies, such as CIPAV and Magic Lantern. More often, however, viruses created by other people or spyware installed by marketing agencies can be used to gain access through the security breaches that they create.[25]

Another method is "cracking" into the computer to gain access over a network. An attacker can then install surveillance software remotely. Servers and computers with permanent broadband connections are most vulnerable to this type of attack.[26] Another source of security cracking is employees giving out information or users using brute force tactics to guess their password.[27]

One can also physically place surveillance software on a computer by gaining entry to the place where the computer is stored and install it from a compact disc, floppy disk, or thumbdrive. This method shares a disadvantage with hardware devices in that it requires physical access to the computer.[28] One well-known worm that uses this method of spreading itself is Stuxnet.[29]

7.1.4 Social network analysis

One common form of surveillance is to create maps of social networks based on data from social networking sites as well as from traffic analysis information from phone call records such as those in the NSA call database,[30] and internet traffic data gathered under CALEA. These social network "maps" are then data mined to extract useful information such as personal interests, friendships and affiliations, wants, beliefs, thoughts, and activities.[31][32][33]

Many U.S. government agencies such as the Defense Advanced Research Projects Agency (DARPA), the National Security Agency (NSA), and the Department of Homeland Security (DHS) are currently investing heavily in research involving social network analysis.[34][35] The intelligence community believes that the biggest threat to the U.S. comes from decentralized, leaderless, geographically dispersed groups. These types of threats are most easily countered by finding important nodes in the network, and removing them. To do this requires a detailed map of the network.[33][36]

Jason Ethier of Northeastern University, in his study of modern social network analysis, said the following of the Scalable Social Network Analysis Program developed by the Information Awareness Office:

> *The purpose of the SSNA algorithms program is to extend techniques of social network analysis to assist with distinguishing potential terrorist cells from legitimate groups of people ... In order to be successful SSNA will require information on the social interactions of the majority of people around the globe. Since the Defense Department cannot easily distinguish between peaceful citizens and terrorists, it will be necessary for them to gather data on innocent civilians as well as on potential terrorists.*
> — Jason Ethier[33]

7.1.5 Monitoring from a distance

It has been shown that it is possible to monitor computers from a distance, with only commercially available equipment, by detecting the radiation emitted by the CRT monitor. This form of computer surveillance, known as

TEMPEST, involves reading electromagnetic emanations from computing devices in order to extract data from them at distances of hundreds of meters.[37][38][39]

IBM researchers have also found that, for most computer keyboards, each key emits a slightly different noise when pressed. The differences are individually identifiable under some conditions, and so it's possible to log key strokes without actually requiring logging software to run on the associated computer.[40][41]

And it has also been shown, by Adi Shamir et al., that even the high frequency noise emitted by a CPU includes information about the instructions being executed.[42]

7.1.6 Policeware and govware

Policeware is software designed to police citizens by monitoring discussion and interaction of its citizens.[43] Within the U.S., Carnivore was a first incarnation of secretly installed e-mail monitoring software installed in Internet service providers' networks to log computer communication, including transmitted e-mails.[44] Magic Lantern is another such application, this time running in a targeted computer in a trojan style and performing keystroke logging. CIPAV, deployed by FBI, is a multi-purpose spyware/trojan.

The "Consumer Broadband and Digital Television Promotion Act" (CBDTA) was a bill proposed in the United States Congress. CBDTPA was known as the "Security Systems and Standards Certification Act" (SSSCA) while in draft form, and was killed in committee in 2002. Had CBDTPA become law, it would have prohibited technology that could be used to read digital content under copyright (such as music, video, and e-books) without Digital Rights Management (DRM) that prevented access to this material without the permission of the copyright holder.[45]

In German-speaking countries, spyware used or made by the government is sometimes called *govware*.[46] Some countries like Switzerland and Germany have a legal framework governing the use of such software.[47][48] Known examples include the Swiss MiniPanzer and MegaPanzer and the German R2D2 (trojan).

7.1.7 Surveillance as an aid to censorship

See also: Internet censorship and Internet censorship circumvention

Surveillance and censorship are different. Surveillance can be performed without censorship, but it is harder to engage in censorship without some form of surveillance.[49] And even when surveillance does not lead directly to censorship, the widespread knowledge or belief that a person, their computer, or their use of the Internet is under surveillance can lead to self-censorship.[50]

In March 2013 Reporters Without Borders issued a *Special report on Internet surveillance* that examines the use of technology that monitors online activity and intercepts electronic communication in order to arrest journalists, citizen-journalists, and dissidents. The report includes a list of "State Enemies of the Internet", Bahrain, China, Iran, Syria, and Vietnam, countries whose governments are involved in active, intrusive surveillance of news providers, resulting in grave violations of freedom of information and human rights. Computer and network surveillance is on the increase in these countries. The report also includes a second list of "Corporate Enemies of the Internet", Amesys (France), Blue Coat Systems (U.S.), Gamma (UK and Germany), Hacking Team (Italy), and Trovicor (Germany), companies that sell products that are liable to be used by governments to violate human rights and freedom of information. Neither list is exhaustive and they are likely to be expanded in the future.[51]

Protection of sources is no longer just a matter of journalistic ethics. Journalists should equip themselves with a "digital survival kit" if they are exchanging sensitive information online, storing it on a computer hard-drive or mobile phone.[51][52] Individuals associated with high profile rights organizations, dissident groups, protest groups, or reform groups are urged to take extra precautions to protect their online identities.[53]

7.1.8 See also

- Anonymizer, a software system that attempts to make network activity untraceable
- Computer surveillance in the workplace
- Cyber spying
- Differential privacy, a method to maximize the accuracy of queries from statistical databases while minimizing the chances of violating the privacy of individuals.
- ECHELON, a signals intelligence (SIGINT) collection and analysis network operated on behalf of Australia, Canada, New Zealand, the United Kingdom, and the United States, also known as AUSCANNZUKUS and Five Eyes
- GhostNet, a large-scale cyber spying operation discovered in March 2009
- List of government surveillance projects

- Mass surveillance
  - China's Golden Shield Project
  - Mass surveillance in Australia
  - Mass surveillance in China
  - Mass surveillance in East Germany
  - Mass surveillance in India
  - Mass surveillance in North Korea
  - Mass surveillance in the United Kingdom
  - Mass surveillance in the United States
- Surveillance
- Surveillance by the United States government:
  - 2013 mass surveillance disclosures, reports about NSA and its international partners' mass surveillance of foreign nationals and U.S. citizens
  - Bullrun (code name), a highly classified NSA program to preserve its ability to eavesdrop on encrypted communications by influencing and weakening encryption standards, by obtaining master encryption keys, and by gaining access to data before or after it is encrypted either by agreement, by force of law, or by computer network exploitation (hacking)
  - Carnivore, a U.S. Federal Bureau of Investigation system to monitor email and electronic communications
  - COINTELPRO, a series of covert, and at times illegal, projects conducted by the FBI aimed at U.S. domestic political organizations
  - Communications Assistance For Law Enforcement Act
  - Computer and Internet Protocol Address Verifier (CIPAV), a data gathering tool used by the U.S. Federal Bureau of Investigation (FBI)
  - Dropmire, a secret surveillance program by the NSA aimed at surveillance of foreign embassies and diplomatic staff, including those of NATO allies
  - Magic Lantern, keystroke logging software developed by the U.S. Federal Bureau of Investigation
  - Mass surveillance in the United States
  - NSA call database, a database containing metadata for hundreds of billions of telephone calls made in the U.S.
  - NSA warrantless surveillance (2001–07)
  - NSA whistleblowers: William Binney, Thomas Andrews Drake, Mark Klein, Edward Snowden, Thomas Tamm, Russ Tice
  - Spying on United Nations leaders by United States diplomats
  - Stellar Wind (code name), code name for information collected under the President's Surveillance Program
  - Tailored Access Operations, NSA's hacking program
  - Terrorist Surveillance Program, an NSA electronic surveillance program
  - Total Information Awareness, a project of the Defense Advanced Research Projects Agency (DARPA)
- TEMPEST, codename for studies of unintentional intelligence-bearing signals which, if intercepted and analyzed, may disclose the information transmitted, received, handled, or otherwise processed by any information-processing equipment

7.1.9 References

[1] Anne Broache. "FBI wants widespread monitoring of 'illegal' Internet activity". *CNET*. Retrieved 25 March 2014.

[2] "Is the U.S. Turning Into a Surveillance Society?". *American Civil Liberties Union*. Retrieved March 13, 2009.

[3] "Bigger Monster, Weaker Chains: The Growth of an American Surveillance Society" (PDF). *American Civil Liberties Union*. January 15, 2003. Retrieved March 13, 2009.

[4] "Anonymous hacks UK government sites over 'draconian surveillance' ", Emil Protalinski, ZDNet, 7 April 2012, retrieved 12 March 2013

[5] Hacktivists in the frontline battle for the internet retrieved 17 June 2012

[6] Diffie, Whitfield; Susan Landau (August 2008). "Internet Eavesdropping: A Brave New World of Wiretapping". *Scientific American*. Retrieved 2009-03-13.

[7] "CALEA Archive -- Electronic Frontier Foundation". *Electronic Frontier Foundation (website)*. Retrieved 2009-03-14.

[8] "CALEA: The Perils of Wiretapping the Internet". *Electronic Frontier Foundation (website)*. Retrieved 2009-03-14.

[9] "CALEA: Frequently Asked Questions". *Electronic Frontier Foundation (website)*. Retrieved 2009-03-14.

[10] Kevin J. Connolly (2003). *Law of Internet Security and Privacy*. Aspen Publishers. p. 131. ISBN 978-0-7355-4273-0.

[11] American Council on Education vs. FCC, Decision, United States Court of Appeals for the District of Columbia Circuit, 9 June 2006. Retrieved 8 September 2013.

[12] Hill, Michael (October 11, 2004). "Government funds chat room surveillance research". USA Today. Associated Press. Retrieved 2009-03-19.

[13] McCullagh, Declan (January 30, 2007). "FBI turns to broad new wiretap method". *ZDNet News*. Retrieved 2009-03-13.

[14] "First round in Internet war goes to Iranian intelligence", Debkafile, 28 June 2009. (subscription required)

[15] O'Reilly, T. (2005). What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software. O'Reilly Media, 1-5.

[16] Fuchs, C. (2011). New Media, Web 2.0 and Surveillance. Sociology Compass, 134-147.

[17] Fuchs, C. (2011). Web 2.0, Presumption, and Surveillance. Surveillance & Society, 289-309.

[18] Muise, A., Christofides, E., & Demsmarais, S. (2014). " Creeping" or just information seeking? Gender differences in partner monitoring in response to jealousy on Facebook. Personal Relationships, 21(1), 35-50.

[19] Story, Louise (November 1, 2007). "F.T.C. to Review Online Ads and Privacy". *New York Times*. Retrieved 2009-03-17.

[20] Butler, Don (January 31, 2009). "Are we addicted to being watched?". *The Ottawa Citizen* (canada.com). Retrieved 26 May 2013.

[21] Soghoian, Chris (September 11, 2008). "Debunking Google's log anonymization propaganda". *CNET News*. Retrieved 2009-03-21.

[22] Joshi, Priyanki (March 21, 2009). "Every move you make, Google will be watching you". *Business Standard*. Retrieved 2009-03-21.

[23] "Advertising and Privacy". *Google (company page)*. 2009. Retrieved 2009-03-21.

[24] "Spyware Workshop: Monitoring Software on Your OC: Spywae, Adware, and Other Software", Staff Report, U.S. Federal Trade Commission, March 2005. Retrieved 7 September 2013.

[25] Aycock, John (2006). *Computer Viruses and Malware*. Springer. ISBN 978-0-387-30236-2.

[26] "Office workers give away passwords for a cheap pen", John Leyden, *The Register*, 8 April 2003. Retrieved 7 September 2013.

[27] "Passwords are passport to theft", *The Register*, 3 March 2004. Retrieved 7 September 2013.

[28] "Social Engineering Fundamentals, Part I: Hacker Tactics", Sarah Granger, 18 December 2001.

[29] "Stuxnet: How does the Stuxnet worm spread?". Antivirus.about.com. 2014-03-03. Retrieved 2014-05-17.

[30] Keefe, Patrick (March 12, 2006). "Can Network Theory Thwart Terrorists?". *New York Times*. Retrieved 14 March 2009.

[31] Albrechtslund, Anders (March 3, 2008). "Online Social Networking as Participatory Surveillance". *First Monday* **13** (3). Retrieved March 14, 2009.

[32] Fuchs, Christian (2009). *Social Networking Sites and the Surveillance Society. A Critical Case Study of the Usage of studiVZ, Facebook, and MySpace by Students in Salzburg in the Context of Electronic Surveillance* (PDF). Salzburg and Vienna: Forschungsgruppe Unified Theory of Information. ISBN 978-3-200-01428-2. Retrieved March 14, 2009.

[33] Ethier, Jason (27 May 2006). "Current Research in Social Network Theory" (PDF). Northeastern University College of Computer and Information Science. Retrieved 15 March 2009.

[34] Marks, Paul (June 9, 2006). "Pentagon sets its sights on social networking websites". *New Scientist*. Retrieved 2009-03-16.

[35] Kawamoto, Dawn (June 9, 2006). "Is the NSA reading your MySpace profile?". *CNET News*. Retrieved 2009-03-16.

[36] Ressler, Steve (July 2006). "Social Network Analysis as an Approach to Combat Terrorism: Past, Present, and Future Research". *Homeland Security Affairs* **II** (2). Retrieved March 14, 2009.

[37] McNamara, Joel (4 December 1999). "Complete, Unofficial Tempest Page". Retrieved 7 September 2013.

[38] Van Eck, Wim (1985). "Electromagnetic Radiation from Video Display Units: An Eavesdropping Risk?" (PDF). *Computers & Security* **4**: 269–286. doi:10.1016/0167-4048(85)90046-X.

[39] Kuhn, M.G. (26–28 May 2004). "Electromagnetic Eavesdropping Risks of Flat-Panel Displays" (PDF). *4th Workshop on Privacy Enhancing Technologies* (Toronto): 23–25.

[40] Asonov, Dmitri; Agrawal, Rakesh (2004), *Keyboard Acoustic Emanations* (PDF), IBM Almaden Research Center

[41] Yang, Sarah (14 September 2005), "Researchers recover typed text using audio recording of keystrokes", *UC Berkeley News*

[42] Adi Shamir & Eran Tromer. "Acoustic cryptanalysis". Blavatnik School of Computer Science, Tel Aviv University. Retrieved 1 November 2011.

[43] Jeremy Reimer (20 July 2007). "The tricky issue of spyware with a badge: meet 'policeware'". Ars Technica.

[44] Hopper, D. Ian (4 May 2001). "FBI's Web Monitoring Exposed". ABC News.

[45] "Consumer Broadband and Digital Television Promotion Act", U.S. Senate bill S.2048, 107th Congress, 2nd session, 21 March 2002. Retrieved 8 September 2013.

[46] "Swiss coder publicises government spy Trojan". News.techworld.com. Retrieved 25 March 2014.

[47] Basil Cupa, Trojan Horse Resurrected: On the Legality of the Use of Government Spyware (Govware), LISS 2013, pp. 419-428

[48] "FAQ – Häufig gestellte Fragen". Ejpd.admin.ch. 2011-11-23. Retrieved 2014-05-17.

[49] "Censorship is inseparable from surveillance", Cory Doctorow, *The Guardian*, 2 March 2012

[50] "Trends in transition from classical censorship to Internet censorship: selected country overviews"

[51] *The Enemies of the Internet Special Edition : Surveillance*, Reporters Without Borders, 12 March 2013

[52] "When Secrets Aren't Safe With Journalists", Christopher Soghoian, *New York Times*, 26 October 2011

[53] *Everyone's Guide to By-passing Internet Censorship*, The Citizen Lab, University of Toronto, September 2007

7.1.10 External links

- "Selected Papers in Anonymity", Free Haven Project, accessed 16 September 2011.

7.2 Operation: Bot Roast

Operation: Bot Roast is an operation by the FBI to track down bot herders, crackers, or virus coders who install malicious software on computers through the Internet without the owners' knowledge, which turns the computer into a zombie computer that then sends out spam to other computers from the compromised computer, making a botnet or network of bot infected computers. The operation was launched because the vast scale of botnet resources poses a threat to national security.[1]

7.2.1 The results

The operation was created to disrupt and disassemble bot herders. In June 2007, the FBI had identified about 1 million computers that were compromised, leading to the arrest of the persons responsible for creating the malware. In the process, owners of infected computers were notified, many of whom were unaware of the exploitation.[1][2]

Some early results of the operation include charges against the following:

- Robert Alan Soloway of Seattle, Washington, pleaded guilty to charges of using botnets to send tens of millions of spam messages touting his website.[1]
- Jeanson James Ancheta plead guilty to controlling thousands of infected computers.[3]
- Jason Michael Downey (pseudonym "Nessun"), founder of the IRC network Rizon, is charged with using botnets to disable other systems.[1]
- Akbot author Owen Walker (pseudonym "AKILL") of New Zealand, was tried for various crimes and discharged by the prosecution in 2008.[4]

7.2.2 See also

- Botnet
- E-mail spam
- Internet crime
- Internet security
- Storm botnet
- Lycos Europe

7.2.3 References

[1] "OPERATION: BOT ROAST 'Bot-herders' Charged as Part of Initiative" (Press release). Federal Bureau of Investigation. 2007-06-13. Retrieved 2012-11-26.

[2] "FBI tries to fight zombie hordes" (Press release). BBC News. 2007-06-14. Retrieved 2007-06-20.

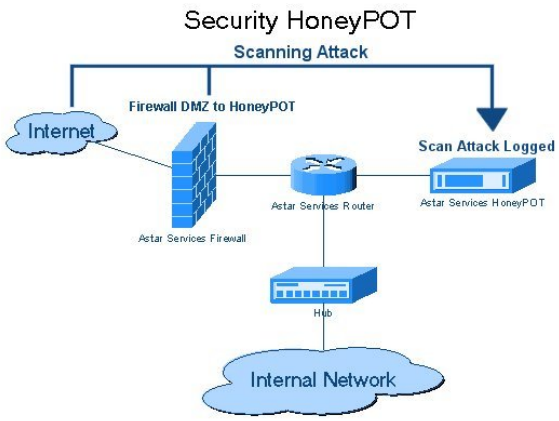
[3] Dan Goodin (13 June 2007). "FBI logs its millionth zombie address". the register. Retrieved 2008-09-26.

[4] Akill pleads guilty to all charges, By Ulrika Hedquist, 1 April, 2008, Computerworld

7.3 Honeypot (computing)

In computer terminology, a **honeypot** is a computer security mechanism set to detect, deflect, or, in some manner, counteract attempts at unauthorized use of information systems. Generally, a honeypot consists of data (for example,

in a network site) that appears to be a legitimate part of the site but is actually isolated and monitored, and that seems to contain information or a resource of value to attackers, which are then blocked. This is similar to the police baiting a criminal and then conducting undercover surveillance, and finally punishing the criminal.[1]



Honeypot diagram to help understand the topic

7.3.1 Types

Honeypots can be classified based on their deployment (use/action) and based on their level of involvement. Based on deployment, honeypots may be classified as

1. production honeypots
2. research honeypots

Production honeypots are easy to use, capture only limited information, and are used primarily by companies or corporations. Production honeypots are placed inside the production network with other production servers by an organization to improve their overall state of security. Normally, production honeypots are low-interaction honeypots, which are easier to deploy. They give less information about the attacks or attackers than research honeypots. **Research honeypots** are run to gather information about the motives and tactics of the Black hat community targeting different networks. These honeypots do not add direct value to a specific organization; instead, they are used to research the threats that organizations face and to learn how to better protect against those threats.[2] Research honeypots are complex to deploy and maintain, capture extensive information, and are used primarily by research, military, or government organizations.

Based on design criteria, honeypots can be classified as:

1. pure honeypots
2. high-interaction honeypots
3. low-interaction honeypots

Pure honeypots are full-fledged production systems. The activities of the attacker are monitored by using a casual tap that has been installed on the honeypot's link to the network. No other software needs to be installed. Even though a pure honeypot is useful, stealthiness of the defense mechanisms can be ensured by a more controlled mechanism.

High-interaction honeypots imitate the activities of the production systems that host a variety of services and, therefore, an attacker may be allowed a lot of services to waste his time. By employing virtual machines, multiple honeypots can be hosted on a single physical machine. Therefore, even if the honeypot is compromised, it can be restored more quickly. In general, high-interaction honeypots provide more security by being difficult to detect, but they are expensive to maintain. If virtual machines are not available, one physical computer must be maintained for each honeypot, which can be exorbitantly expensive. Example: Honeynet.

Low-interaction honeypots simulate only the services frequently requested by attackers. Since they consume relatively few resources, multiple virtual machines can easily be hosted on one physical system, the virtual systems have a short response time, and less code is required, reducing the complexity of the virtual system's security. Example: Honeyd.

Malware honeypots

Malware honeypots are used to detect malware by exploiting the known replication and attack vectors of malware. Replication vectors such as USB flash drives can easily be verified for evidence of modifications, either through manual means or utilizing special-purpose honeypots that emulate drives. Malware increasingly is used to search for and steal cryptocurrencies,[3] which provides opportunities for services such as Bitcoin Vigil to create and monitor honeypots by using small amount of money to provide early warning alerts of malware infection.[4]

Spam versions

Spammers abuse vulnerable resources such as open mail relays and open proxies. Some system administrators have created honeypot programs that masquerade as these abusable resources to discover spammer activity. There are several capabilities such honeypots provide to these administrators and the existence of such fake abusable systems

makes abuse more difficult or risky. Honeypots can be a powerful countermeasure to abuse from those who rely on very high volume abuse (e.g., spammers).

These honeypots can reveal the apparent IP address of the abuse and provide bulk spam capture (which enables operators to determine spammers' URLs and response mechanisms). For open relay honeypots, it is possible to determine the e-mail addresses ("dropboxes") spammers use as targets for their test messages, which are the tool they use to detect open relays. It is then simple to deceive the spammer: transmit any illicit relay e-mail received addressed to that dropbox e-mail address. That tells the spammer the honeypot is a genuine abusable open relay, and they often respond by sending large quantities of relay spam to that honeypot, which stops it.[5] The apparent source may be another abused system—spammers and other abusers may use a chain of abused systems to make detection of the original starting point of the abuse traffic difficult.

This in itself is indicative of the power of honeypots as anti-spam tools. In the early days of anti-spam honeypots, spammers, with little concern for hiding their location, felt safe testing for vulnerabilities and sending spam directly from their own systems. Honeypots made the abuse riskier and more difficult.

Spam still flows through open relays, but the volume is much smaller than in 2001 to 2002. While most spam originates in the U.S.,[6] spammers hop through open relays across political boundaries to mask their origin. Honeypot operators may use intercepted relay tests to recognize and thwart attempts to relay spam through their honeypots. "Thwart" may mean "accept the relay spam but decline to deliver it." Honeypot operators may discover other details concerning the spam and the spammer by examining the captured spam messages.

Open relay honeypots include Jackpot, written in Java by Jack Cleaver; *smtpot.py*, written in Python by Karl A. Krueger;[7] and *spamhole (honeypot)|spamhole*, written in C.[8] The *Bubblegum Proxypot* is an open source honeypot (or "proxypot").[9][10]

Email trap

Main article: Spamtrap

An email address that is not used for any other purpose than to receive spam can also be considered a spam honeypot. Compared with the term "spamtrap", the term "honeypot" might be more suitable for systems and techniques that are used to detect or counterattacks and probes. With a spamtrap, spam arrives at its destination "legitimately"—exactly as non-spam email would arrive.

An amalgam of these techniques is Project Honey Pot, a distributed, open source project that uses honeypot pages installed on websites around the world. These honeypot pages disseminate uniquely tagged spamtrap email addresses and spammers can then be tracked—the corresponding spam mail is subsequently sent to these spamtrap e-mail addresses.

Database honeypot

Databases often get attacked by intruders using SQL Injection. As such activities are not recognized by basic firewalls, companies often use database firewalls for protection. Some of the available SQL database firewalls provide/support honeypot architectures so that the intruder runs against a trap database while the web application remains functional.[11]

7.3.2 Detection

Just as honeypots are weapons against spammers, honeypot detection systems are spammer-employed counterweapons. As detection systems would likely use unique characteristics of specific honeypots to identify them, a great deal of honeypots in use makes the set of unique characteristics larger and more daunting to those seeking to detect and thereby identify them. This is an unusual circumstance in software: a situation in which "versionitis" (a large number of versions of the same software, all differing slightly from each other) can be beneficial. There's also an advantage in having some easy-to-detect honeypots deployed. Fred Cohen, the inventor of the Deception Toolkit, even argues that every system running his honeypot should have a deception port that adversaries can use to detect the honeypot.[12] Cohen believes that this might deter adversaries.

7.3.3 Honeynets

Two or more honeypots on a network form a *honeynet*. Typically, a honeynet is used for monitoring a larger and/or more diverse network in which one honeypot may not be sufficient. Honeynets and honeypots are usually implemented as parts of larger network intrusion detection systems. A *honeyfarm* is a centralized collection of honeypots and analysis tools.[13][14]

The concept of the honeynet first began in 1999 when Lance Spitzner, founder of the Honeynet Project, published the paper "To Build a Honeypot":[15]

"A honeynet is a network of high interaction honeypots that simulates a production network and configured such that all

activity is monitored, recorded and in a degree, discreetly regulated."

7.3.4 Metaphor

The metaphor of a bear being attracted to and stealing honey is common in many traditions, including Germanic and Slavic. Bears were at one time called "honey eaters" instead of by their true name for fear of attracting the threatening animals. The tradition of bears stealing honey has been passed down through stories and folklore, including the well known Winnie the Pooh.[16]

7.3.5 See also

- Canary trap
- Client honeypot
- HoneyMonkey
- Honeytoken
- Network telescope
- Operation Trust
- Tarpit

7.3.6 References and notes

[1] Naveen, Sharanya. "Honeypot". Retrieved 1 June 2016.

[2] Lance Spitzner (2002). *Honeypots tracking hackers*. Addison-Wesley. pp. 68–70. ISBN 0-321-10895-7.

[3] Litke, Pat. "Cryptocurrency-Stealing Malware Landscape". *Secureworks.com*. SecureWorks. Retrieved 9 March 2016.

[4] "Bitcoin Vigil: Detecting Malware Through Bitcoin". cryptocoins news. May 5, 2014.

[5] Edwards, M. "Antispam Honeypots Give Spammers Headaches". Windows IT Pro. Retrieved 11 March 2015.

[6] "Sophos reveals latest spam relaying countries". *Help Net Security*. Help Net Security. 24 July 2006. Retrieved 14 June 2013.

[7] "Honeypot Software, Honeypot Products, Deception Software". *Intrusion Detection, Honeypots and Incident Handling Resources*. Honeypots.net. 2013. Retrieved 14 June 2013.

[8] dustintrammell (27 February 2013). "spamhole – The Fake Open SMTP Relay Beta". *SourceForge*. Dice Holdings, Inc. Retrieved 14 June 2013.

[9] Ec-Council (5 July 2009). *Certified Ethical Hacker: Securing Network Infrastructure in Certified Ethical Hacking*. Cengage Learning. pp. 3–. ISBN 978-1-4354-8365-1. Retrieved 14 June 2013.

[10] Kaushik, Gaurav; Tyagi, Rashmi (2012). "Honeypot : Decoy Server or System Setup Together Information Regarding an Attacker" (PDF). *VSRD International Journal of Computer Science & Information Technology* **2**: 155–166.

[11] "Secure Your Database Using Honeypot Architecture". www.dbcoretech.com. August 13, 2010. Archived from the original on March 8, 2012.

[12] "Deception Toolkit". *All.net*. All.net. 2013. Retrieved 14 June 2013.

[13] Nicholas Weaver, Vern Paxson, Stuart Staniford (2003). "Wormholes and a Honeyfarm: Automatically Detecting Novel Worms" (PowerPoint). *Wormholes and a Honeyfarm: Automatically Detecting Novel Worms*. The ICSI Networking and Security Group. Retrieved 14 June 2013.

[14] Honeynets a Honeynet Definition (PDF) by Ryan Talabis from PhilippineHoneynet.org

[15] "Know Your Enemy: GenII Honeynets Easier to deploy, harder to detect, safer to maintain.". *Honeynet Project*. Honeynet Project. 12 May 2005. Retrieved 14 June 2013.

[16] "The word for "bear"". *www.pitt.edu*. Retrieved 12 Sep 2014.

7.3.7 Further reading

- Lance Spitzner (2002). *Honeypots tracking hackers*. Addison-Wesley. ISBN 0-321-10895-7.

7.3.8 External links

- Distributed Open Proxy Honeypots Project: WASC
- SANS Institute: What is a Honey Pot?
- SANS Institute: Fundamental Honeypotting
- Simwood eSMS SIP Honeypot Project
- PodCast – Episode #2: "HoneyMonkeys" from Security Now!
- Project Honeypot
- The Honeynet Project

7.4 Anti-Spyware Coalition

The **Anti-Spyware Coalition** (**ASC**) was a group formed in 2005 with the goal to build a consensus about definitions and best practices in the debate surrounding spyware.

Composed of anti-spyware software companies, academics, and consumer groups, the ASC seeks to bring together a diverse array of perspective on the problem of controlling spyware and other potentially unwanted technologies.

7.4.1 History

Formed in 2005 after the dissolution of the Consortium of Anti-Spyware Technology Vendors (COAST) which broke up over internal dissent.[1] In April 2005 Ari Schwartz called together the initial group of Anti-Spyware companies; others later joined. A series of documents was published, and feedback solicited. The first set of documents consisted of a definition of spyware and potentially unwanted technologies, and a vendor dispute resolution process. This was followed by a "Risk Model" providing Anti-Spyware vendors with a framework for classifying software. In March 2007 the ASC published the public final draft of their Best Practices document.

7.4.2 References

[1] Cowley, Stacy, "Coast antispyware consortium falls apart", IDG News Service, http://www.infoworld.com/article/05/02/08/HNcoastfallsapart_1.html February 8, 2005 Archived February 28, 2007, at the Wayback Machine.

