

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/296669124>

Wikipedia Handbook of Computer Security and Digital Forensics 2016 – Part I – Computer Security

Book · January 2016

DOI: 10.13140/RG.2.1.1166.8249

READS

5,942

2 authors, including:



[Reiner Creutzburg](#)

Brandenburg University of Applied Sciences

412 PUBLICATIONS 261 CITATIONS

[SEE PROFILE](#)

Wikipedia Handbook of Computer Security and Digital Forensics 2016

Part I – Computer Security

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

1	Introduction	1
1.1	Computer security	1
1.1.1	Vulnerabilities and attacks	1
1.1.2	Systems at risk	2
1.1.3	Impact of security breaches	3
1.1.4	Attacker motivation	4
1.1.5	Computer protection (countermeasures)	4
1.1.6	Notable computer security attacks and breaches	8
1.1.7	Legal issues and global regulation	9
1.1.8	Government	9
1.1.9	Actions and teams in the US	9
1.1.10	International actions	10
1.1.11	National teams	11
1.1.12	Modern warfare	11
1.1.13	The cyber security job market	11
1.1.14	Terminology	12
1.1.15	Scholars	14
1.1.16	See also	14
1.1.17	Further reading	15
1.1.18	References	15
1.1.19	External links	18
2	Related security categories	19
2.1	Internet security	19
2.1.1	Types of security	19
2.1.2	Firewalls	20
2.1.3	Malicious software	21
2.1.4	Denial-of-service attack	21
2.1.5	Phishing	21
2.1.6	Browser choice	21
2.1.7	Application vulnerabilities	21
2.1.8	Internet security products	22
2.1.9	See also	22

2.1.10 References	22
2.1.11 External links	22
2.2 Cyberwarfare	23
2.2.1 Types of threat	23
2.2.2 Motivations	24
2.2.3 Cyberwarfare by country	25
2.2.4 Cyber counterintelligence	28
2.2.5 Controversy over terms	29
2.2.6 Incidents	29
2.2.7 Legality, rules	30
2.2.8 See also	30
2.2.9 Further reading	31
2.2.10 References	31
2.2.11 External links	34
2.3 Information security	35
2.3.1 Overview	35
2.3.2 History	36
2.3.3 Definitions	36
2.3.4 Profession	37
2.3.5 Basic principles	37
2.3.6 Risk management	38
2.3.7 Process	42
2.3.8 Business continuity	44
2.3.9 Laws and regulations	45
2.3.10 Information Security Culture	46
2.3.11 Sources of standards	46
2.3.12 Scholars working in the field	47
2.3.13 See also	47
2.3.14 Further reading	47
2.3.15 Notes and references	48
2.3.16 External links	49
2.4 Network security	49
2.4.1 Network security concepts	49
2.4.2 Security management	49
2.4.3 See also	50
2.4.4 References	50
2.4.5 Further reading	50
2.4.6 External links	51
2.5 Mobile security	51
2.5.1 Challenges of mobile security	51
2.5.2 Attacks based on communication	52

2.5.3	Attacks based on vulnerabilities in software applications	54
2.5.4	Attacks based on hardware vulnerabilities	54
2.5.5	Password cracking	54
2.5.6	Malicious software (malware)	54
2.5.7	Countermeasures	56
2.5.8	See also	61
2.5.9	Notes	61
2.5.10	References	62
2.5.11	Further reading	63
2.5.12	External links	64
3	Threats	65
3.1	Cybercrime	65
3.1.1	Classification	65
3.1.2	Documented cases	68
3.1.3	Combating computer crime	68
3.1.4	See also	69
3.1.5	References	70
3.1.6	Further reading	71
3.1.7	External links	71
3.2	Vulnerability (computing)	72
3.2.1	Definitions	72
3.2.2	Vulnerability and risk factor models	73
3.2.3	Information security management system	74
3.2.4	Classification	74
3.2.5	Causes	74
3.2.6	Vulnerability consequences	75
3.2.7	Vulnerability disclosure	75
3.2.8	Vulnerability disclosure date	75
3.2.9	Identifying and removing vulnerabilities	76
3.2.10	Examples of vulnerabilities	76
3.2.11	See also	76
3.2.12	References	77
3.2.13	External links	77
3.3	Eavesdropping	77
3.3.1	Etymology	78
3.3.2	Techniques	78
3.3.3	References	78
3.3.4	See also	78
3.3.5	External links	79
3.4	Exploit (computer security)	79
3.4.1	Classification	79

3.4.2	See also	79
3.4.3	References	80
3.5	Trojan horse (computing)	80
3.5.1	Purpose and uses	80
3.5.2	Notable examples	81
3.5.3	See also	81
3.5.4	References	81
3.5.5	External links	82
3.6	Computer virus	82
3.6.1	Vulnerabilities and infection vectors	82
3.6.2	Infection targets and replication techniques	83
3.6.3	Stealth strategies	83
3.6.4	Countermeasures	85
3.6.5	Historical development	86
3.6.6	See also	88
3.6.7	References	88
3.6.8	Further reading	90
3.6.9	External links	91
3.7	Virus hoax	91
3.7.1	Identification	91
3.7.2	Action	91
3.7.3	Comparison of computer virus hoaxes	91
3.7.4	Telephone scam	91
3.7.5	Parodies	92
3.7.6	See also	92
3.7.7	References	92
3.7.8	External links	92
3.8	Computer worm	93
3.8.1	History	93
3.8.2	Protecting against dangerous computer worms	94
3.8.3	Worms with good intent	94
3.8.4	See also	94
3.8.5	References	94
3.8.6	External links	95
3.9	Denial-of-service attack	95
3.9.1	Symptoms	95
3.9.2	Attack techniques	96
3.9.3	Attack tools	100
3.9.4	Defense techniques	100
3.9.5	Unintentional denial-of-service	101
3.9.6	Side effects of attacks	102

3.9.7 Legality	102
3.9.8 See also	102
3.9.9 References	103
3.9.10 Further reading	104
3.9.11 External links	105
3.10 Malware	105
3.10.1 Purposes	105
3.10.2 Proliferation	106
3.10.3 Infectious malware: viruses and worms	106
3.10.4 Concealment: Viruses, trojan horses, rootkits, backdoors and evasion	107
3.10.5 Vulnerability to malware	108
3.10.6 Anti-malware strategies	109
3.10.7 Grayware	110
3.10.8 History of viruses and worms	110
3.10.9 Academic research	110
3.10.10 See also	111
3.10.11 References	111
3.10.12 External links	113
3.11 Payload (computing)	113
3.11.1 References	113
3.12 Rootkit	113
3.12.1 History	113
3.12.2 Uses	114
3.12.3 Types	115
3.12.4 Installation and cloaking	117
3.12.5 Detection	117
3.12.6 Removal	119
3.12.7 Public availability	119
3.12.8 Defenses	120
3.12.9 See also	120
3.12.10 Notes	120
3.12.11 References	120
3.12.12 Further reading	123
3.12.13 External links	123
3.13 Keystroke logging	123
3.13.1 Application	123
3.13.2 History	126
3.13.3 Cracking	126
3.13.4 Countermeasures	127
3.13.5 See also	129
3.13.6 References	129

3.13.7 External links	130
4 Defenses	131
4.1 Access control	131
4.1.1 Physical security	131
4.1.2 Computer security	138
4.1.3 Access Control	139
4.1.4 Telecommunication	139
4.1.5 Public policy	140
4.1.6 See also	140
4.1.7 References	140
4.1.8 External links	140
4.2 Application security	141
4.2.1 Methodology	141
4.2.2 Threats, attacks, vulnerabilities, and countermeasures	141
4.2.3 Application threats / attacks	141
4.2.4 Mobile application security	141
4.2.5 Security testing for applications	141
4.2.6 Security certifications	142
4.2.7 Security standards and regulations	142
4.2.8 See also	143
4.2.9 References	143
4.2.10 External links	143
4.3 Antivirus software	144
4.3.1 History	144
4.3.2 Identification methods	146
4.3.3 Issues of concern	148
4.3.4 Performance and other drawbacks	150
4.3.5 Alternative solutions	150
4.3.6 Usage and risks	151
4.3.7 See also	151
4.3.8 References	151
4.3.9 Bibliography	154
4.3.10 External links	154
4.4 Secure coding	155
4.4.1 See also	155
4.4.2 References	155
4.4.3 External links	155
4.5 Secure by design	155
4.5.1 Security by design in practice	155
4.5.2 Server/client architectures	156
4.5.3 See also	156

4.5.4	External links	156
4.6	Security-focused operating system	156
4.6.1	BSD	156
4.6.2	Linux	157
4.6.3	Solaris	159
4.6.4	Microsoft Windows Server	159
4.6.5	Object-capability systems	159
4.6.6	See also	160
4.6.7	References	160
4.6.8	External links	160
4.7	Authentication	161
4.7.1	Methods	161
4.7.2	Factors and identity	162
4.7.3	Product authentication	162
4.7.4	Information content	163
4.7.5	History and state-of-the-art	164
4.7.6	Authorization	164
4.7.7	Access control	165
4.7.8	See also	165
4.7.9	References	166
4.7.10	External links	166
4.8	Multi-factor authentication	166
4.8.1	Authentication Factors	167
4.8.2	Legislation	167
4.8.3	Security	168
4.8.4	Implementation considerations	168
4.8.5	Examples	168
4.8.6	See also	168
4.8.7	References	168
4.8.8	External links	169
4.9	Two-factor authentication	169
4.9.1	Components	169
4.9.2	Mobile phone two-factor authentication	169
4.9.3	List of mobile phone two-factor authentication solution providers	170
4.9.4	References	170
4.9.5	See also	170
4.10	Authorization (computer access control)	170
4.10.1	Overview	171
4.10.2	<i>Authorize v authorise</i>	171
4.10.3	Related interpretations	171
4.10.4	See also	171

4.10.5 References	171
4.11 Data-centric security	172
4.11.1 Key Concepts	172
4.11.2 Technology	172
4.11.3 Data-centric security and cloud computing	172
4.11.4 See Also	172
4.11.5 References	172
4.12 Firewall (computing)	173
4.12.1 History	173
4.12.2 Types	174
4.12.3 See also	176
4.12.4 References	176
4.12.5 External links	177
4.13 Intrusion detection system	177
4.13.1 Terminology	177
4.13.2 HIDS and NIDS	178
4.13.3 Passive and reactive systems	178
4.13.4 Comparison with firewalls	178
4.13.5 Statistical anomaly and signature-based IDSEs	179
4.13.6 Limitations	179
4.13.7 Evasion techniques	179
4.13.8 Development	180
4.13.9 See also	180
4.13.10 Free Intrusion Detection Systems	180
4.13.11 References	181
4.13.12 Further reading	182
4.13.13 External links	182
4.14 Intrusion prevention system	182
4.14.1 Classifications	182
4.14.2 Detection methods	182
4.14.3 See also	182
4.14.4 References	183
4.14.5 External links	183
4.15 Mobile secure gateway	183
4.15.1 Client Library	183
4.15.2 Gateway	183
4.15.3 Host	183
4.15.4 External links	183
5 Text and image sources, contributors, and licenses	184
5.1 Text	184
5.2 Images	198

5.3 Content license	201
-------------------------------	-----

Chapter 1

Introduction

1.1 Computer security

Computer security, also known as **cybersecurity** or **IT security**, is the protection of **information systems** from theft or damage to the hardware, the software, and to the information on them, as well as from disruption or misdirection of the services they provide.^[1] It includes controlling physical access to the hardware, as well as protecting against harm that may come via network access, data and code injection,^[2] and due to malpractice by operators, whether intentional, accidental, or due to them being tricked into deviating from secure procedures.^[3]

The field is of growing importance due to the increasing reliance of computer systems in most societies.^[4] Computer systems now include a very wide variety of “smart” devices, including smartphones, televisions and tiny devices as part of the Internet of Things – and networks include not only the Internet and private data networks, but also Bluetooth, Wi-Fi and other wireless networks.

Computer security covers all the processes and mechanisms by which digital equipment, information and services are protected from unintended or unauthorized access, change or destruction and the process of applying security measures to ensure confidentiality, integrity, and availability of data both in transit and at rest.^[5]

1.1.1 Vulnerabilities and attacks

Main article: **Vulnerability (computing)**

A vulnerability is a system susceptibility or flaw, and many vulnerabilities are documented in the **Common Vulnerabilities and Exposures (CVE)** database and vulnerability management is the cyclical practice of identifying, classifying, remediating, and mitigating vulnerabilities as they are discovered. An *exploitable* vulnerability is one for which at least one working attack or “exploit” exists.

To secure a computer system, it is important to understand the attacks that can be made against it, and these threats can typically be classified into one of the categories below:

Backdoors

A **backdoor** in a computer system, a **cryptosystem** or an algorithm, is any secret method of bypassing normal authentication or security controls. They may exist for a number of reasons, including by original design or from poor configuration. They may also have been added later by an authorized party to allow some legitimate access, or by an attacker for malicious reasons; but regardless of the motives for their existence, they create a vulnerability.

Denial-of-service attack

Main article: **Denial-of-service attack**

Denial of service attacks are designed to make a machine or network resource unavailable to its intended users. Attackers can deny service to individual victims, such as by deliberately entering a wrong password enough consecutive times to cause the victim account to be locked, or they may overload the capabilities of a machine or network and block all users at once. While a network attack from a single IP address can be blocked by adding a new firewall rule, many forms of **Distributed denial of service (DDoS)** attacks are possible, where the attack comes from a large number of points – and defending is much more difficult. Such attacks can originate from the **zombie computers** of a **botnet**, but a range of other techniques are possible including reflection and amplification attacks, where innocent systems are fooled into sending traffic to the victim.

Direct-access attacks

An unauthorized user gaining physical access to a computer is most likely able to directly download data from it. They may also compromise security by making **operating system** modifications, installing software worms, **keyloggers**, or **covert listening devices**. Even when the system is protected by standard security measures, these may be able to be passed by booting another operating system or tool from a **CD-ROM** or other bootable media. **Disk encryption** and **Trusted Platform Module** are designed to prevent these attacks.



Common consumer devices that can be used to transfer data surreptitiously.

Eavesdropping

Eavesdropping is the act of surreptitiously listening to a private conversation, typically between hosts on a network. For instance, programs such as Carnivore and NarusInsight have been used by the FBI and NSA to eavesdrop on the systems of internet service providers. Even machines that operate as a closed system (i.e., with no contact to the outside world) can be eavesdropped upon via monitoring the faint electro-magnetic transmissions generated by the hardware; TEMPEST is a specification by the NSA referring to these attacks.

Spoofing

Spoofing of user identity describes a situation in which one person or program successfully masquerades as another by falsifying data.

Tampering

Tampering describes a malicious modification of products. So-called “Evil Maid” attacks and security services planting of surveillance capability into routers^[6] are examples.

Privilege escalation

Privilege escalation describes a situation where an attacker with some level of restricted access is able to, without authorization, elevate their privileges or access level. So for example a standard computer user may be able to fool the system into giving them access to restricted data; or even to "become root" and have full unrestricted access to a system.

Phishing

Phishing is the attempt to acquire sensitive information such as usernames, passwords, and credit card details directly from users. Phishing is typically carried out by email spoofing or instant messaging, and it often directs users to enter details at a fake website whose look and feel are almost identical to the legitimate one. Preying on a victim's trusting, phishing can be classified as a form of social engineering.

Clickjacking

Clickjacking, also known as “UI redress attack or User Interface redress attack”, is a malicious technique in which an attacker tricks a user into clicking on a button or link on another webpage while the user intended to click on the top level page. This is done using multiple transparent or opaque layers. The attacker is basically “hijacking” the clicks meant for the top level page and routing them to some other irrelevant page, most likely owned by someone else. A similar technique can be used to hijack keystrokes. Carefully drafting a combination of stylesheets, iframes, buttons and text boxes, a user can be led into believing that they are typing the password or other information on some authentic webpage while it is being channeled into an invisible frame controlled by the attacker.

Social engineering and trojans

Main article: Social engineering (security)

See also: Category:Cryptographic attacks

Social engineering aims to convince a user to disclose secrets such as passwords, card numbers, etc. by, for example, impersonating a bank, a contractor, or a customer.^[7]

1.1.2 Systems at risk

Computer security is critical in almost any industry which uses computers.^[8]

Financial systems

Web sites that accept or store credit card numbers and bank account information are prominent hacking targets, because of the potential for immediate financial gain from transferring money, making purchases, or selling the information on the black market. In-store payment systems and ATMs have also been tampered with in order to gather customer account data and PINs.

Utilities and industrial equipment

Computers control functions at many utilities, including coordination of telecommunications, the power grid, nuclear power plants, and valve opening and closing in water and gas networks. The Internet is a potential attack vector for such machines if connected, but the Stuxnet worm demonstrated that even equipment controlled by computers not connected to the Internet can be vulnerable to physical damage caused by malicious commands sent to industrial equipment (in that case uranium enrichment centrifuges) which are infected via removable media. In 2014, the Computer Emergency Readiness Team, a division of the Department of Homeland Security, investigated 79 hacking incidents at energy companies.^[9]

Aviation

The aviation industry is very reliant on a series of complex system which could be attacked.^[10] A simple power outage at one airport can cause repercussions worldwide,^[11] much of the system relies on radio transmissions which could be disrupted,^[12] and controlling aircraft over oceans is especially dangerous because radar surveillance only extends 175 to 225 miles offshore.^[13] There is also potential for attack from within an aircraft.^[14]

The consequences of a successful attack range from loss of confidentiality to loss of system integrity, which may lead to more serious concerns such as exfiltration of data, network and air traffic control outages, which in turn can lead to airport closures, loss of aircraft, loss of passenger life, damages on the ground and to transportation infrastructure. A successful attack on a military aviation system that controls munitions could have even more serious consequences.