7.4.3 External links

- The Anti-Spyware Coalition
- Anti-spyware guidelines get final version

Chapter 8

Text and image sources, contributors, and licenses

8.1 Text

- **Malware** *Source:* https://en.wikipedia.org/wiki/Malware?oldid=729966237 *Contributors:* LC~enwiki, Mav, The Anome, PierreAbbat, Paul~enwiki, Fubar Obfusco, Heron, Edward, Michael Hardy, David Martland, Pnm, Liftarn, Wwwwolf, Shoaler, (, CesarB, Ellywa, David-WBrooks, CatherineMunro, Angela, Darkwind, Ciphergoth, Stefan-S, Evercat, GCarty, Etaoin, RodC, WhisperToMe, Radiojon, Tpbradbury, Bevo, Spikey, Khym Chanur, Finlay McWalter, Rossumcapek, Huangdi, Riddley, Donarreiskoffer, Pigsonthewing, Fredrik, Vespristiano, JosephBarillari, Postdlf, Rfc1394, KellyCoinGuy, DHN, Mandel, Lzur, David Gerard, Centrx, Fennec, Laudaka, Akadruid, Jtg, CarloZottmann, Mintleaf~enwiki, Everyking, Dratman, Mboverload, AlistairMcMillan, Matt Crypto, ChicXulub, Noe, Salasks, Piotrus, Quarl, Rdsmith4, Mikko Paananen, Kevin B12, Icairns, TonyW, Clemwang, Trafton, D6, Monkeyman, Discospinster, Rich Farmbrough, Guanabot, Vague Rant, Vsmith, Sperling, Night Gyr, Bender235, Sc147, JoeSmack, Elwikipedista~enwiki, Sietse Snel, Art LaPella, EurekaLott, One-dimensional Tangent, Xgravity23, Bobo192, Longhair, Billymac00, Smalljim, Unquietwiki, KBi, VBGFscJUn3, Visualize, Minghong, Hfguide, Espoo, Alansohn, Mickeyreiss, Tek022, Patrick Bernier, Arthena, T-1000, !melquiades, JeffreyAtW, Stephen Turner, Snowolf, Wtmitchell, Velella, GL, Uucp, Danhash, Evil Monkey, RainbowOfLight, Xixtas, Dtobias, Richard Arthur Norton (1958-), OwenX, Mindmatrix, Camw, Pol098, Julyo, Zhen-Xjell, Palica, Allen3, Cuvtixo, Elvey, Chun-hian, Jclemens, Reisio, Dpv, Ketiltrout, Rjwilmsi, Collins.mc, Vary, Bruce1ee, Frenchman113, PrivaSeeCrusade, Connorhd, Yamamoto Ichiro, Andrzej P. Wozniak, RainR, FlaBot, Fragglet, Intgr, Chobot, Bornhj, Bdelisle, Cshay, Gwernol, RogerK, Siddhant, YurikBot, Wavelength, RattusMaximus, Aussie Evil, Phantomsteve, Ikester, RussBot, DMahalko, TheDoober, Coyote376, Ptomes, Wimt, Thane, NawlinWiki, Hm2k, Krystyn Dominik, Trovatore, Cleared as filed, Coderzombie, Kingpomba, Ugnius, Amcfreely, Voidxor, Tony1, Alex43223, FlyingPenguins, Chriscoolc, Bota47, Groink, Yudiweb, User27091, Tigalch, Flipjargendy, Romal, American2, Nikkimaria, Theda, Closedmouth, Abune, PrivaSeeCrusader, Dspradau, GraemeL, Cffrost, RealityCheck, RenamedUser jasklddjslak904, Allens, Jasón, NeilN, Mhardcastle, MacsBug, SmackBot, ManaUser, Mmernex, Hal Canary, Hydrogen Iodide, Bigbluefish, WookieInHeat, Stifle, KelleyCook, Bobzchemist, Ericwest, Ccole, HalfShadow, Yamaguchi??, Gilliam, Ohnoitsjamie, Skizzik, PJTraill, Larsroe, Appelshine, Father McKenzie, Jopsen, Jprg1966, Thumperward, Pylori, Mitko, Fluri, SalemY, Ikiroid, Whispering, Ted87, Janipewter, Audriusa, Furby100, Berland, JonHarder, Rrburke, DR04, Mr.Z-man, Radagast83, Cybercobra, Warren, HarisM, URLwatcher02, Drphilharmonic, DMacks, Fredgoat, Salamurai, Kajk, Pilotguy, Clicketyclack, Ged UK, MaliNorway, CFLeon, Howdoesthiswo, Vanished user 9i39j3, Mike1901, Gobonobo, Robofish, Xaldafax, Nagle, Fernando S. Aldado~enwiki, Voceditenore, Ian Dalziel, 16@r, Andypandy.UK, Slakr, NcSchu, Ehheh, Uuhuų, Vernalex, Dl2000, Andreworkney, Fan-1967, Tomwood0, Iridescent, Dreftymac, NativeForeigner, Mere Mortal, UncleDouggie, Cbrown1023, Jitendrasinghjat, Astral9, Mzub, JForget, DJPhazer, Durito, FleetCommand, Americasroof, Powerpugg, Wikkid, Hezzy, JohnCD, Kris Schnee, Jesse Viviano, Xxovercastxx, Kejoxen, Augrunt, Nnp, TheBigA, Cydebot, MC10, Besieged, Gogo Dodo, Pascal.Tesson, Medovina, Shirulashem, DumbBOT, Chrislk02, Optimist on the run, Kozuch, Drewjames, Vanished User jdksfajlasd, Tunheim, Legotech, Thijs!bot, Epbr123, Crockspot, Wikid77, Pstanton, Oldiowl, Technogreek43, Kharitonov, Headbomb, A3RO, Screen317, James086, SusanLesch, Dawnseeker2000, Escarbot, AntiVandalBot, Widefox, Seaphoto, Михајло Анђелковић, SummerPhD, Quintote, Mack2, Storkk, Golgofrinchian, JAnDbot, Mac Lover, Andonic, Entgroupzd, MSBOT, Geniac, Bongwarrior, VoABot II, Kinston eagle, Tedickey, Sugarboogy phalanx, Nyttend, Rich257, Alekjds, 28421u2232nfenfcenc, LorenzoB, DerHexer, MKS, Calltech, XandroZ, Gwern, Kiminatheguardian, Atulsnischal, ClubOranje, MartinBot, CliffC, Ct280, BetBot~enwiki, Aladdin Sane, R'n'B, LedgendGamer, J.delanoy, Svetovid, Phoenix1177, Herbythyme, A Nobody, Jaydge, Compman12, Fomalhaut71, Freejason, Demizh, HiLo48, Chiswick Chap, Kraftlos, Largoplazo, Cometstyles, Tiggerjay, Robert Adrian Dizon, Tiangua1830, Bonadea, Jarry1250, RiseAgainst01, Sacada2, Tfraserteacher, VolkovBot, Jeff G., Nburden, Satani, Wes Pacek, Philip Trueman, Teacherdude56, TXiKiBoT, Oshwah, Floddinn, Muro de Aguas, Zifert, PoM187, Rei-bot, Retiono Virginian, Jackfork, LeaveSleaves, Optigan13, Miketsa, BotKung, Wewillmeetagain, Tmalcomv, Blurpeace, RandomXYZb, Digita, LittleBenW, Logan, Fredtheflyingfrog, S.Örvarr.S, Adaviel, Copana2002, Tom NM, Nubiatech, LarsHolmberg, Sephiroth storm, Yintan, Calabraxthis, Xelgen, Arda Xi, Bentogoa, Happysailor, Flyer22 Reborn, Jojalozzo, Nnkx00, Nosferatus2007, Evaluist, Miniapolis, Lightmouse, Helikophis, Correogsk, Stieg, Samker, Jacob.jose, BfMGH, Dabomb87, Denisarona, Ratemonth, Martarius, ClueBot, Muhammadsb1, NickCT, Vítor Cassol, The Thing That Should Not Be, VsBot, Lawrence Cohen, Wysprgr2005, Frmorrison, Jwihbey, Sam Barsoom, Ottava Rima, Paulcmnt, Excirial, Jusdafax, Dcampbell30, Rhododendrites, Ejsilver26, WalterGR, 7&6=thirteen, Maniago, Jaizovic, Dekisugi, Xme, DanielPharos, Versus22, Callinus, Johnuniq, Rossen4, DumZiBoT, Darkicebot, XLinkBot, BodhisattvaBot, DaL33T, Avoided, Sogle, Mifter, Noctibus, CalumH93, Kei Jo, Ad-

dbot, Xp54321, Cxz111, Arcolz, Mortense, A.qarta, Otisjimmy1, Crazysane, TutterMouse, Ashton1983, CanadianLinuxUser, Leszek Jańczuk, T38291, Noozgroop, CactusWriter, MrOllie, Download, LaaknorBot, Glane23, Ld100, AndersBot, Jasper Deng, Tassedethe, Evildeathmath, Tide rolls, Krano, Teles, Gail, Jarble, Quantumobserver, Crt, Legobot, आशीष भटनागर, Publicly Visible, Luckas-bot, Yobot, Philmikeyz, WikiDan61, Tohd8BohaithuGh1, Ptbotgourou, Fraggle81, Evans1982, Gjohnson9894, Dmarquard, AnomieBOT, DemocraticLuntz, Rubinbot, Roman candles, IRP, Galoubet, RandomAct, Materialscientist, CoMePrAdZ, RevelationDirect, Crimsonmargarine, Frankenpuppy, ArthurBot, Quebec99, Cameron Scott, Xqbot, TheAMmollusc, Mgaskins1207, Capricorn42, 12056, Avastik, Christopher Forster, Masonaxcte, S0aasdf2sf, Almabot, GrouchoBot, Monaarora84, Shirik, RibotBOT, PM800, Dougofborg, Luminique, GliderMaven, Afromayun, Fingerz, FrescoBot, Vikasahil, Mitravaruna, Wikipe-tan, Sky Attacker, PickingGold12, Wouldshed, TurningWork, Jonathansuh, Romangralewicz, HamburgerRadio, Citation bot 1, SL93, Uberian22, Bobmack89x, Pinethicket, I dream of horses, Idemnow, Vicenarian, Rameshngbot, RedBot, Overkill82, Serols, Chan mike, Fumitol, Graham france, Javanx3d, FoxBot, Tgv8925, TobeBot, SchreyP, FFM784, Jesus Presley, TheStrayCat, Neutronrocks, Techienow, Lotje, Wikipandaeng, Dinamik-bot, Vrenator, Mirko051, Aoidh, KP000, Simonkramer, Tbhotch, Lord of the Pit, DARTH SIDIOUS 2, Whisky drinker, Mean as custard, Moshe1962, RjwilmsiBot, Colindiffer, Panda Madrid, Thunder238, Salvio giuliano, Enauspeaker, EmausBot, John of Reading, WikitanvirBot, The Mysterious El Willstro, Winner 42, Wikipelli, K6ka, Porque123, Elison2007, SocialAlex, Connoe, AvicBot, FlippyFlink, Bollyjeff, Dffgd, Azuris, H3llBot, EneMsty12, Simorjay, Paramecium13, Tyuioop, Tolly4bolly, Cit helper, Coasterlover1994, Techpraveen, Rigley, Capnjim123, Ankitcktd, Ego White Tray, Mv Cristi, Pastore Italy, Corb555, Mark Martinec, Helpsome, ClueBot NG, Cwmhiraeth, Rich Smith, PizzaMuncherMan, Lzeltser, PwnFlakes, Matthiaspaul, MelbourneStar, Catlemur, Satellizer, Piast93, Steve dexon, MarsTheGrayAdept, Mesoderm, Widr, Brandobraganza, Rubybarett, BBirke, Calabe1992, BG19bot, Krenair, Harmonicsonic, Vagobot, Sailing to Byzantium, Maln98, PatrickCarbone, Hopsmatch, MusikAnimal, Mark Arsten, Michael Barera, Forbo, JZCL, 220 of Borg, AntanO, BattyBot, JC.Torpey, Spazz rabbit, Cimorcus, Zhaofeng Li, MahdiBot, Stigmatella aurantiaca, Paully72, BYagour, Verzer, Antonio.chuh, ChrisGualtieri, Tech77, Hassim1983, Hnetsec, MadGuy7023, JYBot, Ghostman1947, Toffeepot, EagerToddler39, Dexbot, FoCuSandLeArN, FTLK, Joshy5.crane, Dothack111, Skullmak, 967Bytes, Webbanana1, Rajalakshmi S Kothandaraman, Salwanmohsen, TwoTwoHello, CoLocate, Lugia2453, Himanshu Jha 07, Frosty, UNOwenNYC, SirkusSystems, Sourov0000, Corn cheese, Palmbeachguy, Gtangil, Epicgenius, Dr Dinosaur IV, JimsWorld, Flashgamer001, Camayoc, Msumedh, Kap 7, Nonsenseferret, Bio pox, RonPaul573e, Madsteve 9, Kogmaw, Tankman98, Olivernina, ExtraBart, 38zu.cn, Mooman4158, DavidLeighEllis, Warl0ck, Kharkiv07, Ugog Nizdast, Melody Lavender, Swiftsectioner, Ginsuloft, Chris231989, Wifi router rootkits, Ban embedded cpus for networks, JohnMadden2009, Someone not using his real name, Wifi wiretapping, BornFearz, Dannyruthe, Noyster, FockeWulf FW 190, Ajitkumar.pu, JaconaFrere, Vandraag, SnoozeKing, Newlywoos, Gatundr Burkllg, Alan24308, Worzzy, Lucius.the, Andrei Marzan, Joeleoj123, MLP Eclipse, Stevebrown164, Qwertyxp2000, Andrdema, Dcgh96, A4frees, 1anamada, AnastasiiaGS, Maxwell Verbeek, Oiyarbepsy, WikiGopi, Eurodyne, Shovelhead54, Spamero, Fimatic, Hvaara, Friendshipbracelet, Rubbish computer, Gabiked, ToonLucas22, Nysrtup, Davenill, Akstotz, M8b8f8, Supdiop, KasparBot, Tmar877, Marzan Chowdhury, JJMC89, Jasmineluv, ROMA1maro, Albaniakucov, Shreyassachdeva, Hearmewe, Salarmfd, Chenthil Vel, Kosherpenguinexperience, Phils234, Srilekha selva, Sharanyanaveen, Eno Lirpa, Pranjal01, Atomicselection, Lmiller 22 and Anonymous: 912

- **Computer virus** *Source:* https://en.wikipedia.org/wiki/Computer_virus?oldid=730655930 *Contributors:* Damian Yerrick, AxelBoldt, Peter Winnberg, The Cunctator, LC~enwiki, Lee Daniel Crocker, Brion VIBBER, Mav, Bryan Derksen, The Anome, Taw, Taral, Malcolm Farmer, Tim Chambers, Mark Ryan, Dragon Dave, Greg Lindahl, Fubar Obfusco, William Avery, SimonP, Ben-Zin~enwiki, Heron, Modemac, Frecklefoot, Edward, Michael Hardy, Nixdorf, Pnm, Liftarn, Smkatz, Wwwwolf, Ixfd64, Cyde, TakuyaMurata, Minesweeper, Egil, Ahoerstemeier, Jdforrester, Darkwind, Stefan-S, Rossami, Nikai, BOARshevik, Cratbro, Evercat, Smaffy, Rob Hooft, Samuel~enwiki, GRAHAMUK, Hashar, Dblaisdell, Adam Bishop, Dcoetzee, RickK, Dysprosia, Doradus, WhisperToMe, Zoicon5, Tpbradbury, Jake Nelson, Furrykef, Morwen, SEWilco, Omegatron, Morven, Wetman, Chrisjj, Pakaran, Secretlondon, Jamesday, Rossumcapek, Huangdi, David Stapleton, Chuunen Baka, Gromlakh, Robbot, Paranoid, Sander123, Fredrik, Kizor, RedWolf, Bernhard Bauer, Romanm, Kokiri, Lowellian, Merovingian, Academic Challenger, Goofyheadedpunk, Premeditated Chaos, DHN, Jondel, Hadal, Kostiq, Mattflaschen, Cordell, Carnildo, Tea2min, David Gerard, Baloo Ursidae, Giftlite, DocWatson42, Marius~enwiki, Fennec, Jtg, Lethe, Tom harrison, Ferkelparade, Frevidar, Ayman, Noone~enwiki, Ds13, Guanaco, Patrickdavidson, Dmmaus, Mboverload, Siroxo, AlistairMcMillan, Matt Crypto, Adam McMaster, Utcursch, Knutux, Sonjaaa, Antandrus, Beland, Chinakow, Tbjablin, Wkdewey, Bumm13, Kevin B12, Thevaliant, Sam Hocevar, Cynical, Sillydragon, Crazyeddie, Nataliesiobhan, Joyous!, Jcw69, MakeRocketGoNow, Trafton, Grm wnr, Chmod007, M1ss1ontomars2k4, Kate, Mike Rosoft, Mormegil, Freakofnurture, Monkeyman, Imroy, Jiy, Discospinster, Solitude, Rich Farmbrough, Agnistus, Rhobite, Qwerty1234, EliasAlucard, Lemontea, Stereotek, Bender235, Rubicon, ESkog, ZeroOne, TerraFrost, CMC, JoeSmack, MisterSheik, CanisRufus, Ginnsu, El C, Joanjoc~enwiki, PhilHibbs, Shanes, Sietse Snel, RoyBoy, Jpgordon, Rpresser, Bobo192, Vanished user sdfkjertiwoi1212u5mcake, Stesmo, Smalljim, Clawson, Brendansa, John Vandenberg, Flxmghvgvk, Orbst, Jjk, La goutte de pluie, Jojit fb, Minghong, John Fader, Obradovic Goran, Wrs1864, Sam Korn, Pearle, Benbread, Jakew, Wayfarer, Alan Isherwood, Knucmo2, Jumbuck, Storm Rider, Alansohn, Andrewpmk, Riana, AzaToth, Kurt Shaped Box, Goldom, T-1000, Phocks, Hu, Malo, VladimirKorablin, GregLindahl, Snowolf, Marianocecowski, PaePae, Wtmitchell, Tocsin, Fordan, TaintedMustard, Gdavidp, Paul1337, Docboat, Evil Monkey, TrollVandal, LFaraone, Bsadowski1, SteinbDJ, Alai, LukeSurl, Gofeel, Dan100, Ceyockey, Umapathy, Bruce89, Tom.k, Boothy443, Firsfron, Alvis, Reinoutr, Roboshed, Octernion, RHaworth, Uncle G, Kurzon, MattGiuca, Pol098, MONGO, Miss Madeline, Nfearnley, Jok2000, Wayward, Prashanthns, Dysepsion, Seishirou Sakurazuka, Graham87, Marcg106, Magister Mathematicae, Cuchullain, CivilCasualty, Kbdank71, FreplySpang, JIP, Jclemens, Reisio, Grammarbot, Zoz, Spot Color Process, Ketiltrout, Sjakkalle, Rjwilmsi, Whatcanbrowndo, Xgamer4, Koavf, Isaac Rabinovitch, Tarnas, Kinu, Adjusting, Collins.mc, Commander, Astronaut, Rillian, Tangotango, Bruce1ee, Authr, Raffaele Megabyte, Captain Disdain, ErikHaugen, Gudeldar, Frenchman113, Darksasami, Bubba73, Jdmurray, GregAsche, Sango123, DirkvdM, Antimatt, Leithp, Audunv, Munahaf, RainR, FlaBot, RobertG, Winhunter, Crazycomputers, JiFish, Nivix, RexNL, Gurch, Alexjohnc3, DevastatorIIC, Born2cycle, BitterMan, SteveBaker, Ahunt, Imagine&Engage, Superdude876, Chobot, Visor, Bornhj, Random user 39849958, Digitalme, Peterl, YurikBot, Wavelength, Sceptre, Hairy Dude, Huw Powell, Adam1213, Sigeng, Jtkiefer, WAvegetarian, Anonymous editor, SpuriousQ, Ukdragon37, RadioFan, Stephenb, Gaius Cornelius, CambridgeBayWeather, Wimt, Bullzeye, Lusanaherandraton, Anomalocaris, Shanel, NawlinWiki, Wiki alf, Bachrach44, Dialectric, Tfine80, Jaxl, Harksaw, Dureo, Robert McClenon, Nick, Coderzombie, Raven4x4x, Ugnius, Froth, Amcfreely, FatM1ke, Zwobot, Slaad, Dlyons493, Brucevdk, Pablomartinez, DeadEyeArrow, Bota47, Haemo, JoshuaArgent, Romal, Wknight94, Jcvamp, Mugunth Kumar, FF2010, Zero1328, K.Nevelsteen, Zzuuzz, Imaninjapirate, Theda, Closedmouth, Spondoolicks, Dspradau, BorgQueen, Petri Krohn, GraemeL, Rlove, JoanneB, Mario23, Pursin1, Chrishmt0423, SingDeep, LeonardoRob0t, Rex Nebular, Scoutersig, Amren, Kevin, AVazquezR, Spliffy, Curpsbot-unicodify, RunOrDie, RG2, Eptin, Tyomitch, GrinBot~enwiki, DVD R W, Theroachman, Rahul s55, ???? robot, So Hungry, SpLoT, Veinor, SmackBot, MattieTK, Fireworks, Thomas Ash, Khfan93, DreamTheEndless, KnowledgeOfSelf, TestPilot, Hydrogen Iodide, Aborlan, Chairman S., Frymaster, Ericwest, Onebravemonkey, Born2killx, Xaosflux, PeterSymonds, Macintosh User, Gilliam, Algont, Ohnoitsjamie, Oscarthecat, Skizzik, MikeVella, ERcheck,

Bluebot, Hayson1991, Rkitko, Green meklar, MalafayaBot, Bolmedias, Danielmau, Deli nk, Jerome Charles Potts, EdgeOfEpsilon, DHN-bot~enwiki, The Moose, Sbharris, Hongooi, Brucedes, Antonrojo, Firetrap9254, Cancaseiro, Can't sleep, clown will eat me, Frap, Ultra-Loser, Onorem, Avb, KevM, JonHarder, Yidisheryid, TKD, Manhatten Project 2000, Addshore, SundarBot, Name? I have no name., Maurice45, Sspecter, Krich, PiMaster3, Zrulli, Khukri, Decltype, Nakon, Savidan, Birdfluboy, TedE, Dreadstar, Warren, Insineratehymn, Weregerbil, Last Avenue, Hammer1980, RedViking, RichAromas, Wizardman, Ultraexactzz, Sigma 7, Ck lostsword, Pilotguy, DataGigolo, Kukini, Ged UK, Ugur Basak Bot~enwiki, The undertow, SashatoBot, TjOeNeR, Daishokaioshin, Acebrock, Srikeit, SS2005, Vanished user 9i39j3, Kuru, John, Edetic, Wtwilson3, Slowmover, Sfivaz, Lazylaces, Sir Nicholas de Mimsy-Porpington, Minna Sora no Shita, CaptainVindaloo, Aleenf1, Iron-Gargoyle, Cbk1994, PseudoSudo, Ckatz, A. Parrot, AFOH, Andypandy.UK, Slakr, CommKing, Beetstra, Muadd, Boomshadow, LuYiSi, Martinp23, Mr Stephen, Gerardsylvester, Optakeover, Waggers, Doczilla, Riffic, Elb2000, Evadb, Ryanjunk, Zyborg, Vtt395, KJS77, Levineps, Alan.ca, BranStark, Fredil Yupigo, ILovePlankton, Rigurat, Twas Now, MikeHobday, Courcelles, Astral9, Coffee Atoms, Fdp, Mzub, Tawker-bot2, Prophaniti, Chris55, Emote, Bleavitt, SkyWalker, Weird0, FleetCommand, Jorcoga, Ninetyone, KyraVixen, JohnCD, Jesse Viviano, Sree v, Leujohn, Casper2k3, Kejoxen, Karenjc, Ricecake42, Nmacu, TJDay, Inzy, Cydebot, Ntsimp, Herd of Swine, Fl, Steel, Meno25, Michaelas10, Gogo Dodo, Hebrides, JFreeman, DarthSidious, Dancter, Tawkerbot4, DumbBOT, Eneville, Chrislk02, Phydend, Asenine, Optimist on the run, Omicronpersei8, Daniel Olsen, Lo2u, Bcohea, Gimmetrow, Dartharias, DarkMasterBob, Epbr123, Hervegirod, Computafreak, Sagaciousuk, Andyjsmith, Ambulnick, Glennfcowan, Azkhiri, Hunan131, Oliver202, Philsp, Luigifan, Jojan, Radiokillplay, James086, Doyley, Java13690, Dddddgggg, Brainbox 112, Renamed user 5197261az5af96as6aa, Dfrg.msc, Philippe, Big Bird, Natalie Erin, Oldmanbiker, Dainis, Dzubint, Dantheman531, Mentifisto, Porqin, AntiVandalBot, Fedayee, Luna Santin, Widefox, Ibigenwald lcaven, MHoover, Oducado, Quintote, Jayron32, Qa Plar, TimVickers, Scepia, Petey21, Malcolm, Chill doubt, Ayrin83, Storkk, Myanw, Darrenhusted, Leuko, Kigali1, MER-C, Epeefleche, Avaya1, Jolmos, OhanaUnited, Tengfred, Xeno, Mullibok, Berkeley@gmail.com, Bookinvestor, Coopercmu, LittleOldMe, Acroterion, SteveSims, Tarif Ezaz, Akuyume, Magioladitis, WolfmanSF, KeKe, Pedro, Bongwarrior, VoABot II, MartinDK, Dragon Dan, Jetstreamer, Jrg7891, Master2841, Think outside the box, EhUpMother, Brewhaha@edmc.net, Twsx, Shankargiri, Avicennasis, Midgrid, Bubba hotep, Catgut, Aishwarya .s, Web-Crawling Stickler, Adrian J. Hunter, Allstarecho, Canyouhearmenow, Spellmaster, Omkarcp3, Davis W, Glen, Chris G, DerHexer, Eeera, Markco1, Glennforever, Connor Behan, Alu042, Leliro19, 0612, Murraypaul, Stephenchou0722, Stealthound, Hdt83, MartinBot, CliffC, NAHID, Poeloq, Rhlitonjua, Rettetast, Mike6271, Draknfyre, Keith D, Bemsor, Jonathan Hall, Kostisl, R'n'B, Smial, WelshMatt, DarkGhost89, Siliconov, ItsProgrammable, Artaxiad, Paranomia, J.delanoy, MRFraga, Legitimate Editor, Carre, Rgoodermote, Bitethesilverbullet, Inspigeon, David sancho, Public Menace, MrBell, Eliz81, 12dstring, RAF Regiment, Acalamari, Xbspiro, Ravi.shankar.kgr, JFKenn, Timpeiris, SpigotMap, Compman12, Crakkpot, Skier Dude, Evils Dark, Gurchzilla, JayJasper, Bilbobee, Bubble94, Chriswiki, Psynwavez, HiLo48, Kmoe333, Drahgo, TomasBat, NewEnglandYankee, Srpnor, Dougmarlowe, Rwessel, Hennessey, Patrick, Hanacy, Darthpickle, Jackaranga, Black Hornet, Darr dude, RB972, Jamesontai, Jleske, GGenov, Vanished user 39948282, Anonymouseuser, Sniper120, Kamote321, Gtg204y, RVJ, Jarry1250, Jay.rulzster, Squids and Chips, Specter01010, Praesidium~enwiki, Leethax, Wikieditor06, BauerJ24, Sacada2, Javeed Safai, VolkovBot, TreasuryTag, CWii, Bluegila, Jeff G., Alegjos, Philip Trueman, Qevlarr, TXiKiBoT, Oshwah, NoticeBored, BuickCenturyDriver, Technopat, Hqb, Yuma en, Miranda, Crazypete101, Lord Vaedeon, JayC, Qxz, Shindo9Hikaru, Mattinnson, WraithTDK, Rfcrossp, Anna Lincoln, Codyjames7, Lradrama, Sanjivdinakar, Xaraikex, Tricky Wiki44, JhsBot, Phillyfan1111, Mind23, LeaveSleaves, Tpk5010, Peter Dontchev, Miketsa, Air-con noble, David in DC, Wiae, Csdorman, Da31989, Slipknotmetal, Madhero88, Lolroflomg, Jacky15, Aryeh Grosskopf, Lerdthenerd, Haseo9999, SQL, Tikuko, Gorank4, Falcon8765, Purgatory Fubar, LarsBK, Brianga, Mike4ty4, Bobo The Ninja, LittleBenW, AlleborgoBot, Kehrbykid, FlyingLeopard2014, Kpa4941, Wraithdart, ChipChamp, Repy, Sepetro, Copana2002, Anindianblogger, SieBot, Coffee, J800eb, Dino911, YonaBot, Euryalus, WereSpielChequers, Dawn Bard, Eagleal, BloodDoll, Triwbe, Kkrouni, Bmader, Sephiroth storm, Falcofire, GrooveDog, Jerryobject, Keilana, Flyer22 Reborn, The Evil Spartan, Man It's So Loud In Here, Arbor to SJ, Travis Evans, Askild, Wheres my username, Oxymoron83, Antonio Lopez, KPH2293, Timothy Jacobs, Hobartimus, OKBot, Dillard421, Benji2210, Maelgwnbot, Vice regent, MichaelIvan, Mminocha, Ecthelion83, Miketoloo, Treekids, Jkonline, Lloydpick, Escape Orbit, Into The Fray, Guitaralex, ImageRemovalBot, Martarius, Humpet, ClueBot, Fyyer, The Thing That Should Not Be, TableManners, Voxpuppet, WakaWakaWoo20, Jan1nad, Seriousch, Jotag14, Wysprgr2005, Likepeas, Meekywiki, VQuakr, Mild Bill Hiccup, Boing! said Zebedee, Moniquehls, CounterVandalismBot, Jakebob2, Tumland, Blanchardb, LizardJr8, Bjbutton, Hitherebrian, Otolemur crassicaudatus, MrBosnia, Bfmv-rulez, Puchiko, Rockfang, MindstormsKid, Gunnar Kreitz, DragonBot, Campoftheamericas, Excirial, Alexbot, Jusdafax, PixelBot, Winston365, Lartoven, Posix memalign, Rhododendrites, Cenarium, WalterGR, TheRedPenOfDoom, Tnxman307, Hans Adler, Frozen4322, SchreiberBike, Arvind007, ChrisHodgesUK, BOTarate, El bot de la dieta, DanielPharos, Thingg, Aitias, Scalhotrod, Jester5x5, Versus22, PCHS-NJROTC, MelonBot, Vybr8, Qwfp, Johnuniq, Jpg1954, Egmontaz, Apparition11, Editor2020, SF007, Runefrost, DumZiBoT, Bones000sw, Gkaukonen, TheNameWithNoMan, XLinkBot, Stickee, Rror, Feinoha, Poosebag, Avoided, Mitch Ames, Skarebo, WikHead, SilvonenBot, Me, Myself, and I, SkyLined, Airplaneman, Imapoo, Thatguyflint, Nesky~enwiki, Wnzrf, CalumH93, Flixmhj321, Mojibz, Pacific ocean stiller ocean, Addbot, Xp54321, BreannaFirth, Nuno Brito, Willking1979, Clsdennis2007, Giftiger wunsch, StickRail1, DougsTech, Nz26, Ur nans, AlexWangombe, Ronhjones, Justinpwnz11233, Mr. Wheely Guy, GyroMagician, Ethanpet113, Ashton1983, Rynhom44, CanadianLinuxUser, Leszek Jańczuk, NumbNull, Download, Chamal N, CarsracBot, RTG, Glane23, AndersBot, Sumbuddi, Favonian, Comphelper12, Jasper Deng, 5 albert square, ACM2, Savetheozone, Last5, Tassedethe, Lolbombs, The hippy nerd, Tide rolls, Bfigura's puppy, Lightbot, C933103, Gail, Bro0010, Micke, Maverick1071, GloomyJD, Alexd18, Legobot, Worldbruce, Senator Palpatine, DisillusionedBitterAndKnackered, Washburnmav, Taxisfolder, THEN WHO WAS PHONE?, Golftheman, KamikazeBot, Theornamentalist, Kablakang, ?????, Bility, Kmoultry, AnomieBOT, AmritasyaPutra, Valueyou, La Corona, Rubinbot, 1exec1, ThaddeusB, Arzanish, Jim1138, Galoubet, Dwayne, Danielt998, AdjustShift, Kingpin13, Materialscientist, Racconish, Cameron Scott, Xqbot, Spidern, Cureden, What!?Why?Who?, Gilo1969, S0aasdf2sf, Coretheapple, Sci-Fi Dude, Basvb, Frosted14, Shas1194, Ute in DC, Wizardist, ProtectionTaggingBot, Shirik, Mark Schierbecker, ReformatMe, Amaury, Drvikaschahal, 1nt2, Aurush kazemini, N419BH, Mrbillgates, Lelapindore, Erik9, Sesu Prime, Legobot III, FreeKnowledgeCreator, Komitsuki, Kitaure, Jono20201, Zero Thrust, Scott A Herbert, HamburgerRadio, Intelligentsium, Pinethicket, I dream of horses, Elockid, HRoestBot, Calmer Waters, Bejinhan, RedBot, Phearson, Serols, HaiyaTheWin, Footwarrior, Lineslarge, Reconsider the static, IJBall, Caramelldansener, 3centsoap, Xeroxli, FoxBot, Mercy11, Trappist the monk, LogAntiLog, Lotje, Callanecc, Vrenator, Singlemaltscotch, Zacker150, Extra999, January, Rudy16, Chimpso, Jeffrd10, Nazizombies!, Mufc ftw, Canuckian89, Suffusion of Yellow, Richardsaugust, Tbhotch, TheMesquito, DARTH SIDIOUS 2, Onel5969, Mean as custard, The Utahraptor, RjwilmsiBot, Bento00, Fletcher707, Regancy42, DRAGON BOOSTER, Chroniccommand, Ghymes21, NerdyScienceDude, Deagle AP, DASHBot, Deadlyops, EmausBot, Immunize, Never give in, Ajraddatz, Sevvie, Zerotonin, Dewritech, Jujitsuthirddan, Ben10Joshua, Olof nord, RenamedUser01302013, Vanished user zq46pw21, Elee, Huctwitha, Tommy2010, Wikipelli, Soadfan112, K6ka, Kurenai96, Papelbon467, Elison2007, Hejerik, Rajnish357, AvicBot, ZéroBot, Rhinestone42, Checkingfax, Fæ, Josve05a, Davemaccc, Bryce Carmony, Sgerbic, Kiwi128, Alxndrpaz, Imtoocool9999, Mhammad-alkhalaf, Skate4life18, 2han3, Pan Brerus, K kisses, Bawx, Tyuioop, Jy0Gc3, Wayne

Slam, Yabba67, Openstrings, Quantumor, Techpraveen, Donner60, Autoerrant, Bomazi, Mv Cristi, ChuispastonBot, GermanJoe, Pastore Italy, Dylan Flaherty, Czeror, SlowPhoton, Neil P. Quinn, Brad117, Mikitei, DASHBotAV, Fungo4shezzo, Myfunkybear123456789, ClueBot NG, Horoporo, Matthiaspaul, This lousy T-shirt, Baseball Watcher, Mesoderm, Rezabot, Masssly, Rukario-sama, Widr, Anupmehra, Helpful Pixie Bot, Alexbee2, Lowercase sigmabot, BG19bot, Janendra, Kaltenmeyer, PatrickCarbone, AvocatoBot, Jobin RV, Tobias B. Besemer, Eman2129, Tony Tan, Agentl90, Napsync, BattyBot, Biosthmors, Justincheng12345-bot, David.moreno72, Smileguy91, Cyberbot II, Codeh, Arcandam, Orioncaspar, Dexbot, Cwobeel, Codename Lisa, 967Bytes, Makecat-bot, Lugia2453, Frosty, Arrayoutofbounds, Andyhowlett, Athomeinkobe, Cdwn, Cadillac000, Palmbeachguy, Epicgenius, Jamesmcmahon0, Ayush 691, Melonkelon, EvergreenFir, Ronburgandee, ElHef, DavidLeighEllis, QPT, Babitaarora, Sam Sailor, Dannyruthe, Averruncus, FockeWulf FW 190, Digiwyse, JaconaFrere, Lakun.patra, 32RB17, Monkbot, Horseless Headman, Happy Attack Dog, Thisiswikidome, Pantel, Jazwal, Shuaibshaikh84, Virus victem, Amortias, Shahean Cozad, NQ, The Last Arietta, Suspender guy, Edityouranswer11, ChamithN, Crystallizedcarbon, WikiGopi, Cpt Wise, Vivek56, Charlotte Churche, অনন্দিময়, Abistevenson22, Tech Aid, Pishcal, Juned kadri, IEditEncyclopedia, Jalen stephens, Wredants, Selim Abou Rahal, Chandrayadav227, Deanwalt123, Hamza190949, Miniredeyes999, GeneralizationsAreBad, Ilikecakeandstuff, Bankofworld, KasparBot, Seanpatrickgray, Astolf0, Gautamnarayan, MCFinest Anthony, Mac1817;-), CJwar, Santhosh Sankar, Muttaiyab Ahmad, JJMC89, TheGlatiator, DelugeRPG, Entro3.14, PCMarcLondon, Viney P Sunu, Ameya Kanojia, CAPTAIN RAJU, Flashgernade99, Mb66w, Imdnj, Koopa King 125, Chenthil Vel, Sharanyanaveen, Wikipedia Virus Page, Naeem chakera, Expert Computers, Promestein, Suresuba, John.smithm, GreenC bot, Wpiuaaa, Ardral, Chickadee46, Msearce, Sprops, Joshuambruck1 and Anonymous: 1992

- **Comparison of computer viruses** *Source:* https://en.wikipedia.org/wiki/Comparison_of_computer_viruses?oldid=730616859 *Contributors:* Magnus Manske, Bryan Derksen, AlexWasFirst, Danny, SimonP, Ahoerstemeier, Nikai, Timwi, Jdstroy, WhisperToMe, SD6-Agent, Robbot, Kizor, Yelyos, Auric, Cyrius, Sj, Jtg, Cool Hand Luke, Marcika, Frencheigh, Jfdwolff, ALargeElk, Wmahan, Gubbubu, Gadfium, Jcw69, Trafton, Trevor MacInnis, Noisy, Twinxor, Will2k, ZeroOne, JoeSmack, Neko-chan, Smalljim, Reinyday, A-Day, Minghong, Arthena, Woohookitty, Peng~enwiki, Huhsunqu, Bluemoose, ??????, Fxer, Seishirou Sakurazuka, The wub, Ahunt, Mhking, YurikBot, Wavelength, Tznkai, Member, NawlinWiki, Dialectric, Drivec, Dbfirs, DeadEyeArrow, Romal, Caballero1967, Curpsbot-unicodify, Kungfuadam, Rwwww, CrniBombarder!!!, MacsBug, Gjs238, Gilliam, Renamed user Sloane, A. B., Reaper X, Mulder416, Frap, Whpq, Lambiam, ArglebargleIV, Writtenonsand, Optakeover, HelloAnnyong, Ale jrb, ONUnicorn, TempestSA, Cydebot, Christian75, Sirmylesnagopaleentheda, Wikid77, Deadbeef, Hut 8.5, JamesBWatson, Jéské Couriano, BigChicken, Catgut, Fiasco229, Jim.henderson, Rettetast, J.delanoy, Uncle Dick, Cometstyles, Bonadea, VolkovBot, Hersfold, Philip Trueman, RiverStyx23, Glob.au, Jerryobject, Cyfal, Bughunter2, Mr. Stradivarius, ClueBot, Mod.torrentrealm, The Thing That Should Not Be, Cptmurdok, Jusdafax, WalterGR, Aitias, Vybr8, Skarebo, WikHead, Thatguyflint, Addbot, Melab-1, Tothwolf, CactusWriter, Chzz, 5 albert square, The hippy nerd, Fraggle81, Materialscientist, Gsmgm, Melmann, DJWolfy, Alexbowden, S0aasdf2sf, FrescoBot, HamburgerRadio, Richard, Teaearlygreyhot, DARTH SIDIOUS 2, TechMizer, Grexx, EmausBot, NotAnonymous0, Jasonanaggie, ZéroBot, K kisses, Richardt8649, Donner60, EdoBot, WHZhang, Petrb, ClueBot NG, Gareth Griffith-Jones, MelbourneStar, O.Koslowski, ScottSteiner, Widr, IISignoreDeiPC, Umarhattab, Sni56996, Ajaxfiore, 2Flows, Mogism, Smsubham, AlbertJB, Lugia2453, Graphium, SirkusSystems, SteenthIWbot, 1DAdrianElmer07, Sourov0000, DrMerry, Yea55, Msumedh, Tentinator, Blackbombchu, Numberstar, Ginsuloft, Gooper20, ElusiveOne96, FockeWulf FW 190, FrB.TG, ZebraOnesie, BethNaught, NQ, MMayur315, THE TRUE BEST, Goawry, Budderheadabc, Bobby bolla, S536870912, Henry Newman, Dogecannon, GriffeyJoons24, Lincoln1705, TheDwellerCamp and Anonymous: 224

- **Computer worm** *Source:* https://en.wikipedia.org/wiki/Computer_worm?oldid=730617010 *Contributors:* LC~enwiki, Brion VIBBER, Mav, The Anome, Stephen Gilbert, Koyaanis Qatsi, Malcolm Farmer, PierreAbbat, Daniel Mahu, Paul~enwiki, Fubar Obfusco, Patrick, Nixdorf, Pnm, Wwwwolf, CesarB, Ahoerstemeier, Cyp, Jebba, Jdforrester, UserGoogol, Andres, Evercat, GCarty, Gamma~enwiki, Dj ansi, Hashar, Agtx, Ww, Dysprosia, Fuzheado, WhisperToMe, Wik, Zoicon5, Furrykef, Dcsohl, Wilinckx~enwiki, Robbot, Naddy, Yosri, Jondel, Seth Ilys, Tea2min, David Gerard, Alerante, Fennec, Akadruid, Jtg, Noone~enwiki, Eequor, Fanf, Matt Crypto, Just Another Dan, Maximaximax, Gscshoyru, Trafton, Grunt, Monkeyman, Discospinster, Rich Farmbrough, Rhobite, KneeLess, YUL89YYZ, Bender235, ESkog, JoeSmack, RJHall, PhilHibbs, Sietse Snel, DavidSky, Smalljim, MITalum, Sam Korn, Nsaa, Alansohn, Andrewpmk, Jonathanriley, Staeiou, Bsadowski1, Pauli133, Bobrayner, Newnoise~enwiki, Roboshed, Woohookitty, Mindmatrix, Camw, Guy M, TomTheHand, Isnow, Kralizec!, Palica, RichardWeiss, Jclemens, Rjwilmsi, Matt.whitby, Syndicate, Mcmvanbree, Nguyen Thanh Quang, RainR, Jwkpiano1, Dan Guan, JiFish, RexNL, Ewlyahoocom, King of Hearts, Pstevens, Daev, Chobot, AFA, Bornhj, DVdm, Mogh, YurikBot, Borgx, Kerowren, Barefootguru, Wimt, Wiki alf, Misza13, DeadEyeArrow, Bota47, Jkelly, WAS 4.250, Dspradau, Rs232, Kungfuadam, GrinBot~enwiki, Asterion, DVD R W, Rahul s55, SmackBot, Mmernex, Aim Here, Gamerzworld, David.Mestel, KelleyCook, Object01, Gilliam, Ohnoitsjamie, Martial Law, Biblioteqa, Bluebot, Snori, Miquonranger03, Pomegranite, DHN-bot~enwiki, Firetrap9254, Anabus, Tsca.bot, NYKevin, Can't sleep, clown will eat me, Yidisheryid, Rrburke, Addshore, Celarnor, Jaimie Henry, James McNally, Richard001, Wirbelwind, Weregerbil, SashatoBot, Ian Dalziel, Nic tan33, Ehheh, Optakeover, Waggers, Vernalex, Woodroar, Iridescent, Jason.grossman, Joseph Solis in Australia, Aeons, Mzub, Tawkerbot2, Dlohcierekim, Chetvorno, Makeemlighter, GHe, Jesse Viviano, Augrunt, Oden, Slazenger, Gogo Dodo, ST47, Luckyherb, Hitro-Milanese, Thijs!bot, Epbr123, Wikid77, Luigifan, Powellatlaw, Dawnseeker2000, Mentifisto, AntiVandalBot, Seaphoto, Oducado, Waerloeg, Jenny Wong, Clharker, JAnDbot, Leuko, MER-C, PubliusFL, Coopercmu, Superjag, SteveSims, Yixin1996, Bongwarrior, Rami R, Alekjds, Adrian J. Hunter, DerHexer, Shuini, NMaia, S3000, MartinBot, STBot, Ghatziki, Poeloq, Lilac Soul, Bitethesilverbullet, Herbythyme, Imfo, Uncle Dick, Yonidebot, Milo03, Crimson Instigator, Barts1a, Ignatzmice, Demizh, DJ1AM, Juliancolton, Beezhive, CardinalDan, Idioma-bot, Lights, Deor, Hersfold, Jeff G., Philip Trueman, Dindon~enwiki, Zifert, Technopat, Zman2000, Oxfordwang, LeaveSleaves, Tpk5010, Big-Dunc, RandomXYZb, MDfoo, Falcon8765, Enviroboy, Burntsauce, EJF, Barkeep, SieBot, BotMultichill, Itsme2000, DarkfireInferno, Sephiroth storm, Sat84, Happysailor, Mszegedy, Very cheap, Smaug123, Hello71, Miniapolis, Macy, OKBot, Amrishdubey2005, StaticGull, Mygerardromance, Hamiltondaniel, GioCM, Denisarona, Cellorelio, Minimosher, ClueBot, Traveler100, The Thing That Should Not Be, Lawrence Cohen, Fenwayguy, CrazyChemGuy, Eeekster, Rhododendrites, WalterGR, Dekisugi, DanielPharos, Thingg, Aitias, VIKIPEDIA IS AN ANUS!, XXXSuperSnakeXXX, SoxBot III, Sensiblekid, DumZiBoT, XLinkBot, Skarebo, WikHead, PL290, Noctibus, ZooFari, Jabberwoch, Wnzrf, Addbot, Amanda2423, A.qarta, Fieldday-sunday, Leszek Jańczuk, CactusWriter, MrOllie, Protonk, Chzz, Favonian, Comphelper12, Jasper Deng, Yyaflkaj;fasd;kdfjk, Numbo3-bot, Craigsjones, Tide rolls, Yobot, Amirobot, Nallimbot, Gunnar Hendrich, Tempodivalse, Souch3, A More Perfect Onion, Jim1138, Piano non troppo, Meatabex, Materialscientist, Neurolysis, ArthurBot, MauritsBot, Xqbot, Useingwere, Capricorn42, Avastik, Khruner, Frosted14, RibotBOT, Ulm, AlanNShapiro, Crackitcert, WPANI, Rossd2oo5, DylanBigbear, HamburgerRadio, Uberian22, Intelligentsium, Pinethicket, I dream of horses, Adlerbot, Subzerobubbles, Lotje, Fox Wilson, Vrenator, Wiwiwiwiwiwiwiwiwi, Nattippy99, Adi4094, Reach Out to the Truth, DARTH SIDIOUS 2, Hajatvrc, DASHBot, EmausBot, Orphan Wiki, Gfoley4, Bexz2000,

Wikipelli, Jasonanaggie, JDDJS, Fæ, Kalin.KOZHUHAROV, A930913, Tolly4bolly, W163, MonoAV, DennisIsMe, ChuispastonBot, Ziyad en, ClueBot NG, Henry Stanley, Borkificator, O.Koslowski, Widr, Helpful Pixie Bot, BG19bot, TheTrainEnthusiast, Tobias B. Besemer, Toccata quarta, Mantovanifabiomarco, Glacialfox, Derschueler, Anbu121, BattyBot, Johnthehero, Cyberbot II, ChrisGualtieri, EagerToddler39, Dexbot, Lal Thangzom, Codename Lisa, Webclient101, Djairhorn, Lugia2453, Jamesx12345, Rossumund, Muhammadbabarzaman, Smilieyss, Ginsuloft, Dannyruthe, FockeWulf FW 190, JaconaFrere, Satyajeet vit, Kjerish, KH-1, KasparBot, Gautamnarayan, RippleSax, Compassionate727, CaseyMillerWiki, Fakersenpaipls, Manav garg, Sharanyanaveen, Jumblebumble12, ASire03, Getloader and Anonymous: 518

- **List of computer worms** *Source:* https://en.wikipedia.org/wiki/List_of_computer_worms?oldid=728286805 *Contributors:* AxelBoldt, CesarB, GCarty, Coren, Smjg, Trafton, Rich Farmbrough, ZeroOne, A-Day, Woohookitty, BD2412, The wub, AFA, Petiatil, Rsrikanth05, NawlinWiki, Liastnir, Welsh, Superiority, Theda, Rwwww, SmackBot, Methem~enwiki, Deepak D'Souza, Terronez, TheJC, STBot, Jerryobject, JL-Bot, Tails 155, Gene93k, Billybobpeter15, Propeng, DanielPharos, Zodon, Addbot, Tassedethe, Yobot, AnomieBOT, Wikieditoroftoday, Ciphers, Ulric1313, Kubuswoningen, HamburgerRadio, Killian441, SL93, Full-date unlinking bot, Lotje, Callanecc, Enauspeaker, GoingBatty, Peaceray, Cymru.lass, Donner60, Nhero2006, ClueBot NG, Kasirbot, Frze, Ugncreative Usergname, Mrt3366, Asmcint, FoCuSandLeArN, Mogism, Yea55, Greefan443, NixZiZ, 8Bitly, Chetanhedau409, Henry Newman, Fionaamazing, Ultrastarine and Anonymous: 39
- **Timeline of computer viruses and worms** *Source:* https://en.wikipedia.org/wiki/Timeline_of_computer_viruses_and_worms?oldid=728802782 *Contributors:* AxelBoldt, Bryan Derksen, The Anome, SimonP, Fonzy, DopefishJustin, Nixdorf, Wwwwolf, Ahoerstemeier, Bueller 007, GCarty, Schneelocke, WhisperToMe, Furrykef, Bhuston, Morven, AnonMoos, SD6-Agent, Chuunen Baka, Kizor, RedWolf, Cyvh, Pingveno, Wereon, Mushroom, Michael2, Akadruid, Jtg, Frevidar, Sik0fewl, FrYGuY, Just Another Dan, Coldacid, Utcursch, SarekOfVulcan, Dwedit, Bumm13, AndrewKeenanRichardson, CesarFelipe, Bk0, Fratley, Histrion, SamSim, Anfi, Trafton, Venu62, Rich Farmbrough, Pmsyyz, Mecanismo, Dave souza, LindsayH, MarkS, Bender235, TerraFrost, JoeSmack, Odalcet, Aecis, RoyBoy, Orbst, Unquietwiki, A-Day, Alansohn, Andrewpmk, BodyTag, VladimirKorablin, Lawriebrown, EmmetCaulfield, Wtmitchell, Uucp, Lerdsuwa, Woohookitty, GVOLTT, Camw, Brunnock, Jacobolus, WadeSimMiser, Darkoneko, Leapfrog314, Graham87, Miq, Pmj, Ketiltrout, Rjwilmsi, Leeyc0, Omnieiunium, Pacific Coast Highway, FayssalF, StuartBrady, Windchaser, Ysangkok, Dalef, Ahunt, Butros, Pstevens, Bgwhite, Wavelength, SpikeJones, RussBot, Bhny, Yamara, Hydrargyrum, Singelet, Ozzykhan, Bachrach44, Msikma, Dialectric, Chick Bowen, Seegoon, Moe Epsilon, LodeRunner, Tony1, FlyingPenguins, Jeh, Jeremy Visser, User27091, N. Harmonik, Romal, Nailbiter, WAS 4.250, Nikkimaria, Th1rt3en, AVazquezR, Ben D., Paul Erik, Jeff Silvers, MacsBug, SmackBot, Nicolas Barbier, Dweller, Mmernex, WikiWookie, Onebravemonkey, TrancedOut, Yamaguchi??, Skizzik, Winterheart, Stuart P. Bentley, Drdamour, Movementarian, QTCaptain, Morte, Renamed user Sloane, Colonies Chris, Quesodood, Christan80, Exec8, EOZyo, Brainman, Adamantios, Warren, Clean Copy, Iridescence, Vina-iwbot~enwiki, Unre4L, Gobonobo, Breno, Minna Sora no Shita, Mgiganteus1, SpyMagician, Citicat, DagErlingSmørgrav, NEMT, RekishiEJ, Aeons, No1lakersfan, Linberry, Cydebot, RyanDesign, Mortus Est, Calvacadeofcats, Jack Phoenix, Dchristle, Boemanneke, Arbitrary username, Niubrad, Epbr123, James086, Dawnseeker2000, Utopiah, AntiVandalBot, Luna Santin, Seaphoto, GiM, Myanw, MikeLynch, NapoliRoma, Barek, Time3000, Magioladitis, Alekjds, Styrofoam1994, Cpl Syx, Spellmaster, MartinBot, R'n'B, Uncle Dick, Javawizard, Eliz81, Milo03, Toobaz, Carolfrog, DarkFalls, Little Professor, IngSoc BigBrother, AppleMacReporter, Bubble94, Chris Croy, Action Jackson IV, Duder130, CardinalDan, Fbifriday, Danwh89, Bsroiaadn, Jacroe, Philip Trueman, Eric outdoors, MikeCerm, Hoppy2, QuintusMaximus, Ferengi, UnitedStatesian, Wiae, Umdelt, Rsnbrgr, CommonEditor2345, Truthanado, Ub3rst4r, Logan, SieBot, Giladbr, Serotonin deficient, Jerryobject, Flyer22 Reborn, James.Denholm, Le Pied-bot~enwiki, Ayudante, Fratrep, Ssdfsa, Kortaggio, Radical-Dreamer, ClueBot, Snigbrook, GreenSpigot, Makeshiftman, Mackilicious, Danialbehzadi, Campoftheamericas, Excirial, Sun Creator, Tau666666, Kekasih13, Rtyb, Kolyavmk, DanielPharos, Versus22, Vybr8, DumZiBoT, TheNameWithNoMan, XLinkBot, Pheonex, Jasonxu98, Addbot, Ghettoblaster, Some jerk on the Internet, Shermanbay, Dazza4994, Tothwolf, Fieldday-sunday, NumbNull, Chiefhuggybear, Lightbot, Dr.queso, Aldibibable, Megaman en m, Legobot, Helpfulweasal, ZX81, Yobot, Legobot II, AnomieBOT, Joshweiser, Wikieditoroftoday, Jim1138, Dstever, Bachsau, DirlBot, Ched, J04n, Papercutbiology, BubbleDude22, SheavesOfWheat, CaptainMorgan, Bo98, Mathgod333, Erik9, Jerrysmp, FrescoBot, StaticVision, Navynaveed, Person453, HamburgerRadio, Citation bot 1, Shivam101, LetsPlayMBP, Kenilworth Terrace, Rf.89, Pinethicket, I dream of horses, Jonesey95, BeaverOtter28, Anish9807, Stone789, Lotje, Dinamik-bot, Vrenator, Jfmantis, Mean as custard, Qaxzaaa, RjwilmsiBot, Noommos, Will Hawes, EmausBot, Somebody500, G1smoe, Slightsmile, Bexz2000, K6ka, Universeis42, Alpha Quadrant, Alpha Quadrant (alt), SporkBot, Wayne Slam, Yabba67, W163, Cwillis1964, Nhero2006, DASHBotAV, ClueBot NG, Yourmomblah, Achlysis, Widr, Mawcowboybillsbrick7, MaineChris, Helpful Pixie Bot, PushButtonToGo, Figure.10, Wbm1058, Aladdinsane64, MultWiki, Demon Hill, BG19bot, Cometcaster, Mark Arsten, Tbs541, Telepan, Eman2129, Sn1per, Fylbecatulous, BattyBot, ChrisGualtieri, Cerabot~enwiki, Sriharsh1234, Xscontrib, BetterSkatez, Seanthehero, Yolofamicom22, Wolf0102, Epicgenius, Alexwho314, Melonkelon, Sunjerbob, Monochrome Monitor, Quenhitran, FockeWulf FW 190, Tango4122, KShepAZ, Bandishjoshi, Fixuture, CatcherStorm, OMPIRE, DeskWarrior, Eugene Urameshi, BelleBougie, VulpesVulpes42, Krishna19nov, MineTechJason, Angelawilliam82, FdswefdsfsdfsdahDBFJGDSYAGFDSG, HIPoynter and Anonymous: 454
- **Trojan horse (computing)** *Source:* https://en.wikipedia.org/wiki/Trojan_horse_(computing)?oldid=730376237 *Contributors:* Damian Yerrick, Paul Drye, MichaelTinkler, LC~enwiki, Mav, Bryan Derksen, Zundark, Rjstott, Andre Engels, Gianfranco, Mincus, Heron, R Lowry, Michael Hardy, Voidvector, Pnm, Dori, Ahoerstemeier, Ronz, Darrell Greenwood, Julesd, Glenn, Jiang, Ryuukuro, Timwi, Andrevan, Ww, WhisperToMe, SEWilco, Chuunen Baka, Robbot, Kizor, Schutz, Altenmann, Puckly, Premeditated Chaos, Sunray, Tbutzon, Saforrest, Borislav, Miles, Splatt, Cyrius, GreatWhiteNortherner, Giftlite, Fennec, Brian Kendig, No Guru, Wikibob, Leonard G., ZeroJanvier, AlistairMcMillan, Fanf, Matt Crypto, PlatinumX, SWAdair, SoWhy, Knutux, SURIV, Antandrus, Tbjablin, Kesac, Asriel86, Bumm13, Trafton, Shiftchange, Monkeyman, A-giau, Discospinster, Sperling, Stereotek, JoeSmack, CanisRufus, Shanes, Sietse Snel, One-dimensional Tangent, Yono, Bobo192, Stesmo, Alexandre.tp, Cmdrjameson, Chirag, DCEdwards1966, Haham hanuka, Jjron, Ranveig, Alansohn, Anthony Appleyard, Guy Harris, Andrewpmk, M7, Riana, Sade, Ciaran H, Kesh, Danhash, Evil Monkey, BDD, Versageek, Brookie, Nuno Tavares, Woohookitty, Mindmatrix, TigerShark, Myleslong, Matey~enwiki, Briangotts, Pol098, WadeSimMiser, Easyas12c, Optichan, Gyrae, Mekong Bluesman, Graham87, Jclemens, Enzo Aquarius, Rjwilmsi, JoshuacUK, Blacktoxic, NeonMerlin, ElKevbo, Ttwaring, Aapo Laitinen, AySz88, Andrzej P. Wozniak, RainR, RobertG, JiFish, Bubbleboys, Ewlyahoocom, Alexjohnc3, TheDJ, DevastatorIIC, Ben-w, Gr8dude, M7bot, Ahunt, Chobot, DVdm, Roboto de Ajvol, Angus Lepper, Sceptre, Ytgy111, Kerowren, CambridgeBayWeather, Eleassar, Ptomes, Wimt, NawlinWiki, Wiki alf, Dialectric, RattleMan, Johann Wolfgang, Vincspenc, THB, Ugnius, Nick C, Kenkoo1987, T, Lockesdonkey, Wknight94, Niggurath, Zzuuzz, E Wing, Jogers, GraemeL, Ethan Mitchell, RandallZ, Airconswitch, Suburbancow, CIreland, Jaysscholar, Slampaladino, J2xshandy, Scolaire, SmackBot, Kellen, Ashenai, Unschool, Narson, Bobet, Tarret, KocjoBot~enwiki, Delldot, KelleyCook, Jpvinall, Arsenaldc1988, Yamaguchi??, Gilliam, Ohnoitsjamie, Spamhuntress, Snori, Tree Biting Conspiracy, Miquonranger03, Gareth, LaggedOnUser, Lexlex, DHN-bot~enwiki, Jeffreyarcand, Abaddon314159, Can't sleep, clown will eat me, MyNameIsVlad, Frap, Christan80, KaiserbBot, Rrburke, TKD, Emre D., Nibuod,

Sljaxon, Drphilharmonic, HDow, LeoNomis, Richard0612, Clicketyclack, Neverender 899, SS2005, Kuru, Jidanni, Gobonobo, Sir Nicholas de Mimsy-Porpington, Evan Robidoux, UkNegative, 041744, JHunterJ, George The Dragon, Alethiophile, Waggers, Iridescent, Redskull619, Ivan-Lanin, JoeE, Blehfu, Courcelles, Linkspamremover, Astral9, Mzub, ChrisCork, Switchercat, SkyWalker, JForget, DJPhazer, CmdrObot, Wafulz, Makeemlighter, ParadoX, CWY2190, Rikva, Lishy Guy, Jesse Viviano, INVERTED, Neelix, Funnyfarmofdoom, Equendil, Slazenger, MC10, Red Director, SnootyClaus, Strom, Mr. XYZ, Shirulashem, UnDeRsCoRe, Rudá Almeida, Omicronpersei8, Rocket000, Thijs!bot, Epbr123, Blademaster313, N5iln, Laboye, Vertium, John254, James086, Leon7, Danfreedman, EdJohnston, Mule Man, Dawnseeker2000, Mentifisto, AntiVandalBot, Luna Santin, Widefox, Seaphoto, Oducado, Karthik sripal, Rhuggins-ahammond, JAnDbot, Xhienne, El Dominio, Vaclon, HellDragon, Mishrankur, Freedomlinux, VoABot II, Nyq, Jrg7891, SineWave, GODhack~enwiki, Indon, Cailil, Esanchez7587, Shuini, Didier-Stevens, Charitwo, Gwern, Atulsnischal, MartinBot, Axlq, Jonathan Hall, R'n'B, JohnNapier, J.delanoy, Patsyanks06, Legoboy2000, Catmoon-girl, Didgeman, Mccajor, McSly, RichJizz123, Demizh, Evils Dark, Gurchzilla, AntiSpamBot, Dividing, LeighvsOptimvsMaximvs, Shoessss, Cue the Strings, Andrewcmcardle, Darryl L James, Bonadea, Martial75, Ditre, Anapologetos, ThePointblank, CardinalDan, Burlywood, Deor, VolkovBot, ABF, Jeff G., Sulcage, Rtrace, VasilievVV, Jacroe, Ryan032, Philip Trueman, PGSONIC, Af648, Zidonuke, Dorcots, Floddinn, Drake Redcrest, Rei-bot, Crohnie, Arnon Chaffin, Warrush, Anna Lincoln, Clarince63, Undine235, LeaveSleaves, ^demonBot2, Lukes123, Skittles266, BotKung, Hurleyman, SpecMode, Darkness0110, Madhero88, Peteritism, Haseo9999, Lamro, Falcon8765, Enviroboy, Insanity Incarnate, Why Not A Duck, Spitfire8520, LittleBenW, AlleborgoBot, Logan, PGWG, Numbuh48, Firefoxobsession, Ramesseum, Softpile, Copana2002, SieBot, Teh nubkilr, BotMultichill, Krawi, Josh the Nerd, Caltas, Eagleal, RJaguar3, X-Fi6, Chiroz, Sephiroth storm, Johnnyeagleisrocker, Happysailor, Flyer22 Reborn, Caidh, Oxymoron83, Kosack, Hobartimus, Drsamgo, Bcrom, Hamiltondaniel, AtteOOIE, Snarkosis, The sunder king, Martarius, ClueBot, Jimmyrules1, Damonkeyman889944, Avenged Eightfold, Binksternet, Artichoker, The Thing That Should Not Be, IceUnshattered, Lawrence Cohen, Ndenison, Wysprgr2005, Ascabastion, Freebullets, Zarkthehackeralliance, Mild Bill Hiccup, Piriczki, Infogaufire, CounterVandalismBot, Dandog77, Aabrol19, Dennistang2007, Gunnar Kreitz, Somno, Aua, Excirial, Jusdafax, PixelBot, Eeekster, Bde1982, Rhododendrites, Mac1202, Lunchscale, WalterGR, Doctor It, Jaizovic, DanielPharos, JaneGrey, Taranet, VIKIPEDIA IS AN ANUS!, 7, Ranjithsutari, Berean Hunter, Egmontaz, Alchemist Jack, Polemos~enwiki, XLinkBot, Spitfire, NiveusLuna, Jovianeye, Feinoha, Parallelized, TFOWR, ErkinBatu, Mifter, Alexius08, Noctibus, Addbot, Some jerk on the Internet, Landon1980, A.qarta, Friginator, Markyman12, Ronhjones, Ashton1983, Nirajdoshi, MrOllie, Download, Morning277, Ericzhang789, London-infoman, D.c.camero, Glane23, Exor674, SamatBot, Arteyu, Theman98, Politoed666, Numbo3-bot, Tide rolls, Legion79, Krano, Apteva, Teles, Zorrobot, Jarble, Arbitrarily0, Fdaneels, Koru3, Legobot, Helpfulweasal, Yobot, 2D, Fraggle81, Cflm001, Xxxpivjtxxx, NERVUN, Nallimbot, QueenCake, Sujit.jhare, South Bay, AnomieBOT, KDS4444, DemocraticLuntz, Rubinbot, Captain Quirk, Jim1138, Chuckiesdad, Materialscientist, Arezey, Frankenpuppy, Xqbot, Capricorn42, Robot85, Liorma, Bihco, Jsharpminor, KrisBogdanov, Mlpearc, S0aasdf2sf, GrouchoBot, Megamonkeyextreme, RibotBOT, SassoBot, TrueGlue, Amaury, JulianDelphiki, Shadowjams, SchnitzelMannGreek, Vanatom, Thehelpfulbot, Trojan1223, FrescoBot, Untilabout9am, Daerlun, Clubmaster3, Michael93555, Scottaucoin89, A little insignificant, Haein45, HamburgerRadio, Mitchell virus, Launchballer, Winterst, I dream of horses, Vicenarian, Edderso, Jacobdead, A8UDI, Rihdiugam, Ddspec, Robo Cop, Pcuser42, GWPSP090, Ksanexx, DixonDBot, Lamarmote, Miiszmylove, MichaelRivers, Vrenator, Reaper Eternal, Jeffrd10, Specs112, Vanished user aoiowaiuyr894isdik43, Ciscorx, Minimac, Ameypersonsave, DARTH SIDIOUS 2, Mean as custard, MMS2013, Lowoox, SMARTCUTEFUNNYXD, Brandonprince00, NerdyScienceDude, Limited2fan, Slon02, Skamecrazy123, DASHBot, EmausBot, Super48paul, Fly by Night, Dewritech, L235, Tommy2010, Wikipelli, Dcirovic, K6ka, TheGeomaster, Skaera, Ida Shaw, Dalek32, Traxs7, Eldruin, Newbiepedian, Kiwi128, EneMsty12, AndrewOne, Lolcat56734, Coasterlover1994, Sahimrobot, L Kensington, Donner60, Gary Dee, ClueBot NG, Cwmhiraeth, MuffinMan999, Gareth Griffith-Jones, MelbourneStar, Bped1985, Augustalex, Muon, Braincricket, Mesoderm, Rezabot, Widr, OKIsItJustMe, Alan357319, Madpigeon12, Strike Eagle, Titodutta, W.andrea, Complol2234343, Robbiecee2, Wiki13, MusikAnimal, AvocatoBot, Desenagrator, Mark Arsten, Sbd01, Onewhohelps, Display name 99, Snow Blizzard, MrBill3, Glacialfox, Kelvinruttman, Millennium bug, Tutelary, Niraj.adyyyy, Th4n3r, Hsr.rautela, Adhithyan15, Cyberbot II, ChrisGualtieri, MadGuy7023, JayMyers-NJITWILL, Ghostman1947, Rezonansowy, FoCuSandLeArN, SoledadKabocha, Djairhorn, Lugia2453, JoshLyman2012, Jc86035, Siravneetsingh, Soda drinker, Discuss-Dubious, Sourov0000, Cablewoman, Bugzeeolboy, NimaBoscarino, RootSword, Dave Braunschweig, Epicgenius, CatBallSack, Eyesnore, Gaman0091, DavidLeighEllis, Ugog Nizdast, Khabir123, NottNott, Kushay titanium, Someone not using his real name, Manish2911, Oranjelo100, Dannyruthe, Sathishguru, FockeWulf FW 190, STH235SilverLover, Joseph 0515, Marp pro, Rkpayne, Monkbot, Sidharta.mallick, Filedelinkerbot, Abcdfeghtys, Laura J. Pyle, Biblioworm, TerryAlex, Classofthewise, Earthquake58, ChamithN, HamadPervaiz, Eteethan, Helpguy77, BoxOfChickens, TQuentin, KasparBot, Ceannlann gorm, James the king12, JeremiahY, TeacherWikipedia, OldMcdonald12345, Drakeblair08, Kurousagi, Majneeds2chill, Vansockslayer, Aditya3929, User0071987, Bradley!90432754, Pa19602030fu, Montouesto, Srilekha selva, Voidcub, Promestein, Lucaru, Prince.manjeet, Arshdeep Singh Bhatia, Miles315 and Anonymous: 1203