Consumer devices

Desktop computers and laptops are commonly infected with malware either to gather passwords or financial account information, or to construct a botnet to attack another target. Smart phones, tablet computers, smart watches, and other mobile devices such as Quantified Self devices like activity trackers have also become targets and many of these have sensors such as cameras, microphones, GPS receivers, compasses, and accelerometers which could be exploited, and may collect personal information, including sensitive health information. Wifi, Bluetooth, and cell phone network on any of these devices could be used as attack vectors, and sensors might be remotely activated after a successful breach.^[15]

Home automation devices such as the Nest thermostat are also potential targets.^[15]

Large corporations

Large corporations are common targets. In many cases this is aimed at financial gain through identity theft and involves data breaches such as the loss of millions of clients' credit card details by Home Depot,^[16] Staples,^[17] and Target Corporation.^[18]

Not all attacks are financially motivated however; for example security firm HBGary Federal suffered a serious series of attacks in 2011 from hacktivist group Anonymous in retaliation for the firm's CEO claiming to have infiltrated their group,^{[19][20]} and Sony Pictures was attacked in 2014 where the motive appears to have been to embarrass with data leaks, and cripple the company by wiping workstations and servers.^{[21][22]}

Automobiles

If access is gained to a car's internal controller area network, it is possible to disable the brakes and turn the steering wheel.^[23] Computerized engine timing, cruise control, anti-lock brakes, seat belt tensioners, door locks, airbags and advanced driver assistance systems make these disruptions possible, and self-driving cars go even further. Connected cars may use wifi and bluetooth to communicate with onboard consumer devices, and the cell phone network to contact concierge and emergency assistance services or get navigational or entertainment information; each of these networks is a potential entry point for malware or an attacker.^[23] Researchers in 2011 were even able to use a malicious compact disc in a car's stereo system as a successful attack vector,^[24] and cars with built-in voice recognition or remote assistance features have onboard microphones which could be used for eavesdropping.

A 2015 report by U.S. Senator Edward Markey criticized manufacturers' security measures as inadequate, and also highlighted privacy concerns about driving, location, and diagnostic data being collected, which is vulnerable to abuse by both manufacturers and hackers.^[25]

Government

Government and military computer systems are commonly attacked by activists^{[26][27][28][29]} and foreign powers.^{[30][31][32][33]} Local and regional government infrastructure such as traffic light controls, police and intelligence agency communications, personnel records and financial systems are also potential targets as they are now all largely computerized.

1.1.3 Impact of security breaches

Serious financial damage has been caused by security breaches, but because there is no standard model for es-

timating the cost of an incident, the only data available is that which is made public by the organizations involved. “Several computer security consulting firms produce estimates of total worldwide losses attributable to **virus** and **worm** attacks and to hostile digital acts in general. The 2003 loss estimates by these firms range from \$13 billion (worms and viruses only) to \$226 billion (for all forms of covert attacks). The reliability of these estimates is often challenged; the underlying methodology is basically anecdotal.”^[34]

However, reasonable estimates of the financial cost of security breaches can actually help organizations make rational investment decisions. According to the classic **Gordon-Loeb Model** analyzing the optimal investment level in information security, one can conclude that the amount a firm spends to protect information should generally be only a small fraction of the expected loss (i.e., the **expected value** of the loss resulting from a **cyber/information security breach**).^[35]

1.1.4 Attacker motivation

As with **physical security**, the motivations for breaches of computer security vary between attackers. Some are thrill-seekers or **vandals**, others are activists; or criminals looking for financial gain. State-sponsored attackers are now common and well resourced, but started with amateurs such as **Markus Hess** who hacked for the **KGB**, as recounted by **Clifford Stoll**, in *The Cuckoo’s Egg*.

A standard part of **threat modelling** for any particular system is to identify what might motivate an attack on that system, and who might be motivated to breach it. The level and detail of precautions will vary depending on the system to be secured. A home **personal computer**, **bank** and **classified military network** all face very different threats, even when the underlying technologies in use are similar.

1.1.5 Computer protection (countermeasures)

In computer security a countermeasure is an action, device, procedure, or technique that reduces a **threat**, a **vulnerability**, or an **attack** by eliminating or preventing it, by minimizing the harm it can cause, or by discovering and reporting it so that corrective action can be taken.^{[36][37][38]}

Some common countermeasures are listed in the following sections:

Security measures

A state of computer “security” is the conceptual ideal, attained by the use of the three processes: threat prevention, detection, and response. These processes are

based on various policies and system components, which include the following:

- User account access controls and **cryptography** can protect systems files and data, respectively.
- **Firewalls** are by far the most common prevention systems from a network security perspective as they can (if properly configured) shield access to internal network services, and block certain kinds of attacks through packet filtering. Firewalls can be both hardware- or software-based.
- **Intrusion Detection System (IDS)** products are designed to detect network attacks in-progress and assist in post-attack **forensics**, while **audit trails** and **logs** serve a similar function for individual systems.
- “Response” is necessarily defined by the assessed security requirements of an individual system and may cover the range from simple upgrade of protections to notification of **legal** authorities, counter-attacks, and the like. In some special cases, a complete destruction of the compromised system is favored, as it may happen that not all the compromised resources are detected.

Today, computer security comprises mainly “preventive” measures, like firewalls or an **exit procedure**. A firewall can be defined as a way of filtering network data between a host or a network and another network, such as the **Internet**, and can be implemented as software running on the machine, hooking into the **network stack** (or, in the case of most **UNIX**-based operating systems such as **Linux**, built into the operating system **kernel**) to provide real time filtering and blocking. Another implementation is a so-called **physical firewall** which consists of a separate machine filtering network traffic. Firewalls are common amongst machines that are permanently connected to the **Internet**.

However, relatively few organisations maintain computer systems with effective detection systems, and fewer still have organised response mechanisms in place. As result, as Reuters points out: “Companies for the first time report they are losing more through electronic theft of data than physical stealing of assets”.^[39] The primary obstacle to effective eradication of cyber crime could be traced to excessive reliance on firewalls and other automated “detection” systems. Yet it is basic evidence gathering by using **packet capture appliances** that puts criminals behind bars.

Reducing vulnerabilities

While **formal verification** of the correctness of computer systems is possible,^{[40][41]} it is not yet common. Operating systems formally verified include **seL4**,^[42] and

SYSGO's PikeOS^{[43][44]} – but these make up a very small percentage of the market.

Cryptography properly implemented is now virtually impossible to directly break. Breaking them requires some non-cryptographic input, such as a stolen key, stolen plaintext (at either end of the transmission), or some other extra cryptanalytic information.

Two factor authentication is a method for mitigating unauthorized access to a system or sensitive information. It requires "something you know"; a password or PIN, and "something you have"; a card, dongle, cellphone, or other piece of hardware. This increases security as an unauthorized person needs both of these to gain access.

Social engineering and direct computer access (physical) attacks can only be prevented by non-computer means, which can be difficult to enforce, relative to the sensitivity of the information. Even in a highly disciplined environment, such as in military organizations, social engineering attacks can still be difficult to foresee and prevent.

It is possible to reduce an attacker's chances by keeping systems up to date with security patches and updates, using a security scanner or/and hiring competent people responsible for security. The effects of data loss/damage can be reduced by careful **backing up** and **insurance**.

Security by design

Main article: Secure by design

Security by design, or alternately secure by design, means that the software has been designed from the ground up to be secure. In this case, security is considered as a main feature.

Some of the techniques in this approach include:

- The **principle of least privilege**, where each part of the system has only the privileges that are needed for its function. That way even if an **attacker** gains access to that part, they have only limited access to the whole system.
- **Automated theorem proving** to prove the correctness of crucial software subsystems.
- **Code reviews** and **unit testing**, approaches to make modules more secure where formal correctness proofs are not possible.
- **Defense in depth**, where the design is such that more than one subsystem needs to be violated to compromise the integrity of the system and the information it holds.
- Default secure settings, and design to "fail secure" rather than "fail insecure" (see **fail-safe** for the equivalent in **safety engineering**). Ideally, a secure system should require a deliberate, conscious,

knowledgeable and free decision on the part of legitimate authorities in order to make it insecure.

- **Audit trails** tracking system activity, so that when a security breach occurs, the mechanism and extent of the breach can be determined. Storing audit trails remotely, where they can only be appended to, can keep intruders from covering their tracks.
- **Full disclosure** of all vulnerabilities, to ensure that the "window of vulnerability" is kept as short as possible when bugs are discovered.

Security architecture

The Open Security Architecture organization defines IT security architecture as "the design artifacts that describe how the security controls (security countermeasures) are positioned, and how they relate to the overall information technology architecture. These controls serve the purpose to maintain the system's quality attributes: confidentiality, integrity, availability, accountability and **assurance services**".^[45]

Techopedia defines security architecture as "a unified security design that addresses the necessities and potential risks involved in a certain scenario or environment. It also specifies when and where to apply security controls. The design process is generally reproducible." The key attributes of security architecture are:^[46]

- the relationship of different components and how they depend on each other.
- the determination of controls based on risk assessment, good practice, finances, and legal matters.
- the standardization of controls.

Hardware protection mechanisms

See also: Computer security compromised by hardware failure

While hardware may be a source of insecurity, such as with microchip vulnerabilities maliciously introduced during the manufacturing process,^{[47][48]} hardware-based or assisted computer security also offers an alternative to software-only computer security. Using devices and methods such as **dongles**, **trusted platform modules**, intrusion-aware cases, drive locks, disabling USB ports, and mobile-enabled access may be considered more secure due to the physical access (or sophisticated **backdoor access**) required in order to be compromised. Each of these is covered in more detail below.

- USB **dongles** are typically used in software licensing schemes to unlock software capabilities,^[49] but

they can also be seen as a way to prevent unauthorized access to a computer or other device's software. The dongle, or key, essentially creates a secure encrypted tunnel between the software application and the key. The principle is that an encryption scheme on the dongle, such as **Advanced Encryption Standard** (AES) provides a stronger measure of security, since it is harder to hack and replicate the dongle than to simply copy the native software to another machine and use it. Another security application for dongles is to use them for accessing web-based content such as cloud software or **Virtual Private Networks** (VPNs).^[50] In addition, a USB dongle can be configured to lock or unlock a computer.^[51]

- Trusted platform modules (TPMs) secure devices by integrating cryptographic capabilities onto access devices, through the use of microprocessors, or so-called computers-on-a-chip. TPMs used in conjunction with server-side software offer a way to detect and authenticate hardware devices, preventing unauthorized network and data access.^[52]
- Computer case intrusion detection refers to a push-button switch which is triggered when a computer case is opened. The firmware or BIOS is programmed to show an alert to the operator when the computer is booted up the next time.
- Drive locks are essentially software tools to encrypt hard drives, making them inaccessible to thieves.^[53] Tools exist specifically for encrypting external drives as well.^[54]
- Disabling USB ports is a security option for preventing unauthorized and malicious access to an otherwise secure computer. Infected USB dongles connected to a network from a computer inside the firewall are considered by Network World as the most common hardware threat facing computer networks.^[55]
- Mobile-enabled access devices are growing in popularity due to the ubiquitous nature of cell phones. Built-in capabilities such as Bluetooth, the newer **Bluetooth low energy** (LE), **Near field communication** (NFC) on non-iOS devices and biometric validation such as thumb print readers, as well as **QR code** reader software designed for mobile devices, offer new, secure ways for mobile phones to connect to access control systems. These control systems provide computer security and can also be used for controlling access to secure buildings.^[56]

Secure operating systems

Main article: Security-focused operating system

One use of the term "computer security" refers to technology that is used to implement secure **operating systems**. Much of this technology is based on science developed in the 1980s and used to produce what may be some of the most impenetrable operating systems ever. Though still valid, the technology is in limited use today, primarily because it imposes some changes to system management and also because it is not widely understood. Such ultra-strong secure operating systems are based on **operating system kernel** technology that can guarantee that certain security policies are absolutely enforced in an operating environment. An example of such a **Computer security policy** is the **Bell-LaPadula model**. The strategy is based on a coupling of special **microprocessor** hardware features, often involving the **memory management unit**, to a special correctly implemented operating system kernel. This forms the foundation for a secure operating system which, if certain critical parts are designed and implemented correctly, can ensure the absolute impossibility of penetration by hostile elements. This capability is enabled because the configuration not only imposes a security policy, but in theory completely protects itself from corruption. Ordinary operating systems, on the other hand, lack the features that assure this maximal level of security. The design methodology to produce such secure systems is precise, deterministic and logical.

Systems designed with such methodology represent the state of the art of computer security although products using such security are not widely known. In sharp contrast to most kinds of software, they meet specifications with verifiable certainty comparable to specifications for size, weight and power. Secure operating systems designed this way are used primarily to protect national security information, military secrets, and the data of international financial institutions. These are very powerful security tools and very few secure operating systems have been certified at the highest level (**Orange Book A-1**) to operate over the range of "Top Secret" to "unclassified" (including Honeywell SCOMP, USAF SACDIN, NSA Blacker and Boeing MLS LAN). The assurance of security depends not only on the soundness of the design strategy, but also on the assurance of correctness of the implementation, and therefore there are degrees of security strength defined for COMPUSEC. The **Common Criteria** quantifies security strength of products in terms of two components, security functionality and assurance level (such as EAL levels), and these are specified in a **Protection Profile** for requirements and a **Security Target** for product descriptions. None of these ultra-high assurance secure general purpose operating systems have been produced for decades or certified under Common Criteria.

In USA parlance, the term **High Assurance** usually suggests the system has the right security functions that are implemented robustly enough to protect DoD and DoE classified information. Medium assurance suggests it can protect less valuable information, such as income tax in-

formation. Secure operating systems designed to meet medium robustness levels of security functionality and assurance have seen wider use within both government and commercial markets. Medium robust systems may provide the same security functions as high assurance secure operating systems but do so at a lower assurance level (such as Common Criteria levels EAL4 or EAL5). Lower levels mean we can be less certain that the security functions are implemented flawlessly, and therefore less dependable. These systems are found in use on web servers, guards, database servers, and management hosts and are used not only to protect the data stored on these systems but also to provide a high level of protection for network connections and routing services.

Secure coding

Main article: [Secure coding](#)

If the operating environment is not based on a secure operating system capable of maintaining a domain for its own execution, and capable of protecting application code from malicious subversion, and capable of protecting the system from subverted code, then high degrees of security are understandably not possible. While such secure operating systems are possible and have been implemented, most commercial systems fall in a 'low security' category because they rely on features not supported by secure operating systems (like portability, and others). In low security operating environments, applications must be relied on to participate in their own protection. There are 'best effort' secure coding practices that can be followed to make an application more resistant to malicious subversion.

In commercial environments, the majority of software subversion **vulnerabilities** result from a few known kinds of coding defects. Common software defects include buffer overflows, format string vulnerabilities, integer overflow, and code/command injection. These defects can be used to cause the target system to execute putative data. However, the "data" contain executable instructions, allowing the attacker to gain control of the processor.

Some common languages such as C and C++ are vulnerable to all of these defects (see Seacord, "*Secure Coding in C and C++*").^[57] Other languages, such as Java, are more resistant to some of these defects, but are still prone to code/command injection and other software defects which facilitate subversion.

Another bad coding practice occurs when an object is deleted during normal operation yet the program neglects to update any of the associated memory pointers, potentially causing system instability when that location is referenced again. This is called **dangling pointer**, and the first known exploit for this particular problem was presented in July 2007. Before this publication the problem

was known but considered to be academic and not practically exploitable.^[58]

Unfortunately, there is no theoretical model of "secure coding" practices, nor is one practically achievable, insofar as the code (ideally, read-only) and data (generally read/write) generally tends to have some form of defect.

Capabilities and access control lists

Main articles: [Access control list](#) and [Capability \(computers\)](#)

Within computer systems, two of many security models capable of enforcing privilege separation are **access control lists** (ACLs) and **capability-based security**. Using ACLs to confine programs has been proven to be insecure in many situations, such as if the host computer can be tricked into indirectly allowing restricted file access, an issue known as the **confused deputy problem**. It has also been shown that the promise of ACLs of giving access to an object to only one person can never be guaranteed in practice. Both of these problems are resolved by capabilities. This does not mean practical flaws exist in all ACL-based systems, but only that the designers of certain utilities must take responsibility to ensure that they do not introduce flaws.

Capabilities have been mostly restricted to research operating systems, while commercial OSs still use ACLs. Capabilities can, however, also be implemented at the language level, leading to a style of programming that is essentially a refinement of standard object-oriented design. An open source project in the area is the **E language**.

The most secure computers are those not connected to the Internet and shielded from any interference. In the real world, the most secure systems are operating systems where **security** is not an add-on.

Response to breaches

Responding forcefully to attempted **security breaches** (in the manner that one would for attempted physical security breaches) is often very difficult for a variety of reasons:

- Identifying attackers is difficult, as they are often in a different **jurisdiction** to the systems they attempt to breach, and operate through proxies, temporary anonymous dial-up accounts, wireless connections, and other anonymising procedures which make backtracing difficult and are often located in yet another jurisdiction. If they successfully breach security, they are often able to delete **logs** to cover their tracks.
- The sheer number of attempted attacks is so large that organisations cannot spend time pursuing each attacker (a typical home user with a permanent (e.g.,

cable modem) connection will be attacked at least several times per day,^[59] so more attractive targets could be presumed to see many more). Note however, that most of the sheer bulk of these attacks are made by automated vulnerability scanners and computer worms.

- Law enforcement officers are often unfamiliar with information technology, and so lack the skills and interest in pursuing attackers. There are also budgetary constraints. It has been argued that the high cost of technology, such as DNA testing, and improved forensics mean less money for other kinds of law enforcement, so the overall rate of criminals not getting dealt with goes up as the cost of the technology increases. In addition, the identification of attackers across a network may require logs from various points in the network and in many countries, the release of these records to law enforcement (with the exception of being voluntarily surrendered by a network administrator or a system administrator) requires a search warrant and, depending on the circumstances, the legal proceedings required can be drawn out to the point where the records are either regularly destroyed, or the information is no longer relevant.

1.1.6 Notable computer security attacks and breaches

Some illustrative examples of different types of computer security breaches are given below.

Robert Morris and the first computer worm

Main article: Morris worm

In 1988, only 60,000 computers were connected to the Internet, and most were mainframes, minicomputers and professional workstations. On November 2, 1988, many started to slow down, because they were running a malicious code that demanded processor time and that spread itself to other computers – the first internet "computer worm".^[60] The software was traced back to 23-year-old Cornell University graduate student Robert Tappan Morris, Jr. who said 'he wanted to count how many machines were connected to the Internet'.^[60]

Rome Laboratory

In 1994, over a hundred intrusions were made by unidentified crackers into the Rome Laboratory, the US Air Force's main command and research facility. Using trojan horses, hackers were able to obtain unrestricted access to Rome's networking systems and remove traces of their activities. The intruders were able to obtain

classified files, such as air tasking order systems data and furthermore able to penetrate connected networks of National Aeronautics and Space Administration's Goddard Space Flight Center, Wright-Patterson Air Force Base, some Defense contractors, and other private sector organizations, by posing as a trusted Rome center user.^[61]

TJX loses 45.7m customer credit card details

In early 2007, American apparel and home goods company TJX announced that it was the victim of an unauthorized computer systems intrusion^[62] and that the hackers had accessed a system that stored data on credit card, debit card, check, and merchandise return transactions.^[63]

Stuxnet attack

The computer worm known as Stuxnet reportedly ruined almost one-fifth of Iran's nuclear centrifuges^[64] by disrupting industrial programmable logic controllers (PLCs) in a targeted attack generally believed to have been launched by Israel and the United States^{[65][66][67][68]} although neither has publicly acknowledged this.

Global surveillance disclosures

Main article: Global surveillance disclosures (2013–present)

In early 2013, massive breaches of computer security by the NSA were revealed, including deliberately inserting a backdoor in a NIST standard for encryption^[69] and tapping the links between Google's data centres.^[70] These were disclosed by NSA contractor Edward Snowden.^[71]

Target and Home Depot breaches

In 2013 and 2014, a Russian/Ukrainian hacking ring known as "Rescator" broke into Target Corporation computers in 2013, stealing roughly 40 million credit cards,^[72] and then Home Depot computers in 2014, stealing between 53 and 56 million credit card numbers.^[73] Warnings were delivered at both corporations, but ignored; physical security breaches using self checkout machines are believed to have played a large role. "The malware utilized is absolutely unsophisticated and uninteresting," says Jim Walter, director of threat intelligence operations at security technology company McAfee – meaning that the heists could have easily been stopped by existing antivirus software had administrators responded to the warnings. The size of the thefts has resulted in major attention from state and Federal United States authorities and the investigation is ongoing.

1.1.7 Legal issues and global regulation

Conflict of laws in cyberspace has become a major cause of concern for computer security community. Some of the main challenges and complaints about the antivirus industry are the lack of global web regulations, a global base of common rules to judge, and eventually punish, **cyber crimes** and **cyber criminals**. There is no global cyber law and cybersecurity treaty that can be invoked for enforcing global cybersecurity issues.

International legal issues of cyber attacks are complicated in nature. Even if an antivirus firm locates the cyber criminal behind the creation of a particular **virus** or piece of malware or form of **cyber attack**, often the local authorities cannot take action due to lack of laws under which to prosecute.^{[74][75]} Authorship attribution for cyber crimes and cyber attacks is a major problem for all law enforcement agencies.

"[Computer viruses] switch from one country to another, from one jurisdiction to another — moving around the world, using the fact that we don't have the capability to globally police operations like this. So the Internet is as if someone [had] given free plane tickets to all the online criminals of the world."^[74] Use of dynamic DNS, **fast flux** and **bullet proof servers** have added own complexities to this situation.

1.1.8 Government

The role of the government is to make **regulations** to force companies and organizations to protect their systems, infrastructure and information from any **cyber attacks**, but also to protect its own national infrastructure such as the national **power-grid**.^[76]

The question of whether the government should intervene or not in the regulation of the **cyberspace** is a very polemical one. Indeed, for as long as it has existed and by definition, the cyberspace is a **virtual space** free of any government intervention. Where everyone agree that an improvement on **cybersecurity** is more than vital, is the government the best actor to solve this issue? Many government officials and experts think that the government should step in and that there is a crucial need for regulation, mainly due to the failure of the private sector to solve efficiently the cybersecurity problem. R. Clarke said during a panel discussion at the **RSA Security Conference** in **San Francisco**, he believes that the "industry only responds when you threaten regulation. If industry doesn't respond (to the threat), you have to follow through."^[77] On the other hand, executives from the private sector agree that improvements are necessary but think that the government intervention would affect their ability to innovate efficiently.

1.1.9 Actions and teams in the US

Legislation

The 1986 18 U.S.C. § 1030, more commonly known as the **Computer Fraud and Abuse Act** is the key legislation. It prohibits unauthorized access or damage of "protected computers" as defined in 18 U.S.C. § 1030(e)(2).

Although various other measures have been proposed, such as the "Cybersecurity Act of 2010 – S. 773" in 2009, the "International Cybercrime Reporting and Cooperation Act – H.R.4962"^[78] and "Protecting Cyberspace as a National Asset Act of 2010 – S.3480"^[79] in 2010 – none of these has succeeded.

Executive order 13636 Improving Critical Infrastructure Cybersecurity was signed February 12, 2013.

Agencies

Homeland Security The Department of Homeland Security has a dedicated division responsible for the response system, **risk management** program and requirements for cybersecurity in the United States called the **National Cyber Security Division**.^{[80][81]} The division is home to US-CERT operations and the National Cyber Alert System.^[81] The National Cybersecurity and Communications Integration Center brings together government organizations responsible for protecting computer networks and networked infrastructure.^[82]

FBI The third priority of the **Federal Bureau of Investigation** (FBI) is to: "*Protect the United States against cyber-based attacks and high-technology crimes*",^[83] and they, along with the **National White Collar Crime Center** (NW3C), and the **Bureau of Justice Assistance** (BJA) are part of the multi-agency task force, **The Internet Crime Complaint Center**, also known as IC3.^[84]

In addition to its own specific duties, the FBI participates alongside non-profit organizations such as **InfraGard**.^{[85][86]}

Department of Justice In the criminal division of the **United States Department of Justice** operates a section called the **Computer Crime and Intellectual Property Section**. The CCIPS is in charge of investigating **computer crime** and **intellectual property** crime and is specialized in the search and seizure of **digital evidence** in computers and networks.^[87]

USCYBERCOM The **United States Cyber Command**, also known as **USCYBERCOM**, is tasked with the defense of specified Department of Defense information networks and "*ensure US/Allied freedom of action in cyberspace and deny the same to our adversaries*".^[88] It has no role in the protection of civilian networks.^{[89][90]}

FCC

The U.S. Federal Communications Commission's role in cybersecurity is to strengthen the protection of critical communications infrastructure, to assist in maintaining the reliability of networks during disasters, to aid in swift recovery after, and to ensure that first responders have access to effective communications services.^[91]

Computer Emergency Readiness Team

Computer Emergency Response Team is a name given to expert groups that handle computer security incidents. In the US, two distinct organization exist, although they do work closely together.

- US-CERT: part of the National Cyber Security Division of the United States Department of Homeland Security.^[92]
- CERT/CC: created by the Defense Advanced Research Projects Agency (DARPA) and run by the Software Engineering Institute (SEI).

1.1.10 International actions

Many different teams and organisations exist, including:

- The Forum of Incident Response and Security Teams (FIRST) is the global association of CSIRTs.^[93] The US-CERT, AT&T, Apple, Cisco, McAfee, Microsoft are all members of this international team.^[94]
- The Council of Europe helps protect societies worldwide from the threat of cybercrime through the Convention on Cybercrime.^[95]
- The purpose of the Messaging Anti-Abuse Working Group (MAAWG) is to bring the messaging industry together to work collaboratively and to successfully address the various forms of messaging abuse, such as spam, viruses, denial-of-service attacks and other messaging exploitations.^[96] France Telecom, Facebook, AT&T, Apple, Cisco, Sprint are some of the members of the MAAWG.^[96]
- ENISA : The European Network and Information Security Agency (ENISA) is an agency of the European Union with the objective to improve network and information security in the European Union.

Germany

Berlin starts National Cyber Defense Initiative On June 16, 2011, the German Minister for Home Affairs, officially opened the new German NCAZ (National Center for Cyber Defense) Nationales Cyber-Abwehrzentrum, which is located in Bonn. The NCAZ

closely cooperates with BSI (Federal Office for Information Security) Bundesamt für Sicherheit in der Informationstechnik, BKA (Federal Police Organisation) Bundeskriminalamt (Deutschland), BND (Federal Intelligence Service) Bundesnachrichtendienst, MAD (Military Intelligence Service) Amt für den Militärischen Abschirmdienst and other national organisations in Germany taking care of national security aspects. According to the Minister the primary task of the new organisation founded on February 23, 2011, is to detect and prevent attacks against the national infrastructure and mentioned incidents like Stuxnet.

South Korea

Following cyberattacks in the first half of 2013, when government, news-media, television station, and bank websites were compromised, the national government committed to the training of 5,000 new cybersecurity experts by 2017. The South Korean government blamed its northern counterpart for these attacks, as well as incidents that occurred in 2009, 2011,^[97] and 2012, but Pyongyang denies the accusations.^[98]

India

Some provisions for cybersecurity have been incorporated into rules framed under the Information Technology Act 2000.

The National Cyber Security Policy 2013 is a policy framework by Department of Electronics and Information Technology (DeitY) which aims to protect the public and private infrastructure from cyber attacks, and safeguard “information, such as personal information (of web users), financial and banking information and sovereign data”.

The Indian Companies Act 2013 has also introduced cyber law and cyber security obligations on the part of Indian directors.

Canada

On October 3, 2010, Public Safety Canada unveiled Canada’s Cyber Security Strategy, following a Speech from the Throne commitment to boost the security of Canadian cyberspace.^{[99][100]} The aim of the strategy is to strengthen Canada’s “cyber systems and critical infrastructure sectors, support economic growth and protect Canadians as they connect to each other and to the world.”^[101] Three main pillars define the strategy: securing government systems, partnering to secure vital cyber systems outside the federal government, and helping Canadians to be secure online.^[101] The strategy involves multiple departments and agencies across the Government of Canada.^[102] The Cyber Incident Management Framework for Canada outlines these responsibilities,

and provides a plan for coordinated response between government and other partners in the event of a cyber incident.^[103] The Action Plan 2010–2015 for Canada's Cyber Security Strategy outlines the ongoing implementation of the strategy.^[104]

Public Safety Canada's Canadian Cyber Incident Response Centre (CCIRC) is responsible for mitigating and responding to threats to Canada's critical infrastructure and cyber systems. The CCIRC provides support to mitigate cyber threats, technical support to respond and recover from targeted cyber attacks, and provides online tools for members of Canada's critical infrastructure sectors.^[105] The CCIRC posts regular cyber security bulletins on the Public Safety Canada website.^[106] The CCIRC also operates an online reporting tool where individuals and organizations can report a cyber incident.^[107] Canada's Cyber Security Strategy is part of a larger, integrated approach to critical infrastructure protection, and functions as a counterpart document to the National Strategy and Action Plan for Critical Infrastructure.^[102]

On September 27, 2010, Public Safety Canada partnered with STOP.THINK.CONNECT, a coalition of non-profit, private sector, and government organizations dedicated to informing the general public on how to protect themselves online.^[108] On February 4, 2014, the Government of Canada launched the Cyber Security Co-operation Program.^[109] The program is a \$1.5 million five-year initiative aimed at improving Canada's cyber systems through grants and contributions to projects in support of this objective.^[110] Public Safety Canada aims to begin an evaluation of Canada's Cyber Security Strategy in early 2015.^[102] Public Safety Canada administers and routinely updates the GetCyberSafe portal for Canadian citizens, and carries out Cyber Security Awareness Month during October.^[111]

1.1.11 National teams

Here are the main computer emergency response teams around the world. Every country have their own team to protect network security. February 27, 2014, the Chinese network security and information technology leadership team is established. The leadership team will focus on national security and long-term development, co-ordination of major issues related to network security and information technology economic, political, cultural, social, and military and other fields of research to develop network security and information technology strategy, planning and major macroeconomic policy promote national network security and information technology law, and constantly enhance security capabilities.

Europe

CSIRTs in Europe collaborate in the TERENA task force TF-CSIRT. TERENA's Trusted Introducer service

provides an accreditation and certification scheme for CSIRTs in Europe. A full list of known CSIRTs in Europe is available from the Trusted Introducer website.

Other countries

- CERT Brazil, member of FIRST (Forum for Incident Response and Security Teams)
- CARNet CERT, Croatia, member of FIRST
- AE CERT, United Arab Emirates
- SingCERT, Singapore
- CERT-LEXSI, France, Canada, Singapore

1.1.12 Modern warfare

Main article: [Cyberwarfare](#)

Cybersecurity is becoming increasingly important as more information and technology is being made available on cyberspace. There is growing concern among governments that cyberspace will become the next theatre of warfare. As Mark Clayton from the *Christian Science Monitor* described in an article titled "The New Cyber Arms Race":

In the future, wars will not just be fought by soldiers with guns or with planes that drop bombs. They will also be fought with the click of a mouse a half a world away that unleashes carefully weaponized computer programs that disrupt or destroy critical industries like utilities, transportation, communications, and energy. Such attacks could also disable military networks that control the movement of troops, the path of jet fighters, the command and control of warships.^[112]

This has led to new terms such as *cyberwarfare* and *cyberterrorism*. More and more critical infrastructure is being controlled via computer programs that, while increasing efficiency, exposes new vulnerabilities. The test will be to see if governments and corporations that control critical systems such as energy, communications and other information will be able to prevent attacks before they occur. As Jay Cross, the chief scientist of the Internet Time Group, remarked, "Connectedness begets vulnerability."^[112]

1.1.13 The cyber security job market

Cyber Security is a fast-growing^[113] field of IT concerned with reducing organizations' risk of hack or data breach.

Commercial, government and non-governmental organizations all employ cybersecurity professionals. However, the use of the term “cybersecurity” is more prevalent in government job descriptions.^[114]

Typical cybersecurity job titles and descriptions include:^[115]

Security Analyst Analyzes and assesses vulnerabilities in the infrastructure (software, hardware, networks), investigates available tools and countermeasures to remedy the detected vulnerabilities, and recommends solutions and best practices. Analyzes and assesses damage to the data/infrastructure as a result of security incidents, examines available recovery tools and processes, and recommends solutions. Tests for compliance with security policies and procedures. May assist in the creation, implementation, and/or management of security solutions.

Security Engineer

Performs security monitoring, security and data/logs analysis, and forensic analysis, to detect security incidents, and mounts incident response. Investigates and utilizes new technologies and processes to enhance security capabilities and implement improvements. May also review code or perform other security engineering methodologies.

Security Architect

Designs a security system or major components of a security system, and may head a security design team building a new security system.

Security Administrator

Installs and manages organization-wide security systems. May also take on some of the tasks of a security analyst in smaller organizations.

Chief Information Security Officer (CISO)

A high-level management position responsible for the entire information security division/staff. The position may include hands-on technical work.

Chief Security Officer (CSO)

A high-level management position responsible for the entire security division/staff. A newer position now deemed needed as security risks grow.

Security Consultant/Specialist/Intelligence

Broad titles that encompass any one or all of the other roles/titles, tasked with protecting computers, networks, software, data, and/or information systems against viruses, worms, spyware, malware, intrusion detection, unauthorized access, denial-of-service attacks, and an ever increasing list of attacks by hackers acting as individuals or as part of organized crime or foreign governments.

Student programs are also available to people interested in beginning a career in cybersecurity.^{[116][117]} Meanwhile, a flexible and effective option for information security professionals of all experience levels to keep studying is online security training, including webcasts.^{[118][119][120]}

1.1.14 Terminology

The following terms used with regards to engineering secure systems are explained below.

- Access authorization restricts access to a computer to group of users through the use of authentication systems. These systems can protect either the whole computer – such as through an interactive login screen – or individual services, such as an FTP server. There are many methods for identifying and authenticating users, such as passwords, identification cards, and, more recently, smart cards and biometric systems.
- Anti-virus software consists of computer programs that attempt to identify, thwart and eliminate computer viruses and other malicious software (malware).
- Applications with known security flaws should not be run. Either leave it turned off until it can be patched or otherwise fixed, or delete it and replace it with some other application. Publicly known flaws are the main entry used by worms to automatically break into a system and then spread to other systems connected to it. The security website Secunia provides a search tool for unpatched known flaws in popular products.
- Authentication techniques can be used to ensure that communication end-points are who they say they are.
- Automated theorem proving and other verification tools can enable critical algorithms and code used in secure systems to be mathematically proven to meet their specifications.
- Backups are a way of securing information; they are another copy of all the important computer files kept in another location. These files are kept on hard disks, CD-Rs, CD-RWs, tapes and more recently on the cloud. Suggested locations for backups are a fire-proof, waterproof, and heat proof safe, or in a separate, offsite location than that in which the original files are contained. Some individuals and companies also keep their backups in safe deposit boxes inside bank vaults. There is also a fourth option, which involves using one of the file hosting services that backs up files over the Internet for both business and individuals, known as the cloud.

- Backups are also important for reasons other than security. Natural disasters, such as earthquakes, hurricanes, or tornadoes, may strike the building where the computer is located. The building can be on fire, or an explosion may occur. There needs to be a recent backup at an alternate secure location, in case of such kind of disaster. Further, it is recommended that the alternate location be placed where the same disaster would not affect both locations. Examples of alternate disaster recovery sites being compromised by the same disaster that affected the primary site include having had a primary site in **World Trade Center I** and the recovery site in **7 World Trade Center**, both of which were destroyed in the **9/11** attack, and having one's primary site and recovery site in the same coastal region, which leads to both being vulnerable to hurricane damage (for example, primary site in New Orleans and recovery site in Jefferson Parish, both of which were hit by **Hurricane Katrina** in 2005). The backup media should be moved between the geographic sites in a secure manner, in order to prevent them from being stolen.
- Capability** and access control list techniques can be used to ensure privilege separation and mandatory access control. This section discusses their use.
- Chain of trust** techniques can be used to attempt to ensure that all software loaded has been certified as authentic by the system's designers.
- Confidentiality** is the nondisclosure of information except to another authorized person.^[121]
- Cryptographic** techniques can be used to defend data in transit between systems, reducing the probability that data exchanged between systems can be intercepted or modified.
- Cyberwarfare** is an Internet-based conflict that involves politically motivated attacks on information and information systems. Such attacks can, for example, disable official websites and networks, disrupt or disable essential services, steal or alter classified data, and cripple financial systems.
- Data integrity** is the accuracy and consistency of stored data, indicated by an absence of any alteration in data between two updates of a data record.^[122]
- Encryption** is used to protect the message from the eyes of others. **Cryptographically** secure ciphers are designed to make any practical attempt of breaking infeasible. **Symmetric-key** ciphers are suitable for bulk encryption using shared keys, and **public-key** encryption using **digital certificates** can provide a

This is secret stuff, PSE do not...

→ 5a0 (k\$hQ% ...

→ This is secret stuff, PSE do not...

Cryptographic techniques involve transforming information, scrambling it so it becomes unreadable during transmission. The intended recipient can unscramble the message; ideally, eavesdroppers cannot.

practical solution for the problem of securely communicating when no key is shared in advance.