- **Rootkit** *Source:* https://en.wikipedia.org/wiki/Rootkit?oldid=728068929 *Contributors:* Zundark, Fubar Obfusco, William Avery, SimonP, Stevertigo, Frecklefoot, JohnOwens, Nixdorf, Pnm, Liftarn, Zanimum, Penmachine, Tregoweth, Ahoerstemeier, Haakon, Nikai, Schneelocke, Emperorbma, Timwi, Aarontay, Ww, Olego, Fuzheado, Markhurd, Echoray, Furrykef, Taxman, Bevo, Rossumcapek, Phil Boswell, Robbot, Scott McNay, Henrygb, Auric, Zidane2k1, Paul G, Tea2min, Unfree, David Gerard, Alison, JimD, Ezhiki, Cloud200, AlistairMcMillan, Saucepan, Taka, Alobodig, Deewiant, Creidieki, Pascalv, Adashiel, Squash, Brianhe, ElTyrant, Rich Farmbrough, Agnistus, Jayc, Bender235, CanisRufus, Twilight (renamed), Kwamikagami, PhilHibbs, Spoon!, Femto, Perfecto, Stesmo, Smalljim, Chasmo, Mpvdm, Adrian~enwiki, Giraffedata, Yonkie, Bawolff, Helix84, Espoo, Jhfrontz, Polarscribe, CyberSkull, JohnAlbertRigali, Hookysun, Phocks, BanyanTree, Earpol, RJFJR, RainbowOfLight, Kazvorpal, RyanGerbil10, Japanese Searobin, Dtobias, Defixio, Alvis, CCooke, OwenX, Woohookitty, David Haslam, Steven Luo, Shevek, Pol098, Apokrif, Btmiller, Easyas12c, Midnightblaze, SDC, Umofomia, Xiong Chiamiov, RichardWeiss, Graham87, BD2412, Rjwilmsi, TitaniumDreads, Syndicate, Arisa, Randolph, RainR, Flarn2006, FlaBot, RobertG, Stoph, JiFish, Harmil, Mark Luszniak, Arunkoshy, Mordien, Intgr, Mimithebrain, Dbpigeon, Bgwhite, Martin Hinks, Poorsod, FrankTobia, Elfguy, Uriah923, YurikBot, Wavelength, Hairy Dude, Diesonne, AVM, Chrisjustinparr, IByte, Hydrargyrum, NawlinWiki, Wiki alf, Mipadi, Ian Cheese, Ejdzej, Stephen e nelson, Cleared as filed, Nick, Raven4x4x, JackHe, Mysid, FoolsWar, Bota47, Nescio, Ninly, Maxwell's Demon, Mateo LeFou, Theda, Closedmouth, Arthur Rubin, Reyk, Roothorick, AnimeJanai, Solarusdude, Jacqui M, That Guy, From That Show!, SmackBot, Mmernex, Estoy Aquí, Reedy, Mate.tamasko, Unyoyega, KelleyCook, Iph, SimonZerafa, Ohnoitsjamie, Chris the speller, Bluebot, Gspbeetle, Thumperward, Ben.the.mole, Octahedron80, DARQ MX, DHN-bot~enwiki, Jmax-, 1(), Frap, Onorem, Tim Pierce, Sommers, Ukrained, Whpq, Michael-Billington, DMacks, J.Christopher.Wells, AndyBQ, A5b, Mitchumch, N-dy, Clicketyclack, FrostyBytes, Tasc, Tthtlc, Peyre, Simon Solts, Xionbox, LAlawMedMBA, IvanLanin, CapitalR, Prpower, Phoenixrod, Courcelles, Tawkerbot2, Davidbspalding, FatalError, Zarex, Cyrus XIII, Megaboz, Jokes Free4Me, Jesse Viviano, Chrismo111, Racooper, Myasuda, Equendil, A876, GrahamGRA, Tryl, Пrate, Fetternity, Mewsterus, Etaon, Ambulnick, Marek69, Tocharianne, Dawnseeker2000, AntiVandalBot, Widefox, Obiwankenobi, Czj, Sjledet, Lfstevens, Bscottbrown,

AndreasWittenstein, TuvicBot, Hiddenstealth, NapoliRoma, MER-C, Minitrue, QuantumEngineer, Karsini, BCube, Repku, Raanoo, Drugonot, Chevinki, Nyttend, Cl36666, Denorios, Stromdal, Alekjds, Hamiltonstone, Cpl Syx, XandroZ, Stephenchou0722, R27smith200245, MartinBot, Eshafto, CobraBK, Fethers, R'n'B, Nono64, Ash, Felipe1982, CraZ, Pharaoh of the Wizards, UBeR, Uncle Dick, Maurice Carbonaro, Public Menace, Leeked, Andy5421, It Is Me Here, Peppergrower, Crakkpot, DavisNT, Wng z3r0, Marekz, Cometstyles, Gemini1980, ArneWynand, VolkovBot, Ashcan Rantings, Senachie, Soliloquial, TXiKiBoT, Sphinx2k, CanOfWorms, Miketsa, UnitedStatesian, Natg 19, Haseo9999, Willbrydo, Suzaku Medli, Ceranthor, Ggpur, MrChupon, SieBot, Technobreath, Sephiroth storm, Edans.sandes, Windowsvistafan, Aly89, General Synopsis, Fyyre, Clearshield, Capitalismojo, Bogwhistle, BfMGH, Guest141, Martarius, ClueBot, The Thing That Should Not Be, TheRasIsBack, Mild Bill Hiccup, Fossguy, Tai Ferret, Socrates2008, Crywalt, PixelBot, JunkyBox, Rhododendrites, Holden yo, NuclearWarfare, Mrkt23, Pinkevin, Htfiddler, DanielPharos, Floul1, Johnuniq, SF007, Uuddii, Pelican eats pigeon, XLinkBot, Dsimic, Thatguyflint, Addbot, Willking1979, Kongr43gpen, Sergey AMTL, Elsendero, TutterMouse, Cst17, MrOllie, OlEnglish, Fiftyquid, Luckas-bot, Yobot, Fraggle81, GateKeeper, Golftheman, Alipie42, AnomieBOT, NoKindOfName, Bluerasberry, Materialscientist, Nutsterrt, Citation bot, ArthurBot, LilHelpa, Avastik, S0aasdf2sf, Notwej, GrouchoBot, Kernel.package, Theareher4, Trafford09, Sophus Bie, XLCior, Shadowjams, FrescoBot, WPANI, Ozhu, Wmcleod, HamburgerRadio, Citation bot 1, JoeSmoker, Winterst, Pinethicket, Jonesey95, Shultquist, Gim3x, OMGWEEGEE2, Rbt0, Trappist the monk, Techienow, Vanished user aoiowaiuyr894isdik43, Onel5969, TjBot, Alph Bot, EmausBot, John of Reading, WikitanvirBot, Timtempleton, Heracles31, Dewritech, Janiko, P3+J3^u!, Dcirovic, ZéroBot, Herman Shurger, Basheersubei, Mike735150, IceCreamForEveryone, Bender17, Chicklette1, Diflame, Macwhiz, Nhero2006, DASHBotAV, Pianosa, ClueBot NG, Biterankle, Morgankevinj huggle, Matthiaspaul, MelbourneStar, Zakblade2000, Barry McGuiness, Helpful Pixie Bot, BG19bot, Strovonsky, Rijinatwiki, Abagi2, Johndavidthomas, BattyBot, Tkbx, StarryGrandma, Cyberbot II, ChrisGualtieri, Draculamilktoast, Cadava14, Dexbot, Codename Lisa, Noul Edge, SoledadKabocha, Cryptodd, CaSJer, A Certain Lack of Grandeur, Ginsuloft, Oranjelo100, Dust482, Monkbot, Vieque, BethNaught, Ahollypak, Shinydiscoball, Jithendran Subburaj, TQuentin, Azlan 6473, KasparBot, Ceannlann gorm, Montouesto, Sharanyanaveen and Anonymous: 588

- **Backdoor (computing)** *Source:* https://en.wikipedia.org/wiki/Backdoor_(computing)?oldid=729436061 *Contributors:* Damian Yerrick, The Anome, Arvindn, Dwheeler, Wshun, Voidvector, Pnm, Ixfd64, (, Iluvcapra, Ronz, Jebba, Nikai, Ww, Furrykef, Thue, Khym Chanur, Movermover, RedWolf, Lowellian, Danutz, KellyCoinGuy, Tea2min, David Gerard, Graeme Bartlett, Gtrmp, Fennec, Mintleaf~enwiki, Tom harrison, Leonard G., Cloud200, AlistairMcMillan, Eckhart Wörner~enwiki, LiDaobing, Robert Brockway, Am088, Icairns, Ojw, Monkeyman, GoodStuff~enwiki, Rich Farmbrough, FT2, MCBastos, Smyth, CanisRufus, Sietse Snel, Euyyn, Smalljim, Ral315, Kdau, Woohookitty, RHaworth, Flamingspinach, Stefanomione, Scratchy, Marudubshinki, BD2412, Rjwilmsi, Commander, Allynfolksjr, RainR, Flarn2006, FlaBot, JiFish, Quuxplusone, Daev, Bgwhite, YurikBot, Borgx, Cybercat, Hairy Dude, Gene.arboit, Stephenb, Bullzeye, Wiki alf, Matir, Fabulous Creature, Anetode, Vlad, Bota47, Arthur Rubin, Urchin, RealityCheck, Luk, SmackBot, Mmernex, Ultramandk, KelleyCook, Xaosflux, Nbarth, Lmsilva~enwiki, Bisected8, Wonderstruck, The undertow, SashatoBot, Harryboyles, Xandi, Lee Carre, Doceddi, CWY2190, Tim1988, DumbBOT, Thijs!bot, Oerjan, KeithPenguin, Gioto, Widefox, AndreasWittenstein, JAnDbot, V. Szabolcs, VoABot II, Connor Behan, Gwern, CliffC, RP88, Axlq, Maurice Carbonaro, Milo03, Daedalus CA, Katalaveno, Berserkerz Crit, KCinDC, Mike V, Bonadea, Ale2006, TXiKiBoT, Baumfreund-FFM, Rei-bot, FironDraak, Xeno8, Rep07, Jroptimus, SieBot, Sephiroth storm, Jojalozzo, Soulweaver, Geoff Plourde, ClueBot, Excirial, Socrates2008, Christopherlmarshall, Zac439, RaceGT, Rhododendrites, DanielPharos, Rror, BlackDeath3, Stemaboatlion, Addbot, TIAA Is An Acronym, SDJ, ZX81, Yobot, THEN WHO WAS PHONE?, AnomieBOT, Materialscientist, Jeffrey Mall, Censorship Workaround, A Quest For Knowledge, GliderMaven, Aldebrn, FrescoBot, Safinaskar, HamburgerRadio, I dream of horses, Calmer Waters, Full-date unlinking bot, Cnwilliams, Trappist the monk, Rooseycheeksdrown, Reaper Eternal, RjwilmsiBot, Dewritech, Pro translator, Zezen, Erianna, Schnoatbrax, Nhero2006, Gary Dee, ClueBot NG, LeoVeo, BG19bot, Dipankan001, Phoenixia1177, Garamond Lethe, Dexbot, Codename Lisa, Hmains-bot1, Openmikenite, Dr Dinosaur IV, Shashank16392, Comp.arch, JadeGuardian, Kahtar, Tqe1999, Monkbot, Hannasnow, Molly tharrington, KasparBot, Anarchyte, Marty-the-Bluetooth, CaseyMillerWiki, Pickyt and Anonymous: 145

- **Zombie (computer science)** *Source:* https://en.wikipedia.org/wiki/Zombie_(computer_science)?oldid=730705578 *Contributors:* R Lowry, Jeryb1961, Przepla, Ww, Greenrd, Furrykef, Tempshill, Khym Chanur, Finlay McWalter, Alerante, Everyking, Fanf, Gyrofrog, Publunch, Mennonot, ArnoldReinhold, JoeSmack, Shanes, La goutte de pluie, Wrs1864, OGoncho, Friviere, DreamGuy, Velella, Rebroad, Danhash, Axeman89, Richard Arthur Norton (1958-), Mindmatrix, Shal~enwiki, Mandrke, Wayward, Xiong, Elvey, FreplySpang, Tangotango, Sango123, Colenso, DVdm, HJKeats, YurikBot, Wavelength, Hairy Dude, EricGiguere, John Bokma, THB, TransUtopian, Intershark, Closedmouth, BorgQueen, Jeremy Butler, SmackBot, Kilo-Lima, Delldot, Commander Keane bot, Persian Poet Gal, Snori, MalafayaBot, Dethme0w, Tsca.bot, Frap, JonHarder, TKD, Andrew Jackson, Richard001, BullRangifer, Polonium, Mion, Ceoil, Ehheh, Iridescent, Ouzo~enwiki, Chetvorno, Davidbspalding, DavidTangye, Dgw, Jesse Viviano, Epbr123, RevolverOcelotX, Neil916, JustAGal, AntiVandalBot, Majorly, Seaphoto, BenTremblay, Streamcipher, Sloclops, Dragfyre, Nposs, MartinBot, Tgeairn, Uncle Dick, TomCat4680, Katalaveno, Advancewars177, VolkovBot, Anonymous Dissident, Dictouray, Lambyte, Enviroboy, Sam.cantrell, Copana2002, Radon210, Mscwriter, Altzinn, ClueBot, Alexbot, Socrates2008, PixelBot, Romney yw, PCHS-NJROTC, XLinkBot, Airplaneman, Addbot, Ronhjones, Favonian, Tassedethe, David0811, Windward1, Luckas-bot, Marshall Williams2, AnomieBOT, Teethmonkey, Kingpin13, Law, Materialscientist, Xqbot, Inferno, Lord of Penguins, GrouchoBot, Cyberssecurity, Anhydrobiosis, RibotBOT, SassoBot, Bugefun, JL 09, Romangralewicz, Ribky, Ericjang, MichaelXX2, RoryJSK, MastiBot, F1niner, Loveanimevampiregrl, Rocketpooch, EmausBot, Tommy2010, Evrestsebretawhtoot, ZéroBot, Traxs7, Bieraaa, ClueBot NG, Imjohnnyknoxville, Widr, Helpful Pixie Bot, Mophedd, Mark Arsten, Marthelati, Junganghansik, Pavel.rydl, Enterprisey, Hamidrezahami, Shobeirvakilian, Ekips39, Lsmll, Datdyat, A Certain Lack of Grandeur, Seafax, Miltos658, FockeWulf FW 190, Yasuo Miyakawa, Heejinmun, Spuphex and Anonymous: 128

- **Man-in-the-middle attack** *Source:* https://en.wikipedia.org/wiki/Man-in-the-middle_attack?oldid=729420098 *Contributors:* Damian Yerrick, Wesley, Ant, Graft, Tzartzam, Edward, Michael Hardy, Tenbaset, Shellreef, Kku, Paddu, Kingturtle, Whkoh, Nikai, GCarty, Ehn, Ventura, Ww, The Anomebot, Birkett, BenRG, Ssd, Matt Crypto, Pgan002, Sonjaaa, Quarl, Gazpacho, ArnoldReinhold, Bender235, Violetriga, Bobo192, Kjkolb, Nroets, Fg, Rebroad, RJFJR, Martian, Woohookitty, Robert K S, Ruud Koot, Lance W. Haverkamp, Turnstep, MrDrew508, BD2412, Phantom784, Ronocdh, Ravik, FlaBot, Margosbot~enwiki, Dinoen, Brownh2o, Jrtayloriv, Brendan Moody, Sstrader, Bgwhite, EamonnPKeane, YurikBot, Wavelength, Jimp, Gene.arboit, Pacaro, Arado, Mike411, DavidJablon, FiggyBee, FireFury, Closedmouth, Josh3580, Jetman123, Ralp, A bit iffy, SmackBot, Mmernex, McGeddon, KVDP, BiT, Ohnoitsjamie, Hmains, PJTraill, Justforasecond, Sct72, Abaddon314159, Jahiegel, Frap, AMK152, Nakon, EdgeHM, Nagle, Slasher-fun, Whisperwolf, Hu12, ChrisCork, Sakurambo, Jesse Viviano, Penbat, Phatom87, Cydebot, Doomed Rasher, Neustradamus, Saintrain, Nonagonal Spider, Lotte Monz, Widefox, K7aay, CobraWiki, Dougher, AndreasWittenstein, JAnDbot, MER-C, Doctorhawkes, .anacondabot, Magioladitis, Whoop whoop, NeutronIC, Brandon Hixson, Thompson.matthew, Robertducon, Eliz81, Brupat, Touisiau, NewEnglandYankee, TrampledUF, Homer Landskirty, Holme053, Philip Trueman, Billinghurst, Bf-

page, Quietbritishjim, SieBot, Sephiroth storm, Yahastu, Reporter2007, Skippydo, HighInBC, ClueBot, Rodhullandemu, Ggia, Textureglitch, Marjaliisa, Mild Bill Hiccup, Trivialist, Socrates2008, M4gnum0n, Jc00, Pro mathesh812004, DanielPharos, Leobold1, Johnuniq, XLinkBot, SilvonenBot, Mifter, Dsimic, Addbot, !Silent, TutterMouse, FreakingOut, Lightbot, Wireless friend, Legobot, Luckas-bot, Yobot, Bunnyhop11, TaBOT-zerem, Mindbuilder, AnomieBOT, DemocraticLuntz, Götz, Jim1138, Materialscientist, ArthurBot, Xqbot, MarkWarren, GrouchoBot, SassoBot, PorkoltLover60, Haeinous, 1nsecure, Jonesey95, Hamtechperson, Jandalhandler, Lotje, Vrenator, Arkelweis, ErikvanB, EmausBot, WikitanvirBot, Wikipelli, Dcirovic, Pokeme444, MithrandirAgain, Lamf 0009, Quondum, Bar-abban, Natalie Perna, Erianna, Donner60, Wangxuan8331800, ClueBot NG, Dfarrell07, Vlhsrp, Rezabot, Widr, Roller958, Johnny C. Morse, Hammadi2100, Wow501Zers, DBigXray, BG19bot, Island Monkey, BattyBot, Justincheng12345-bot, Mdann52, Fmcarthy, JONI2012, Frosty, Elvenflow, Palmbeachguy, Shee Zahra, Akashksunny13, Meteor sandwich yum, 0sm0s1z, Alpha3031, Nyashinski, Tcoshea, Mfnpka, Oiyarbepsy, ChamithN, Jessetaylor84, Jordanwolf326, Bronze2018, GeneralizationsAreBad, Shackhorn, RippleSax, Barath Rajendran, Blondenerd, Qzd, Celso Velasquez and Anonymous: 208

- **Man-in-the-browser** *Source:* https://en.wikipedia.org/wiki/Man-in-the-browser?oldid=723457689 *Contributors:* Moxfyre, Mindmatrix, Kmg90, BD2412, Rjwilmsi, Arthur Rubin, Frap, Mistress Selina Kyle, RomanSpa, Alaibot, Utopiah, Widefox, AndreasWittenstein, Froid, Monkeyjunky, Sunderland06, Remember the dot, Qu3a, Funandtrvl, Flopster2, Calliopejen1, SimonTrew, JL-Bot, Aua, DanielPharos, Pmdgolden, Addbot, AnomieBOT, Grolltech, ErikvanB, Gsgriffin, Dcirovic, ZéroBot, Erianna, Donner60, John Smith 104668, Pastore Italy, ClueBot NG, Organ feaster, MeanMotherJr, Zhaofeng Li, Cyberbot II, Fmcarthy, FockeWulf FW 190, ZovianLord, Prigaleutji and Anonymous: 25
- **Clickjacking** *Source:* https://en.wikipedia.org/wiki/Clickjacking?oldid=730134259 *Contributors:* Julesd, Chealer, Nurg, Cloud200, Rchandra, Ary29, Mormegil, Tristan Schmelcher, Stesmo, Chmeee, Dismas, Woohookitty, Mindmatrix, Daira Hopwood, NoamNelke, Rjwilmsi, Bgwhite, ChrisBoyle, Deku-shrub, Rwalker, SmackBot, Rtc, C.Fred, Martylunsford, Mistress Selina Kyle, Clean Copy, Netpagz, Kuru, Eastlaw, H4l9k, Safalra, Chad.hutchins, Andrew Clark, Widefox, Guy Macon, Mathfreq, Eliz81, Dc197, Tarunbk, JavierMC, Kitzur, Regnareb, Curb Safe Charmer, Amegghiuvirdura, WikiLaurent, Wikiold1, Shoebhakim, Doloco, XLinkBot, Duncan, Avoided, MystBot, Jabberwoch, Sweeper tamonten, Addbot, PatrickFlaherty, MrOllie, Farmercarlos, Exor674, 84user, Luckas-bot, Yobot, Dhaun, AnomieBOT, Happyrabbit, Michael-Coates, IllestFlip, SassoBot, Affinemesh94464, Shlominar, X7q, Haeinous, CousinJohn, RjwilmsiBot, DexDor, Offnfopt, Dwvisser, Timtempleton, Dewritech, Peaceray, Slightsmile, 15turnsm, ZéroBot, H3llBot, Mark Martinec, Petrb, ClueBot NG, Macdonjo, Crazymonkey1123, Helpful Pixie Bot, BG19bot, Deepanker70, BattyBot, Cimorcus, Cyberbot II, ChrisGualtieri, Dr Dinosaur IV, Ruby Murray, Dedobl1, Saectar, Abdf882c25e08d9ba219fe33f17591fe, ToonLucas22, ACookieBreak, Sharanyanaveen, Teddylev, Neermuzic, TheMagzuz, Pppery and Anonymous: 91
- **Privacy-invasive software** *Source:* https://en.wikipedia.org/wiki/Privacy-invasive_software?oldid=719845083 *Contributors:* Fubar Obfusco, Pnm, Andycjp, Riana, Stuartyeates, Marasmusine, Rjwilmsi, Koavf, RussBot, SmackBot, Fvguy72, Stifle, OSborn, DJPhazer, Wikid77, Gioto, Widefox, Andypayne, R'n'B, Kl4m, Rprpr, Morsing, Bichon, Pascualangulo, Xp54321, Tassedethe, AnomieBOT, Drilnoth, HamburgerRadio, Jonesey95, Trappist the monk, TheGreenMartian, Rxavier~enwiki, Ego White Tray, BG19bot, Cyberbot II, AK456, LightandDark2000, Sourov0000, Jjujje, DoomCult and Anonymous: 13
- **Adware** *Source:* https://en.wikipedia.org/wiki/Adware?oldid=727866163 *Contributors:* Damian Yerrick, The Epopt, Bryan Derksen, Stephen Gilbert, Malcolm Farmer, Fubar Obfusco, Ellmist, Tannin, Zanimum, Ellywa, Snoyes, Cadr, Scott, Jeandré du Toit, Evercat, Mxn, BRG, Mydogategodshat, Emperorbma, Aarontay, Radiojon, Pakaran, 1984, Moondyne, Wereon, Jor, Mmeiser, Lzur, DocWatson42, Mintleaf~enwiki, Gracefool, AlistairMcMillan, Spe88, SWAdair, Alexf, Slowking Man, OverlordQ, Quarl, Russell E, Mysidia, Sam Hocevar, Neutrality, Trevor MacInnis, Monkeyman, Discospinster, Rhobite, Supercoop, JoeSmack, Danieljackson, Dannyman, Jpgordon, Apecat, Circeus, ZayZayEM, Johnteslade, Visualize, Siliconsoul, Alansohn, GRider, Jannev~enwiki, RainbowOfLight, Mattbrundage, Firsfron, Woohookitty, Hello5959us, TigerShark, Camw, Pol098, Easyas12c, Schzmo, Sega381, The Nameless, Graham87, Deltabeignet, Reisio, Rjwilmsi, XP1, Davidron, Troymccluresf, Rebug, RainR, FlaBot, Pogoman, Nivix, Pathoschild, Superchad, Ewlyahoocom, Tijuana Brass, ApolloBoy, Cpcheung, Onebyte, Chobot, Antilived, Hahnchen, YurikBot, Aleahey, Kafziel, Hede2000, Gaius Cornelius, Havok, Member, Wiki alf, Brian Crawford, Amcfreely, Mieciu K, Bota47, Romal, Richardcavell, Icedog, Closedmouth, Reyk, GraemeL, Brianlucas, Kevin, Emc2, Wootonius, Che829, ViperSnake151, Destin, Luk, SmackBot, F, Unyoyega, Vermorel, Brossow, Ericwest, Ccole, Ohnoitsjamie, Lakshmin, Kazkaskazkasako, Chris the speller, MalafayaBot, Octahedron80, DHN-bot~enwiki, William Allen Simpson, Neo139, Kindall, VMS Mosaic, Huon, Valwen, Weregerbil, Sljaxon, DMacks, Nevyan, L337p4wn, AmiDaniel, Tor Stein~enwiki, Minna Sora no Shita, NongBot~enwiki, White Agent, Alistair.phillips1, Zorxd, Caiaffa, Vernalex, Hu12, B7T, Lonyo, Blakegripling ph, IvanLanin, Linkspamremover, Astral9, Mzub, Tawkerbot2, Morryau, FleetCommand, Ale jrb, Wikkid, Grungen, Cydebot, Gogo Dodo, Frosty0814snowman, JB196, Dipics, Albi90, Got milk, TheJC, Thijs!bot, Hasan.Z, Daniel, 0dd1, Wikikiki~enwiki, Antidemon, Tinnytintin, Dawnseeker2000, Silver Edge, Escarbot, AntiVandalBot, Dylan Lake, Dreaded Walrus, Barek, MER-C, Goncalo, OhanaUnited, TAnthony, Entgroupzd, Snesfm~enwiki, SiobhanHansa, Doshindude, Magioladitis, Prof.rick, Bongwarrior, VoABot II, Ishikawa Minoru, CoolChris, Violetness, Adrian J. Hunter, Glen, Rydra Wong, Wi-king, Gwern, Stephenchou0722, CliffC, Axlq, DavidSTaylor, Lilac Soul, Joannna, Eden5995, GandalfDaGraay, Compman12, Literacola, Healy6991, Major123, Kraftlos, Ajfweb, Bonadea, Darthnader37, Vmaldia, MarkChambers, Hanserer, Jeff G., TXiKiBoT, Dominic Dryden, Steve Wise, Zifert, Technopat, ElinorD, Blazemonkey, Sanfranman59, Hythlodayalmond, DizzyITTech, Abynion08, PowerCycle, Newsaholic, Jake Beech, Brianga, Hezekiah957, Doc James, Fredtheflyingfrog, GamesSmash, SieBot, Wallajam, IHateMalware, RJaguar3, Xenobiologista, Sephiroth storm, Tazchook, Flyer22 Reborn, Arbor to SJ, Momo san, Nosferatus2007, Oxymoron83, Antonio Lopez, MrWikiMiki, Phykyloman~enwiki, Xaver David, Samker, Triedtool, ImageRemovalBot, ClueBot, Chuckbronson45, Hutcher, DavidGGG, MalwareSmarts, The Thing That Should Not Be, Thanatos465, Hellosandimas, Deciptamacon, Braksus, Aalolzore, ElSaxo, Excirial, Socrates2008, Marianian, WalterGR, 7&6=thirteen, DKproductions, Manco Capac, BOTarate, Goodvac, Mikon8er, XLinkBot, Alexius08, Noctibus, Gggh, Debsalmi, SuperSmashBros.Brawl777, Otisjimmy1, Ronhjones, Trapped34, Fieldday-sunday, Fluffernutter, Mstrfishy, Mario CUSENZA, Cst17, MrOllie, CarsracBot, BrianKnez, Gail, Legobot, Luckas-bot, Yobot, Henryrogers, Macgeeksta, Tohd8BohaithuGh1, ArchonMagnus, Lapucelle, Vini 17bot5, Jim1138, Ulric1313, Materialscientist, Mylife2702, Xqbot, TheAMmollusc, Avastik, Zirix, Jaypvip, S0aasdf2sf, Kernel.package, Thurak13, Shadowjams, 54together, WPANI, HamburgerRadio, DenisDollfus, MondalorBot, K3nt0456, Allenkelly, Jbarg, SchreyP, Toothpaste95, Irish1348, Lotje, Aoidh, Suffusion of Yellow, RjwilmsiBot, Ripchip Bot, DASHBot, EmausBot, Immunize, GoingBatty, Elee, Slightsmile, Tommy2010, Cmlloyd1969, K6ka, JamesQ2010, Listmeister, Jz797john, Bar-abban, Staszek Lem, Mercadorios, Vale Len, FloridaShawn123, Status, Gary Dee, ClueBot NG, Cwmhiraeth, Shaddim, Deenyah, Widr, BG19bot, BertStiles, Hallows AG, Eternityglacier, Mav12222, CitationCleanerBot, Dimon879, Cre8tin, ChrisGualtieri, Ghostman1947, Codename Lisa, SoledadKabocha, Makecat-bot, Sourov0000, Smortypi, Ugog Nizdast, Mickel1982, Aromatt, Joeleoj123, Suryaaknu, Mucca danna, Julietdeltalima, GreenC bot, Fazeel Ahmed Jasir and Anonymous: 426