- Endpoint security software helps networks to prevent exfiltration (data theft) and virus infection at network entry points made vulnerable by the prevalence of potentially infected portable computing devices, such as laptops and mobile devices, and external storage devices, such as USB drives.^[123]
- Firewalls** are an important method for control and security on the Internet and other networks. A network firewall can be a communications processor, typically a router, or a dedicated server, along with firewall software. A firewall serves as a gatekeeper system that protects a company's intranets and other computer networks from intrusion by providing a filter and safe transfer point for access to and from the Internet and other networks. It screens all network traffic for proper passwords or other security codes and only allows authorized transmission in and out of the network. Firewalls can deter, but not completely prevent, unauthorized access (hacking) into computer networks; they can also provide some protection from online intrusion.
- Honey pots** are computers that are either intentionally or unintentionally left vulnerable to attack by crackers. They can be used to catch crackers or fix vulnerabilities.
- Intrusion-detection systems** can scan a network for people that are on the network but who should not be there or are doing things that they should not be doing, for example trying a lot of passwords to gain access to the network.
- A **microkernel** is the near-minimum amount of software that can provide the mechanisms to implement an operating system. It is used solely to provide very low-level, very precisely defined machine code upon which an operating system can be developed. A simple example is the early '90s GEMSOS (Gemini Computers), which provided extremely low-level machine code, such as "segment" management, atop which an operating system could be built. The theory (in the case of "segments") was that—rather than have the operating system itself worry about

mandatory access separation by means of military-style labeling—it is safer if a low-level, independently scrutinized module can be charged **solely** with the management of individually labeled segments, be they memory “segments” or file system “segments” or executable text “segments.” If software below the visibility of the operating system is (as in this case) charged with labeling, there is no theoretically viable means for a clever hacker to subvert the labeling scheme, since the operating system *per se* does **not** provide mechanisms for interfering with labeling: the operating system is, essentially, a client (an “application,” arguably) atop the microkernel and, as such, subject to its restrictions.

- **Pinging** The ping application can be used by potential crackers to find if an IP address is reachable. If a cracker finds a computer, they can try a port scan to detect and attack services on that computer.
- Social engineering awareness keeps employees aware of the dangers of social engineering and/or having a policy in place to prevent social engineering can reduce successful breaches of the network and servers.

1.1.15 Scholars

- Salvatore J. Stolfo
- Ross J. Anderson
- Annie Anton
- Adam Back
- Daniel J. Bernstein
- Matt Blaze
- Stefan Brands
- L. Jean Camp
- Lance Cottrell
- Lorrie Cranor
- Dorothy E. Denning
- Peter J. Denning
- Cynthia Dwork
- Deborah Estrin
- Joan Feigenbaum
- Ian Goldberg
- Shafi Goldwasser
- Lawrence A. Gordon
- Peter Gutmann

- Paul Kocher
- Monica S. Lam
- Butler Lampson
- Brian LaMacchia
- Carl Landwehr
- Kevin Mitnick
- Peter G. Neumann
- Susan Nycum
- Roger R. Schell
- Bruce Schneier
- Dawn Song
- Gene Spafford
- Joseph Steinberg
- Willis Ware
- Moti Yung

1.1.16 See also

- Attack tree
- CAPTCHA
- CERT
- CertiVox
- Cloud computing security
- Comparison of antivirus software
- Computer insecurity
- Computer security model
- Content security
- Countermeasure (computer)
- Cyber security standards
- Dancing pigs
- Data loss prevention products
- Data security
- Differentiated security
- Disk encryption
- Exploit (computer security)
- Fault tolerance
- Human-computer interaction (security)

- Identity Based Security
- Identity management
- Identity theft
- Information Leak Prevention
- Information Security Awareness
- Internet privacy
- ISO/IEC 15408
- IT risk
- List of Computer Security Certifications
- Mobile security
- Network security
- Network Security Toolkit
- Next-Generation Firewall
- Open security
- OWASP
- Penetration test
- Physical information security
- Presumed security
- Privacy software
- Proactive Cyber Defence
- Risk cybernetics
- Sandbox (computer security)
- Separation of protection and security
- Software Defined Perimeter
- Cyber Insurance

1.1.17 Further reading

- Chwan-Hwa (John) Wu and J. David Irwin, *Introduction to Computer Networks and Cybersecurity* (Boca Raton: CRC Press, 2013), ISBN 978-1466572133.
- Newton Lee, *Counterterrorism and Cybersecurity: Total Information Awareness (Second Edition)* (Switzerland: Springer International Publishing, 2015), ISBN 978-3-319-17243-9.
- P. W. Singer and Allan Friedman, *Cybersecurity and Cyberwar: What Everyone Needs to Know* (Oxford: Oxford University Press, 2014), ISBN 978-0199918119.
- Peter Kim, *The Hacker Playbook: Practical Guide To Penetration Testing* (Seattle: CreateSpace Independent Publishing Platform, 2014), ISBN 978-1494932633.

1.1.18 References

- [1] Gasser, Morrie (1988). *Building a Secure Computer System* (PDF). Van Nostrand Reinhold. p. 3. ISBN 0-442-23022-2. Retrieved 6 September 2015.
- [2] “Definition of computer security”. *Encyclopedia*. Ziff Davis, PCMag. Retrieved 6 September 2015.
- [3] Rouse, Margaret. “Social engineering definition”. TechTarget. Retrieved 6 September 2015.
- [4] “Reliance spells end of road for ICT amateurs”, May 07, 2013, The Australian
- [5] <http://www.evollution.com/opinions/cybersecurity-understanding-online-threat/>
- [6] Gallagher, Sean (May 14, 2014). “Photos of an NSA “upgrade” factory show Cisco router getting implant”. Ars Technica. Retrieved August 3, 2014.
- [7] Arcos Sergio. “Social Engineering” (PDF).
- [8] J. C. Willemssen, “FAA Computer Security”. GAO/T-AIMD-00-330. Presented at Committee on Science, House of Representatives, 2000.
- [9] Pagliery, Jose. “Hackers attacked the U.S. energy grid 79 times this year”. *CNN Money*. Cable News Network. Retrieved 16 April 2015.
- [10] P. G. Neumann, “Computer Security in Aviation,” presented at International Conference on Aviation Safety and Security in the 21st Century, White House Commission on Safety and Security, 1997.
- [11] J. Zellan, *Aviation Security*. Hauppauge, NY: Nova Science, 2003, pp. 65–70.
- [12] <http://www.securityweek.com/air-traffic-control-systems-vulnerabilities-could-make-unfriendly-skies-blac>
- [13] <http://www.npr.org/blogs/alltechconsidered/2014/08/04/337794061/hacker-says-he-can-break-into-airplane-systems-using-in-flight-wi-fi>
- [14] <http://www.reuters.com/article/2014/08/04/us-cybersecurity-hackers-airplanes-idUSKBN0G40WQ20140804>
- [15] <http://www.npr.org/blogs/alltechconsidered/2014/08/06/338334508/is-your-watch-or-thermostat-a-spy-cyber-security-firms-are-on-it>
- [16] Melvin Backman (18 September 2014). “Home Depot: 56 million cards exposed in breach”. CNNMoney.
- [17] “Staples: Breach may have affected 1.16 million customers’ cards”. Fortune.com. December 19, 2014. Retrieved 2014-12-21.
- [18] “Target security breach affects up to 40M cards”. Associated Press via Milwaukee Journal Sentinel. 19 December 2013. Retrieved 21 December 2013.
- [19] Bright, Peter (February 15, 2011). “Anonymous speaks: the inside story of the HBGary hack”. ArsTechnica.com. Retrieved March 29, 2011.

- [20] Anderson, Nate (February 9, 2011). “How one man tracked down Anonymous—and paid a heavy price”. Arstechnica.com. Retrieved March 29, 2011.
- [21] Palilery, Jose (December 24, 2014). “What caused Sony hack: What we know now”. CNN Money. Retrieved January 4, 2015.
- [22] James Cook (December 16, 2014). “Sony Hackers Have Over 100 Terabytes Of Documents. Only Released 200 Gigabytes So Far”. *Business Insider*. Retrieved December 18, 2014.
- [23] [http://www.vox.com/2015/1/18/7629603/
car-hacking-dangers](http://www.vox.com/2015/1/18/7629603/car-hacking-dangers)
- [24] <http://www.autosec.org/pubs/cars-usenixsec2011.pdf>
- [25] [http://www.markey.senate.gov/imo/media/doc/
2015-02-06_MarkeyReport-Tracking_Hacking_
CarSecurity%202.pdf](http://www.markey.senate.gov/imo/media/doc/2015-02-06_MarkeyReport-Tracking_Hacking_CarSecurity%202.pdf)
- [26] “Internet strikes back: Anonymous’ Operation Megupload explained”. *RT*. January 20, 2012. Archived from the original on May 5, 2013. Retrieved May 5, 2013.
- [27] “Gary McKinnon profile: Autistic ‘hacker’ who started writing computer programs at 14”. *The Daily Telegraph* (London). 23 January 2009.
- [28] “Gary McKinnon extradition ruling due by 16 October”. BBC News. September 6, 2012. Retrieved September 25, 2012.
- [29] Law Lords Department (30 July 2008). “House of Lords – Mckinnon V Government of The United States of America and Another”. Publications.parliament.uk. Retrieved 30 January 2010. 15. ... alleged to total over \$700,000
- [30] “NSA Accessed Mexican President’s Email”, October 20, 2013, Jens Glüsing, Laura Poitras, Marcel Rosenbach and Holger Stark, spiegel.de
- [31] Sanders, Sam (4 June 2015). “Massive Data Breach Puts 4 Million Federal Employees’ Records At Risk”. *NPR*. Retrieved 5 June 2015.
- [32] Liptak, Kevin (4 June 2015). “U.S. government hacked; feds think China is the culprit”. *CNN*. Retrieved 5 June 2015.
- [33] Sean Gallagher. “Encryption “would not have helped” at OPM, says DHS official”.
- [34] Cashell, B., Jackson, W. D., Jickling, M., & Webel, B. (2004). The Economic Impact of Cyber-Attacks. Congressional Research Service, Government and Finance Division. Washington DC: The Library of Congress.
- [35] Gordon, Lawrence; Loeb, Martin (November 2002). “The Economics of Information Security Investment”. *ACM Transactions on Information and System Security* 5 (4): 438–457. doi:10.1145/581271.581274.
- [36] RFC 2828 Internet Security Glossary
- [37] CNSS Instruction No. 4009 dated 26 April 2010
- [38] InfosecToday Glossary
- [39] “Firms lose more to electronic than physical theft”. Reuters.
- [40] Harrison, J. (2003). “Formal verification at Intel”. pp. 45–54. doi:10.1109/LICS.2003.1210044.
- [41] Formal verification of a real-time hardware design. Portal.acm.org (1983-06-27). Retrieved on April 30, 2011.
- [42] “Abstract Formal Specification of the seL4/ARMv6 API” (PDF). Retrieved May 19, 2015.
- [43] Christoph Baumann, Bernhard Beckert, Holger Blasum, and Thorsten Bormer Ingredients of Operating System Correctness? Lessons Learned in the Formal Verification of PikeOS
- [44] “Getting it Right” by Jack Ganssle
- [45] Definitions: IT Security Architecture. SecurityArchitecture.org, Jan, 2006
- [46] Jannsen, Cory. “Security Architecture”. Techopedia. Janalta Interactive Inc. Retrieved 9 October 2014.
- [47] *The Hacker in Your Hardware: The Next Security Threat* August 4, 2010 Scientific American
- [48] Waksman, Adam; Sethumadhavan, Simha (2010), “Tamper Evident Microprocessors” (PDF), *Proceedings of the IEEE Symposium on Security and Privacy* (Oakland, California)
- [49] “Sentinel HASP HL”. E-Spin. Retrieved 2014-03-20.
- [50] “Token-based authentication”. SafeNet.com. Retrieved 2014-03-20.
- [51] “Lock and protect your Windows PC”. TheWindowsClub.com. Retrieved 2014-03-20.
- [52] James Greene (2012). “Intel Trusted Execution Technology: White Paper” (PDF). Intel Corporation. Retrieved 2013-12-18.
- [53] “SafeNet ProtectDrive 8.4”. SCMagazine.com. 2008-10-04. Retrieved 2014-03-20.
- [54] “Secure Hard Drives: Lock Down Your Data”. PC-Mag.com. 2009-05-11.
- [55] “Top 10 vulnerabilities inside the network”. Network World. 2010-11-08. Retrieved 2014-03-20.
- [56] “Forget IDs, use your phone as credentials”. Fox Business Network. 2013-11-04. Retrieved 2014-03-20.
- [57] “Secure Coding in C and C++, Second Edition”. Cert.org. Retrieved 2013-09-25.
- [58] New hacking technique exploits common programming error. SearchSecurity.com, July 2007
- [59] “Computer Security”. cyberlawsindia.net. Retrieved 2015-10-23.
- [60] Jonathan Zittrain, 'The Future of The Internet', Penguin Books, 2008

- [61] Information Security. United States Department of Defense, 1986
- [62] "THE TJX COMPANIES, INC. VICTIMIZED BY COMPUTER SYSTEMS INTRUSION; PROVIDES INFORMATION TO HELP PROTECT CUSTOMERS" (Press release). The TJX Companies, Inc. 2007-01-17. Retrieved 2009-12-12.
- [63] Largest Customer Info Breach Grows. MyFox Twin Cities, 29 March 2007.
- [64] "The Stuxnet Attack On Iran's Nuclear Plant Was 'Far More Dangerous' Than Previously Thought". Business Insider. 20 November 2013.
- [65] Reals, Tucker (24 September 2010). "Stuxnet Worm a U.S. Cyber-Attack on Iran Nukes?". CBS News.
- [66] Kim Zetter (17 February 2011). "Cyberwar Issues Likely to Be Addressed Only After a Catastrophe". Wired. Retrieved 18 February 2011.
- [67] Chris Carroll (18 October 2011). "Cone of silence surrounds U.S. cyberwarfare". Stars and Stripes. Retrieved 30 October 2011.
- [68] John Bumgarner (27 April 2010). "Computers as Weapons of War" (PDF). IO Journal. Retrieved 30 October 2011.
- [69] "Can You Trust NIST?".
- [70] "New Snowden Leak: NSA Taped Google, Yahoo Data Centers", Oct 31, 2013, Lorenzo Franceschi-Bicchieri, mashable.com
- [71] Seipel, Hubert. "Transcript: ARD interview with Edward Snowden". *La Foundation Courage*. Retrieved 11 June 2014.
- [72] "Missed Alarms and 40 Million Stolen Credit Card Numbers: How Target Blew It"
- [73] "Home Depot says 53 million emails stolen"
- [74] "Mikko Hypponen: Fighting viruses, defending the net". TED.
- [75] "Mikko Hypponen – Behind Enemy Lines". Hack In The Box Security Conference.
- [76] "Ensuring the Security of Federal Information Systems and Cyber Critical Infrastructure and Protecting the Privacy of Personally Identifiable Information". Government Accountability Office. Retrieved November 3, 2015.
- [77] Kirby, Carrie (June 24, 2011). "Former White House aide backs some Net regulation / Clarke says government, industry deserve 'F' in cybersecurity". *The San Francisco Chronicle*.
- [78] "Text of H.R.4962 as Introduced in House: International Cybercrime Reporting and Cooperation Act – U.S. Congress". OpenCongress. Retrieved 2013-09-25.
- [79] Archived July 4, 2015 at the Wayback Machine
- [80] "National Cyber Security Division". U.S. Department of Homeland Security. Retrieved June 14, 2008.
- [81] "FAQ: Cyber Security R&D Center". U.S. Department of Homeland Security S&T Directorate. Retrieved June 14, 2008.
- [82] AFP-JiJi, "U.S. boots up cybersecurity center", October 31, 2009.
- [83] "Federal Bureau of Investigation – Priorities". Federal Bureau of Investigation.
- [84] Internet Crime Complaint Center
- [85] "Infragard, Official Site". *Infragard*. Retrieved 10 September 2010.
- [86] "Robert S. Mueller, III -- InfraGard Interview at the 2005 InfraGard Conference". *Infragard (Official Site) -- "Media Room"*. Retrieved 9 December 2009.
- [87] "CCIPS".
- [88] U.S. Department of Defense, Cyber Command Fact Sheet, May 21, 2010 http://www.stratcom.mil/factsheets/Cyber_Command/
- [89] "Speech:". Defense.gov. Retrieved 2010-07-10.
- [90] Shachtman, Noah. "Military's Cyber Commander Swears: "No Role" in Civilian Networks", The Brookings Institution, 23 September 2010.
- [91] "FCC Cybersecurity". FCC.
- [92] Verton, Dan (January 28, 2004). "DHS launches national cyber alert system". *Computerworld* (IDG). Retrieved 2008-06-15.
- [93] "FIRST website".
- [94] "First members".
- [95] "European council".
- [96] "MAAWG".
- [97] "South Korea seeks global support in cyber attack probe". *BBC Monitoring Asia Pacific*. 7 March 2011.
- [98] Kwanwoo Jun (23 September 2013). "Seoul Puts a Price on Cyberdefense". *Wall Street Journal*. Dow Jones & Company, Inc. Retrieved 24 September 2013.
- [99] "Government of Canada Launches Canada's Cyber Security Strategy". *Market Wired*. 3 October 2010. Retrieved 1 November 2014.
- [100] "Canada's Cyber Security Strategy".
- [101] "Canada's Cyber Security Strategy". *Public Safety Canada*. Government of Canada. Retrieved 1 November 2014.
- [102] "Action Plan 2010–2015 for Canada's Cyber Security Strategy". *Public Safety Canada*. Government of Canada. Retrieved 3 November 2014.