- **Spyware** *Source:* https://en.wikipedia.org/wiki/Spyware?oldid=728774425 *Contributors:* The Epopt, WojPob, LC~enwiki, Eloquence, Vicki Rosenzweig, Mav, Zundark, Berek, Toby Bartels, Fubar Obfusco, SimonP, Ellmist, R Lowry, Modemac, KF, Frecklefoot, Edward, Willsmith, Fred Bauder, Pnm, Tannin, Wwwwolf, Tgeorgescu, Karada, Ahoerstemeier, DavidWBrooks, Haakon, Mac, Arwel Parry, Notheruser, Darkwind, Mcfly85, Julesd, Cgs, Glenn, Bogdangiusca, Slusk, Phenry, Evercat, Raven in Orbit, Mydogategodshat, Guaka, Aarontay, Mbstone, RickK, Dysprosia, WhisperToMe, Wik, Pedant17, Jake Nelson, Grendelkhan, Saltine, ZeWrestler, Sabbut, Wernher, Bevo, Joy, Khym Chanur, Fvw, Raul654, Pakaran, Jamesday, Denelson83, PuzzletChung, Aenar, Robbot, Paranoid, Senthil, ChrisO~enwiki, Korath, Tomchiukc, Vespristiano, Moondyne, ZimZalaBim, Psychonaut, Yelyos, Modulatum, Lowellian, Mirv, JustinHall, Stewartadcock, Academic Challenger, Texture, Meelar, LGagnon, DHN, Hadal, Dehumanizer, Wereon, Michael Snow, Boarder8925, ElBenevolente, Anthony, Mmeiser, Lzur, Tea2min, Alerante, Alexwcovington, DocWatson42, Fennec, Inter, Lupin, Ferkelparade, Everyking, Kadzuwo~enwiki, Rookkey, Frencheigh, FrYGuY, Gracefool, Daniel Brockman, Zoney, Pascal666, AlistairMcMillan, Spe88, SWAdair, Golbez, Justzisguy, Gadfium, Shibboleth, Toytoy, CryptoDerk, GeneralPatton, Quadell, Antandrus, Beland, OverlordQ, The Trolls of Navarone, Piotrus, Quarl, Khaosworks, MFNickster, Kesac, Jesster79, Maximaximax, SeanProctor, Bumm13, Kevin B12, Sam Hocevar, Sridev, TonyW, Rantaro, Neutrality, Joyous!, Jcw69, Adashiel, JamesTeterenko, Grunt, Guppyfinsoup, Mike Rosoft, Maryevelyn, Tom X. Tobin, Monkeyman, Poccil, Imroy, Maestro25, Naryathegreat, Discospinster, Twinxor, Rich Farmbrough, Rhobite, Andros 1337, MCBastos, Clawed, YUL89YYZ, LindsayH, Mani1, Tinus, Pavel Vozenilek, Martpol, Paul August, SpookyMulder, Bender235, ESkog, JoeSmack, Violetriga, Brendandonhue, CanisRufus, *drew, Fireball~enwiki, Mwanner, Perspective, Aude, Spoon!, Femto, Incognito, ZooCrewMan, Sole Soul, Bobo192, Longhair, Meggar, Flxmghvgvk, Mikemsd, Chessphoon, Unquietwiki, Alpheus, Jag123, Alexs letterbox, Visualize, Minghong, Wrs1864, Haham hanuka, Jonathunder, SPUI, ClementSeveillac, Nkedel, Espoo, Danski14, Alansohn, JYolkowski, Cronus, GRider, Interiot, Arthena, Rd232, Jeltz, Andrewpmk, Plumbago, Zippanova, T-1000, Kocio, InShaneee, DavidCWG, Idont Havaname, Blobglob, Wtmitchell, BanyanTree, Uucp, Yuckfoo, Evil Monkey, W7KyzmJt, Kusma, Jsorensen, Someoneinmyheadbutit'snotme, Zootm, Kerry7374, Mikenolte, 4c27f8e656bb34703d936fc59ede9a, Kyrin, Bobrayner, Weyes, Boothy443, Kelly Martin, Woohookitty, LostAccount, Mindmatrix, Vorash, TigerShark, Scriberius, LOL, Nuggetboy, Localh77, Daniel Case, Baysalc, Snotty (renamed), WadeSimMiser, Drongo, Schzmo, BlaiseFEgan, Rchamberlain, Zzyzx11, Leemeng, Wayward, ??????, Zhen-Xjell, Stefanomione, Karam.Anthony.K, Zpb52, Palica, Allen3, MassGalactusUniversum, Graham87, Marskell, Deltabeignet, Magister Mathematicae, BD2412, Roger McCoy, RadioActive~enwiki, MauriceJFox3, Jclemens, Icey, Josh Parris, Canderson7, Sjakkalle, Seidenstud, Coemgenus, Baeksu, Eyu100, Dannysalerno, Amire80, Carbonite, Harro5, Nneonneo, Oblivious, Roivas, Creative210, OKtosiTe, Hermione1980, AySz88, Yamamoto Ichiro, Teddythetank, Eexlebots, RainR, Titoxd, FlaBot, Ecb29, Ian Pitchford, RobertG, Otnru, HowardLeeHarkness, Nihiltres, Arlondiluthel, JiFish, Avalyn, JYOuyang, Klosterdev, Rune.welsh, RexNL, Gurch, Quuxplusone, Intgr, Bmicomp, Noxious Ninja, Butros, King of Hearts, KaintheScion, Scoops, Bornhj, DVdm, Ariele, Voodoom, Bgwhite, YurikBot, Wavelength, Aleahey, Splintercellguy, Kencaesi, Kafziel, Adam1213, Pleonic, Hede2000, Bhny, Richjkl, Paul Quirk, Admiral Roo, Kirill Lokshin, Pvasiliadis, Van der Hoorn, Akamad, Chensiyuan, Amanaplanacanalpanama, Stephenb, Manop, Barefootguru, Coyote376, Gaius Cornelius, CambridgeBayWeather, Kyorosuke, Member, Wimt, MarcK, Crazyman, Wiki alf, Dialectric, God Of All, AlMac, RazorICE, Irishguy, Brian Crawford, Kynes, Rmky87, Ugnius, Amcfreely, Misza13, Tony1, FlyingPenguins, Zephalis, Pablomartinez, DeadEyeArrow, Bota47, Xpclient, Flipjargendy, Romal, Wknight94, Graciella, Zzuuzz, Encephalon, Gorgonzilla, Bayerischermann, AtOMiCNebula, Theda, Abune, Reyk, Dspradau, Sean Whitton, BorgQueen, GraemeL, Shawnc, Peter, QmunkE, Emc2, JLaTondre, MagneticFlux, Che829, Bluezy, Katieh5584, Kungfuadam, Plethorapw, NeilN, Leuk he, Kingboyk, Destin, Mardus, SkerHawx, That Guy, From That Show!, SG, Attilios, Veinor, MacsBug, Firewall-guy, SmackBot, Colinstu, Estoy Aquí, Justinstroud, KnowledgeOfSelf, Royalguard11, CompuHacker, Georgeryp, C.Fred, Blue520, Davewild, Matthuxtable, Stifle, El-Dakio, Delldot, KelleyCook, ProveIt, Vilerage, Ccole, Kaunietis25, Yamaguchi??, Gilliam, Ohnoitsjamie, Jushi, Oscarthecat, Skizzik, Chaojoker, ERcheck, Gary09202000, Chris the speller, Parajuris, Skintigh, Chemturion, Thumperward, Christopher denman, SchfiftyThree, Delink, Octahedron80, DHN-bot~enwiki, Darth Panda, Trimzulu, Jmax-, Can't sleep, clown will eat me, Frap, Episteme-jp, Nixeagle, JonHarder, Korinkami, Rablari Dash, Homestarmy, Xyzzyplugh, Jax9999, Midnightcomm, Mr.Z-man, Gabi S., Cybercobra, Engwar, Nakon, Valenciano, GhostDancer, Monotonehell, Warren, Weregerbil, Polonium, Sbluen, Sljaxon, Twain777, Fredgoat, Jeremyb, Kotjze, Nevyan, MOO, Risker, DataGigolo, Clicketyclack, SashatoBot, Rory096, Swatjester, JethroElfman, Heimstern, Tor Stein~enwiki, Xaldafax, Minna Sora no Shita, Abdomination, Llamadog903, PseudoSudo, LebanonChild, Chrisch, Mr. Vernon, Andypandy.UK, Jcmiras, Alistairphillips, Alistair.phillips1, Darklord.dave, MrArt, Mphill14, SandyGeorgia, Camp3rstrik3r, Jam01, Rip-Saw, Vernalex, Michael.koe, Sifaka, Jnk, Iridescent, Lonyo, JoeBot, Cowicide, Gholam, 10014derek, JHP, J Di, IvanLanin, Igoldste, Cbrown1023, RekishiEJ, AGK, Linkspamremover, Astral9, Kanecain, Mzub, Tawkerbot2, Morryau, Jasrocks, SMRPG, Clintmsand, Alestrial, George100, Dan1679, SkyWalker, J Milburn, JForget, FleetCommand, Anon user, Wikkid, Xlegiofalco, Ewc21, DevinCook, Pockle, Raceprouk, Green caterpillar, El aprendelenguas, Kejoxen, Herenthere, CJBot, Angelsfreeek, Kribbeh, Phatom87, TheBigA, Cydebot, Treybien, Steel, Gogo Dodo, Mroesler, Tiger williams, Bigjake, Shirulashem, Christian75, Codetiger, DumbBOT, TheJC, Omicronpersei8, Zalgo, Lo2u, Jed keenan, Satori Son, Tortillovsky, FrancoGG, Thijs!bot, Epbr123, Wikid77, Ilpalozzo, Supermario99, Daniel, Wikikiki~enwiki, Nonagonal Spider, Who123, Rcandelori, Jojan, Moulder, West Brom 4ever, A3RO, Cool Blue, Grayshi, CharlotteWebb, Nick Number, Wai Wai, Wikidenizen, Dawnseeker2000, Natalie Erin, Silver Edge, Escarbot, CamperStrike, Andykitchen, Mentifisto, Mr.Fraud, AntiVandalBot, Operator link, Luna Santin, Ownlyanangel, Schooop, Anotherpongo, Dylan Lake, Kmesserly, Shlomi Hillel, Pixelface, Jenny Wong, Falconleaf, Alevine-eantflick, Qwerty Binary, Ingolfson, JAnDbot, Hiddenstealth, Ginza, Barek, Epeefleche, BCube, Bhaddow, D. Kapusta, Dcooper, The elephant, Entgroupzd, MadMom2, Kipholbeck, SteveSims, Magioladitis, Bongwarrior, VoABot II, Mike5906, Abbadox, Yandman, Dfense, XPOTX, Tedickey, Twsx, Mikey129, LonelyWolf, Alekjds, Violetness, Robotman1974, Allstarecho, Cpl Syx, Fang 23, Bugtrio, Fayul, Glen, Myststix, NMaia, Gwern, Atulsnischal, Ksero, Gundato, Hdt83, MartinBot, M3tal H3ad, CliffC, BetBot~enwiki, Flamingpanda, Axlq, Skipatek, Lcaa9, Ittan, R'n'B, I2omani, Bgold4, RaccoonFox, J.delanoy, Fakir005, Trusilver, Deonwilliams, Neon white, Singing guns, Dispenser, Justinm1978, LordAnubisBOT, 2IzSz, Thomas Larsen, Compman12, Freejason, Demizh, Jwright1, Legendsword, AntiSpamBot, WikiChip, TomasBat, Bushcarrot, NewEnglandYankee, Hellohellohello007, فـنـدلـيز, Fsf~enwiki, Juliancolton, WarFox, Atama, Teggis, Redrocket, Wiki989, Mguy, Kiyo o, VolkovBot, ChrisPerardi, Jeff G., Tesscass, Dajahew1, TXiKiBoT, Oshwah, Zidonuke, Moogwrench, KevinTR, Rei-bot, GcSwRhIc, Shindo9Hikaru, Oxfordwang, Anna Lincoln, Melsaran, Martin451, LeaveSleaves, Alexarankteam, Master Bigode, Wiae, Copper20, Trickiality, Bercyon, Billinghurst, The Negotiator, Haseo9999, Flamesrule89, Lamro, Willbrydo, Digita, Mickelln, LittleBenW, AlleborgoBot, Fredtheflyingfrog, Fabioejp, EmxBot, Overtheblock, Superfly789, SieBot, Techwrite, Spartan, Backpackkk, Backpack123, Gorx, Jack Merridew, IHateMalware, Dawn Bard, Schwartz, Ken, Sephiroth storm, WJerome, Arda Xi, Pdub567, Oda Mari, Arbor to SJ, Jojalozzo, Nosferatus2007, Oxymoron83, Faradayplank, AngelOfSadness, Wjemather, ImageRemovalBot, Loren.wilton, ClueBot, Mr. pesci, GorillaWarfare, Fyyer, The Thing That Should Not Be, College222, Darthveda, Drmies, Mild Bill Hiccup, Braksus, Mackmar, Milenamm, Absmith111, Tokyogamer, Christineokelly, Bichon, Emperordarius, Igorberger, Rhododendrites, WalterGR, WWriter, Anti328, DanielPharos, Morriske, Apparition11, SF007, DumZiBoT, Adams527, Mikon8er, XLinkBot,

ICaNbEuRsOuLjAgIrL, Skarebo, SilvonenBot, Alexius08, Noctibus, Dubmill, Addbot, Deepmath, Clsdennis2007, Wowrocker2, Joost Kieviet, SuperSmashBros.Brawl777, AndrewJNeis, Christos2121, 15lsoucy, Ronhjones, Leszek Jańczuk, Skyezx, MrOllie, Glane23, Chzz, Debresser, Favonian, Mike A Quinn, Evildeathmath, Lightbot, OlEnglish, Qwertyytrewqqwerty, Fiilott, Luckas-bot, Yobot, Sdalk208, TaBOT-zerem, Voyage34, Aaronit~enwiki, Egosintrick, THEN WHO WAS PHONE?, SeanTheBest949, Writerjohan, KamikazeBot, Kypriano~enwiki, Fortmadder, TJDishaw6, Quentinv57, AnomieBOT, Keepitreal74, Roman candles, Jim1138, MinnetonkaCZ, IRP, Galoubet, Piano non troppo, AdjustShift, Wasisnt, Yachtsman1, Ulric1313, Materialscientist, Danno uk, The Firewall, Xqbot, Sionus, Capricorn42, Dubboy1969, Avastik, Junkcops, Katcrane, Halstonm, PraeceptorIP, Mlpearc, S0aasdf2sf, Ragityman, Danalpha31, Kurtdriver, BubbleDude22, Prunesqualer, Mathonius, IShadowed, Vuletrox, Luminique, Fastguy397, FrescoBot, VS6507, DigitalMonster, Cykloman15, Flakmonkey24, HamburgerRadio, Yodaddy4276, Jammy467, Pinethicket, Idemnow, Jacobdead, Chucknorriss007, JNorman704, Ngyikp, Brian Everlasting, SpaceFlight89, RandomStringOfCharacters, OMGWEEGEE2, Reconsider the static, MichaelRivers, Sahil16, Dinamik-bot, Vrenator, Halti1328, Sammonaran, Jeffrd10, Diannaa, JV Smithy, Thunerb, Tbhotch, Luis8750, Onel5969, Mean as custard, RjwilmsiBot, VernoWhitney, Buggie111, Xvunrealvx, Nabahat, EmausBot, John of Reading, Marmbrus, Bob22234, Dewritech, RA0808, L235, Tommy2010, Wikipelli, K6ka, Boysfood, Zach eastburn, Roflcopter23, EneMsty12, L0ngpar1sh, Wayne Slam, Isarra, Staszek Lem, Hidbaty223, Janesilentbob, Donner60, Damirgraffiti, FloridaShawn123, GrayFullbuster, Gary Dee, Jschwa12, ClueBot NG, Karlson2k, Cntras, Braincricket, 123Hedgehog456, O.Koslowski, Chikkey007, Widr, Neilacharya, Pattiewillford, Rubybarett, Icallitvera, DBigXray, Kwolton, Jordan james elder, PatrickCarbone, Larda, MusikAnimal, EmadIV, Brilubic2, YolentaShield, Cre8tin, Mechanic1545, VanEman, RobertEdingerPHD, Egyptianmorrow, J3zzy1998DBZ, Jeremy112233, Cyan.aqua, Squishy901, Cyberbot II, Rms1524, ZappaOMati, EuroCarGT, Dexbot, Cwobeel, Codename Lisa, Jamiedude2002, Geniusmanship, SFK2, Sourov0000, Corn cheese, Allne1972, François Robere, Melonkelon, Eyesnore, Yuvalg9, Jameii123, Muhammadbabarzaman, MountRainier, Majidmec, Babitaarora, Ymd2004, Someone not using his real name, Jianhui67, Kahtar, Dannyruthe, Mickel1982, 7Sidz, Saectar, Thibaut120094, BethNaught, Qwertyxp2000, Wii fi fi, KH-1, Zaixar, 7thwave1, JackDaniels11, Julietdeltalima, Silien2002, MagyVi, Securitysentry, Tripboom, KasparBot, Jiesenpan, JJMC89, Lux-hibou, Chenthil Vel, Srilekha selva, Topapp, Promestein, FrontsInFront, Getechfeed, Felixshariv, GreenC bot, Sameer092, Sisir123, Mohamdosama87, Chrish Jordan and Anonymous: 1380

- **Botnet** *Source:* https://en.wikipedia.org/wiki/Botnet?oldid=730721327 *Contributors:* The Anome, Fubar Obfusco, Jtk, DonDaMon, Edward, Pnm, Baylink, Plop, Dean p foster, Julesd, Dynabee, Kaihsu, Pedant17, Furrykef, Dimadick, Tbutzon, Walloon, Alerante, Gtrmp, Rick Block, Gracefool, Khalid hassani, Alvestrand, Ianneub, Olivier Debre, Moxfyre, Slavik0329, Freakofnurture, Rich Farmbrough, Bender235, Dewet, RJHall, Tjic, Bobo192, Jjmerelo~enwiki, Aquillion, Kjkolb, Krellis, Hooperbloob, ClementSeveillac, Joolz, BodyTag, InShaneee, Juhtolv, Kusma, BDD, Bsdlogical, Yurivict, Feezo, Simetrical, Woohookitty, Mindmatrix, Carlos Porto, Shello, Mihai Damian, Pol098, CiTrusD, JediKnyghte, Josh Parris, Rjwilmsi, PHenry, Yamamoto Ichiro, FlaBot, Latka, Gurch, Intgr, Zebediah49, Benlisquare, Dadu~enwiki, YurikBot, Wavelength, Samuel Wiki, StuffOfInterest, The Literate Engineer, NawlinWiki, Mosquitopsu, Scs, Deku-shrub, Flipjargendy, Romal, Abune, Rurik, Fsiler, Katieh5584, One, SmackBot, Narson, McGeddon, Brick Thrower, KelleyCook, Eiler7, Mcld, Gilliam, Ohnoitsjamie, Chris the speller, Kurykh, TimBentley, Jcc1, Sinicixp, DHN-bot~enwiki, Emurphy42, Jmax-, Can't sleep, clown will eat me, Trinite, Blah2, Mitsuhirato, Frap, JonHarder, Hitoride~enwiki, Luno.org, Rockpocket, Kuru, Euchiasmus, Ivucica, Ehheh, Ttul, Dl2000, Hu12, DabMachine, HisSpaceResearch, Iridescent, Winkydink, KimChee, Powerslide, DavidTangye, Kylu, Dgw, Jesse Viviano, Hserus, RagingR2, Abdullahazzam, Grahamrichter, Mzima, Mato, Gogo Dodo, DumbBOT, Optimist on the run, Zokum, Kozuch, Tobias382, Ferris37, Mbell, Ckhung, Aiko, Bobblehead, OrenBochman, Binarybits, Sidasta, Luna Santin, Tohnayy, Luxomni, Lfstevens, Mscullin, AndreasWittenstein, SemperSecurus, Husond, Sheitan, Struthious Bandersnatch, Andreas Toth, Magioladitis, VoABot II, Nyttend, Upholder, Boffob, Daniel.birket, Ryan1918, Forensicsguy, MartinBot, SasaMaker, LittleOldMe old, Boston, J.delanoy, EscapingLife, Skiidoo, Eliz81, Milo03, Mtxf, Buhadram, Fomalhaut71, Crakkpot, Jwh335, STBotD, Sbanker, VolkovBot, LokiClock, Franck Dernoncourt, Philip Trueman, TXiKiBoT, Stagefrog2, Brian Helsinki, Lambyte, Calculuslover800, Ephix, InFAN1ty, C45207, Senpai71, Michael Frind, Logan, Derekcslater, Sephiroth storm, Yintan, Android Mouse, Exert, Jonahtrainer, KoshVorlon, Lightmouse, Dracker, Denisarona, Escape Orbit, The sunder king, Mr. Granger, Jaimee212, Church, ClueBot, GorillaWarfare, Abhinav, Vacio, Ravivr, Lawrence Cohen, Konsumkind, Pwitham, Paul Abrahams, Mild Bill Hiccup, DnetSvg, Dante brevity, Rprpr, Julesbarbie, Excirial, Gulmammad, Dralokyn, Rhododendrites, SchreiberBike, DanielPharos, D.Cedric, BlueDevil, Herunar, XLinkBot, Dark Mage, Stickee, Little Mountain 5, WikHead, Jadtnr1, A little mollusk, Addbot, Ramu50, A.qarta, Burkestar, Enkrona, Zellfaze, Tothwolf, Linktopast30, Scientus, MrOllie, Danpoulton, Hintss, Jarble, Luckas-bot, Yobot, Ptbotgourou, AnomieBOT, Jim1138, Yachtsman1, Materialscientist, Hcps-spottsgr, LykMurph, ArthurBot, Quebec99, Xqbot, THWoodman, DataWraith, BebyB, S0aasdf2sf, GrouchoBot, Kyng, Chaheel Riens, FrescoBot, W Nowicki, Chilrreh, HamburgerRadio, 10metreh, Skyerise, Bugsguy, Pastafarian32, GlowBee, Fishsicles, Lbwilliams, Trappist the monk, Dundonite, Lotje, Dragan2~enwiki, Tbhotch, Jfmantis, Onel5969, Liamzebedee, Ripchip Bot, EmausBot, Jackson McArthur, Cmartincaj, Heracles31, ScottyBerg, Dewritech, JohnValeron, RenamedUser01302013, K6ka, Marshviperx, Martinibra, Daonguyen95, A930913, H3llBot, Ivhtbr, Erianna, Staszek Lem, TyA, The guy on da moon, Cyberdog958, Schnoatbrax, Rigley, Donner60, TravisMunson1993, Whoop whoop pull up, Mjbmrbot, Gary Dee, ClueBot NG, Magicman3894, MelbourneStar, Satellizer, Abecedarius, Guive37, Twillisjr, Mgnicholas, Mesoderm, O.Koslowski, Helpful Pixie Bot, Harley16ss, TRANA1-NJITWILL, Lifemaestro, Hewhoamareismyself, Fredo699, Vagobot, DaveB549, Paulbeeb, ElphiBot, MusikAnimal, Socal212, Affinanti3, Szary89, AdventurousSquirrel, Zune0112, Jbarre10, Gyvachius, Tetraflexagon, Haleycat, Cyberbot II, Deimos747, Faisal ALbarrak, Oknitram, Chengshuotian, Padenton, Superkc, Waqob, FoCuSandLeArN, Oneplusnine, Agent766, Axesrotoor, Jakedtc, Compdewd, FockeWulf FW 190, FrB.TG, Herpingdo, JaconaFrere, Impssswoon, TheEpTic, Jamesmarkchan, AnonArme, Fl4meb0tnet, Professornova, Anotherdaylate, Crystallizedcarbon, Spagheti, Ceannlann gorm, MusikBot, BrainSquared, Yasuo Miyakawa, Kurousagi, UttamSINHA, JohnStew826, Hacktivist117, Sharanyanaveen, Dogeipedia, GreenC bot, Jeaser91, Shraypuri, Ps765330, Jicetus and Anonymous: 484

- **Keystroke logging** *Source:* https://en.wikipedia.org/wiki/Keystroke_logging?oldid=730722265 *Contributors:* Derek Ross, LC~enwiki, The Anome, SimonP, R Lowry, Edward, Lir, Pnm, Ixfd64, Ellywa, Ronz, Angela, Kingturtle, Aimaz, Rossami, Evercat, Samw, GCarty, Guaka, Aarontay, Ww, Dysprosia, WhisperToMe, Markhurd, Tschild, Furrykef, Nv8200pa, Omegatron, Jamesday, Catskul, Blugill, Lowellian, Hadal, Wereon, David Gerard, DavidCary, Laudaka, Jason Quinn, AlistairMcMillan, Solipsist, Antandrus, Beland, OverlordQ, Lynda Finn, Mike Rosoft, Discospinster, Rich Farmbrough, ArnoldReinhold, Xezbeth, ZeroOne, JoeSmack, Sietse Snel, RoyBoy, Femto, Adambro, Yono, Bobo192, Nigelj, Stesmo, Wisdom89, Dteare, Starchild, Alansohn, Danhash, Bobrayner, Woohookitty, Unixer, Armando, Pol098, WadeSimMiser, Firien, Dbutler1986, Graham87, JIP, Rjwilmsi, DickClarkMises, FlaBot, Weihao.chiu~enwiki, Latka, JiFish, Intgr, Runescape Dude, Salvatore Ingala, Peterl, Whosasking, Tiimage, YurikBot, Wavelength, Borgx, FlareNUKE, Lincolnite, Conscious, Hede2000, SpuriousQ, Rsrikanth05, Wimt, Mipadi, Bob Stromberg, Vivaldi, Tony1, Occono, Palpalpalpal, DeadEyeArrow, Closedmouth, GraemeL, Egumtow, Stefan yavorsky, Baxil, Veinor, A bit iffy, SmackBot, Royalguard11, Hydrogen Iodide, Gnangarra, J.J.Sagnella, Ohnoitsjamie, Skizzik, Chris the

speller, Optikos, @modi, MK8, DHN-bot~enwiki, Colonies Chris, Firetrap9254, KojieroSaske, SheeEttin, Frap, Skidude9950, Ww2censor, Flask215, Khoikhoi, Engwar, Nakon, Gamgee, Kalathalan, Clicketyclack, James Allison, Ckatz, Tuanmd, Redboot, Ehheh, Njb, Mets501, H, Kvng, Mike Doughney, Pauric, Sander Säde, On1ine, Jeremy Banks, JForget, Dycedarg, Jesse Viviano, Gogo Dodo, Corpx, Alexdw, Odie5533, Tawkerbot4, Bposert, DumbBOT, SJ2571, Njan, Alexey M., Epbr123, FTAAP, Snydley, RamiroB, Sheng.Long 200X, Druiloor, AntiVandalBot, Luna Santin, Seaphoto, Fayenatic london, Zorgkang, Spydex, Qwerty Binary, Dreaded Walrus, JAnDbot, Thylacinus cynocephalus, Tony Myers, Barek, Bakasuprman, A1ecks, Hut 8.5, Isthisthingon, Techie guru, .anacondabot, Magioladitis, Jaysweet, Ukuser, JNW, Cheezyd, Confiteordeo, Fedia, Wikivda, Wikire, MartinBot, STBot, CliffC, Jonathan.lampe@standardnetworks.com, Anaxial, Keith D, Nono64, $pider, Tresmius, Slash, J.delanoy, Pharaoh of the Wizards, Cyrus abdi, Thomas Larsen, Samtheboy, Noogenesis, VolkovBot, TreasuryTag, MemeGeneScene, Jeff G., Philip Trueman, TXiKiBoT, Mrdave2u, Zifert, A4bot, Glarosa, Isis4563, Madhero88, Dirkbb, Turgan, Jjjccc~enwiki, ChewyCaligari, Rock2e, Resurgent insurgent, Cool110110, SieBot, Triwbe, Sephiroth storm, Nmviw, Arda Xi, OsamaBinLogin, Banditauron, Tombomp, Clearshield, Dillard421, ArchiSchmedes, ClueBot, Wilbur1337, The Thing That Should Not Be, AsymptoteG, Garyzx, Dotmax, Blanchardb, Asalei, Socrates2008, Rhododendrites, Technobadger, Manasjyoti, Arjayay, Drwhofor, Shin-chan01, El bot de la dieta, DanielPharos, Berean Hunter, Johnuniq, SF007, Noname6562, Darkicebot, Against the current, XLinkBot, Spitfire, Stickee, Rror, Dom44, Lamantine, WikHead, Dsimic, Tustin2121, Addbot, Mortense, Movingboxes, Rhinostopper, MrOllie, Etracksys, Matt5075, Networkintercept, Favonian, ChenzwBot, Sureshot327, Tide rolls, MuZemike, Luckas-bot, Yobot, 2D, Bigtophat, Navy blue84, AnomieBOT, Andrewrp, Kingpin13, Ulric1313, Materialscientist, Are you ready for IPv6?, ℍuman, HkBattousai, GB fan, LilHelpa, Xqbot, Dragonshardz, Jeffrey Mall, Reallymoldycheese, Automaite, Ezen, S0aasdf2sf, Aceclub, RadiX, GrouchoBot, IslandLumberJack, Mark Schierbecker, Krypton3, 78.26, Aenus, Mountielee, Prari, FrescoBot, WPANI, Clubmaster3, DigitalMonster, PeramWiki, Nathancac, Waller540, HamburgerRadio, Italick, Redrose64, Tom.Reding, Rajtuhin, Serols, MKFI, AgentG, Reconsider the static, Ao5357, Lotje, Vrenator, Mean as custard, F11f12f13, Sloppyjosh, Forenti, DASHBot, J36miles, EmausBot, Manishfusion1, GoingBatty, Wikipelli, LinuxAngel, FlippyFlink, John Cline, Ida Shaw, Traxs7, S3cr3tos, Δ, Ego White Tray, AlexNEAM, ClueBot NG, Matthiaspaul, O.Koslowski, Mactech1984, Lolpopz1234, Marsmore, Nbudden, BG19bot, IraChesterfield, Samiam111~enwiki, Guesst4094, Carliitaeliza, MeanMotherJr, BattyBot, Abgelcartel, Jfd34, Millennium bug, Lloydliske, EagerToddler39, Codename Lisa, Webclient101, Klabor74, Zhiweisun, Jaericsmith, Sourov0000, Corn cheese, Way2veers, Yuvalg9, MountRainier, JadeGuardian, Kennethaw88, Lvanwaes, Mover07, Jianhui67, Dannyruthe, NewWorldOdor, FockeWulf FW 190, Janeandrew01, Michael Dave, CNMall41, Jamesmakeon, Bobsd12, Wasill37, KH-1, Crystallizedcarbon, Scyrusk, Devwebtel, JoanaRivers, ScottDNelson, Jhfhey, Awmarks, Muhammad mobeen83, BD2412bot, Wikijagon, Sharanyanaveen, Ivory Dream, White Arabian Filly, The Voidwalker, Chrisprice07, Phillipsreggie999, James Samwise Ganji, JAMES MCGUMMERY, Deshawn992, Deshawn991, Jameshardwell, Khorkhe, Luigiboy260, Richardparker119, Supportpc and Anonymous: 564

- **Form grabbing** *Source:* https://en.wikipedia.org/wiki/Form_grabbing?oldid=721557005 *Contributors:* The Anome, Pnm, Discospinster, Rtc, Breno, Yugsdrawkcabeht, EoGuy, Nnemo, Muhandes, UnCatBot, Addbot, Amirobot, Raven1977, IslandLumberJack, EmausBot, Widr, BG19bot, Quantumbenxh, Codename Lisa, FockeWulf FW 190, Rong Zhou, Nhiggins2013, Prabragu and Anonymous: 16

- **Web threat** *Source:* https://en.wikipedia.org/wiki/Web_threat?oldid=705324657 *Contributors:* Edward, Pascal666, Espoo, Mindmatrix, Mistress Selina Kyle, Soap, Alaibot, Widefox, Obiwankenobi, Largoplazo, Kmverdi, JL-Bot, Sgroupace, Dthomsen8, Favonian, Yobot, AnomieBOT, CXCV, Sophus Bie, Calmer Waters, HiW-Bot, W163, Pastore Italy, Nhero2006, Zakblade2000, Helpful Pixie Bot, Mdann52, FockeWulf FW 190, Greenmow and Anonymous: 6

- **Dialer** *Source:* https://en.wikipedia.org/wiki/Dialer?oldid=726793940 *Contributors:* William Avery, SimonP, KF, Norm, Sannse, Ronz, Jimfbleak, Dysprosia, E23~enwiki, Pakaran, Hadal, Alf Boggis, Xorx77, SWAdair, Pgan002, SebastianBreier~enwiki, Dirus, Sam Hocevar, Neutrality, Brianhe, CanisRufus, Art LaPella, Smalljim, Rd232, Josh Parris, Vegaswikian, Margosbot~enwiki, Avalyn, Srleffler, Hede2000, Perry Middlemiss, Yudiweb, Bluezy, KnightRider~enwiki, SmackBot, Slashme, PJTraill, Frap, Mosca, Warren, Pgillman, Euchiasmus, General Ization, Shattered, Defireman, Pavithran, Koffie, Thijs!bot, Pogogunner, Codecheetah, Cspec chaplin, Jhansonxi, Coffee4binky, Japo, Calltech, Stephenchou0722, STBot, BetBot~enwiki, TyrS, Vlozqmdp, Steel1943, LittleBenW, Austriacus, PeterCanthropus, Scarian, Wageslave, Pxma, SchreiberBike, Addbot, Scientus, Rishishringan, Lightbot, Teles, FrescoBot, HamburgerRadio, Full-date unlinking bot, Mean as custard, EmausBot, Dewritech, Access Denied, Rostz, Kale31899, ClueBot NG, Teresa.jose, Widr, Rikrobson, BG19bot, BattyBot, Rix Ryskamp, Mikez711, TheJJJunk, Khazar2, Codename Lisa, Faizan, Alexorton23, Prospectingnut, Kapil budhiraja, Aatifriaz1, Zamaster4536, Chriskallen and Anonymous: 58