- [103] “Cyber Incident Management Framework For Canada”. *Public Safety Canada*. Government of Canada. Retrieved 3 November 2014.
- [104] “Action Plan 2010–2015 for Canada’s Cyber Security Strategy”. *Public Safety Canada*. Government of Canada. Retrieved 1 November 2014.
- [105] “Canadian Cyber Incident Response Centre”. *Public Safety Canada*. Retrieved 1 November 2014.
- [106] “Cyber Security Bulletins”. *Public Safety Canada*. Retrieved 1 November 2014.
- [107] “Report a Cyber Security Incident”. *Public Safety Canada*. Government of Canada. Retrieved 3 November 2014.
- [108] “Government of Canada Launches Cyber Security Awareness Month With New Public Awareness Partnership”. *Market Wired* (Government of Canada). 27 September 2012. Retrieved 3 November 2014.
- [109] “Cyber Security Cooperation Program”. *Public Safety Canada*. Retrieved 1 November 2014.
- [110] “Cyber Security Cooperation Program”. *Public Safety Canada*.
- [111] “GetCyberSafe”. *Get Cyber Safe*. Government of Canada. Retrieved 3 November 2014.
- [112] Clayton, Mark. “The new cyber arms race”. *The Christian Science Monitor*. Retrieved 16 April 2015.
- [113] “The Growth of Cybersecurity Jobs”. Mar 2014. Retrieved 24 April 2014.
- [114] de Silva, Richard (11 Oct 2011). “Government vs. Commerce: The Cyber Security Industry and You (Part One)”. Defence IQ. Retrieved 24 Apr 2014.
- [115] “Department of Computer Science”. Retrieved April 30, 2013.
- [116] “(Information for) Students”. NICCS (US National Initiative for Cybercareers and Studies). Retrieved 24 April 2014.
- [117] “Current Job Opportunities at DHS”. U.S. Department of Homeland Security. Retrieved 2013-05-05.
- [118] “Cybersecurity Training & Exercises”. U.S. Department of Homeland Security. Retrieved 2015-01-09.
- [119] “Cyber Security Awareness Free Training and Webcasts”. MS-ISAC (Multi-State Information Sharing & Analysis Center. Retrieved 9 January 2015.
- [120] “Security Training Courses”. LearnQuest. Retrieved 2015-01-09.
- [121] “Confidentiality”. Retrieved 2011-10-31.
- [122] “Data Integrity”. Retrieved 2011-10-31.
- [123] “Endpoint Security”. Retrieved 2014-03-15.

1.1.19 External links

- Computer security at DMOZ

Chapter 2

Related security categories

2.1 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.

2.1.1 Types of security

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.^[3]

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.^[4] 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.^[5] 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.^[6]

2.1.2 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.

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.

2.1.3 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.

2.1.4 Denial-of-service attack

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.^[7]

2.1.5 Phishing

Main article: **Phishing**

Phishing is where 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.^{[8][9]} Insurance group **RSA** said that phishing accounted for worldwide losses of \$1.5 Billion in 2012.^[10]

2.1.6 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,^[11] is considered extremely insecure^[12] 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)^[11] and vulnerabilities are exploited in many different browsers.^{[13][14][15]}

2.1.7 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.

2.1.8 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.^[16]

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.^[17] 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^[18] as of at least 2012.

2.1.9 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)

2.1.10 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] Margaret Rouse (September 2005). "What is a security token?". SearchSecurity.com. Retrieved 2014-02-14.
- [4] "Virtual Private Network". NASA. Retrieved 2014-02-14.

- [5] Asgaut Eng (1996-04-10). "Network Virtual Terminal". The Norwegian Institute of Technology ppv.org. Retrieved 2014-02-14.
- [6] "What Is a Message Authentication Code?". Wisegeek.com. Retrieved 2013-04-20.
- [7] "Information Security: A Growing Need of Businesses and Industries Worldwide". University of Alabama at Birmingham Business Program. Retrieved 20 November 2014.
- [8] Ramzan, Zulfikar (2010). "Phishing attacks and countermeasures". In Stamp, Mark & Stavroulakis, Peter. *Handbook of Information and Communication Security*. Springer. ISBN 9783642041174.
- [9] 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.
- [10] "2012 Global Losses From Phishing Estimated At \$1.5 Bn". FirstPost. February 20, 2013. Retrieved December 21, 2014.
- [11] "Browser Statistics". W3Schools.com. Retrieved 2011-08-10.
- [12] Bradly, Tony. "It's Time to Finally Drop Internet Explorer 6". PCWorld.com. Retrieved 2010-11-09.
- [13] Messmer, Ellen and NetworkWorld (2010-11-16). "Google Chrome Tops 'Dirty Dozen' Vulnerable Apps List". PCWorld.com. Retrieved 2010-11-09.
- [14] Keizer, Greg (2009-07-15). "Firefox 35 vulnerability confirmed". PCWorld.com. Retrieved 2010-11-09.
- [15] Skinner, Carrie-Ann. "Opera Plugs "Severe" Browser Hole". PC World.com. Retrieved 2010-11-09.
- [16] Larkin, Eric (2008-08-26). "Build Your Own Free Security Suite". Retrieved 2010-11-09.
- [17] Rebbapragada, Narasu. "All-in-one Security". PC World.com. Retrieved 2010-11-09.
- [18] "Free products for PC security". 2015-10-08.

2.1.11 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)
- Massive spying by the US on many countries, revealed by Edward Snowden.
- After the NSA's spying on Germany's Chancellor Angela Merkel was revealed the Chancellor compared the NSA with the Stasi.^[13]
- NSA recording nearly every cell phone conversation in the Bahamas without the Bahamian government's permission,^[14] and similar programmes in Kenya, the Philippines, Mexico and Afghanistan.^[15]
- The "Titan Rain" probes of American defence contractors computer systems since 2003.^[16]
- The Office of Personnel Management data breach, in the US, widely attributed to China.^{[17][18]}

Sabotage

Computers and satellites that coordinate other activities are vulnerable components of a system and could lead to the disruption of equipment. Compromise of military systems, such as C4ISTAR components that are responsible for orders and communications could lead to their interception or malicious replacement. Power, water, fuel, communications, and transportation infrastructure all may be vulnerable to disruption. According to Clarke, the civilian realm is also at risk, noting that the security breaches have already gone beyond stolen credit card numbers, and that potential targets can also include the electric power grid, trains, or the stock market.^[19]

In mid July 2010, security experts discovered a malicious software program called Stuxnet that had infiltrated factory computers and had spread to plants around the world. It is considered "the first attack on critical industrial infrastructure that sits at the foundation of modern economies," notes *The New York Times*.^[20]

Denial-of-service attack Main article: Denial-of-service attack

In computing, a denial-of-service attack (DoS attack) or distributed denial-of-service attack (DDoS attack) is an attempt to make a machine or network resource unavailable to its intended users. Perpetrators of DoS attacks typically target sites or services hosted on high-profile web servers such as banks, credit card payment gateways, and even root nameservers. DoS attacks may not be limited to computer-based methods, as strategic physical attacks against infrastructure can be just as devastating. For example, cutting undersea communication cables may severely cripple some regions and countries with regards to their information warfare ability.

Electrical power grid The federal government of the United States admits that the electric power grid is sus-

2.2 Cyberwarfare

"Cyberwar" redirects here. For the video game, see [Cyberwar \(video game\)](#). For the 2004 movie also known as Cyber Wars, see [Avatar \(2004 film\)](#).

Not to be confused with Electronic warfare or software wars.

Cyberwarfare has been defined as "actions by a nation-state to penetrate another nation's computers or networks for the purposes of causing damage or disruption,"^{[1]:6} but other definitions also include non-state actors, such as terrorist groups, companies, political or ideological extremist groups, hacktivists, and transnational criminal organizations.^{[2][3][4][5][6]}

Some governments have made it an integral part of their overall military strategy, with some having invested heavily in cyberwarfare capability.^{[7][8][9][10]}

2.2.1 Types of threat

Cyberattacks, where damage or disruption is caused are the main concern.^[11]

Cyber espionage, which can provide the information needed to launch a successful attack.

Espionage

Traditional espionage is not an act of war, nor is cyber-espionage,^[12] and both are generally assumed to be ongoing between major powers.

Despite this assumption, some incidents can cause serious tensions between nations, and are often described as "attacks". For example:

ceptible to cyberwarfare.^{[21][22]} The United States Department of Homeland Security works with industry to identify vulnerabilities and to help industry enhance the security of control system networks, the federal government is also working to ensure that security is built in as the next generation of “smart grid” networks are developed.^[23] In April 2009, reports surfaced that China and Russia had infiltrated the U.S. electrical grid and left behind software programs that could be used to disrupt the system, according to current and former national security officials.^[24] The North American Electric Reliability Corporation (NERC) has issued a public notice that warns that the electrical grid is not adequately protected from cyber attack.^[25] China denies intruding into the U.S. electrical grid.^{[26][27]} One countermeasure would be to disconnect the power grid from the Internet and run the net with droop speed control only.^{[28][29]} Massive power outages caused by a cyber attack could disrupt the economy, distract from a simultaneous military attack, or create a national trauma.

Howard Schmidt, former Cyber-Security Coordinator of the US, commented on those possibilities:^[30]

It's possible that hackers have gotten into administrative computer systems of utility companies, but says those aren't linked to the equipment controlling the grid, at least not in developed countries. [Schmidt] has never heard that the grid itself has been hacked.

2.2.2 Motivations

Military

In the U.S., General Keith B. Alexander, first head of the recently formed USCYBERCOM, told the Senate Armed Services Committee that computer network warfare is evolving so rapidly that there is a “mismatch between our technical capabilities to conduct operations and the governing laws and policies. Cyber Command is the newest global combatant and its sole mission is cyberspace, outside the traditional battlefields of land, sea, air and space.” It will attempt to find and, when necessary, neutralize cyberattacks and to defend military computer networks.^[31]

Alexander sketched out the broad battlefield envisioned for the computer warfare command, listing the kind of targets that his new headquarters could be ordered to attack, including “traditional battlefield prizes – command-and-control systems at military headquarters, air defense networks and weapons systems that require computers to operate.”^[31]

One cyber warfare scenario, *Cyber ShockWave*, which was wargamed on the cabinet level by former administration officials, raised issues ranging from the National Guard to the power grid to the limits of statutory authority.^{[32][33][34][35]}

The distributed nature of internet based attacks means that it is difficult to determine motivation and attacking party, meaning that it is unclear when a specific act should be considered an act of war.^[36]

Examples of cyberwarfare driven by political motivations can be found worldwide. In 2008, Russia began a cyber attack on the Georgian government website, which was carried out along with Georgian military operations in South Ossetia. In 2008, Chinese 'nationalist hackers' attacked CNN as it reported on Chinese repression on Tibet.^[37]

Jobs in cyberwarfare have become increasingly popular in the military. The United States Navy actively recruits for cyber warfare engineers.^[38] The US Army has their Cyber Command where they actively recruit for cryptologic network warfare specialists.

Civil

Potential targets in internet sabotage include all aspects of the Internet from the **backbones** of the web, to the **Internet Service Providers**, to the varying types of data communication mediums and network equipment. This would include: web servers, enterprise information systems, client server systems, communication links, network equipment, and the desktops and laptops in businesses and homes. Electrical grids and telecommunication systems are also deemed vulnerable, especially due to current trends in automation.

Hacktivism

Politically motivated hacktivism, involves the subversive use of **computers** and **computer networks** to promote an agenda, and can potential extend to attacks, theft and virtual sabotage that could be seen as cyberwarfare - or mistaken for it.^[39]

Private sector

Computer hacking represents a modern threat in ongoing **industrial espionage** and as such is presumed to widely occur. It is typical that this type of crime is underreported. According to McAfee's George Kurtz, corporations around the world face millions of cyberattacks a day. “Most of these **attacks** don't gain any media attention or lead to strong political statements by victims.”^[40] This type of crime is usually financially motivated.

Non-profit research

But not all examinations with the issue of cyberwarfare are achieving profit or personal gain. There are still institutes and companies like the University of Cincinnati or the Kaspersky Security Lab which are trying to increase

the sensibility of this topic by researching and publishing of new security threats.

2.2.3 Cyberwarfare by country

The Internet security company **McAfee** stated in their 2007 annual report that approximately 120 countries have been developing ways to use the Internet as a weapon and target financial markets, government computer systems and utilities.^[41]

Cyberwarfare in China

Main article: Cyberwarfare in the People's Republic of China

See also: Chinese intelligence activity abroad, Chinese intelligence operations in the United States and Chinese Information Operations and Information Warfare

Diplomatic cables highlight US concerns that China is using access to Microsoft source code and 'harvesting the talents of its private sector' to boost its offensive and defensive capabilities.^[42]

A 2008 article in the *Culture Mandala: The Bulletin of the Centre for East-West Cultural and Economic Studies* by Jason Fritz alleges that the Chinese government from 1995 to 2008 was involved in a number of high-profile cases of espionage, primarily through the use of a "decentralized network of students, business people, scientists, diplomats, and engineers from within the Chinese Diaspora".^[43] A defector in Belgium, purportedly an agent, claimed that there were hundreds of spies in industries throughout Europe, and on his defection to Australia Chinese diplomat Chen Yonglin said there were over 1,000 such in that country. In 2007, a Russian executive was sentenced to 11 years for passing information about the rocket and space technology organization to China. Targets in the United States have included 'aerospace engineering programs, space shuttle design, C4ISR data, high-performance computers, Nuclear weapon design, cruise missile data, semiconductors, integrated circuit design, and details of US arms sales to Taiwan'.^[43]

While China continues to be held responsible for a string of cyber-attacks on a number of public and private institutions in the United States, India, Russia, Canada, and France, the Chinese government denies any involvement in cyber-spying campaigns. The administration maintains the position that China is not the threat but rather the victim of an increasing number of cyber-attacks. Most reports about China's cyber warfare capabilities have yet to be confirmed by the Chinese government.^[44]

According to Fritz, China has expanded its cyber capabilities and military technology by acquiring foreign military technology.^[45] Fritz states that the Chinese government uses "new space-based surveillance and intelligence

gathering systems, Anti-satellite weapon, anti-radar, infrared decoys, and false target generators" to assist in this quest, and that they support their "informationization" of the their military through "increased education of soldiers in cyber warfare; improving the information network for military training, and has built more virtual laboratories, digital libraries and digital campuses."^[45] Through this informationization, they hope to prepare their forces to engage in a different kind of warfare, against technically capable adversaries.^[46] Many recent news reports link China's technological capabilities to the beginning of a new 'cyber cold war'.^[47]

In response to reports of cyberattacks by China against the United States, Amitai Etzioni of the Institute for Communitarian Policy Studies has suggested that China and the United States agree to a policy of mutually assured restraint with respect to cyberspace. This would involve allowing both states to take the measures they deem necessary for their self-defense while simultaneously agreeing to refrain from taking offensive steps; it would also entail vetting these commitments.^[48]

Cyberwarfare in Germany

In 2013, Germany revealed the existence of their 60-person Computer Network Operation unit.^[49] The German intelligence agency, **BND**, announced it was seeking to hire 130 "hackers" for a new "cyber defence station" unit. In March 2013, BND president Gerhard Schindler announced that his agency had observed up to five attacks a day on government authorities, thought mainly to originate in China. He confirmed the attackers had so far only accessed data and expressed concern that the stolen information could be used as the basis of future sabotage attacks against arms manufacturers, telecommunications companies and government and military agencies.^[50] Shortly after Edward Snowden leaked details of the U.S. **National Security Agency**'s cyber surveillance system, German Interior Minister Hans-Peter Friedrich announced that the BND would be given an additional budget of 100 million Euros to increase their cyber surveillance capability from 5% of total internet traffic in Germany to 20% of total traffic, the maximum amount allowed by German law.^[51]

Cyberwarfare in India

See also: National Cyber Security Policy 2013

The Department of Information Technology created the **Indian Computer Emergency Response Team (CERT-In)** in 2004 to thwart cyber attacks in India.^[52] That year, there were 23 reported cyber security breaches. In 2011, there were 13,301. That year, the government created a new subdivision, the **National Critical Information Infrastructure Protection Centre (NCIIPC)** to thwart at-

tacks against energy, transport, banking, telecom, defence, space and other sensitive areas.

The Executive Director of the Nuclear Power Corporation of India (NPCIL) stated in February 2013 that his company alone was forced to block up to ten targeted attacks a day. CERT-In was left to protect less critical sectors.

A high-profile cyber attack on 12 July 2012 breached the email accounts of about 12,000 people, including those of officials from the Ministry of External Affairs, Ministry of Home Affairs, Defence Research and Development Organisation (DRDO), and the Indo-Tibetan Border Police (ITBP).^[52] A government-private sector plan being overseen by National Security Advisor (NSA) Shivshankar Menon began in October 2012, and intends to beef up India's cyber security capabilities in the light of a group of experts findings that India faces a 470,000 shortfall of such experts despite the country's reputation of being an IT and software powerhouse.^[53]

In February 2013, Information Technology Secretary J. Satyanarayana stated that the NCIIPC was finalizing policies related to national cyber security that would focus on domestic security solutions, reducing exposure through foreign technology.^[52] Other steps include the isolation of various security agencies to ensure that a synchronised attack could not succeed on all fronts and the planned appointment of a National Cyber Security Coordinator. As of that month, there had been no significant economic or physical damage to India related to cyber attacks.

Cyberwarfare in Iran

Main article: Cyberwarfare in Iran

See also: Iranian Cyber Army

Further information: Operation Olympic Games, Operation Ababil, Operation Cleaver and Operation Newscaster

Iran has been both victim and predator of several cyberwarfare operations. Iran is considered an emerging military power in the field.^[54]

Cyberwarfare in South Korea

Main article: 2013 South Korea cyberattack

With ongoing tensions on the Korean Peninsula, South Korea's defense ministry stated that South Korea was going to improve cyber-defense strategies in hopes of preparing itself from possible cyber attacks. In March 2013, South Korea's major banks – Shinhan Bank, Woori Bank and NongHyup Bank – as well as many broadcasting stations – KBS, YTN and MBC – were hacked and more than 30,000 computers were affected; it is one of the biggest attacks South Korea has faced in years.^[55] Al-

though it remains uncertain as to who was involved in this incident, there has been immediate assertions that North Korea is connected, as it threatened to attack South Korea's government institutions, major national banks and traditional newspapers numerous times – in reaction to the sanctions it received from nuclear testing and to the continuation of Foal Eagle, South Korea's annual joint military exercise with the United States. North Korea's cyber warfare capabilities raise the alarm for South Korea, as North Korea is increasing its manpower through military academies specializing in hacking. Current figures state that South Korea only has 400 units of specialized personnel, while North Korea has more than 3,000 highly trained hackers; this portrays a huge gap in cyber warfare capabilities and sends a message to South Korea that it has to step up and strengthen its Cyber Warfare Command forces. Therefore, in order to be prepared from future attacks, South Korea and the United States will discuss further about deterrence plans at the Security Consultative Meeting (SCM). At SCM, they plan on developing strategies that focuses on accelerating the deployment of ballistic missiles as well as fostering its defense shield program, known as the Korean Air and Missile Defense.^[56]

Cyberwarfare in the Netherlands

In the Netherlands Cyber Defense is nationally coordinated by the National Cyber Security Center (NCSC).^[57] The Dutch Ministry of Defense laid out a cyber strategy in 2011.^[58] The first focus is to improve the cyber defense handled by the Joint IT branch (JIVC). To improve intel operations the intel community in the Netherlands (including the military intel organization MIVD) has set up the Joint Sigint Cyber Unit (JSCU). The ministry of Defense is furthermore setting up an offensive cyber force, called Defensie Cyber Command (DCC),^[59] which will be operational in the end of 2014.