- **Internet bot** *Source:* https://en.wikipedia.org/wiki/Internet_bot?oldid=730685635 *Contributors:* Ken Arromdee, Tregoweth, Haakon, Ronz, Dino, Furrykef, Cleduc, Robbot, Pigsonthewing, Costello, Alan Liefting, Jacoplane, Joe Sewell, Wronkiew, Utcursch, Knutux, Antandrus, Zantolak, Shrimppesto, Jcw69, Discospinster, Shanes, RoyBoy, Deathawk, Foobaz, Opusaug, Alansohn, Kelly Martin, Lincspoacher, Uncle G, Cos~enwiki, Apokrif, RichardWeiss, Pentawing, Rjwilmsi, Tizio, Koavf, Kinu, Goldfndr, Crazynas, FlaBot, Nihiltres, David H Braun (1964), Junior kimura, Imnotminkus, Bornhj, Korg, YurikBot, Whoisjohngalt, TheTrueSora, Shanel, NawlinWiki, Guy Hatton, Robdurbar, Zwobot, Aaron Schulz, BOT-Superzerocool, Denis C., Ilmaisin, Werdna, Jverkoey, Joeeste, Mike Dillon, GraemeL, ViperSnake151, Serendipodous, Sycthos, Yvwv, Crystallina, SmackBot, Narson, PJM, Xaosflux, Improbcat, Bluebot, Keegan, JDowning, Tigerhawkvok, Octahedron80, Nbarth, DHNbot~enwiki, Namkim, Frap, JonHarder, VMS Mosaic, Mr.Z-man, Soosed, MichaelBillington, Mtmelendez, Pilotguy, Synthe, Rory096, Yellow up, Sevensevenseven22, Putnamehere3145, Filippowiki, The Sting, Hvn0413, Lukis100, Ultimaga, Manifestation, LaMenta3, Landeyda, Hu12, PaulGS, Mousaynon, Martin Kozák, Phoenixrod, Tawkerbot2, George100, Vanisaac, Picaroon, DaveHepler08, WeggeBot, JenG24, A876, Agentdenim, Sempai, Optimist on the run, Cowpriest2, Mikewax, UberScienceNerd, JamesBrownJr, Thijs!bot, Epbr123, Daa89563, Mercury~enwiki, Marek69, John254, Neil916, Navigatr85, AntiVandalBot, SmileFace, JAnDbot, Dereckson, MER-C, Shermanmonroe, BranER, Xeno, SteveSims, VoABot II, LunaticBeatnik, CheMechanical, Giggy, Orangemonster2k1, DerHexer, Ryan1918, R27smith200245, Hdt83, MartinBot, NochnoiDozor, Yaron K., J.delanoy, Captain panda, Bitbotbotbot, Skiidoo, McSly, Mikael Häggström, PocklingtonDan, Mikevde, Hersfold, Jeff G., A.Ward, Dajwilkinson, Oshwah, Debeaux, Monkey Bounce, Ocolon, Melsaran, Optigan13, Jamelan, FlyingLeopard2014, Lilbunnifufu101, SieBot, Coffee, Mikemoral, Keilana, James3uk, Askild, CharlesGillingham, Remiaubert, Caaaggf, Luminus9, ClueBot, Lawrence Cohen, Watadoo, Lensicon~enwiki, MsMagoo, DragonBot, Panyd, PixelBot, Abrech, Dk 1969, Cecelegy, Ivantheterrible1234, NinetyCharacters, XLinkBot, Emmette Hernandez Coleman, PseudoOne, Jovianeye, Noctibus, Addbot, Lithoderm, Tothwolf, AkhtaBot, Gremel123, Vishnava, Leszek Jańczuk, Chamal N, Debresser, Jasper Deng, Arbitrarily0, Michaello, Swarm, Luckas-bot, Yobot, JackPotte, Ptbotgourou, It's Been Emotional, AnomieBOT, Angry Mushi, AlotToLearn, Ai24081983, Piano non troppo, Bendiben, Law, Brightgalrs, 12056, Hedge777, Bandishbhoir, Elemesh, SD5, FrescoBot, Holysharky, Occamsrazorwit, Sae1962, HamburgerRadio, A F K When Needed, Kusnob, Strontium86, Lotje,

Ggoere, PurplePiper, Ammodramus, EmausBot, John of Reading, WikitanvirBot, ZéroBot, Emphrase, Denver & Rio Grande, Erianna, Brandmeister, Donner60, ChuispastonBot, GermanJoe, Rainbowroad6w, Nhero2006, Llightex, ClueBot NG, Subbyte, BG19bot, Bair175, Mark Arsten, Slashinmaine, Seoyazilimlari, Larryh6, Pikachu Bros., Oldschooldsl, Wikpoint, Justincheng12345-bot, YFdyh-bot, Laberkiste, Dobie80, Enterprisey, Mogism, PeacefulPlanet3, SFK2, Me, Myself, and I are Here, Gogo Rulez, DudeWithAFeud, Lakun.patra, Nickolay.mateev, Jameshine, Jaqoc, Lgats, Alxtye, Hhm8, MRD2014, Percylipinski, KH-1, Hechengzhi, YITYNR, Mkwiipe, Sajid saj Gour, Rebeccaprofile, TheDopestEva, Bcrouchjrff, Proud User, Sorainnosia, WaexuWiki, Boomer Vial, Constanstin, Addusimu and Anonymous: 272

- **Scareware** *Source:* https://en.wikipedia.org/wiki/Scareware?oldid=729968038 *Contributors:* Edward, Andres, Nurg, Xanzzibar, Ebear422, Grm wnr, Andrewferrier, Evice, CanisRufus, Kappa, TheParanoidOne, Jivlain, Kgrr, Mandarax, NeonMerlin, Robert A West, Nentuaby, SmackBot, C.Fred, Robocoder, Breno, Goodnightmush, Sabertooth, Uruiamme, Prolog, Barek, TheOtherSiguy, AlmostReadytoFly, Michael Goodyear, Maurice Carbonaro, DoubleZeta, SovereignGFC, VolkovBot, Fences and windows, TXiKiBoT, Darkrevenger, Jamelan, The Seventh Taylor, Thisismyrofl, LittleBenW, Logan, Kbrose, Moonriddengirl, WereSpielChequers, Alexbot, Socrates2008, Callinus, Mikon8er, Nathan Johnson, ErkinBatu, MystBot, Addbot, Zellfaze, Sillyfolkboy, GastonRabbit, Megaman en m, Luckas-bot, Yobot, UltraMagnus, Metalhead94, Xqbot, Jergling, Vanished user oweironvoweiuo0239u49regt8j3849hjtowiefj234, ???, Cantons-de-l'Est, Tabledhote, FrescoBot, LucienBOT, Haein45, HamburgerRadio, Anonymous07921, Winterst, Starbox, Train2104, Deadrat, Jonkerz, Lotje, Teenboi001, Chris Rocen, WikitanvirBot, Angrytoast, Dewritech, Illogicalpie, Pooh110andco, ZéroBot, Fæ, The Nut, Wabbott9, Staszek Lem, Taistelu-Jaska, DASHBotAV, Roambassador, ClueBot NG, Vibhijain, BG19bot, Northamerica1000, Mccallister8, Flexo013, Security Shield, Farqad, Codename Lisa, Mogism, Sourov0000, 069952497a, Mannir muhammad, Sharanyanaveen and Anonymous: 71
- **Rogue security software** *Source:* https://en.wikipedia.org/wiki/Rogue_security_software?oldid=730594476 *Contributors:* Fubar Obfusco, Julesd, WhisperToMe, IceKarma, Mazin07, Senthil, Nurg, Jfire, Vsmith, Bender235, Unquietwiki, Charonn0, Espoo, Anthony Appleyard, Ron Ritzman, Woohookitty, RHaworth, Ruud Koot, MyFavoriteMartin, JIP, Rjwilmsi, Koavf, Quietust, Bubba73, Kri, Aussie Evil, NawlinWiki, FlyingPenguins, Real decimic, Nailbiter, Kevin, SmackBot, Jasy jatere, SmackEater, Ohnoitsjamie, Rmaster1200, Chris the speller, Improbcat, Trimzulu, JonHarder, Azumanga1, Nakon, Dreadstar, Kukini, Fanx, Cableguytk, Green Giant, JHunterJ, Peyre, TheFarix, Blakegripling ph, RekishiEJ, Cryptic C62, FatalError, Fredvries, CmdrObot, Ivan Pozdeev, Jackzhp, Iuio, Greystork, Davidnason, Sadaphal, DevinCook, Poowis, Jesse Viviano, Seven of Nine, Gogo Dodo, Ameliorate!, Satori Son, Malleus Fatuorum, Nonagonal Spider, Dalahäst, Pogogunner, Obiwankenobi, AjaaniSherisu, Barek, Mark Grant, Geniac, Rhodilee, VoABot II, WikiMax, Michael Goodyear, Schumi555, Inclusivedisjunction, Gwern, CliffC, Falazure, R'n'B, CommonsDelinker, .1337., Pharaoh of the Wizards, Skier Dude, AppleMacReporter, Chaosraiden, TanookiMario257, Talon Kelson, Bonadea, VolkovBot, N3cr0m4nc3r, Philip Trueman, Jart351, Kok3388, Haseo9999, Synthebot, LittleBenW, AlleborgoBot, PedroDaGr8, Kbrose, Raiden X, Malcolmxl5, IHateMalware, Sephiroth storm, Flyer22 Reborn, EwanMclean2005, Es101, Phykyloman~enwiki, Jessejaksin, ClueBot, Chuckbronson45, LizardJr8, Chiazwhiz, Regardless143, Excirial, Socrates2008, Anon lynx, NuclearWarfare, Purplewowies, Autoplayer, Htfiddler, DanielPharos, Callinus, SF007, DumZiBoT, Mikon8er, Ginnrelay, Sigrulfr, Wikiuser100, Jennysue, Texasrexbobcat, Addbot, Xp54321, Trapped34, Flo 1, Smithereen, GastonRabbit, Yobot, Tohd8BohaithuGh1, Marvel Freak, Jponiato, KieranC15, SilverEyePro, Killiondude, Obersachsebot, CPU01, Jeffrey Mall, Junkcops, ???, A1a33, S0aasdf2sf, КоЯп, BubbleDude22, AndrewCrogonklol, Pre-Cautioned Watcher, Beaconblack, Uvula!, Fastguy397, WPANI, Stars1408, ObbySnadles, HamburgerRadio, Citation bot 1, MaxwellHerme, ASCMSEM1, Jacobdead, Σ, Starbox, MichaelRivers, Deadrat, Lotje, Someone 123, Kanzler31, Born2bgratis, Spyware exspert, Sky17565292, Ebe123, Progprog, Techguy197, Kunal0315, Napeyga, Champion, Dlowe2224, Ego White Tray, DClason93, Kldukes, Rich Smith, Skylar130, Helpful Pixie Bot, Kinaro, IlSignoreDeiPC, Terryforsdyke, Meatsgains, David.moreno72, JC.Torpey, HybridBiology, Farqad, Tangaling, Cyberbot II, Dexbot, Zeeyanwiki, Codename Lisa, Webclient101, Kephir, Soda drinker, Sourov0000, MasterONEz, Thatonewikiguy, Marufsau, PersistedUser, Lemondoge, JohnStew826, Revantteotia, Sharanyanaveen, Logonemeri, Keni, GreenC bot, KATMAKROFAN and Anonymous: 267
- **Ransomware** *Source:* https://en.wikipedia.org/wiki/Ransomware?oldid=730620766 *Contributors:* Ubiquity, Nealmcb, Karada, Furrykef, ZeWrestler, JonathanDP81, HaeB, Pengo, Bradeos Graphon, Sam Hocevar, Monkeyman, FT2, Demitsu, Art LaPella, Bobo192, 23skidoo, Transfinite, Ricky81682, Andrew Gray, Quintin3265, Someoneinmyheadbutit'snotme, Firsfron, Pol098, Apokrif, BD2412, Rjwilmsi, Roboto de Ajvol, RobotE, RussBot, Hydrargyrum, MightyGiant, Rsrikanth05, Futurix, Deku-shrub, Caerwine, Nailbiter, Sandstein, Paul1149, ViperSnake151, Harrisony, Rtc, Howardchu, GraemeMcRae, Chris the speller, Bluebot, Snori, Frap, JonHarder, Mosca, JzG, General Ization, Bourgeoisdude, WilliamJE, Bobamnertiopsis, Steel, Gogo Dodo, Underpants, Thijs!bot, Mdriver1981, Dawnseeker2000, GiM, Arsenikk, Mac Lover, Bull3t, Davidpage, Magioladitis, JNW, CodeCat, Alekjds, David Eppstein, Inclusivedisjunction, Connor Behan, SecVortex, Public Menace, Fluteboy, WWGB, VolkovBot, Jackfork, Wiae, Jponnoly, LittleBenW, Michael Frind, Logan, The Thing That Should Not Be, Mild Bill Hiccup, Socrates2008, PixelBot, Arjayay, DanielPharos, Toxicbreakfast, XLinkBot, Avoided, ErkinBatu, Addbot, MrOllie, Download, Jasper Deng, Alwaystech, Luckas-bot, Yobot, GamerPro64, AnomieBOT, Tezpur4u, КоЯп, Dhuruvan, FrescoBot, WPANI, Ahmer Jamil Khan, Citation bot 1, Zunter, MastiBot, Thegod2112, Deadrat, Lotje, Seahorseruler, Ivanvector, Topsfield99, Tbhotch, RjwilmsiBot, EmausBot, Dewritech, ZéroBot, Wingman4l7, ClueBot NG, Xxjackjackxx, برسام, Reify-tech, Oddbodz, Egle0702, BG19bot, Jassy pal, BattyBot, Lions of Inquiry, Ostrava001, IjonTichyIjonTichy, Dexbot, Codename Lisa, Webclient101, Joshtaco, Dozzzzzzzzzing off, ZAApplebee, SMB99thx, Oranjelo100, Williamwalace33389789, FockeWulf FW 190, Racer Omega, Wincent77, Fixuture, Abudibaa, The Infobox Strikes Again!, Cybersecurity101, Wolly Lee, Macofe, ElectronicKing888, 0xF8E8, DariusMcSean, Kidsankyran, User000name, Amccann421, PersistedUser, Sudipmus, Yasuo Miyakawa, Zoomy921, Hoang the Hoangest, Natasha Miranda, Chenthil Vel, Fivestarts, 69 Man Shark 69, Eurocus47, Bsdpuffy, Bby1-psu, Epicawesomewolf, Binary Fission, Sspywareoff, Tani.oliver, Shafqat571, Rossjamesfrank and Anonymous: 106
- **Linux malware** *Source:* https://en.wikipedia.org/wiki/Linux_malware?oldid=729219902 *Contributors:* SimonP, Merphant, Comte0, Norm, Paddu, Berengar~enwiki, Dysprosia, Markhurd, AnonMoos, Rursus, SchmuckyTheCat, Mattflaschen, Giftlite, Mark Richards, Malbear, Chowbok, Bradlegar, Thorwald, Rich Farmbrough, Bender235, Nigelj, Chbarts, Charonn0, Conny, Bart133, Snowolf, Danaman5, Sether, Linum~enwiki, Deeahbz, Radiant!, Marudubshinki, JIP, Rjwilmsi, Gudeldar, Melancholie, Ahunt, WriterHound, Wolfmankurd, Hendrixski, Hydrargyrum, Lesfer, Cryptic, ScottyWZ, Booch, Rfsmit, Shimei, Mike Selinker, Vandalbot on wheels!, Rwwww, Macoafi, Luk, SmackBot, InverseHypercube, Geira, KVDP, Dark Apostrophe, EdgeOfEpsilon, JGXenite, Audriusa, Modest Genius, Can't sleep, clown will eat me, Frap, Ultra-Loser, Gothmog.es, Ruolin59, Niczar, Hvn0413, Stephenjudge, Laurens-af, Bandan, Derekgillespie, Megatronium, Andrewtappert, Leevanjackson, Inzy, Cydebot, Cwhii, Gogo Dodo, B, Kozuch, Thijs!bot, Jdm64, SteveSims, Xb2u7Zjzc32, Stuart Morrow, Tedickey, Xavierorr, The cattr, Maddog39, Stephenchou0722, Aliendude5300, Lifanxi~enwiki, G Velasco, Archolman, Fakewcfrog, Althepal, Lietk12, Homo logos, X!, Mythrill, Wikkedit, Weejock, TXiKiBoT, Theosch, Miranda, Natg 19, Draconx, Unknown Unknowns, EmxBot, Winchelsea, Jerryobject, Infestor, Jayvince, Pini-pini, Jay Turner, Treekids, Kortaggio, Martarius, ClueBot, The Thing That Should Not Be, Niceguyedc, Cavalary, Rprpr,

KenJackson.US, Ai ja nai, Grandmasterfc, Alexbot, Zomno, WalterGR, TobiasPersson, BOTarate, Atallcostsky, SF007, XLinkBot, IsmaelLuceno, Addbot, LPfi, Anklyne, Unibond, Serola, Mattventura, AnomieBOT, Kotika98, TheTrueCorrector, DeG84, Vlad003, Akemi Loli Mokoto, HamburgerRadio, Skyerise, RedBot, Jonkerz, Lotje, Rpt0, Vrenator, EmausBot, Dewritech, Cheeseman182, Wikipelli, Thenotonly, Ryanm84, Kenny Strawn, Warezbb, ClueBot NG, Lexy username, Mesoderm, Runnermule, Helpful Pixie Bot, Wb4whd, BattyBot, Cyberbot II, Pokajanje, Sc9953, FockeWulf FW 190, Hibbarnt, BlindForay, Fleetwoodta, Monkbot, WikiGopi, Bbustill, Hariuser123, Chenthil Vel, Arthur2968, Msearce and Anonymous: 184

- **Palm OS viruses** *Source:* https://en.wikipedia.org/wiki/Palm_OS_viruses?oldid=721562570 *Contributors:* JoeSmack, RJHall, TheParanoidOne, Psantora, Korg, Koffieyahoo, Dialectric, Aaron Brenneman, Garion96, J7, EatsWontons, CmdrObot, Cydebot, Obiwankenobi, PEAR, White 720, Fredrick day, HideandLeek, Hornet35, Alexius08, Yobot, AnomieBOT, HamburgerRadio, DARTH SIDIOUS 2, Access Denied, Nhero2006, Civican, Cyberbot II, FockeWulf FW 190, Alphaslucas and Anonymous: 15
- **Mobile Malware** *Source:* https://en.wikipedia.org/wiki/Mobile_Malware?oldid=724620156 *Contributors:* Dreamyshade, Ronz, Markhurd, Secretlondon, Everyking, Gadfium, RossPatterson, El C, LtNOWIS, Cdc, Mindmatrix, David Haslam, Zpetro, Feydey, Fourdee, Borgx, Sceptre, Rwalker, Dv82matt, Rajeevdb, Veinor, SmackBot, McGeddon, Gilliam, Couchand, JoeBot, Ikajaste, Cydebot, ST47, Alaibot, Kozuch, Dgies, AntiVandalBot, Obiwankenobi, Barry26, MER-C, Awilley, Hut 8.5, Gapple, Anaxial, Keith D, Grmarkam, VolkovBot, SimplyCellPhone, Oshiri, Canaima, Natg 19, Enigmaman, SieBot, Avanzare, Jojalozzo, Ericeverson, ClueBot, General Trelane, Sun Creator, Jaizovic, DanielPharos, Parallelized, SilvonenBot, PL290, Addbot, Writermonique, Innv, Leszek Jańczuk, Yobot, Fraggle81, AnomieBOT, JackieBot, Leo45555, Kernel.package, FrescoBot, Remotelysensed, Recognizance, HamburgerRadio, Buddy23Lee, Mean as custard, Connoe, 1Veertje, Donner60, Nhero2006, ClueBot NG, Steve dexon, Jeraphine Gryphon, BG19bot, Virtualerian, Neøn, MusikAnimal, Marcustoh3, UNOwenNYC, Dermoid, Gtangil, Oriol.corroto, Lemnaminor, Sxs957, Didigodot, Rbandx, JamesMoose, Muneezarafiq, AhmadQn, Waqaee, A.sky245, Binojvj96, MightyCool56, Gnagendrareddi, TutuMarie07, Kahtar, Tonywalmeida, GBK415, Akash Bedi13, Highlander9535, FockeWulf FW 190, Paulexyn0, Claire321.1728, Habberwock, Rebecca Ren, Portmanteau88, Monkbot, Baltergeist, Jeomarisaez, KH-1, Chenthil Vel, Parthimurugesan, Sharanyanaveen, Faad1, Realwikiexpert, Search4sure, Khan The REbel and Anonymous: 74
- **Macro virus** *Source:* https://en.wikipedia.org/wiki/Macro_virus?oldid=730155913 *Contributors:* The Cunctator, LC~enwiki, Lee Daniel Crocker, Fubar Obfusco, Ellmist, Pnm, Jdforrester, WhisperToMe, Munford, Tempshill, Floydian, J D, Yosri, David Gerard, Fennec, Laudaka, Utcursch, Oneiros, Sietse Snel, Pearle, Alansohn, CyberSkull, AzaToth, Mac Davis, Bellhalla, Masterjamie, E090, Rjwilmsi, FlaBot, YurikBot, Hede2000, NawlinWiki, ColinFine, Emmyceru, SmackBot, Gilliam, Skizzik, Chris the speller, Bluebot, Thumperward, Martijn Hoekstra, Mistress Selina Kyle, Bejnar, KenWalker, Jokes Free4Me, Cydebot, Epbr123, TurboForce, Themeparkfanatic, Mentifisto, Ajolliffe bmack, Bongwarrior, Alekjds, Kayau, Gwern, Schmloof, Ertyseidel, J.delanoy, DarkFalls, Cooldude7273, CardinalDan, JRS, VolkovBot, Vrac, Nagy, Malcolmxl5, Tombomp, ClueBot, Trivialist, DanielPharos, Spitfire, Little Mountain 5, Abedia~enwiki, Wnzrf, Clsdennis2007, Tide rolls, Legobot, Yobot, Okami85, AnomieBOT, Materialscientist, Pitke, Neurolysis, Mark Schierbecker, George2001hi, Recognizance, D'ohBot, HamburgerRadio, Pinethicket, KuduIO, LibertyOrDeath, Orange Suede Sofa, Nhero2006, ClueBot NG, Mark Arsten, Leo72, FockeWulf FW 190, Chenthil Vel, Promestein, Napalm123 and Anonymous: 92
- **ANTI (computer virus)** *Source:* https://en.wikipedia.org/wiki/ANTI_(computer_virus)?oldid=609078054 *Contributors:* Pmj, Umais Bin Sajjad, Arjunkarappayil, BIcurious3334 and CyberXRef
- **INIT 1984** *Source:* https://en.wikipedia.org/wiki/INIT_1984?oldid=663329141 *Contributors:* Ketiltrout, Chaser, SmackBot, Chris the speller, Ktr101, DanielPharos, Addbot, Fluffernutter, CactusWriter, Miracle Pen, ZéroBot, BattyBot and Anonymous: 2
- **MacMag** *Source:* https://en.wikipedia.org/wiki/MacMag?oldid=710248445 *Contributors:* Drbreznjev, SmackBot, Mike Evangelist, Sable232, JamesBWatson, Jjblackshear, WOSlinker, Airplaneman, Yobot, HamburgerRadio, Miracle Pen, ClueBot NG, Helpful Pixie Bot, Reddogsix, Thisiswikidome, Tsungri and Anonymous: 13
- **MDEF** *Source:* https://en.wikipedia.org/wiki/MDEF?oldid=571003709 *Contributors:* Sable232, WOSlinker, Yobot, HamburgerRadio and Miracle Pen
- **NVIR** *Source:* https://en.wikipedia.org/wiki/NVIR?oldid=663016873 *Contributors:* Maury Markowitz, Cmdrjameson, A-Day, Wtmitchell, Dpv, Yobot, Miracle Pen, Marcin Łukasz Kiejzik and Anonymous: 8
- **Scores (computer virus)** *Source:* https://en.wikipedia.org/wiki/Scores_(computer_virus)?oldid=711901625 *Contributors:* WilliamKF, Microtherion, Dialectric, Sable232, WOSlinker, Yobot, AnomieBOT, LilHelpa, Supertramp78, HamburgerRadio, Rezonansowy, FockeWulf FW 190 and Anonymous: 2
- **SevenDust (computer virus)** *Source:* https://en.wikipedia.org/wiki/SevenDust_(computer_virus)?oldid=710626262 *Contributors:* Cmdrjameson, A-Day, Stephan Leeds, Dpv, Shippy Mandy, Takeel, Cruj, AppleMacReporter, Jerryobject, Mygerardromance, BashM, Emperordarius, Yobot, The Earwig, AnomieBOT, Miracle Pen, Look2See1, Archos331, Carliitaeliza, Hmainsbot1, VirtualRash and Anonymous: 15
- **KeyRaider** *Source:* https://en.wikipedia.org/wiki/KeyRaider?oldid=726505353 *Contributors:* Arthur Rubin, Jersey92, I dream of horses, AdagioRibbit and Anonymous: 1
- **Wirelurker** *Source:* https://en.wikipedia.org/wiki/Wirelurker?oldid=726506731 *Contributors:* BD2412, Mercurywoodrose, Rankersbo, GoingBatty, Sam Sailor, Koons5159 and AdagioRibbit
- **XcodeGhost** *Source:* https://en.wikipedia.org/wiki/XcodeGhost?oldid=727693605 *Contributors:* Fuzheado, Rich Farmbrough, Elvey, Dravecky, Totie, Yobot, Dcirovic, BG19bot, Old Naval Rooftops, FockeWulf FW 190, Arthur2968, Qzd, Riadhluke and Anonymous: 5
- **Brain Test** *Source:* https://en.wikipedia.org/wiki/Brain_Test?oldid=730033492 *Contributors:* Rich Farmbrough, Rathfelder, CuriousEric, Magioladitis, R'n'B, Tokyogirl79, Parallelized, AnomieBOT and FockeWulf FW 190
- **Dendroid (Malware)** *Source:* https://en.wikipedia.org/wiki/Dendroid_(Malware)?oldid=722353875 *Contributors:* Parallelized, BG19bot and FockeWulf FW 190
- **DroidKungFu** *Source:* https://en.wikipedia.org/wiki/DroidKungFu?oldid=728134804 *Contributors:* WikiGopi, My Chemistry romantic and OnionRing