Cyberwarfare in Russia

Main article: Cyberwarfare in Russia

Cyberwarfare in the UK

MI6 reportedly infiltrated an Al Qaeda website and replaced the recipe for a pipe bomb with the recipe for making cupcakes.^[60]

On 12 November 2013, financial organisations in London conducted cyber war games dubbed 'Waking Shark 2'^[61] to simulate massive internet-based attacks against bank and other financial organisations. The Waking Shark 2 cyber war games followed a similar exercise in Wall Street.^[62]

Cyberwarfare in the United States

Main article: Cyberwarfare in the United States

Cyberwarfare in the United States is a part of the American military strategy of proactive cyber defence and the use of cyberwarfare as a platform for attack.^[63] The new United States military strategy makes explicit that a cyberattack is *casus belli* just as a traditional act of war.^[64]

In 2013 Cyberwarfare was, for the first time, considered a larger threat than Al Qaeda or terrorism, by many U.S. intelligence officials.^[65] Representative Mike Rogers, chairman of the U.S. House Permanent Select Committee on Intelligence, for instance, said in late July 2013, that "most Americans" do not realize that the United States is currently in the middle of a "cyber war."^[66]

U.S. government security expert Richard A. Clarke, in his book *Cyber War* (May 2010), defines "cyberwarfare" as "actions by a nation-state to penetrate another nation's computers or networks for the purposes of causing damage or disruption."^{[1]:6} *The Economist* describes cyberspace as "the fifth domain of warfare,"^[67] and William J. Lynn, U.S. Deputy Secretary of Defense, states that "as a doctrinal matter, the Pentagon has formally recognized cyberspace as a new domain in warfare . . . [which] has become just as critical to military operations as land, sea, air, and space."^[7]

In 2009, President Barack Obama declared America's digital infrastructure to be a "strategic national asset," and in May 2010 the Pentagon set up its new U.S. Cyber Command (USCYBERCOM), headed by General Keith B. Alexander, director of the National Security Agency (NSA), to defend American military networks and attack other countries' systems. The EU has set up ENISA (European Union Agency for Network and Information Security) which is headed by Prof. Udo Helmbrecht and there are now further plans to significantly expand ENISA's capabilities. The United Kingdom has also set up a cyber-security and "operations centre" based in Government Communications Headquarters (GCHQ), the British equivalent of the NSA. In the U.S. however, Cyber Command is only set up to protect the military, whereas the government and corporate infrastructures are primarily the responsibility respectively of the Department of Homeland Security and private companies.^[67]

In February 2010, top American lawmakers warned that the "threat of a crippling attack on telecommunications and computer networks was sharply on the rise."^[68] According to The Lipman Report, numerous key sectors of the U.S. economy along with that of other nations, are currently at risk, including cyber threats to public and private facilities, banking and finance, transportation, manufacturing, medical, education and government, all of which are now dependent on computers for daily

operations.^[68] In 2009, President Obama stated that "cyber intruders have probed our electrical grids."^[69]

The Economist writes that China has plans of "winning informationised wars by the mid-21st century". They note that other countries are likewise organizing for cyberwar, among them Russia, Israel and North Korea. Iran boasts of having the world's second-largest cyber-army.^[67] James Gosler, a government cybersecurity specialist, worries that the U.S. has a severe shortage of computer security specialists, estimating that there are only about 1,000 qualified people in the country today, but needs a force of 20,000 to 30,000 skilled experts.^[70] At the July 2010 Black Hat computer security conference, Michael Hayden, former deputy director of national intelligence, challenged thousands of attendees to help devise ways to "reshape the Internet's security architecture", explaining, "You guys made the cyberworld look like the north German plain."^[71]

In January 2012, Mike McConnell, the former director of national intelligence at the National Security Agency under President George W. Bush told the Reuters news agency that the U.S. has already launched attacks on computer networks in other countries.^[72] McConnell did not name the country that the U.S. attacked but according to other sources it may have been Iran.^[72] In June 2012 *the New York Times* reported that President Obama had ordered the cyber attack on Iranian nuclear enrichment facilities.^[73]

In August 2010, the U.S. for the first time warned publicly about the Chinese military's use of civilian computer experts in clandestine cyber attacks aimed at American companies and government agencies. The Pentagon also pointed to an alleged China-based computer spying network dubbed GhostNet that was revealed in a research report last year.^[74] The Pentagon stated:

"The People's Liberation Army is using "information warfare units" to develop viruses to attack enemy computer systems and networks, and those units include civilian computer professionals. Commander Bob Mehal, will monitor the PLA's buildup of its cyberwarfare capabilities and will continue to develop capabilities to counter any potential threat."^[75]

The United States Department of Defense sees the use of computers and the Internet to conduct warfare in cyberspace as a threat to national security. The United States Joint Forces Command describes some of its attributes:

Cyberspace technology is emerging as an "instrument of power" in societies, and is becoming more available to a country's opponents, who may use it to attack, degrade, and disrupt communications and the flow of information. With low barriers to entry, coupled

with the anonymous nature of activities in cyberspace, the list of potential adversaries is broad. Furthermore, the globe-spanning range of cyberspace and its disregard for national borders will challenge legal systems and complicate a nation's ability to deter threats and respond to contingencies.^[76]

In February 2010, the **United States Joint Forces Command** released a study which included a summary of the threats posed by the internet.^[76]

With very little investment, and cloaked in a veil of anonymity, our adversaries will inevitably attempt to harm our national interests. Cyberspace will become a main front in both irregular and traditional conflicts. Enemies in cyberspace will include both states and non-states and will range from the unsophisticated amateur to highly trained professional hackers. Through cyberspace, enemies will target industry, academia, government, as well as the military in the air, land, maritime, and space domains. In much the same way that airpower transformed the battlefield of World War II, cyberspace has fractured the physical barriers that shield a nation from attacks on its commerce and communication. Indeed, adversaries have already taken advantage of computer networks and the power of information technology not only to plan and execute savage acts of terrorism, but also to influence directly the perceptions and will of the U.S. Government and the American population.

In 24 November 2014. The Sony hack was a release of confidential data belonging to Sony Pictures Entertainment (SPE)

American "*Kill switch bill*"

On 19 June 2010, United States Senator Joe Lieberman (I-CT) introduced a bill called "Protecting Cyberspace as a National Asset Act of 2010",^[77] which he co-wrote with Senator Susan Collins (R-ME) and Senator Thomas Carper (D-DE). If signed into law, this controversial bill, which the American media dubbed the "*Kill switch bill*", would grant the President emergency powers over parts of the Internet. However, all three co-authors of the bill issued a statement that instead, the bill "[narrowed] existing broad Presidential authority to take over telecommunications networks".^[78]

The United States has used cyberattacks for tactical advantage in Afghanistan.^[79]

2.2.4 Cyber counterintelligence

Cyber counter-intelligence are measures to identify, penetrate, or neutralize foreign operations that use cyber means as the primary tradecraft methodology, as well as foreign intelligence service collection efforts that use traditional methods to gauge cyber capabilities and intentions.^[80]

- On 7 April 2009, **The Pentagon** announced they spent more than \$100 million in the last six months responding to and repairing damage from cyber attacks and other computer network problems.^[81]
- On 1 April 2009, U.S. lawmakers pushed for the appointment of a White House cyber security "czar" to dramatically escalate U.S. defenses against cyber attacks, crafting proposals that would empower the government to set and enforce security standards for private industry for the first time.^[82]
- On 9 February 2009, the **White House** announced that it will conduct a review of the nation's cyber security to ensure that the **Federal government of the United States** cyber security initiatives are appropriately integrated, resourced and coordinated with the **United States Congress** and the private sector.^[83]
- In the wake of the **2007 cyberwar** waged against Estonia, **NATO** established the **Cooperative Cyber Defence Centre of Excellence (CCD CoE)** in Tallinn, Estonia, in order to enhance the organization's cyber defence capability. The center was formally established on 14 May 2008, and it received full accreditation by NATO and attained the status of International Military Organization on 28 October 2008.^[84] Since **Estonia** has led international efforts to fight cybercrime, the **United States Federal Bureau of Investigation** says it will permanently base a computer crime expert in Estonia in 2009 to help fight international threats against computer systems.^[85]

One of the hardest issues in cyber counterintelligence is the problem of "Attribution". Unlike conventional warfare, figuring out who is behind an attack can be very difficult.^[86] However Defense Secretary Leon Panetta has claimed that the United States has the capability to trace attacks back to their sources and hold the attackers "accountable".^[87]

In 2015, the Department of Defense released an updated cyber strategy memorandum detailing the present and future tactics deployed in the service of defense against cyberwarfare. In this memorandum, three cybermissions are laid out. The first cybermission seeks to arm and maintain existing capabilities in the area of cyberspace, the second cybermission focuses on prevention of cyberwarfare, and the third cybermission includes strate-

gies for retaliation and preemption (as distinguished from prevention).^[91]

2.2.5 Controversy over terms

There is debate on whether the term “cyberwarfare” is accurate.

Eugene Kaspersky, founder of Kaspersky Lab, concludes that "cyberterrorism" is a more accurate term than "cyberwar." He states that "with today's attacks, you are clueless about who did it or when they will strike again. It's not cyber-war, but cyberterrorism."^[88] He also equates large-scale **cyber weapons**, such as **Flame** and **NetTraveler** which his company discovered, to **biological weapons**, claiming that in an interconnected world, they have the potential to be equally destructive.^{[88][89]}

In October 2011 the *Journal of Strategic Studies*, a leading journal in that field, published an article by Thomas Rid, “Cyber War Will Not Take Place” which argued that all politically motivated cyber attacks are merely sophisticated versions of sabotage, espionage, or subversion^[90] - and that it is unlikely that cyber war will occur in the future.

Howard Schmidt, an American cybersecurity expert, argued in March 2010 that “there is no cyberwar... I think that is a terrible metaphor and I think that is a terrible concept. There are no winners in that environment.”^[90]

Other experts, however, believe that this type of activity already constitutes a war. The warfare analogy is often seen intended to motivate a militaristic response when that is not necessarily appropriate. Ron Deibert, of Canada’s Citizen Lab, has warned of a “militarization of cyberspace.”^[91]

The European cybersecurity expert Sandro Gaycken argued for a middle position. He considers cyberwar from a legal perspective an unlikely scenario, due to the reasons lined out by Rid (and, before him, Sommer),^[92] but the situation looks different from a strategic point of view. States have to consider military-led cyber operations an attractive activity, within and without war, as they offer a large variety of cheap and risk-free options to weaken other countries and strengthen their own positions. Considered from a long-term, geostrategic perspective, cyber offensive operations can cripple whole economies, change political views, agitate conflicts within or among states, reduce their military efficiency and equalize the capacities of high-tech nations to that of low-tech nations, and use access to their critical infrastructures to blackmail them.^[93]

2.2.6 Incidents

- On 21 November 2011, it was widely reported in the U.S. media that a hacker had destroyed a water

pump at the Curran-Gardner Township Public Water District in Illinois.^[94] However, it later turned out that this information was not only false, but had been inappropriately leaked from the Illinois Statewide Terrorism and Intelligence Center.^[95]

- On 6 October 2011, it was announced that Creech AFB's **drone** and Predator fleet's command and control data stream had been **keylogged**, resisting all attempts to reverse the exploit, for the past two weeks.^[96] The Air Force issued a statement that the virus had “posed no threat to our operational mission”.^[97]
- In July 2011, the South Korean company SK Communications was hacked, resulting in the theft of the personal details (including names, phone numbers, home and email addresses and resident registration numbers) of up to 35 million people. A trojaned software update was used to gain access to the SK Communications network. Links exist between this hack and other malicious activity and it is believed to be part of a broader, concerted hacking effort.^[98]
- Operation Shady RAT** is an ongoing series of cyber attacks starting mid-2006, reported by Internet security company McAfee in August 2011. The attacks have hit at least 72 organizations including governments and defense contractors.^[99]
- On 4 December 2010, a group calling itself the **Pakistan Cyber Army** hacked the website of India’s top investigating agency, the Central Bureau of Investigation (CBI). The **National Informatics Center** (NIC) has begun an inquiry.^[100]
- On 26 November 2010, a group calling itself the **Indian Cyber Army** hacked the websites belonging to the Pakistan Army and the others belong to different ministries, including the Ministry of Foreign Affairs, Ministry of Education, Ministry of Finance, Pakistan Computer Bureau, Council of Islamic Ideology, etc. The attack was done as a revenge for the Mumbai terrorist attacks.^[101]
- In October 2010, Iain Lobban, the director of the **Government Communications Headquarters** (GCHQ), said Britain faces a “real and credible” threat from cyber attacks by hostile states and criminals and government systems are targeted 1,000 times each month, such attacks threatened Britain’s economic future, and some countries were already using cyber assaults to put pressure on other nations.^[102]
- In September 2010, **Iran** was attacked by the Stuxnet worm, thought to specifically target its **Natanz nuclear enrichment facility**. The worm is said to be the most advanced piece of malware ever discovered and significantly increases the profile of cyberwarfare.^{[103][104]}

- In July 2009, there were a series of coordinated denial of service attacks against major government, news media, and financial websites in South Korea and the United States.^[105] While many thought the attack was directed by North Korea, one researcher traced the attacks to the United Kingdom.^[106]
- Russian, South Ossetian, Georgian and Azerbaijani sites were attacked by hackers during the 2008 South Ossetia War.^[107]
- In 2007 the website of the Kyrgyz Central Election Commission was defaced during its election. The message left on the website read “This site has been hacked by Dream of Estonian organization”. During the election campaigns and riots preceding the election, there were cases of Denial-of-service attacks against the Kyrgyz ISPs.^[108]
- In September 2007, Israel carried out an airstrike on Syria dubbed Operation Orchard. U.S. industry and military sources speculated that the Israelis may have used cyberwarfare to allow their planes to pass undetected by radar into Syria.^{[109][110]}
- In April 2007, Estonia came under cyber attack in the wake of relocation of the Bronze Soldier of Tallinn.^[111] The largest part of the attacks were coming from Russia and from official servers of the authorities of Russia.^[112] In the attack, ministries, banks, and media were targeted.^{[113][114]} This attack on Estonia, a seemingly small Baltic nation, was so effective because of how most of the nation is run online. Estonia has implemented an e-government, where bank services, political elections and taxes are all done online. This attack really hurt Estonia’s economy and the people of Estonia. At least 150 people were injured on the first day due to riots in the streets.^[115]
- In the 2006 war against Hezbollah, Israel alleges that cyber-warfare was part of the conflict, where the Israel Defense Forces (IDF) intelligence estimates several countries in the Middle East used Russian hackers and scientists to operate on their behalf. As a result, Israel attached growing importance to cyber-tactics, and became, along with the U.S., France and a couple of other nations, involved in cyber-war planning. Many international high-tech companies are now locating research and development operations in Israel, where local hires are often veterans of the IDF’s elite computer units.^[116] Richard A. Clarke adds that “our Israeli friends have learned a thing or two from the programs we have been working on for more than two decades.”^{[1]:8}

It should also be worthy to note that a Spy Drone was shot down by a team Egyptian White/Grey Hat Hackers near the Libyan borders, which was later delivered to the Egyptian government.

2.2.7 Legality, rules

Various parties have attempted to come up with international legal frameworks to clarify what is and is not acceptable, but none have yet to be widely accepted.

The **Tallinn Manual**, published in 2013, is an academic, non-binding study on how international law, in particular the *jus ad bellum* and *international humanitarian law*, apply to cyber conflicts and *cyber warfare*. It was written at the invitation of the **Tallinn-based NATO Cooperative Cyber Defence Centre of Excellence** by an international group of approximately twenty experts between 2009 and 2012.

The **Shanghai Cooperation Organisation** (members of which include China and Russia) defines cyberwar to include dissemination of information “harmful to the spiritual, moral and cultural spheres of other states”. In September 2011, these countries proposed to the UN Secretary General a document called “International code of conduct for information security”.^[117]

In contrast, the United States’ approach focuses on physical and economic damage and injury, putting political concerns under *freedom of speech*. This difference of opinion has led to reluctance in the West to pursue global cyber arms control agreements.^[118] However, American General Keith B. Alexander did endorse talks with Russia over a proposal to limit military attacks in cyberspace.^[119] In June 2013, Barack Obama and **Vladimir Putin** agreed to install a secure *Cyberwar-Hotline* providing “a direct secure voice communications line between the US cybersecurity coordinator and the Russian deputy secretary of the security council, should there be a need to directly manage a crisis situation arising from an **ICT** security incident.” (White House quote)^[120]

A Ukrainian professor of International Law, Alexander Merezhko, has developed a project called the International Convention on Prohibition of Cyberwar in Internet. According to this project, cyberwar is defined as the use of Internet and related technological means by one state against political, economic, technological and information sovereignty and independence of any other state. Professor Merezhko’s project suggests that the Internet ought to remain free from warfare tactics and be treated as an international landmark. He states that the Internet (cyberspace) is a “common heritage of mankind.”^[121]

2.2.8 See also

- Air Force Cyber Command (Provisional)
- Automated teller machine#Device operation integrity
- Cyber-arms industry
- Cyber-collection
- Cyber spying

- IT risk
- iWar
- List of cyber-attacks
- List of cyber attack threat trends
- Penetration test
- Signals intelligence
- Duqu
- United States Cyber Command
 - Air Force Cyber Command
 - United States Army Cyber Command
 - Fleet Cyber Command
 - Marine Corps Cyberspace Command
- Fifth Dimension Operations

2.2.9 Further reading

- Andress, Jason. Winterfeld, Steve. (2011). *Cyber Warfare: Techniques, Tactics and Tools for Security Practitioners*. Syngress. ISBN 1-59749-637-5
- Brenner, S. (2009). *Cyber Threats: The Emerging Fault Lines of the Nation State*. Oxford University Press. ISBN 0-19-538501-2
- Carr, Jeffrey. (2010). *Inside Cyber Warfare: Mapping the Cyber Underworld*. O'Reilly. ISBN 978-0-596-80215-8
- Cordesman, Anthony H., Cordesman, Justin G. *Cyber-threats, Information Warfare, and Critical Infrastructure Protection*, Greenwood Publ. (2002)
- Costigan, Sean S.; Perry, Jake (2012). *Cyberspaces and global affairs*. Farnham, Surrey: Ashgate. ISBN 9781409427544.
- Gaycken, Sandro. (2012). *Cyberwar – Das Wettrüsten hat längst begonnen*. Goldmann/Randomhouse. ISBN 978-3442157105
- Geers, Kenneth. (2011). *Strategic Cyber Security*. NATO Cyber Centre. *Strategic Cyber Security*, ISBN 978-9949-9040-7-5, 169 pages
- Shane Harris (2014). *@War: The Rise of the Military-Internet Complex*. Eamon Dolan/Houghton Mifflin Harcourt. ISBN 978-0544251793.
- Hunt, Edward (2012). “US Government Computer Penetration Programs and the Implications for Cyberwar”. *IEEE Annals of the History of Computing* 34 (3): 4–21. doi:10.1109/mahc.2011.82.