- **Shedun** *Source:* https://en.wikipedia.org/wiki/Shedun?oldid=722342409 *Contributors:* Rich Farmbrough, Derek R Bullamore, PamD, Parallelized, Yobot, AnomieBOT and FockeWulf FW 190
- **Anti-keylogger** *Source:* https://en.wikipedia.org/wiki/Anti-keylogger?oldid=730108775 *Contributors:* Danhash, Intgr, Pseudomonas, Paul Magnussen, Nikkimaria, Tom Morris, Frap, Optakeover, Anshuk, Barek, JohnBlackburne, WurmWoode, Socrates2008, Yobot, SwisterTwister, AnomieBOT, Mark Schierbecker, 78.26, INeverCry, Mean as custard, RjwilmsiBot, DASHBot, JDDJS, Snotbot, Widr, Nbudden, BG19bot, Panurgentleman, IraChesterfield, Khazar2, Mover07, Dannyruthe, WikiGopi, MagyVi, Aladir, JeremiahY, Muhammad mobeen83, Mobeen1234 and Anonymous: 15
- **Antivirus software** *Source:* https://en.wikipedia.org/wiki/Antivirus_software?oldid=729236424 *Contributors:* Bryan Derksen, Zundark, Danny, Fubar Obfusco, William Avery, DennisDaniels, Edward, Pnm, Tannin, Tgeorgescu, Minesweeper, CesarB, Ronz, Yaronf, Rlandmann, Whkoh, Stefan-S, Nikai, IMSoP, RickK, Pedant17, Furrykef, Tempshill, Omegatron, Pakaran, Shantavira, Robbot, Chealer, Boffy b, Calimero, RedWolf, Altenmann, KellyCoinGuy, Iaen, Delpino, Lzur, David Gerard, Fabiform, Graeme Bartlett, Laudaka, Eran, Noone~enwiki, Rick Block, AlistairMcMillan, Solipsist, Wmahan, Utcursch, SoWhy, Beland, Piotrus, Cynical, Gscshoyru, TonyW, Hobart, Eisnel, Discospinster, Rich Farmbrough, Bender235, ESkog, JoeSmack, Evice, Aecis, Chungy, PhilHibbs, Sietse Snel, Femto, Perfecto, Stesmo, Longhair, Orbst, Richi, TheProject, Troels Nybo~enwiki, Timsheridan, Hagerman, Alansohn, CyberSkull, Conan, PatrickFisher, Babajobu, Stephen Turner, Snowolf, Wtmitchell, Downlode, Rotring, Nightstallion, Umapathy, Woohookitty, Mindmatrix, Armando, Robwingfield, Pol098, Urod, Isnow, Kralizec!, Pictureuploader, Palica, Matturn, Cuvtixo, Kbdank71, Yurik, Ryan Norton, Rjwilmsi, DirkvdM, RainR, FlaBot, JiFish, RexNL, Gurch, Davide-Andrea, ChongDae, Born2cycle, Melancholie, Ahunt, Peterl, Gwernol, YurikBot, Wavelength, Borgx, Grizzly37, Wfried, Arado, TheDoober, Pi Delport, SpuriousQ, Akhristov, Claunia, NawlinWiki, Hm2k, Badagnani, Arichnad, CecilWard, Vlad, Bota47, Bokonon~enwiki, Bazooka-Joe, GraemeL, Peter, Caballero1967, Fourohfour, Hirebrand, Jaysbro, Eptin, [illegible] robot, Dunxd, Cumbiagermen, Firewall-guy, SmackBot, Although, JurgenHadley, J7, Dxco, Relaxing, Easygoeasycome, Gilliam, JorgePeixoto, Lakshmin, Gary09202000, Chris the speller, Egladkih, Morte, EncMstr, Jerome Charles Potts, Bigs slb, DHN-bot~enwiki, Uniwares, Darth Panda, Frap, JonHarder, Korinkami, 03vaseyj, SundarBot, Cybercobra, Valenciano, Mwtoews, Ihatetoregister, Oo7jeep, Gobonobo, Capmo, NongBot~enwiki, 16@r, Erotml, Beetstra, Doczilla, Qu4rk, Cronos Warchild, Caiaffa, Hu12, DabMachine, SimonD, Phantomnecro, UncleDouggie, CapitalR, Kirill Chiryasov, Courcelles, Tawkerbot2, Fdssdf, FleetCommand, CmdrObot, BENNYSOFT, Jesse Viviano, NaBUru38, Chrisahn, Cydebot, Gogo Dodo, Xxhopingtearsxx, AcceleratorX, Tawkerbot4, Khattab01~enwiki, Ohadgliksman, The Mad Bomber, SpK, Neustradamus, Mikewax, TAG.Odessa, Dimo414, Thijs!bot, Jdivakarla, Leedeth, LemonMan, Saibo, Dalahäst, RickinBaltimore, TurboForce, Dawnseeker2000, Mentifisto, AntiVandalBot, Sjconradmchedrawe, Gökhan, Serpent's Choice, JAnDbot, Kaobear, Meinsla, MER-C, Tushard mwti, TAnthony, .anacondabot, Raanoo, Penubag, Bongwarrior, Lotusv82, Proland, The Kinslayer, JohnLai, Gomm, Xeolyte, Chris G, DerHexer, Hdt83, MartinBot, STBot, CliffC, FDD, Icenine378, CommonsDelinker, Emilinho~enwiki, J.delanoy, Pharaoh of the Wizards, Dinoguy1000, Public Menace, Jesant13, Turbulencepb, Neon white, Ripdog2121, Tokyogirl79, 5theye, Patrickjk, AntiSpamBot, Dougmarlowe, DadaNeem, Pandawelch, White 720, Jamesontai, Idioma-bot, Javeed Safai, Melovfemale, VolkovBot, AlnoktaBOT, Philip Trueman, DoorsAjar, TXiKiBoT, Oshwah, Emedlin1, Mujdat61, Vipinhari, Technopat, Anonymous Dissident, Qxz, Corvus cornix, LeaveSleaves, Natg 19, Tmalcomv, Haseo9999, C45207, Ngantengyuen, LittleBenW, Fredtheflyingfrog, Lonwolve, Wrldwzrd89, Sahilm, Derekcslater, Newspartnergroup, Swaq, Sephiroth storm, Yintan, Miremare, Mothmolevna, Jerryobject, Flyer22 Reborn, PolarBot, Nosferatus2007, Askild, Topicle, OKBot, Plati, Samker, PrimeYoshi, Escape Orbit, Arnos78, Martarius, Tanvir Ahmmed, Leahtwosaints, ClueBot, Kl4m, The Thing That Should Not Be, IceUnshattered, Trotline, Spuernase, Freebullets, Mild Bill Hiccup, Ka vijay, LizardJr8, ChandlerMapBot, Georgest23, Rockfang, DragonBot, Excirial, Socrates2008, Pavix, Tyler, Pladook, Jotterbot, JamieS93, ChrisHodgesUK, DanielPharos, Versus22, Johnuniq, SoxBot III, SF007, Sensiblekid, XLinkBot, Rror, Mavenkatesh, Svarya, HexaChord, Addbot, Xp54321, Wizho, Mortense, Nuno Brito, Softfreak, Sergey AMTL, Vatrena ptica, CanadianLinuxUser, Fluffernutter, Ankitguptajaipur, Kueensrÿche, NjardarBot, WorldlyWebster, MrOllie, CarsracBot, FluffyWhiteCat, Womanitoba, ChenzwBot, Jasper Deng, Mike A Quinn, Katharine908, Tide rolls, Luckas Blade, Teles, Luckas-bot, Yobot, THEN WHO WAS PHONE?, Wonderfl, AnomieBOT, Jim1138, DMWuCg, Roastingpan, Bluerasberry, Materialscientist, Police267, Kalamkaar, Eumolpo, Cameron Scott, Misi91, Avun, XZeroBot, Sputink, Rwmoekoe, S0aasdf2sf, Frosted14, SassoBot, ReformatMe, Mathonius, VB.NETLover, TheRyan95, Shadowjams, Diablosblizz, Samwb123, G7yunghi, FrescoBot, GunAlchemist, WPANI, Yuyujoke, Mi8ka, HJ Mitchell, Craig Pemberton, Franklin.online2006, Expertour, HamburgerRadio, Redrose64, SuperAntivirus, Marnegro, Pinethicket, HRoestBot, Skyerise, Paulsterne, A8UDI, Ma2001, Kostes32, One666, Seam123, AntonST, Σ, Meaghan, Salvidrim!, Ravensburg13, Cnwilliams, Trappist the monk, Lamarmote, Miiszmylove, LogAntiLog, Lotje, Wikipandaeng, Vrenator, TBloemink, 777sms, Neshemah, Diannaa, Hornlitz, Execter, Teenboi001, Mean as custard, RjwilmsiBot, Ripchip Bot, Panda Madrid, Enauspeaker, DASHBot, EmausBot, John of Reading, WikitanvirBot, Immunize, Philtweir, Heracles31, Dinhtuydzao, Ibbn, Ryanxo, Tommy2010, Cmlloyd1969, Dcirovic, Emenid, Elison2007, Fæ, Mats131, ElationAviation, Makecat, Skyinfo, Yabba67, Rickraptor707, Diflame, ChuispastonBot, GermanJoe, Pastore Italy, EdoBot, Kandr8, Petrb, ClueBot NG, Lzeltser, TheKaneDestroyer, Jack Greenmaven, Satellizer, LK20, Dfarrell07, Multiwikiswat, Piyush1992, JuventiniFan, Malijinx, Widr, Hsinghsarao, Joseph843, Helpful Pixie Bot, Dwe0008, HMSSolent, Krenair, Jeza87, Janendra, Arthurnyc, AvocatoBot, Thekillerpenguin, Teksquisite, Irfanshaharuddin, TheMw2Genius, Kremnin, Newmen1020304050, BattyBot, Justincheng12345-bot, David.moreno72, JC.Torpey, Divonnais, Farqad, IddiKlu, Nisha1987, Rohaneknathshinde459, Garamond Lethe, JYBot, Dark Silver Crow, Codename Lisa, Cryptodd, Pcguru66, K1ngXSp3c1al, Lugia2453, Kumarworld2, Sourov0000, Seo100, Me, Myself, and I are Here, M.R.V model, Gautamcool12, Faizan, I am One of Many, Ryan889, Matt.Sharp98, Jakec, Eddymck1, Ashajose0002, Assumelation, Ginsuloft, Quenhitran, Dannyruthe, Noyster, MetalFusion81, Robevans123, Monkbot, CodyHofstetter, TerryAlex, Xpasindu123, Thetechgirl, KH-1, Randomuser0122, ChamithN, WikiGopi, Jacbizer, DpkgDan, Puffle7275, Deanwalt123, Rom broke, Drop knowhow, Seanpatrickgray, The Professor123, Natasha Miranda, Chenthil Vel, Cyberstip, Chenthil Vel Murugan 1986, Srilekha selva, Sharanyanaveen, Dilipancb, XxXMinecraftProsnipzXxX, Satish0143, Lolo1299008, Plmnbvcxz12, Lytebyte and Anonymous: 682
- **Browser security** *Source:* https://en.wikipedia.org/wiki/Browser_security?oldid=724865133 *Contributors:* Thue, Pmsyyz, Bender235, Enric Naval, Savvo, Bobrayner, Xaosflux, Chris the speller, Frap, Mistress Selina Kyle, Dl2000, Dawnseeker2000, Widefox, Obiwankenobi, The Transhumanist, Public Menace, Ross Fraser, Vrac, LittleBenW, DavidBourguignon, Uncle Milty, Excirial, Socrates2008, Carl.antuar, TutterMouse, Yobot, Nyat, AnomieBOT, Omnipaedista, Haeinous, Yashartha Chaturvedi, Mabande, Helpful Pixie Bot, Chmarkine, BattyBot, Fred.Pendleton, Mdann52, GoShow, SurfPatrol, Corn cheese, Eyesnore, Tsarihan, Maura Driscoll, A Certain Lack of Grandeur, Alxfarr, Dannyruthe, Eslong1026, Seankclark, WikiGopi, Greenmow, Dina de Robles and Anonymous: 16
- **Internet security** *Source:* https://en.wikipedia.org/wiki/Internet_security?oldid=730538917 *Contributors:* Aarontay, Evgeni Sergeev, ZeWrestler, Tuomas, Frencheigh, AlistairMcMillan, Matt Crypto, SWAdair, Wmahan, CaribDigita, Elroch, TonyW, Discospinster, Rich Farmbrough, KneeLess, Xezbeth, JoeSmack, CanisRufus, *drew, MBisanz, Apollo2011, La goutte de pluie, Minghong, Pearle, Smoothy, Wtmitchell,

Jim Mikulak, Versageek, Bsdlogical, Johntex, Mindmatrix, Deltabeignet, Kevinkinnett, Willlangford, Vary, Gurch, Vec, Intgr, Chobot, DVdm, Bgwhite, Peterl, Elfguy, Wavelength, Argav, Alan216, Phantomsteve, Sarranduin, Akamad, Stephenb, Pseudomonas, Moe Epsilon, Amcfreely, HopeSeekr of xMule, S33k3r, KGasso, Th1rt3en, GraemeL, Thespian, DoriSmith, LSnK, Carlosguitar, Draicone, KnowledgeOfSelf, McGeddon, Verne Equinox, Jjalexand, Thumperward, N2f, Baa, Can't sleep, clown will eat me, Frap, Tcwolf, Nixeagle, JonHarder, Rrburke, Parent5446, Ianmacm, Mistress Selina Kyle, Ninnnu~enwiki, A. Parrot, Beetstra, Ehheh, A Clown in the Dark, Mere Mortal, FleetCommand, CmdrObot, Equendil, Cydebot, Gogo Dodo, Tiger williams, ErrantX, Wikid77, Edupedro, Mojo Hand, John254, Dawnseeker2000, Obiwankenobi, Jerrypcjr, SiobhanHansa, Raanoo, Io Katai, Gstroot, JohnLai, 28421u2232nfenfcenc, Gomm, Stephenchou0722, CliffC, Sjjupadhyay~enwiki, Lutz.hausmann, Trusilver, Jesant13, Dbiel, JudyJohn, WikiBone, Karrade, Sbanker, Wiki-ay, Philip Trueman, Zifert, Natg 19, Bryan.dollery, Dirkbb, Mr Nic, LittleBenW, Debog, Kbrose, SieBot, WereSpielChequers, Caltas, DaBler, Bananastalktome, Oxymoron83, Jruderman, Fornaeffe, Yuva raju raj, ClueBot, DavidGGG, The Thing That Should Not Be, EoGuy, Anapazapa, Excirial, Dcampbell30, Spock of Vulcan, Arjayay, La Pianista, Versus22, DumZiBoT, XLinkBot, Mitch Ames, Good Olfactory, Addbot, Xp54321, Mortense, Grayfell, Debsalmi, Kongr43gpen, CanadianLinuxUser, Chzz, Yobot, QueenCake, AnomieBOT, DemocraticLuntz, 1exec1, JDavis680, AdjustShift, Bestija11, Ulric1313, Materialscientist, Waltzmoore66, Citation bot, ArthurBot, Itbuddy, LilHelpa, TheAMmollusc, Mgf12rw, Nasnema, Craftyminion, BritishWatcher, S0aasdf2sf, Seldridge99, Fjollig arme, FrescoBot, Nageh, Chevymontecarlo, D'ohBot, A little insignificant, HamburgerRadio, Singaporesuperboy, DrilBot, Smeago, Jusses2, In3go, Piandcompany, Jujutacular, Cnwilliams, Intsafetycenter, Train2104, Lotje, Vrenator, TBloemink, DARTH SIDIOUS 2, VernoWhitney, EmausBot, WikitanvirBot, Timtempleton, Andyype, Zollerriia, Dcirovic, K6ka, MathMaven, 30, Riittaajo, Fæ, Zloyvolsheb, Enikuo, SporkBot, Rameez-NJITWILL, W163, Sbmeirow, Donner60, Autoerrant, Tyros1972, ChuispastonBot, Humannetwork, Makeet, 28bot, ClueBot NG, Snocman, Cntras, Widr, Karl 334, Helpful Pixie Bot, Electriccatfish2, Titodutta, Jeraphine Gryphon, BG19bot, Mleoking, Muneeb2000, Altaïr, Valentinocourt, Cin316, RroccoMaroc, BKoteska, MeanMotherJr, Motta.allo, Cyberbot II, Sakmaz, Dexbot, مون| بش‌يري, Corn cheese, Palmbeachguy, The Mol Man, Tentinator, Wuerzele, Matty.007, PCRepairGuy, Gaodong, Mickel1982, Chenthilvelmurugana, SiavooshPayandehAzad, Thisara1990, BrettofMoore, SandorSchwartz, Joewalter, Greenmow, XamieA, Mathewherz, AmberHenely, Chenthil Vel, Srilekha selva, Sharanyanaveen, DatGuy, Qzd, Philipheight, Thinusnaa, Robrobrob2, Neetiwariesh and Anonymous: 316

- **Mobile security** *Source:* https://en.wikipedia.org/wiki/Mobile_security?oldid=724978004 *Contributors:* Dreamyshade, Edward, Ronz, Phil Boswell, Bearcat, Bumm13, Giraffedata, Ceyockey, Feezo, Mindmatrix, Mattisha, Tabletop, Rjwilmsi, Ahunt, Bgwhite, Wavelength, Sarysa, Gilliam, Ohnoitsjamie, Deli nk, Kwestin, Shritwod, Drogers.uk, Dawnseeker2000, Visik, Obiwankenobi, Magioladitis, JamesBWatson, Cesarth~enwiki, Jesant13, Katharineamy, Bonadea, SylviaStanley, Flyer22 Reborn, Capitalismojo, Josang, Socrates2008, Stypex, Boleyn, Addbot, Yobot, WikiDan61, AnomieBOT, Rubinbot, Mark Schierbecker, 78.26, January2009, FrescoBot, Vertago1, Biker Biker, Jonesey95, OMGWEEGEE2, Cnwilliams, Chris Caven, RjwilmsiBot, Rsmaah, EmausBot, WikitanvirBot, Timtempleton, K6ka, Midas02, SporkBot, ClueBot NG, Pietrogpjmu, Widr, Helpful Pixie Bot, Electriccatfish2, BG19bot, Lifeformnoho, Yasht101, Cyberbot II, ChrisGualtieri, MikeDNJITWILL, Dexbot, Rickbcaol, 967Bytes, Lugia2453, Me, Myself, and I are Here, Securevoicecalling, Gtangil, Mbqt31, Tsarihan, Dannyruthe, Monkbot, RicardoHeinz, Dsprc, Nitin93flanker, Alesteska, Andreilow, Ujjwal Sahay, Anarchyte, Marieswiss12, Cmason1234, Lisahamilton90, Muhammad mobeen83, Chenthil Vel, Wikijagon, Bootsandmountains, GreenC bot, Peerlyst2016, SuperFetched777, AFlannery and Anonymous: 49

- **Network security** *Source:* https://en.wikipedia.org/wiki/Network_security?oldid=730566090 *Contributors:* Ed Brey, ZimZalaBim, Mike Rosoft, Discospinster, Wrp103, Bobo192, Nsaa, Arthena, Woohookitty, Mindmatrix, Aarghdvaark, BD2412, Kbdank71, Seraphimblade, Ttwaring, Nivix, Ewlyahoocom, Intgr, Chobot, DVdm, Bgwhite, FrankTobia, Epabhith, Irishguy, Nick C, Deku-shrub, Vanished user 8488293, Zzuuzz, JonnyJinx, BorgQueen, Victor falk, SmackBot, Gilliam, Ohnoitsjamie, Rmosler2100, Deli nk, Kungming2, Rrburke, Radicle, Nakon, Weregerbil, Philpraxis~enwiki, Will Beback, Kuru, DavidBailey, Robofish, Gorgalore, Aqw3R, Ehheh, TastyPoutine, Kvng, Iridescent, Shoeofdeath, Dthvt, Cbrown1023, Leujohn, E smith2000, MaxEnt, Sutekh.destroyer, Shandon, Ayzmo, Thijs!bot, Epbr123, TheFearow, JustAGal, Michael A. White, Dawnseeker2000, AntiVandalBot, Obiwankenobi, Seaphoto, Marokwitz, Prolog, Ellenaz, AndreasWittenstein, Res2216firestar, Andonic, PhilKnight, Tqbf, Raanoo, VoABot II, AuburnPilot, AlephGamma, Rohasnagpal, Elinruby, Stephenchou0722, STBot, CliffC, Jonathan Hall, R'n'B, Obscurans, Pharaoh of the Wizards, Jesant13, Ginsengbomb, Smyle.dips, ElectricValkyrie, DH85868993, Ken g6, Vanished user 39948282, CardinalDan, Wiki-ay, Philip Trueman, Oshwah, Mdeshon, M2petite, GidsR, Falcon8765, FlyingLeopard2014, SieBot, Liamoshan, RJaguar3, Quest for Truth, Flyer22 Reborn, Mscwriter, Dodger67, Hariva, ManOnPipes, ClueBot, Arcitsol, Donalcampbell, Drmies, Mild Bill Hiccup, Boing! said Zebedee, Krmarshall, Somno, Jswd, Eeekster, Leonard^Bloom, Radiosband, Rhododendrites, Tnspartan, Ranjithsutari, Johnuniq, SoxBot III, SF007, XLinkBot, Stickee, Nepenthes, BaroloLover, Mitch Ames, ErkinBatu, Nakamura36, Sindbad72, HexaChord, Sweeper tamonten, Addbot, Willking1979, Dmsynx, Ronhjones, TutterMouse, Jncraton, CanadianLinuxUser, T38291, Cst17, MrOllie, Download, Тивеполник, Tide rolls, Ghalleen, Bodies3819, Batman2472, CaliMan88, Rogger.Ferguson, Sgarchik, Pheebalicious, AnomieBOT, JDavis680, Dwayne, Piano non troppo, AdjustShift, Rwhalb, Ulric1313, Materialscientist, Aneah, Stationcall, ArthurBot, Xtremejames183, TheAMmollusc, Simsianity, Impakti, FreshBreak, Tedzdog, Pradameinhoff, SassoBot, SCΛRECROW, Seldridge99, Jcj04864, E0steven, Coffeerules9999, Citation bot 1, Farazv, FoxBot, Lotje, Vrenator, Dmuellenberg, Reach Out to the Truth, Mean as custard, RjwilmsiBot, Sundar.ciet, Ripchip Bot, Tremaster, Happyisenough, Timtempleton, Primefac, Sumsum2010, Take your time, Dcirovic, Yurisk, Ocaasi, Slimkaos, De.vos.katja, Donner60, Orange Suede Sofa, TYelliot, Sepersann, Mjbmrbot, Eliwins, ClueBot NG, Girraj study, JetBlast, Jaket911, TZM.Tronix, Netsecurityauthors, Sn23-NJITWILL, Denningr, Rahulbud, HMSSolent, Calabe1992, BG19bot, Hallows AG, Kyendell, Annapawiki, Mark Arsten, Piano1900, Nashrul Hakiem, Sheenaancy, Millennium bug, W.D., Mdann52, Mrt3366, Yy.sujin, YFdyhbot, Prelude after noon, Jags707, SimonWiseman, Codename Lisa, Patrick.bausemer, Pete Mahen, Junkyardsparkle, Me, Myself, and I are Here, Vanischenu from public computers, Phamnhatkhanh, Csteinb, Jakec, Ahoora62, FockeWulf FW 190, Softwareguy2013, Hard ToOp, Monkbot, Tabowen, Owais Khursheed, Oushee, TerryAlex, Sin.akshat, Robbiegray-mv, Greenmow, TheoMessin, Tmar877, Eroticgiraffeselfies, Yoloswag1224, Mrs.Guidoisa (fill in the blank), Mrs. Guido is a (fill in the bank), RoadWarrior445, Srilekha selva, Sharanyanaveen, Christy Max, LanceRishad and Anonymous: 313

- **Defensive computing** *Source:* https://en.wikipedia.org/wiki/Defensive_computing?oldid=693911879 *Contributors:* Pnm, Bender235, Woohookitty, BD2412, RussBot, Bachrach44, Matticus78, Cs-wolves, Frap, MarshBot, RainbowCrane, Dfense, Verkhovensky, CliffC, R'n'B, EdOnRoll, ClueBot, DanielPharos, MrOllie, Erik9, Superking300, FrescoBot, Turko88, Michaeldobson13, Geohac, Gorthian and Anonymous: 14

- **Firewall (computing)** *Source:* https://en.wikipedia.org/wiki/Firewall_(computing)?oldid=730638842 *Contributors:* Paul~enwiki, Nealmcb, Michael Hardy, Pnm, Egil, Ahoerstemeier, Copsewood, Haakon, Jebba, Sugarfish, Rl, Dcoetzee, Jay, DJ Clayworth, Taxman, Bevo, Topbanana,

Joy, Khym Chanur, Robbot, ZimZalaBim, Danutz, Auric, Jondel, Hadal, Diberri, Tea2min, Pabouk, Giftlite, Yama, Everyking, Rchandra, AlistairMcMillan, Eequor, Matthäus Wander, Wiki Wikardo, DemonThing, Wmahan, Stevietheman, ConradPino, Antandrus, Ricky~enwiki, Mitaphane, Biot, Deewiant, Joyous!, Hax0rw4ng, Asqueella, Mernen, Grand Edgemaster, Monkeyman, Discospinster, Fabioj, Wk muriithi, EliasAlucard, Smyth, YUL89YYZ, Deelkar, DonDiego, Pmetzger, El C, Mwanner, Dols, Spearhead, Linkoman, RoyBoy, Femto, Jpgordon, Bobo192, Smalljim, Enric Naval, Viriditas, Giraffedata, Danski14, Alansohn, Anthony Appleyard, Interiot, Malo, Wtmitchell, Velella, L33th4x0rguy, Rick Sidwell, IMeowbot, Henry W. Schmitt, TheCoffee, DSatz, Kenyon, Brookie, Zntrip, Andem, Nuno Tavares, Angr, OwenX, Woohookitty, Karnesky, Mindmatrix, Dzordzm, Bazsi~enwiki, Kralizec!, Prashanthns, DESiegel, Turnstep, Ashmoo, Graham87, Chun-hian, Kbdank71, FreplySpang, Jclemens, Rjwilmsi, OneWeirdDude, Eptalon, NeonMerlin, ElKevbo, Sferrier, Dmccreary, Gurch, DevastatorIIC, Intgr, Alphachimp, OpenToppedBus, Ahunt, Marcuswittig, DVdm, FeldBum, Bgwhite, Theymos, YurikBot, Wavelength, Borgx, TexasAndroid, Quentin X, Sceptre, Alan216, MMuzammils, RussBot, Mattgibson, Lincolnite, Pi Delport, Stephenb, Manop, Rsrikanth05, Wimt, Capi, NawlinWiki, ENeville, Trevor1, Rebel, Mortein, Cryptosmith, Jpbowen, Voidxor, Bkil, Zwobot, Bucketsofg, Black Falcon, Mcicogni, CraigB, Nlu, Wknight94, Rwxrwxrwx, Dse, JonnyJinx, Closedmouth, E Wing, Pb30, ILRainyday, Chriswaterguy, Talyian, Cffrost, Anclation~enwiki, Maxamegalon2000, Bswilson, A13ean, SmackBot, Unschool, Rbmcnutt, KnowledgeOfSelf, C.Fred, Od Mishehu, Eskimbot, Vilerage, Info lover, Xaosflux, Gilliam, Ohnoitsjamie, Lakshmin, Bluebot, DStoykov, Jprg1966, Thumperward, Mcj220, Snori, Oli Filth, Prasan21, Lubos, Elagatis, DavidChipman, DHN-bot~enwiki, Da Vynci, Anabus, Suicidalhamster, Abaddon314159, Can't sleep, clown will eat me, Frap, Chlewbot, JonHarder, Yorick8080, Fynali, Celarnor, Meandtheshell, Ntolkin, Aldaron, Nachico, Elcasc, HarisM, Skrewz~enwiki, Phoenix314, LeoNomis, FerzenR, Andrei Stroe, Ugur Basak Bot~enwiki, The undertow, Harryboyles, Eldraco, Mattloaf1, Melody Concerto, Beetstra, Boomshadow, Feureau, Peyre, Hu12, Hetar, BranStark, BananaFiend, Jhi247, Robbie Cook, Newone, GDallimore, Pmattos~enwiki, Tawkerbot2, Chetvorno, SkyWalker, JForget, FleetCommand, Ale jrb, Megaboz, JohnCD, Topspinslams, Kgentryjr, Random name, Lazulilasher, WeggeBot, Josemi, Nnp, Equendil, Phatom87, Cydebot, T Houdijk, Mashby, UncleBubba, Gogo Dodo, Tbird1965, Hamzanaqvi, Guitardemon666, Πrate, Omicronpersei8, Thijs!bot, Danhm, Epbr123, Barticus88, Kubanczyk, Dschrader, Pajz, Randilyn, Simeon H, Marek69, SGGH, Chrisdab, CharlotteWebb, Wai Wai, Dawnseeker2000, AntiVandalBot, RoMo37, Davidoff, Purpleslog, Isilanes, Vendettax, LegitimateAndEvenCompelling, Dougher, ShyShocker, DoogieConverted, Dman727, Deadbeef, Acrosser, JAnDbot, Sheridp, MER-C, Seddon, Lucy1981, Tushard mwti, Kjwu, Jahoe, Raanoo, VoABot II, AtticusX, Maheshkumaryadav, Swpb, Djdancy, Hps@hps, Cellspark, Twsx, Dean14, AlephGamma, Gstroot, LeinaD natipaC, Hans Persson, Nposs, Greg Grahame, Just James, DerHexer, Rtouret, Hbent, Jalara, XandroZ, Seba5618, Tommysander, MartinBot, CliffC, LeonTang, R'n'B, Ash, PrestonH, Tgeairn, J.delanoy, NightFalcon90909, Shawniverson, Ans-mo, Jigesh, L'Aquatique, !Darkfire!6'28'14, Molly-in-md, KCinDC, STBotD, Equazcion, Red Thrush, Beezhive, Halmstad, SoCalSuperEagle, Idioma-bot, Zeroshell, Jramsey, Timotab, VolkovBot, Mike.batters, Jeff G., Indubitably, AlnoktaBOT, VasilievVV, Venom8599, Philip Trueman, Apy886, Jackrockstar, Cedric dlb, Ulrichlang, OlavN, Anna Lincoln, Corvus cornix, David.bar, Sanfranman59, Justin20, LeaveSleaves, Seb az86556, Lolsalad, Yk Yk Yk, Phirenzic, Why Not A Duck, Brianga, MrChupon, JasonTWL, EmxBot, Hoods11, SieBot, EQ5afN2M, Jchandlerhall, YonaBot, Sephiroth storm, Yintan, Miremare, Calabraxthis, Milan Kerslager, Android Mouse, Hokiehead, JSpung, Hazawazawaza, Goodyhusband, Doctorfluffy, Oxymoron83, Nuttycoconut, Tombomp, C'est moi, Mygerardromance, Altzinn, WikiLaurent, Bryon575, Ilpostinouno, Berford, Escape Orbit, Loren.wilton, ClueBot, Rumping, Snigbrook, CorenSearchBot, The Thing That Should Not Be, Jan1nad, SecPHD, Arakunem, Jobeard, Njmanson, Niceguyedc, Blanchardb, Harland1, ChandlerMapBot, Bencejoful, Jusdafax, Tim874536, Dcampbell30, Estirabot, Shiro jdn, Aurora2698, Peter.C, Mxbuck, Creed1928, ChrisHodgesUK, BOTarate, La Pianista, 9Nak, Aitias, Certes, Apparition11, Vanished user uih38riiw4hjlsd, Sensiblekid, DumZiBoT, BarretB, Wordwizz, Gnowor, Booster4324, Gonzonoir, Rror, NellieBly, SP1R1TM4N, Badgernet, Alexius08, Noctibus, WikiDao, Thatguyflint, Osarius, Wyatt915, Addbot, Wikialoft, RPHv, Some jerk on the Internet, Captain-tucker, Otisjimmy1, Crazysane, TutterMouse, Lets Enjoy Life, Vishnava, CanadianLinuxUser, Leszek Jańczuk, Sysy909, Cst17, MrOllie, Roseurey, Emailtonaved, Chzz, Debresser, Muheer, LinkFA-Bot, Tide rolls, Lightbot, OlEnglish, Krano, Iune, Bluebusy, WikiDreamer Bot, Shawnj99, Luckas-bot, Yobot, Terronis, Fraggle81, Amirobot, Fightingirishfan, AnomieBOT, JDavis680, Jlavepoze, Tcosta, Killiondude, Jim1138, Gascreed, Piano non troppo, Elieb001, Gc9580, Fahadsadah, Kyleflaherty, Flewis, Materialscientist, Citation bot, Aneah, Neurolysis, Obersachsebot, Xqbot, TheAMmollusc, Duesseljan, Addihockey10, JimVC3, Capricorn42, CoolingGibbon, 4twenty42o, Jmprtice, Ched, GrouchoBot, Backpackadam, Prunesqualer, RibotBOT, SassoBot, EddieNiedzwiecki, Thearcher4, Doulos Christos, =Josh.Harris, Gnuish, Chaheel Riens, Jaraics, Dan6hell66, G7yunghi, Prari, FrescoBot, Nageh, WPANI, Kamathvasudev, Galorr, Smile4ever, Expertour, Lukevenegas, DivineAlpha, Grapht, Pinethicket, I dream of horses, HRoestBot, Serols, Meaghan, Richard, MrBenCai, December21st2012Freak, Cougar w, Weylinp, Danshelb, TobeBot, WilliamSun, FunkyBike1, Vrenator, Clarkcj12, Stephenman882, Bangowiki, Mwalsh34, Eponymosity, Tbhotch, Gaiterin, DARTH SIDIOUS 2, Hugger and kisser, Dbrooksgta, Teenboi001, Aviv007, Regancy42, VernoWhitney, DASHBot, Chuck369, EmausBot, WikitanvirBot, Timtempleton, Super48paul, Solarra, Winner 42, Dcirovic, K6ka, Aejr120, Shuipzv3, Athn, Ebrambot, Kandarp.pande.kandy, Sg313d, Cit helper, IntelligentComputer, Rafiwiki, Flyinghatchet, OisinisiO, NTox, Cubbyhouse, Zabanio, DASHBotAV, Sepersann, 28bot, Socialservice, ClueBot NG, AAriel42, Lord Roem, Vakanuvis789, 123Hedgehog456, Vlhsrp, Widr, Johnny C. Morse, Debby5.0, HMSSolent, Titodutta, Kanwar47, Wbm1058, Wiki13, Silvrous, Dentalplanlisa, Euphoria42, Zune0112, Paulwray97, Nperrakis, Klilidiplomus, Sk8erPrince, Cimorcus, Fastcatz, CGuerrero-NJITWILL, Cvarta, PhilipFoulkes, Dexbot, Sendar, Jmitola, SimonWiseman, Codename Lisa, Avinash7075, Pete Mahen, CaSJer, Junkyardsparkle, Jamesx12345, Rob.bosch, VikiED, Palmbeachguy, Epicgenius, Camayoc, Melonkelon, Anupasinha.20, Praemonitus, SamoaBot, EvergreenFir, DavidLeighEllis, Indiesingh, Ginsuloft, ScotXW, Harshad1310, Nyashinski, Monkbot, The Original Filfi, Dsprc, Darshansham, TJH2018, WikiGopi, Jeremy.8910, ScrapIronIV, Kenkutengu, AMLIMSON, Miraclexix, Amccann421, Jerodlycett, KasparBot, RippleSax, ProprioMe OW, Risc64, Natasha Miranda, Radhwann, 0600K078, Chenthil Vel, Sharanyanaveen, Entranced98, Dakshin t and Anonymous: 997

- **Intrusion detection system** *Source:* https://en.wikipedia.org/wiki/Intrusion_detection_system?oldid=730108418 *Contributors:* Mav, Harperf, Michael Hardy, Willsmith, (, Ellywa, Ronz, Julesd, Smaffy, Schneelocke, Dwo, Hashar, Joy, Astronautics~enwiki, Tbutzon, Wereon, Tea2min, Lady Tenar, Gtrmp, Mintleaf~enwiki, Ngardiner, Rick Block, Cloud200, Falcon Kirtaran, Chowbok, Utcursch, Kusunose, Hellisp, Bluefoxicy, Olivier Debre, Aaryna, Sysy, Discospinster, Bender235, Sietse Snel, GattoRandagio, Tmh, Obradovic Goran, Espoo, Guy Harris, Denoir, Tje, Rick Sidwell, M3tainfo, H2g2bob, Btornado, Nuno Tavares, Krille, Isnow, Jclemens, Rjwilmsi, Aapo Laitinen, Margosbot~enwiki, Intgr, Windharp, Chobot, Bgwhite, Joonga, Borgx, Michael@thelander.com, Wolfmankurd, Boonebytes, Msoos, Falcon9x5, Abune, Josh3580, LeonardoRob0t, NeilN, Simmondp, SmackBot, Mmernex, JurgenHadley, Charlesrh, Unforgettableid, Jprg1966, Mhecht, Frap, JonHarder, Midnightcomm, UU, Grover cleveland, Radagast83, Weregerbil, Kolmigabrouil, Drphilharmonic, DMacks, Kuru, Itsgeneb, Tomhubbard, MarkSutton, Slakr, Yms, Visnup, EdC~enwiki, Kvng, FleetCommand, Sir Vicious, Difference engine, Cydebot, Mikebrand, Studerby, Cs california, Malleus Fatuorum, Jojan, Dawnseeker2000, AndreasWittenstein, JAnDbot, PhilKnight, Caffeinepuppy, BigChicken, Ronbarak, Pete Wall, CliffC, Tamer ih~enwiki, CraigMonroe, Haiauphixu, NewEnglandYankee, Sgorton, Derekrogerson, Wilson.canadian, Wlgrin, VolkovBot, Tburket,