- Janczewski, Lech; Colarik, Andrew M. *Cyber Warfare and Cyber Terrorism* IGI Global (2008)
- Rid, Thomas (2011) “Cyber War Will Not Take Place,” *Journal of Strategic Studies*, doi:10.1080/01402390.2011.608939
- Ventre, D. (2007). *La guerre de l'information*. Hermes-Lavoisier. 300 pages
- Ventre, D. (2009). *Information Warfare*. Wiley – ISTE. ISBN 978-1-84821-094-3
- Ventre, D. (Edit.) (2010). *Cyberguerre et guerre de l'information. Stratégies, règles, enjeux*. Hermes-Lavoisier. ISBN 978-2-7462-3004-0
- Ventre, D. (2011). *Cyberespace et acteurs du cyber-conflit*. Hermes-Lavoisier. 288 pages
- Ventre, D. (Edit.) (2011). *Cyberwar and Information Warfare*. Wiley. 460 pages
- Ventre, D. (2011). *Cyberattaque et Cyberdéfense*. Hermes-Lavoisier. 336 pages
- Ventre, D. (Edit.) (2012). *Cyber Conflict. Competing National Perspectives*. Wiley-ISTE. 330 pages
- Woltag, Johann-Christoph: 'Cyber Warfare' in Rüdiger Wolfrum (Ed.) *Max Planck Encyclopedia of Public International Law* (Oxford University Press 2012).

2.2.10 References

- [1] Clarke, Richard A. *Cyber War*, HarperCollins (2010) ISBN 9780061962233
- [2] Blitz, James (1 November 2011). “Security: A huge challenge from China, Russia and organised crime”. Financial Times. Retrieved 6 June 2015.
- [3] Arquilla, John (1999). “Can information warfare ever be just?”. *Ethics and Information Technology* 1 (3): 203–212. doi:10.1023/A:1010066528521.
- [4] Collins, Sean (April 2012). “Stuxnet: the emergence of a new cyber weapon and its implications”. *Journal of Policing, Intelligence and Counter Terrorism* 7 (1). Retrieved 6 June 2015.
- [5] “Critical infrastructure vulnerable to attack, warns cyber security expert”. *gsnmagazine.com*. Government Security News. 2014. Retrieved 6 June 2015.
- [6] Maniscalchi, Jago (4 September 2011). “What is Cyber-war?”. Retrieved 6 June 2015.
- [7] Lynn, William J. III. “Defending a New Domain: The Pentagon’s Cyberstrategy”, *Foreign Affairs*, Sept/Oct. 2010, pp. 97–108
- [8] Clapper, James R. “Worldwide Threat Assessment of the US Intelligence Community ”, Senate Armed Services Committee, Feb. 26, 2015 p. 1

- [9] Lisa Lucile Owens, Justice and Warfare in Cyberspace, *The Boston Review* (2015), available at
- [10] Poole-Robb, Stuart. "Turkish blackout sparks fears of cyber attack on the West", ITProPortal.com, May 19, 2015
- [11] "Cyberattacks, Terrorism Top U.S. Security Threat Report". *NPR.org*. 12 March 2013.
- [12] "A Note on the Laws of War in Cyberspace", James A. Lewis, April 2010
- [13] Rayman, Noah (December 18, 2013). "Merkel Compared NSA To Stasi in Complaint To Obama". *Time*. Retrieved February 1, 2014.
- [14] Devereaux, Ryan; Greenwald, Glenn; Poitras, Laura (May 19, 2014). "Data Pirates of the Caribbean: The NSA Is Recording Every Cell Phone Call in the Bahamas". *The Intercept*. First Look Media. Retrieved May 21, 2014.
- [15] Schonfeld, Zach (May 23, 2014). "The Intercept Wouldn't Reveal a Country the U.S. Is Spying On, So WikiLeaks Did Instead". *Newsweek*. Retrieved May 26, 2014.
- [16] Bodmer, Kilger, Carpenter, & Jones (2012). Reverse Deception: Organized Cyber Threat Counter-Exploitation. New York: McGraw-Hill Osborne Media. ISBN 0071772499, ISBN 978-0071772495
- [17] Sanders, Sam (4 June 2015). "Massive Data Breach Puts 4 Million Federal Employees' Records At Risk". *NPR*. Retrieved 5 June 2015.
- [18] Liptak, Kevin (4 June 2015). "U.S. government hacked; feds think China is the culprit". *CNN*. Retrieved 5 June 2015.
- [19] "Clarke: More defense needed in cyberspace" *HometownAnnapolis.com*, 24 September 2010
- [20] "Malware Hits Computerized Industrial Equipment". *The New York Times*, 24 September 2010
- [21] Shiels, Maggie. (9 April 2009) BBC: Spies 'infiltrate US power grid'. BBC News. Retrieved 8 November 2011.
- [22] Meserve, Jeanne (8 April 2009). "Hackers reportedly have embedded code in power grid". CNN. Retrieved 8 November 2011.
- [23] "US concerned power grid vulnerable to cyber-attack". In.reuters.com (9 April 2009). Retrieved 8 November 2011.
- [24] Gorman, Siobhan. (8 April 2009) Electricity Grid in U.S. Penetrated By Spies. *The Wall Street Journal*. Retrieved 8 November 2011.
- [25] NERC Public Notice. (PDF). Retrieved 8 November 2011.
- [26] Xinhua: China denies intruding into the U.S. electrical grid. 9 April 2009
- [27] 'China threat' theory rejected. *China Daily* (9 April 2009). Retrieved 8 November 2011.
- [28] ABC News: Video. ABC News. (20 April 2009). Retrieved 8 November 2011.
- [29] Disconnect electrical grid from Internet, former terror czar Clarke warns. *The Raw Story* (8 April 2009). Retrieved 8 November 2011.
- [30] "White House Cyber Czar: 'There Is No Cyberwar'". *Wired*, 4 March 2010
- [31] "Cyber-War Nominee Sees Gaps in Law", *The New York Times*, 14 April 2010
- [32] Cyber ShockWave Shows U.S. Unprepared For Cyber Threats. Bipartisanpolicy.org. Retrieved 8 November 2011.
- [33] Drogin, Bob (17 February 2010). "In a doomsday cyber attack scenario, answers are unsettling". *Los Angeles Times*.
- [34] Ali, Sarmad (16 February 2010). "Washington Group Tests Security in 'Cyber ShockWave'". *The Wall Street Journal*.
- [35] Cyber ShockWave CNN/BPC wargame: was it a failure?. *Computerworld* (17 February 2010). Retrieved 8 November 2011.
- [36] Steve Ragan Report: The Cyber ShockWave event and its aftermath. *The Tech Herald*. 16 February 2010
- [37] Lee, Andy (1 May 2012). "International Cyber Warfare: Limitations and Possibilities". Jeju Peace Institute.
- [38] U.S. Navy Recruiting - Cyber Warfare Engineer.
- [39] Denning, D. E. (2008). The ethics of cyber conflict. The Handbook of Information and Computer Ethics. 407–429.
- [40] "Google Attack Is Tip Of Iceberg", McAfee Security Insights, 13 January 2010
- [41] Government-sponsored cyberattacks on the rise, McAfee says. *Network World* (29 November 2007). Retrieved 8 November 2011.
- [42] "US embassy cables: China uses access to Microsoft source code to help plot cyber warfare, US fears". *The Guardian* (London). 4 December 2010. Retrieved 31 December 2010.
- [43] "How China will use cyber warfare to leapfrog in military competitiveness". *Culture Mandala: The Bulletin of the Centre for East-West Cultural and Economic Studies* 8 (1 [October 2008]). p. 37. Retrieved January 2013.
- [44] "China to make mastering cyber warfare A priority (2011)". Washington, D.C.: NPR. Retrieved January 2013.
- [45] "How China will use cyber warfare to leapfrog in military competitiveness". *Culture Mandala: The Bulletin of the Centre for East-West Cultural and Economic Studies* 8 (1 [October 2008]). p. 42. Retrieved January 2013.

- [46] "How China will use cyber warfare to leapfrog in military competitiveness". *Culture Mandala: The Bulletin of the Centre for East-West Cultural and Economic Studies* 8 (1 [October 2008]). p. 43. Retrieved January 2013.
- [47] "Washington, Beijing In Cyber-War Standoff". Yahoo! News. 12 February 2013. Retrieved January 2013.
- [48] Etzioni, Amitai (20 September 2013). "MAR: A Model for US-China Relations", *The Diplomat*.
- [49] "Germany's 60-person Computer Network Operation (CNO) unit has been practicing for cyber war for years."
- [50] "Hackers wanted to man front line in cyber war", *The Local*, 24 March 2013
- [51] "Germany to invest 100 million euros on internet surveillance: report", Kazinform, 18 June 2013
- [52] "Beware of the bugs: Can cyber attacks on India's critical infrastructure be thwarted?". *BusinessToday*. Retrieved January 2013.
- [53] "5 lakh cyber warriors to bolster India's e-defence". *The Times of India* (India). 16 October 2012. Retrieved 18 October 2012.
- [54] "Iran's military is preparing for cyber warfare". Flash//CRITIC Cyber Threat News. Retrieved 18 March 2015.
- [55] Lee, Se Young. "South Korea raises alert after hackers attack broadcasters, banks". *Global Post*. Retrieved 6 April 2013.
- [56] Kim, Eun-jung. "S. Korean military to prepare with U.S. for cyber warfare scenarios". Yonhap News Agency. Retrieved 6 April 2013.
- [57] "Nationale Cyber Security Centrum – NCSC".
- [58] "Defensie Cyber Strategie".
- [59] "Cyber commando".
- [60] "Al Qaeda rocked by apparent cyberattack. But who did it?". *The Christian Science Monitor*.
- [61] "Attack the City: why the banks are 'war gaming'".
- [62] "Wall Street banks learn how to survive in staged cyber attack". Reuters. 21 October 2013.
- [63] American Forces Press Service: Lynn Explains U.S. Cybersecurity Strategy. Defense.gov. Retrieved 8 November 2011.
- [64] "Pentagon to Consider Cyberattacks Acts of War". *The New York Times*. 31 May 2006
- [65] Dilanian, Ken. "Cyber-attacks a bigger threat than Al Qaeda, officials say", *Los Angeles Times*, 12 March 2013
- [66] "Intelligence Chairman: U.S. Fighting Cyber War 'Every Day'", *PJ Media*, 29 July 2013
- [67] "Cyberwar: War in the Fifth Domain" *Economist*, 1 July 2010
- [68] The Lipman Report, 15 October 2010
- [69] Clarke, Richard. "China's Cyberassault on America", *The Wall Street Journal*, 15 June 2011
- [70] "Cyberwarrior Shortage Threatens U.S. Security". NPR, 19 July 2010
- [71] "U.S. military cyberwar: What's off-limits?" CNET, 29 July 2010
- [72] "US Launched Cyber Attacks on Other Nations". RT, 26 January 2012.
- [73] Sanger, David E. "Obama Order Sped Up Wave of Cyber-attacks Against Iran." *The New York Times*, 1 June 2012.
- [74] ANNUAL REPORT TO CONGRESS Military and Security Developments Involving the People's Republic of China 2010. US Defense Department (PDF). Retrieved 8 November 2011.
- [75] AP: Pentagon takes aim at China cyber threat
- [76] "The Joint Operating Environment", Joint Forces Command, 18 February 2010, pp. 34–36
- [77] A Bill. To amend the Homeland Security Act of 2002 and other laws to enhance the security and resiliency of the cyber and communications infrastructure of the United States.. Senate.gov. 111th Congress 2D Session
- [78] Senators Say Cybersecurity Bill Has No 'Kill Switch', *Information Week*, 24 June 2010. Retrieved 25 June 2010.
- [79] Satter, Raphael. "US general: We hacked the enemy in Afghanistan.". Associated Press, 24 August 2012.
- [80] DOD – Cyber Counterintelligence. Dtac.mil. Retrieved 8 November 2011.
- [81] Pentagon Bill To Fix Cyber Attacks: ,0M. CBS News. Retrieved 8 November 2011.
- [82] "Senate Legislation Would Federalize Cybersecurity". *The Washington Post*. Retrieved 8 November 2011.
- [83] "White House Eyes Cyber Security Plan". CBS News (10 February 2009). Retrieved 8 November 2011.
- [84] CCD COE – Cyber Defence. Ccdcoe.org. Retrieved 8 November 2011.
- [85] Associated Press (11 May 2009) FBI to station cyber-crime expert in Estonia. *Boston Herald*. Retrieved 8 November 2011.
- [86] Reed, John. "Is the 'holy grail' of cyber security within reach?". *Foreign Policy Magazine*, 6 September 2012.
- [87] Carroll, Chris. "US can trace cyberattacks, mount pre-emptive strikes, Panetta says". *Stars and Stripes*, 11 October 2012.
- [88] "Latest viruses could mean 'end of world as we know it,' says man who discovered Flame", *The Times of Israel*, 6 June 2012
- [89] "Cyber espionage bug attacking Middle East, but Israel untouched — so far", *The Times of Israel*, 4 June 2013

- [90] Rid, Thomas (October 2011). "Cyber War Will Not Take Place". *Journal of Strategic Studies* 35: 5–32. doi:10.1080/01402390.2011.608939. Retrieved 21 October 2011.
- [91] Deibert, Ron (2011). "Tracking the emerging arms race in cyberspace". *Bulletin of the Atomic Scientists* 67 (1): 1–8. doi:10.1177/0096340210393703.
- [92] Sommer, Peter (January 2011). "Reducing Systemic Cybersecurity Risk" (PDF). *OECD Multi-Disciplinary Issues*. Retrieved 21 May 2012.
- [93] Gaycken, Sandro (2010). "Cyberwar – Das Internet als Kriegsschauplatz".
[94] Mathew J. Schwartz (21 November 2011). "Hacker Apparently Triggers Illinois Water Pump Burnout". *Information Week*.
- [95] Kim Zetter (30 November 2011). "Exclusive: Comedy of Errors Led to False 'Water-Pump Hack' Report". *Wired*.
- [96] U.S. drone and predator fleet is being keylogged. *Wired*, October 2011. Retrieved 6 October 2011
- [97] Hennigan, W.J. "Air Force says drone computer virus poses 'no threat'". *Los Angeles Times*, 13 October 2011.
- [98] "SK Hack by an Advanced Persistent Threat" (PDF). Command Five Pty Ltd. Retrieved 24 September 2011.
- [99] Jim Finkle (3 August 2011). "State actor seen in "enormous" range of cyber attacks". Reuters. Retrieved 3 August 2011.
- [100] "Hacked by 'Pakistan cyber army', CBI website still not restored". Ndtv.com (4 December 2010). Retrieved 8 November 2011.
- [101] "36 government sites hacked by 'Indian Cyber Army'". *The Express Tribune*. Retrieved 8 November 2011.
- [102] Britain faces serious cyber threat, spy agency head warns. *The Globe and Mail* (13 October 2010). Retrieved 8 November 2011.
- [103] AFP (1 October 2010). Stuxnet worm brings cyber warfare out of virtual world. Google. Retrieved 8 November 2011.
- [104] Ralph Langner: Cracking Stuxnet, a 21st-century cyber weapon | Video on. Ted.com. Retrieved 8 November 2011.
- [105] Sudworth, John. (9 July 2009) "New cyberattacks hit South Korea". BBC News. Retrieved 8 November 2011.
- [106] Williams, Martin. UK, Not North Korea, Source of DDOS Attacks, Researcher Says. *PC World*.
- [107] Danchev, Dancho (11 August 2008). "Coordinated Russia vs Georgia cyberattack". ZDNet. Retrieved 25 November 2008.
- [108] Website of Kyrgyz Central Election Commission hacked by Estonian hackers, Regnum, 14 December 2007
- [109] Fulghum, David A. "Why Syria's Air Defenses Failed to Detect Israelis", *Aviation Week & Space Technology*, 3 October 2007. Retrieved 3 October 2007.
- [110] Fulghum, David A. "Israel used electronic attack in air strike against Syrian mystery target", *Aviation Week & Space Technology*, 8 October 2007. Retrieved 8 October 2007.
- [111] "War in the fifth domain. Are the mouse and keyboard the new weapons of conflict?". *The Economist*. 1 July 2010. Retrieved 2 July 2010. Important thinking about the tactical and legal concepts of cyber-warfare is taking place in a former Soviet barracks in Estonia, now home to NATO's "centre of excellence" for cyber-defence. It was established in response to what has become known as "Web War 1", a concerted denial-of-service attack on Estonian government, media and bank web servers that was precipitated by the decision to move a Soviet-era war memorial in central Tallinn in 2007.
- [112] Estonia accuses Russia of 'cyber attack'. *The Christian Science Monitor*. (17 May 2007). Retrieved 8 November 2011.
- [113] Ian Traynor, 'Russia accused of unleashing cyberwar to disable Estonia', *The Guardian*, 17 May 2007
- [114] Boyd, Clark. (17 June 2010) "Cyber-war a growing threat warn experts". BBC News. Retrieved 8 November 2011.
- [115] Scott J. Shackelford, From Nuclear War to Net War: Analogizing Cyber Attacks in International Law, 27 Berkeley J. Int'l Law. 192 (2009).
- [116] "Israel Adds Cyber-Attack to IDF", Military.com, 10 February 2010
- [117] Russian Embassy to the UK . Retrieved 25 May 2012.
- [118] Tom Gjelten (23 September 2010). "Seeing The Internet As An 'Information Weapon'". NPR. Retrieved 23 September 2010.
- [119] Gorman, Siobhan. (4 June 2010) WSJ: U.S. Backs Talks on Cyber Warfare. *The Wall Street Journal*. Retrieved 8 November 2011.
- [120] Sean Gallagher, US, Russia to install "cyber-hotline" to prevent accidental cyberwar, ArsTechnica, 18 June 2013
- [121] Український центр політичного менеджменту – Зміст публікації – Конвенция о запрещении использования кибервойны. Politik.org.ua. Retrieved 8 November 2011.