Butseriouslyfolks, TXiKiBoT, Qxz, Ferengi, Clangin, Nitin.skd, Softtest123, Madhero88, Finity, Hotmixclass, Dremeda, Phisches, Shahmirj, Jerryobject, SouthLake, Jasonhatwiki, ClueBot, Lokipro, PolarYukon, Fossguy, Anubis1055, Gunnar Kreitz, CrazyChemGuy, Dcampbell30, Footballfan190, Pete71, SoxBot III, Jovianeye, Tackat, Addbot, Some jerk on the Internet, Ronhjones, J7387438, MrOllie, Chrismcnab, Oakleeman, Bond0088, Yobot, Omarmalali, AnomieBOT, Jim1138, Hiihammuk, Materialscientist, Aneah, Xqbot, Capricorn42, Kpcatch6, Jesse5656, Kernel.package, Locobot, Shadowjams, Aghsajjy, Weyesr1, Access-bb, RedBot, Banej, Lotje, Offnfopt, Autumnalmonk, Ddasune, EmausBot, WikitanvirBot, Mjdtjm, Dewritech, RA0808, Qrsdogg, Thecheesykid, ZéroBot, Lamf 0009, Paklan, Boundary11, RISCO Group, Bomazi, Nhero2006, Cgt, ClueBot NG, Andrew.philip.thomas, Alima86, Bernolákovčina, Rezabot, Widr, Helpful Pixie Bot, Karthikjain, Deltaray3, Titodutta, Wbm1058, BG19bot, NewsAndEventsGuy, PhnomPencil, Kagundu, RobVeggett, Archos331, Baszerr, Cristianocosta, Mogism, Cerabot~enwiki, Bgibbs2, Namnatulco, Spicyitalianmeatball, Frosty, SFK2, Me, Myself, and I are Here, Jose Manuel Caballero, KBhokray, Rajamca66, Acc12345acc, Ajpolino, Paul2520, Rsschomburg, Freddyap, Drkhataniar, Monkbot, Sofia Koutsouveli, S166865h, Malayks, GeoffreyT2000, 115ash, Crystallizedcarbon, Pblowry, Jchango301, CAPTAIN RAJU, Ahmed embedded, Pickyt, Pranuvinu1 and Anonymous: 308

- **Data loss prevention software** *Source:* https://en.wikipedia.org/wiki/Data_loss_prevention_software?oldid=728794734 *Contributors:* Nealmcb, Yaronf, Disdero, Psychonaut, Jfire, Cloud200, Utcursch, Ary29, Enric Naval, Hooperbloob, Wtmitchell, Stuartyeates, Rjwilmsi, Zubi, Intgr, RussBot, Eric Sellars, JLaTondre, SmackBot, C.Fred, Xaosflux, Gilliam, Frap, Huon, Bj96, Dekket, Jhales71, Nuwewsco, DPdH, AndreasWittenstein, Tqbf, Magioladitis, JNW, Nikfar, CliffC, Nono64, Ryan Postlethwaite, Qu3a, Cralar, Kyle the bot, OrenT, Rbrisco, Entbark, EdOnRoll, Vfeditor, The Thing That Should Not Be, Lancehooper, Mild Bill Hiccup, Blanchardb, Auntof6, M4gnum0n, Vericept, WmunroeIII, Tnspartan, Erikgellatly, Turajski~enwiki, DanielPharos, Tmsmith67, Dthomsen8, MystBot, Addbot, Noafdcap, Knguyeniii, Yobot, AnomieBOT, Ansanelli, Ghaney0328, Gabriel1907, Jgeorge60, Omnipaedista, Dataresolve, Aelley, Lescarolnyc, FrescoBot, Arny81, Feldermouse, Jandalhandler, Everrob, Aoidh, RjwilmsiBot, EmausBot, WikitanvirBot, Jblauch, Nealw1590, Liorrokach, Nixelpixel, Dineshkumar Ponnusamy, Nhero2006, Subn4u, Toochka, ClueBot NG, Sheltazar, Bstranger, Zamsonj, Snotbot, Gail2011, Heomap1983, Elleboogie12, Wikisian, Lucas.samaras, DakineMan, Dreamteamone, Ncn720, Ratongus, Ytic nam, Wikisamg, Bpani, Carlicoleman, Tonydees, RockInSpain, Mikefromnyc, Secrius, Joecaruso123, Felygus, Monkbot, DowneyIT, Philipdesouza`, KH-1, ChamithN, Nfernandes1, DemetriusGiannopoulos, Oliverlevin, Mahuta33, Psharma9 and Anonymous: 145
- **Computer and network surveillance** *Source:* https://en.wikipedia.org/wiki/Computer_and_network_surveillance?oldid=717554670 *Contributors:* Ubiquity, Modster, Pnm, Kku, Notheruser, Kingturtle, Smack, Novum, Ww, Fredrik, RedWolf, Gracefool, Wmahan, Andycjp, Beland, TonyW, Corti, Rich Farmbrough, ZeroOne, CanisRufus, Stesmo, ZayZayEM, Valar, Alansohn, Eleland, ReyBrujo, Versageek, Bobrayner, Mindmatrix, Rjwilmsi, Haya shiloh, Jrtayloriv, Intgr, Bgwhite, RussBot, Gardar Rurak, Bachrach44, Joel7687, Emijrp, Zzuuzz, Arthur Rubin, Rurik, Rwwww, SmackBot, Bluebot, GoldDragon, Adpete, Can't sleep, clown will eat me, Frap, Gamgee, Mion, Clicketyclack, Beetstra, Sander Säde, CmdrObot, JohnCD, Dreaded Walrus, Ingolfson, Techie guru, Magioladitis, Bongwarrior, Fedia, Elinruby, Atulsnischal, AVRS, CliffC, CommonsDelinker, Maurice Carbonaro, Hodja Nasreddin, Crakkpot, Demizh, Jeepday, AntiSpamBot, NewEnglandYankee, Andy Marchbanks, TreasuryTag, ServeNow, Pwnage8, PeetMoss, Qworty, MunkyJuce69, Svick, ClueBot, The Thing That Should Not Be, IoptaBan, WalterGR, Mlaffs, DumZiBoT, XLinkBot, Jasonma84, Melab-1, Ente75, MrOllie, Jarble, Legobot, Yobot, Vividupper66, AnomieBOT, LilHelpa, PansikMaZer, Ellipi, Alialiac, FrescoBot, 2dsea, M2545, PigFlu Oink, Lotje, Mean as custard, RjwilmsiBot, Rollins83, John of Reading, Érico, H3llBot, W163, Ego White Tray, ClueBot NG, North Atlanticist Usonian, BG19bot, Sprinting faster, MusikAnimal, Meclee, BattyBot, Chris-Gualtieri, Artem12345, TimMouraveiko, Jemappelleungarcon, I am One of Many, Ms. Anthropic, Majidmec, Someone not using his real name, Dannyruthe, Coreyemotela, Fixuture, Itsalleasy, 7Sidz, Saectar, The Infobox Strikes Again!, Monkbot, Ennykonto, Digitalzoo, Wallace McDonald, KH-1, Qin Xue, Starfire2999, Ghost Lourde, Johngreenaway, Hampton11235, Brendapallister, MagyVi, Nextstepsailing, CaseyMillerWiki, Cjllacuna, CAPTAIN RAJU, Nathan Bachman, Dueletva and Anonymous: 72
- **Operation: Bot Roast** *Source:* https://en.wikipedia.org/wiki/Operation%3A_Bot_Roast?oldid=675359256 *Contributors:* Pnm, B.d.mills, Neutrality, Klemen Kocjancic, Dachannien, Ricky81682, Yuckfoo, Pauli133, Rjwilmsi, Arthur Rubin, NeilN, SmackBot, C.Fred, KD5TVI, Snori, Reaper X, Onorem, Risker, Eastlaw, Esemono, Gioto, Public Menace, RagnaParadise, SpigotMap, AllGloryToTheHypnotoad, Ephix, Silent52, Blueking12, FlamingSilmaril, ClueBot, Professional Internet User, Lawrence Cohen, Ottre, DanielPharos, AndreNatas, JBsupreme, Bunnyhop11, LifeIsPain, Erik9, HamburgerRadio, Dashren2001, Helpful Pixie Bot, Seafax, BlazeT3ck, DDosFreedomofspeed and Anonymous: 18
- **Honeypot (computing)** *Source:* https://en.wikipedia.org/wiki/Honeypot_(computing)?oldid=727324962 *Contributors:* Bryan Derksen, David Merrill, Fubar Obfusco, Ghakko, Stevertigo, Frecklefoot, Patrick, Dcljr, Paul A, Tregoweth, Ahoerstemeier, DavidWBrooks, RickK, Glimz~enwiki, Steinsky, HappyDog, Furrykef, Saltine, LMB, VeryVerily, Calieber, Korath, Donreed, Henrygb, KellyCoinGuy, Tom harrison, Nachi~enwiki, Kate, Mike Rosoft, Kenj0418, Pmsyyz, Vsmith, Ponder, Perfecto, Duk, Mmckenzie, Wrs1864, Jakelee, Espoo, Anthony Appleyard, Arthena, Stillnotelf, Richard Arthur Norton (1958-), Mindmatrix, Apokrif, Cbdorsett, SCEhardt, Bubeck, Weevil, Marudubshinki, BD2412, RxS, Sjö, NeonMerlin, FlaBot, Erikina, Ewlyahoocom, Wongm, DVdm, Hall Monitor, BlueJaeger, Eraserhead1, RussBot, Tyler.szabo, Barefootguru, Kimchi.sg, Bovineone, Lao Wai, Ok80324, Robchurch, Eurosong, Graciella, Arthur Rubin, RenamedUser jaskldjslak904, Potterra, SmackBot, Mmernex, InverseHypercube, Blue520, KVDP, TheDoctor10, Gary09202000, EncMstr, Dvrasp~enwiki, Reeveorama, Uniwares, JonHarder, Seduisant, Radagast83, B jonas, Minasbeede, SnappingTurtle, Pwjb, James Mohr, Juux, Entereczek, TenPoundHammer, Kuru, Todd661, Cielomobile, Lee Carre, Hu12, CP\M, Wafulz, Nielsprovos, A876, Riker2000, Aristeo, Chrislk02, TIMe110, Thijs!bot, Rusl, LateToTheGame, Dawnseeker2000, AntiVandalBot, Gmarsden, Dreaded Walrus, Ingolfson, Wasell, Pierre Monteux, Magioladitis, Bongwarrior, LunaticBeatnik, Nyttend, Loqi, Indon, Anuzis, Cseifert, Gwern, Logzorg, MartinBot, Boyton, Kateshortforbob, Public Menace, Victuallers, King ging, TXiKiBoT, Lechatjaune, DJFrankie2468, Melsaran, BotKung, Natg 19, Vbvbvb~enwiki, Dirkbb, Insanity Incarnate, LittleBenW, SieBot, Mstarrr, AlanUS, Brylie, Maralia, Dust Filter, Monkeyburg, Jon1412, ClueBot, Kai-Hendrik, Syhon, Michaelsherin, Trivialist, Sv1xv, Socrates2008, Igorberger, Lartoven, Sarsaparilla, Chrisarnesen, Chuckgo, DumZiBoT, Jimyoo~enwiki, TreyGeek, Addbot, Luckas-bot, Yobot, Amirobot, M9.justin, AnomieBOT, VanishedUser sdu9aya9fasdsopa, Jim1138, Aaagmnr, PizzaofDoom, Kernel.package, January2009, Zebforge, Shadowjams, Thorenn, Aldy, I dream of horses, RedBot, Chris Caven, Amphicoelias, Vrenator, Styxnsoon, Sirkablaam, RjwilmsiBot, EmausBot, ScottyBerg, Dewritech, Steko, Listmeister, Mir5, Wikfr, Cameron11598, Ehendrix6, Petrb, Greatzsp, ClueBot NG, Helpful Pixie Bot, HMSSolent, BG19bot, BhustonOH, Wfseg, Rohit7401, Soulparadox, Faizan, Ben-Yeudith, Machdelu, Meteor sandwich yum, Clepsisoft, 24kanika, ChamithN, Julietdeltalima, Billyriobr, BU Rob13, JohnStew826, Dzampino, Sharanyanaveen and Anonymous: 290
- **Anti-Spyware Coalition** *Source:* https://en.wikipedia.org/wiki/Anti-Spyware_Coalition?oldid=705075715 *Contributors:* Topbanana, Northgrove, Bobblewik, EagleOne, RJFJR, Pol098, OCNative, Rjwilmsi, SchuminWeb, Ugur Basak, Dialectric, Seegoon, BirgitteSB, Carabinieri,

SmackBot, N2e, Bercyon, Enigmafyv, Socrates2008, Paul.j.richardson, Pcap, AnomieBOT, Paranspringnote, Werieth, L0ngpar1sh, Helpful Pixie Bot, Cyberbot II and Anonymous: 3

8.2 Images

- **File:1988-03_Universal_Peace_virus_on_my_Mac.jpg** *Source:* https://upload.wikimedia.org/wikipedia/commons/9/98/1988-03_Universal_Peace_virus_on_my_Mac.jpg *License:* CC BY-SA 3.0 *Contributors:* Own work *Original artist:* Mike Evangelist
- **File:2010-05-14-USCYBERCOM_Logo.jpg** *Source:* https://upload.wikimedia.org/wikipedia/commons/3/3a/2010-05-14-USCYBERCOM_Logo.jpg *License:* Public domain *Contributors:* Department of Defense *Original artist:* http://www.defense.gov/home/features/2010/0410_cybersec/images/cybercom_seal_large1.jpg Department of Defense
- **File:Ambox_current_red.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/9/98/Ambox_current_red.svg *License:* CC0 *Contributors:* self-made, inspired by Gnome globe current event.svg, using Information icon3.svg and Earth clip art.svg *Original artist:* Vipersnake151, penubag, Tkgd2007 (clock)
- **File:Ambox_important.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/b/b4/Ambox_important.svg *License:* Public domain *Contributors:* Own work, based off of Image:Ambox scales.svg *Original artist:* Dsmurat (talk · contribs)
- **File:Baidu_Pan.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/c/cf/Baidu_Pan.png *License:* CC BY-SA 4.0 *Contributors:* Own work *Original artist:* Arthur2968
- **File:Beast_RAT_client.jpg** *Source:* https://upload.wikimedia.org/wikipedia/commons/9/9a/Beast_RAT_client.jpg *License:* Public domain *Contributors:* Own work *Original artist:* V.H.
- **File:Boundless_Informant_data_collection.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/5/5b/Boundless_Informant_data_collection.svg *License:* CC0 *Contributors:* BlankMap-World6.svg
Original artist: Rezonansowy
- **File:CIA.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/2/25/Seal_of_the_Central_Intelligence_Agency.svg *License:* Public domain *Contributors:* http://www.law.cornell.edu/uscode/50/403m.html *Original artist:* United States federal government
- **File:CPU_ring_scheme.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/2/25/CPU_ring_scheme.svg *License:* CC-BY-SA-3.0 *Contributors:* This vector image was created with Inkscape. *Original artist:* User:Sven, original Author User:Cljk
- **File:Circle_of_spam.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/0/08/Circle_of_spam.svg *License:* CC-BY-SA-3.0 *Contributors:* own work, based on png version from English-language Wikipedia (by Fubar Obfusco & Admrboltz) *Original artist:* odder
- **File:ClamAV0.95.2.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/2/2f/ClamAV0.95.2.png *License:* GPL *Contributors:* my PC running Ubuntu 9.04 *Original artist:* SourceFire
- **File:ClamTK3.08.jpg** *Source:* https://upload.wikimedia.org/wikipedia/commons/2/26/ClamTK3.08.jpg *License:* GPL *Contributors:* Own work (own screenshot) *Original artist:* Dave Mauroni
- **File:ClamWin_Wine_screenshot.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/0/08/ClamWin_Wine_screenshot.png *License:* GPL *Contributors:* Transferred from en.wikipedia to Commons by Mardus. *Original artist:* The original uploader was SF007 at English Wikipedia
- **File:Clicking_Upload_button_on_abovetopsecret_triggers_clickjacking_warning_on_NoScript_141128-cropped.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/7/7c/Clicking_Upload_button_on_abovetopsecret_triggers_clickjacking_warning_on_NoScript_141128-cropped.png *License:* CC0 *Contributors:* self *Original artist:* <https://noscript.net/ + self>
- **File:Commons-logo.svg** *Source:* https://upload.wikimedia.org/wikipedia/en/4/4a/Commons-logo.svg *License:* CC-BY-SA-3.0 *Contributors:* ? *Original artist:* ?
- **File:Computer-aj_aj_ashton_01.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/d/d7/Desktop_computer_clipart_-_Yellow_theme.svg *License:* CC0 *Contributors:* https://openclipart.org/detail/105871/computeraj-aj-ashton-01 *Original artist:* AJ from openclipart.org
- **File:Conficker.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/5/53/Conficker.svg *License:* CC BY-SA 3.0 *Contributors:* Own work *Original artist:* Gppande
- **File:Crystal_Clear_app_browser.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/f/fe/Crystal_Clear_app_browser.png *License:* LGPL *Contributors:* All Crystal icons were posted by the author as LGPL on kde-look *Original artist:* Everaldo Coelho and YellowIcon
- **File:Crystal_Clear_device_cdrom_unmount.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/1/10/Crystal_Clear_device_cdrom_unmount.png *License:* LGPL *Contributors:* All Crystal Clear icons were posted by the author as LGPL on kde-look; *Original artist:* Everaldo Coelho and YellowIcon;
- **File:Crystal_kchart.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/2/28/Crystal_kchart.png *License:* LGPL *Contributors:* All Crystal icons were posted by the author as LGPL on kde-look *Original artist:* Everaldo Coelho and YellowIcon
- **File:Desktop_computer_clipart_-_Yellow_theme.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/d/d7/Desktop_computer_clipart_-_Yellow_theme.svg *License:* CC0 *Contributors:* https://openclipart.org/detail/105871/computeraj-aj-ashton-01 *Original artist:* AJ from openclipart.org
- **File:Edit-clear.svg** *Source:* https://upload.wikimedia.org/wikipedia/en/f/f2/Edit-clear.svg *License:* Public domain *Contributors:* The *Tango! Desktop Project*. *Original artist:*

  The people from the Tango! project. And according to the meta-data in the file, specifically: "Andreas Nilsson, and Jakub Steiner (although minimally)."

- **File:EvilTwinWireless_en.jpg** *Source:* https://upload.wikimedia.org/wikipedia/en/4/4a/EvilTwinWireless_en.jpg *License:* CC-BY-SA-3.0 *Contributors:*
  I derived this from Thomas.Baguette's work, using Paint.NET to edit the strings with similar fonts. *Original artist:*
  User:sarysa
- **File:Fbi_duquesne.jpg** *Source:* https://upload.wikimedia.org/wikipedia/commons/0/07/Fbi_duquesne.jpg *License:* Public domain *Contributors:* ? *Original artist:* ?
- **File:Firewall.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/5/5b/Firewall.png *License:* CC BY-SA 3.0 *Contributors:* Feito por mim *Original artist:* Bruno Pedrozo
- **File:Flag_of_Australia.svg** *Source:* https://upload.wikimedia.org/wikipedia/en/b/b9/Flag_of_Australia.svg *License:* Public domain *Contributors:* ? *Original artist:* ?
- **File:Flag_of_Canada.svg** *Source:* https://upload.wikimedia.org/wikipedia/en/c/cf/Flag_of_Canada.svg *License:* PD *Contributors:* ? *Original artist:* ?
- **File:Flag_of_France.svg** *Source:* https://upload.wikimedia.org/wikipedia/en/c/c3/Flag_of_France.svg *License:* PD *Contributors:* ? *Original artist:* ?
- **File:Flag_of_Germany.svg** *Source:* https://upload.wikimedia.org/wikipedia/en/b/ba/Flag_of_Germany.svg *License:* PD *Contributors:* ? *Original artist:* ?
- **File:Flag_of_New_Zealand.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/3/3e/Flag_of_New_Zealand.svg *License:* Public domain *Contributors:* http://www.mch.govt.nz/files/NZ%20Flag%20-%20proportions.JPG *Original artist:* Zscout370, Hugh Jass and many others
- **File:Flag_of_the_United_Kingdom.svg** *Source:* https://upload.wikimedia.org/wikipedia/en/a/ae/Flag_of_the_United_Kingdom.svg *License:* PD *Contributors:* ? *Original artist:* ?
- **File:Folder_Hexagonal_Icon.svg** *Source:* https://upload.wikimedia.org/wikipedia/en/4/48/Folder_Hexagonal_Icon.svg *License:* Cc-by-sa-3.0 *Contributors:* ? *Original artist:* ?
- **File:Graphiquemalware_en.jpg** *Source:* https://upload.wikimedia.org/wikipedia/en/6/6b/Graphiquemalware_en.jpg *License:* CC-BY-SA-3.0 *Contributors:*
  I derived this from Thomas.Baguette's work, using Paint.NET to edit the strings with similar fonts. *Original artist:*
  User:sarysa
- **File:Gufw_10.04.4.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/b/ba/Gufw_10.04.4.png *License:* GPL *Contributors:* http://gufw.tuxfamily.org *Original artist:* ?
- **File:Honeypot_diagram.jpg** *Source:* https://upload.wikimedia.org/wikipedia/commons/7/76/Honeypot_diagram.jpg *License:* CC BY-SA 3.0 *Contributors:* Own work *Original artist:* 24kanika
- **File:IAO-logo.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/d/d1/IAO-logo.png *License:* Public domain *Contributors:* from http://www.darpa.mil/iao/ (Currently down, see archived page: [1] *Original artist:* USGov-Military
- **File:Internet_map_1024_-_transparent,_inverted.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/3/3f/Internet_map_1024_-_transparent%2C_inverted.png *License:* CC BY 2.5 *Contributors:* Originally from the English Wikipedia; description page is/was here. *Original artist:* The Opte Project
- **File:Juniper_networks_backdoor_admin_password_hidden_in_code.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/e/e3/Juniper_networks_backdoor_admin_password_hidden_in_code.png *License:* CC BY-SA 4.0 *Contributors:* Own work *Original artist:* Zezen
- **File:Keylogger-hardware-PS2-example-connected.jpg** *Source:* https://upload.wikimedia.org/wikipedia/commons/d/dc/Keylogger-hardware-PS2-example-connected.jpg *License:* GFDL *Contributors:* http://www.weboctopus.nl/webshop/img/p/59-430-large.jpg *Original artist:* http://www.weboctopus.nl
- **File:Keylogger-hardware-PS2.jpg** *Source:* https://upload.wikimedia.org/wikipedia/commons/1/11/Keylogger-hardware-PS2.jpg *License:* Copyrighted free use *Contributors:* http://www.keylogger-keyloggers.nl/images/keylogger_company_keylogger_hardware_PS2.jpg *Original artist:* www.keylogger-keyloggers.nl
- **File:Keylogger-screen-capture-example.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/2/22/Keylogger-screen-capture-example.png *License:* MPL 1.1 *Contributors:* Own work *Original artist:* own work
- **File:Keylogger-software-logfile-example.jpg** *Source:* https://upload.wikimedia.org/wikipedia/commons/c/c4/Keylogger-software-logfile-example.jpg *License:* GPL *Contributors:* Own work in combination with the keylogger program http://pykeylogger.sourceforge.net/ and the text editor http://notepad-plus.sourceforge.net/ *Original artist:* Own work
- **File:MalwareEffect.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/a/a6/MalwareEffect.png *License:* GFDL *Contributors:* reproduction et modification d'un graphique *Original artist:* jeff000
- **File:Malware_logo.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/f/ff/Malware_logo.svg *License:* LGPL *Contributors:* Skull and crossbones.svg (valid SVG)
  Original artist: Skull and crossbones.svg: Silsor
- **File:Malware_statics_2011-03-16-en.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/e/ec/Malware_statics_2011-03-16-en.svg *License:* CC BY-SA 3.0 *Contributors:*
- Malware_statics_2011-03-16-es.svg *Original artist:* Malware_statics_2011-03-16-es.svg: Kizar

- **File:Man_in_the_middle_attack.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/e/e7/Man_in_the_middle_attack.svg *License:* CC BY-SA 3.0 *Contributors:* Own work *Original artist:* Miraceti
- **File:Merge-arrow.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/a/aa/Merge-arrow.svg *License:* Public domain *Contributors:* ? *Original artist:* ?
- **File:Mergefrom.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/0/0f/Mergefrom.svg *License:* Public domain *Contributors:* ? *Original artist:* ?
- **File:Metropolitan_Police_ransomware_scam.jpg** *Source:* https://upload.wikimedia.org/wikipedia/en/c/ca/Metropolitan_Police_ransomware_scam.jpg *License:* ? *Contributors:*
  http://nakedsecurity.sophos.com/2012/02/13/metropolitan-police-malware-warning/ *Original artist:*
  - Screenshot: Sophos
- **File:Monitor_padlock.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/7/73/Monitor_padlock.svg *License:* CC BY-SA 3.0 *Contributors:* Own work (Original text: *self-made*) *Original artist:* Lunarbunny (talk)
- **File:Morris_Worm.jpg** *Source:* https://upload.wikimedia.org/wikipedia/commons/b/b6/Morris_Worm.jpg *License:* CC BY-SA 2.0 *Contributors:* Museum of Science - Morris Internet Worm *Original artist:* Go Card USA from Boston, USA
- **File:National_Security_Agency.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/3/3f/Seal_of_the_United_States_National_Security_Agency.svg *License:* Public domain *Contributors:* www.nsa.gov *Original artist:* U.S. Government
- **File:Netfilter-packet-flow.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/3/37/Netfilter-packet-flow.svg *License:* CC BY-SA 3.0 *Contributors:* Own work, Origin SVG PNG *Original artist:* Jan Engelhardt
- **File:P2P-network.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/3/3f/P2P-network.svg *License:* Public domain *Contributors:* Own work *Original artist:* User:Mauro Bieg
- **File:Portal-puzzle.svg** *Source:* https://upload.wikimedia.org/wikipedia/en/f/fd/Portal-puzzle.svg *License:* Public domain *Contributors:* ? *Original artist:* ?
- **File:Privacy-Invasive_Software_Classification.png** *Source:* https://upload.wikimedia.org/wikipedia/en/0/02/Privacy-Invasive_Software_Classification.png *License:* GFDL *Contributors:* ? *Original artist:* ?
- **File:Question_book-new.svg** *Source:* https://upload.wikimedia.org/wikipedia/en/9/99/Question_book-new.svg *License:* Cc-by-sa-3.0 *Contributors:*
  Created from scratch in Adobe Illustrator. Based on Image:Question book.png created by User:Equazcion *Original artist:*
  Tkgd2007
- **File:Rkhunter_Ubuntu.png** *Source:* https://upload.wikimedia.org/wikipedia/en/5/5c/Rkhunter_Ubuntu.png *License:* ? *Contributors:*
  Screenshot taken in Ubuntu
  Original artist:
  Michael Boelen et al
- **File:Rkhunter_on_Mac_OS_X.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/c/c0/Rkhunter_on_Mac_OS_X.png *License:* GPL *Contributors:* Transferred from en.wikipedia to Commons by IngerAlHaosului using CommonsHelper. *Original artist:* The original uploader was CyberSkull at English Wikipedia
- **File:RootkitRevealer.png** *Source:* https://upload.wikimedia.org/wikipedia/en/9/9c/RootkitRevealer.png *License:* Fair use *Contributors:*
  http://blogs.technet.com/b/markrussinovich/archive/2005/10/31/sony-rootkits-and-digital-rights-management-gone-too-far.aspx *Original artist:* ?
- **File:Seal_of_the_United_States_Department_of_Homeland_Security.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/8/8a/Seal_of_the_United_States_Department_of_Homeland_Security.svg *License:* Public domain *Contributors:* http://www.uscg.mil/ *Original artist:* DHS, as noted below.
- **File:Server-based-network.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/f/fb/Server-based-network.svg *License:* LGPL *Contributors:* derived from the Image:Computer n screen.svg which is under the GNU LGPL *Original artist:* User:Mauro Bieg
- **File:SpySheriffPopUp.png** *Source:* https://upload.wikimedia.org/wikipedia/en/b/bd/SpySheriffPopUp.png *License:* ? *Contributors:*
  http://vil.nai.com/vil/content/v_135033.htm *Original artist:* ?
- **File:Stachledraht_DDos_Attack.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/3/3f/Stachledraht_DDos_Attack.svg *License:* LGPL *Contributors:* All Crystal icons were posted by the author as LGPL on kde-look *Original artist:* Everaldo Coelho and YellowIcon
- **File:Strawhorse.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/f/fb/Strawhorse.png *License:* CC BY-SA 4.0 *Contributors:* Own work *Original artist:* Arthur2968
- **File:Stylized_eye.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/4/4c/Stylized_eye.svg *License:* CC0 *Contributors:* Own work *Original artist:* camelNotation
- **File:Symbol_book_class2.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/8/89/Symbol_book_class2.svg *License:* CC BY-SA 2.5 *Contributors:* Mad by Lokal_Profil by combining: *Original artist:* Lokal_Profil
- **File:Symbol_list_class.svg** *Source:* https://upload.wikimedia.org/wikipedia/en/d/db/Symbol_list_class.svg *License:* Public domain *Contributors:* ? *Original artist:* ?

- **File:Text_document_with_red_question_mark.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/a/a4/Text_document_with_red_question_mark.svg *License:* Public domain *Contributors:* Created by bdesham with Inkscape; based upon Text-x-generic.svg from the Tango project. *Original artist:* Benjamin D. Esham (bdesham)
- **File:Translation_to_english_arrow.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/8/8a/Translation_to_english_arrow.svg *License:* CC-BY-SA-3.0 *Contributors:* Own work, based on :Image:Translation_arrow.svg. Created in Adobe Illustrator CS3 *Original artist:* tkgd2007
- **File:US-CentralSecurityService-Seal.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/6/6a/US-CentralSecurityService-Seal.svg *License:* Public domain *Contributors:* Extracted from PDF version of 50th Anniversary Brochure (direct PDF URL [1]). *Original artist:* U.S. Government
- **File:US-DeptOfJustice-Seal.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/5/54/Seal_of_the_United_States_Department_of_Justice.svg *License:* Public domain *Contributors:* Extracted from PDF file available here. *Original artist:* U.S. government
- **File:US-FBI-ShadedSeal.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/7/70/US-FBI-ShadedSeal.svg *License:* Public domain *Contributors:* Extracted from PDF version of a DNI 100-day plan followup report (direct PDF URL here). *Original artist:* Federal Bureau of Investigation
- **File:Virus_Blaster.jpg** *Source:* https://upload.wikimedia.org/wikipedia/commons/e/ec/Virus_Blaster.jpg *License:* Public domain *Contributors:* http://nuevovirus.info/virus-blaster/ *Original artist:* admin
- **File:Wiki_letter_w.svg** *Source:* https://upload.wikimedia.org/wikipedia/en/6/6c/Wiki_letter_w.svg *License:* Cc-by-sa-3.0 *Contributors:* ? *Original artist:* ?
- **File:Wiki_letter_w_cropped.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/1/1c/Wiki_letter_w_cropped.svg *License:* CC-BY-SA-3.0 *Contributors:* This file was derived from Wiki letter w.svg:
Original artist: Derivative work by Thumperward
- **File:Wikibooks-logo-en-noslogan.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/d/df/Wikibooks-logo-en-noslogan.svg *License:* CC BY-SA 3.0 *Contributors:* Own work *Original artist:* User:Bastique, User:Ramac et al.
- **File:Wiktionary-logo-v2.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/0/06/Wiktionary-logo-v2.svg *License:* CC BY-SA 4.0 *Contributors:* Own work *Original artist:* Dan Polansky based on work currently attributed to Wikimedia Foundation but originally created by Smurrayinchester
- **File:Windows_ActiveX_security_warning_(malware).png** *Source:* https://upload.wikimedia.org/wikipedia/en/7/71/Windows_ActiveX_security_warning_%28malware%29.png *License:* ? *Contributors:* ? *Original artist:* ?

8.3 Content license

- Creative Commons Attribution-Share Alike 3.0