2.2.11 External links

Books

- Bodmer, Kilger, Carpenter, & Jones (2012). Reverse Deception: Organized Cyber Threat Counter-Exploitation. New York: McGraw-Hill Osborne Media. ISBN 0071772499, ISBN 978-0071772495

Websites

- NATO Cooperative Cyber Defence Centre of Excellence (CCDCOE)
- Cyberwar Twitter feed from Richard Stiennon

Videos

- “Sabotaging the System” video, “60 Minutes”, 8 November 2009, CBS News, 15 minutes

Articles

- ABC: Former White House security advisor warns of cyber war
- Wall Street Journal: Fighting Wars in Cyberspace
- Will There Be An Electronic Pearl Harbor, PC World by Ira Winkler, 1 December 2009
- Senate panel: 80 percent of cyberattacks preventable, Wired, 17 November 2009
- Consumer Reports Online Security Guide
- Cyberwarfare reference materials
- Duncan Gardham, 26 June 2009, Hackers recruited to fight 'new cold war', Telegraph UK
- Stefano Mele, Jun 2013, Cyber-Weapons: Legal and Strategic Aspects (version 2.0)
- Stefano Mele, 30 September 2010, Cyberwarfare and its damaging effects on citizens
- History of Cyber Warfare
- Cybersecurity: Authoritative Reports and Resources, US Congressional Research Service
- Why the USA is Losing The Cyberwar Against China, by Joseph Steinberg, VentureBeat, 9 November 2011
- Michael Riley and Ashlee Vance, 20 July 2011, Cyber Weapons: The New Arms Race
- The Digital Arms Race: NSA Preps America for Future Battle, Der Spiegel, January 2015

2.3 Information security

Information security, sometimes shortened to **InfoSec**, is the practice of defending information from unauthorized access, use, disclosure, disruption, modification, perusal, inspection, recording or destruction. It is a general term that can be used regardless of the form the data may take (e.g. electronic, physical).^[1]

2.3.1 Overview

IT security Sometimes referred to as computer security, Information Technology security is information security applied to technology (most often some form of computer system). It is worthwhile to note that a computer does not necessarily mean a home desktop. A computer is any device with a processor and some memory. Such devices can range from non-networked standalone devices as simple as calculators, to networked mobile computing devices such as smartphones and tablet computers. IT security specialists are almost always found in any major enterprise/establishment due to the nature and value of the data within larger businesses. They are responsible for keeping all of the technology within the company secure from malicious cyber attacks that often attempt to breach into critical private information or gain control of the internal systems.

Information assurance The act of ensuring that data is not lost when critical issues arise. These issues include, but are not limited to: natural disasters, computer/server malfunction, physical theft, or any other instance where data has the potential of being lost. Since most information is stored on computers in our modern era, information assurance is typically dealt with by IT security specialists. One of the most common methods of providing information assurance is to have an off-site backup of the data in case one of the mentioned issues arise.

Threats

Computer system threats come in many different forms. Some of the most common threats today are software attacks, theft of intellectual property, identity theft, theft of equipment or information, sabotage, and information extortion. Most people have experienced software attacks of some sort. Viruses, worms, phishing attacks, and trojan horses are a few common examples of software attacks. The theft of intellectual property has also been an extensive issue for many businesses in the IT field. Intellectual property is the ownership of property usually consisting of some form of protection. Theft of software is probably the most common in IT businesses today. Identity theft is the attempt to act as someone else usually to obtain that person's personal information or to take advantage of their access to vital information. Theft of equipment or information is becoming more prevalent today due to the fact that most devices today are mobile. Cell phones are prone to theft and have also become far more desirable as the amount of data capacity increases. Sabotage usually consists of the destruction of an organization's website in an attempt to cause loss of confidence to its customers. Information extortion consists of theft of a company's property or information as an attempt to receive a payment in exchange for returning the information or property back to its owner. There are many ways

to help protect yourself from some of these attacks but one of the most functional precautions is user carefulness.

Governments, military, corporations, financial institutions, hospitals and private businesses amass a great deal of confidential information about their employees, customers, products, research and financial status. Most of this information is now collected, processed and stored on electronic computers and transmitted across networks to other computers.

Should confidential information about a business' customers or finances or new product line fall into the hands of a competitor or a **black hat hacker**, a business and its customers could suffer widespread, irreparable financial loss, as well as damage to the company's reputation. Protecting confidential information is a business requirement and in many cases also an ethical and legal requirement. Hence a key concern for organizations today is to derive the optimal information security investment. The renowned **Gordon-Loeb Model** actually provides a powerful mathematical economic approach for addressing this critical concern.^[2]

For the individual, information security has a significant effect on **privacy**, which is viewed very differently in different cultures.

The field of information security has grown and evolved significantly in recent years. There are many ways of gaining entry into the field as a career. It offers many areas for specialization including securing network(s) and allied infrastructure, securing applications and databases, security testing, information systems auditing, business continuity planning and digital forensics.

2.3.2 History

Since the early days of communication, diplomats and military commanders understood that it was necessary to provide some mechanism to protect the confidentiality of correspondence and to have some means of detecting tampering. Julius Caesar is credited with the invention of the **Caesar cipher** c. 50 B.C., which was created in order to prevent his secret messages from being read should a message fall into the wrong hands, but for the most part protection was achieved through the application of procedural handling controls. Sensitive information was marked up to indicate that it should be protected and transported by trusted persons, guarded and stored in a secure environment or strong box. As postal services expanded, governments created official organizations to intercept, decipher, read and reseal letters (e.g. the UK Secret Office and Deciphering Branch in 1653).

In the mid-19th century more complex **classification systems** were developed to allow governments to manage their information according to the degree of sensitivity. The British Government codified this, to some extent, with the publication of the **Official Secrets Act** in 1889.

By the time of the First World War, multi-tier classification systems were used to communicate information to and from various fronts, which encouraged greater use of code making and breaking sections in diplomatic and military headquarters. In the United Kingdom this led to the creation of the **Government Code and Cypher School** in 1919. Encoding became more sophisticated between the wars as machines were employed to scramble and unscramble information. The volume of information shared by the Allied countries during the Second World War necessitated formal alignment of classification systems and procedural controls. An arcane range of markings evolved to indicate who could handle documents (usually officers rather than men) and where they should be stored as increasingly complex safes and storage facilities were developed. Procedures evolved to ensure documents were destroyed properly and it was the failure to follow these procedures which led to some of the greatest intelligence coups of the war (e.g. U-570).

The end of the 20th century and early years of the 21st century saw rapid advancements in telecommunications, computing hardware and software, and data encryption. The availability of smaller, more powerful and less expensive computing equipment made **electronic data processing** within the reach of small business and the home user. These computers quickly became interconnected through the **Internet**.

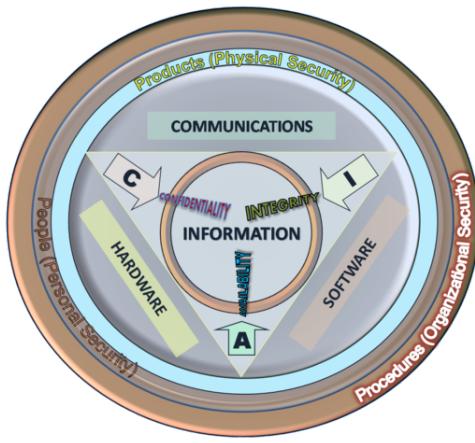
The rapid growth and widespread use of electronic data processing and **electronic business** conducted through the Internet, along with numerous occurrences of international **terrorism**, fueled the need for better methods of protecting the computers and the information they store, process and transmit. The academic disciplines of **computer security** and **information assurance** emerged along with numerous professional organizations – all sharing the common goals of ensuring the security and reliability of information systems.

2.3.3 Definitions

The definitions of InfoSec suggested in different sources are summarized below (adopted from).^[3]

1. “Preservation of confidentiality, integrity and availability of information. Note: In addition, other properties, such as authenticity, accountability, non-repudiation and reliability can also be involved.” (ISO/IEC 27000:2009)^[4]
2. “The protection of information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction in order to provide confidentiality, integrity, and availability.” (CNSS, 2010)^[5]
3. “Ensures that only authorized users (confidentiality) have access to accurate and complete information (integrity) when required (availability).” (ISACA, 2008)^[6]
4. “Information Security is the process of protecting

thinness, non-repudiation, accountability and auditability." (Cherdantseva and Hilton, 2013)^[3]



Information Security Attributes: or qualities, i.e., Confidentiality, Integrity and Availability (CIA). Information Systems are composed in three main portions, hardware, software and communications with the purpose to help identify and apply information security industry standards, as mechanisms of protection and prevention, at three levels or layers: physical, personal and organizational. Essentially, procedures or policies are implemented to tell people (administrators, users and operators) how to use products to ensure information security within the organizations.

the intellectual property of an organisation." (Pipkin, 2000)^[7]

5. "...information security is a risk management discipline, whose job is to manage the cost of information risk to the business." (McDermott and Geer, 2001)^[8]

6. "A well-informed sense of assurance that information risks and controls are in balance." (Anderson, J., 2003)^[9]

7. "Information security is the protection of information and minimises the risk of exposing information to unauthorised parties." (Venter and Elof, 2003)^[10]

8. "Information Security is a multidisciplinary area of study and professional activity which is concerned with the development and implementation of security mechanisms of all available types (technical, organisational, human-oriented and legal) in order to keep information in all its locations (within and outside the organisation's perimeter) and, consequently, information systems, where information is created, processed, stored, transmitted and destroyed, free from threats."

Threats to information and information systems may be categorised and a corresponding security goal may be defined for each category of threats. A set of security goals, identified as a result of a threat analysis, should be revised periodically to ensure its adequacy and conformance with the evolving environment. The currently relevant set of security goals may include: *confidentiality, integrity, availability, privacy, authenticity & trustworthiness, non-repudiation, accountability and auditability*.

2.3.4 Profession

Information security is a stable and growing profession. Information security professionals are very stable in their employment; more than 80 percent had no change in employer or employment in the past year, and the number of professionals is projected to continuously grow more than 11 percent annually from 2014 to 2019.^[11]

2.3.5 Basic principles

Key concepts

The CIA triad of **confidentiality, integrity, and availability** is at the heart of information security.^[12] (The members of the classic InfoSec triad — confidentiality, integrity and availability — are interchangeably referred to in the literature as security attributes, properties, security goals, fundamental aspects, information criteria, critical information characteristics and basic building blocks.) There is continuous debate about extending this classic trio.^[3] Other principles such as Accountability^[13] have sometimes been proposed for addition – it has been pointed out that issues such as Non-Repudiation do not fit well within the three core concepts.

In 1992 and revised in 2002, the **OECD**'s Guidelines for the Security of Information Systems and Networks^[14] proposed the nine generally accepted principles: Awareness, Responsibility, Response, Ethics, Democracy, Risk Assessment, Security Design and Implementation, Security Management, and Reassessment. Building upon those, in 2004 the **NIST**'s Engineering Principles for Information Technology Security^[15] proposed 33 principles. From each of these derived guidelines and practices.

In 2002, Donn Parker proposed an alternative model for the classic CIA triad that he called the six atomic elements of information. The elements are confidentiality, possession, integrity, authenticity, availability, and utility. The merits of the Parkerian hexad are a subject of debate amongst security professionals.

In 2013, based on a thorough analysis of Information Assurance and Security (IAS) literature, the IAS-octave was proposed as an extension of the CIA-triad.^[16] The IAS-octave includes Confidentiality, Integrity, Availability, Accountability, Auditability, Authenticity/Trustworthiness, Non-repudiation and Privacy. The completeness and accuracy of the IAS-octave was evaluated via a series of interviews with IAS academics and experts. The IAS-octave is one of the dimensions of a **Reference Model of Information Assurance and Security (RMIAS)**, which summarizes the IAS knowledge in

one all-encompassing model.

Confidentiality In information security, confidentiality “is the property, that information is not made available or disclosed to unauthorized individuals, entities, or processes” (Excerpt ISO27000).

Integrity In information security, data integrity means maintaining and assuring the accuracy and completeness of data over its entire life-cycle.^[17] This means that data cannot be modified in an unauthorized or undetected manner. This is not the same thing as referential integrity in databases, although it can be viewed as a special case of consistency as understood in the classic **ACID** model of transaction processing. Information security systems typically provide message integrity in addition to data confidentiality.

Availability For any information system to serve its purpose, the information must be **available** when it is needed. This means that the computing systems used to store and process the information, the security controls used to protect it, and the communication channels used to access it must be functioning correctly. **High availability** systems aim to remain available at all times, preventing service disruptions due to power outages, hardware failures, and system upgrades. Ensuring availability also involves preventing **denial-of-service attacks**, such as a flood of incoming messages to the target system essentially forcing it to shut down.^[18]

Non-repudiation In law, **non-repudiation** implies one’s intention to fulfill their obligations to a contract. It also implies that one party of a transaction cannot deny having received a transaction nor can the other party deny having sent a transaction. Note: This is also regarded as part of **Integrity**.

It is important to note that while technology such as cryptographic systems can assist in non-repudiation efforts, the concept is at its core a legal concept transcending the realm of technology. It is not, for instance, sufficient to show that the message matches a digital signature signed with the sender’s private key, and thus only the sender could have sent the message and nobody else could have altered it in transit. The alleged sender could in return demonstrate that the digital signature algorithm is vulnerable or flawed, or allege or prove that his signing key has been compromised. The fault for these violations may or may not lie with the sender himself, and such assertions may or may not relieve the sender of liability, but the assertion would invalidate the claim that the signature necessarily proves authenticity and integrity and thus prevents repudiation.

2.3.6 Risk management

Main article: **Risk management**

The *Certified Information Systems Auditor (CISA) Review Manual 2006* provides the following definition of risk management: “Risk management is the process of identifying **vulnerabilities** and **threats** to the information resources used by an organization in achieving business objectives, and deciding what **countermeasures**, if any, to take in reducing risk to an acceptable level, based on the value of the information resource to the organization.”^[19]

There are two things in this definition that may need some clarification. First, the *process* of risk management is an ongoing, iterative process. It must be repeated indefinitely. The business environment is constantly changing and new **threats** and **vulnerabilities** emerge every day. Second, the choice of **countermeasures** (controls) used to manage risks must strike a balance between productivity, cost, effectiveness of the countermeasure, and the value of the informational asset being protected.

Risk analysis and risk evaluation processes have their limitations since, when security incidents occur, they emerge in a context, and their rarity and even their uniqueness give rise to unpredictable threats. The analysis of these phenomena which are characterized by breakdowns, surprises and side-effects, requires a theoretical approach which is able to examine and interpret subjectively the detail of each incident.^[20]

Risk is the likelihood that something bad will happen that causes harm to an informational asset (or the loss of the asset). A **vulnerability** is a weakness that could be used to endanger or cause harm to an informational asset. A **threat** is anything (man-made or act of nature) that has the potential to cause harm.

The likelihood that a threat will use a vulnerability to cause harm creates a risk. When a threat does use a vulnerability to inflict harm, it has an impact. In the context of information security, the impact is a loss of availability, integrity, and confidentiality, and possibly other losses (lost income, loss of life, loss of real property). It should be pointed out that it is not possible to identify all risks, nor is it possible to eliminate all risk. The remaining risk is called “residual risk”.

A **risk assessment** is carried out by a team of people who have knowledge of specific areas of the business. Membership of the team may vary over time as different parts of the business are assessed. The assessment may use a subjective qualitative analysis based on informed opinion, or where reliable dollar figures and historical information is available, the analysis may use quantitative analysis.

The research has shown that the most vulnerable point in most information systems is the human user, operator, designer, or other human.^[21] The ISO/IEC 27002:2005 Code of practice for information security management

recommends the following be examined during a risk assessment:

- security policy,
- organization of information security,
- asset management,
- human resources security,
- physical and environmental security,
- communications and operations management,
- access control,
- information systems acquisition, development and maintenance,
- information security incident management,
- business continuity management, and
- regulatory compliance.

In broad terms, the risk management process consists of:

1. Identification of assets and estimating their value. Include: people, buildings, hardware, software, data (electronic, print, other), supplies.
2. Conduct a threat assessment. Include: Acts of nature, acts of war, accidents, malicious acts originating from inside or outside the organization.
3. Conduct a vulnerability assessment, and for each vulnerability, calculate the probability that it will be exploited. Evaluate policies, procedures, standards, training, physical security, quality control, technical security.
4. Calculate the impact that each threat would have on each asset. Use qualitative analysis or quantitative analysis.
5. Identify, select and implement appropriate controls. Provide a proportional response. Consider productivity, cost effectiveness, and value of the asset.
6. Evaluate the effectiveness of the control measures. Ensure the controls provide the required cost effective protection without discernible loss of productivity.

For any given risk, management can choose to **accept the risk** based upon the relative low value of the asset, the relative low frequency of occurrence, and the relative low impact on the business. Or, leadership may choose to **mitigate the risk** by selecting and implementing appropriate control measures to reduce the risk. In some cases, the risk can be **transferred** to another business by buying insurance or outsourcing to another business.^[22] The reality of some risks may be disputed. In such cases leadership may choose to **deny the risk**.

Controls

Main article: [security controls](#)

Selecting proper controls and implementing those will initially help an organization to bring down risk to acceptable levels. Control selection should follow and should be based on the risk assessment. Controls can vary in nature but fundamentally they are ways of protecting the confidentiality, integrity or availability of information. ISO/IEC 27001:2005 has defined 133 controls in different areas, but this is not exhaustive. Organizations can implement additional controls according to requirement of the organization. ISO 27001:2013 has cut down the number of controls to 113. From 08.11.2013 the technical standard of information security in place is: ABNT NBR ISO/IEC 27002:2013.^[23]

Administrative Administrative controls (also called procedural controls) consist of approved written policies, procedures, standards and guidelines. Administrative controls form the framework for running the business and managing people. They inform people on how the business is to be run and how day-to-day operations are to be conducted. Laws and regulations created by government bodies are also a type of administrative control because they inform the business. Some industry sectors have policies, procedures, standards and guidelines that must be followed – the Payment Card Industry Data Security Standard (PCI DSS) required by Visa and MasterCard is such an example. Other examples of administrative controls include the corporate security policy, password policy, hiring policies, and disciplinary policies.

Administrative controls form the basis for the selection and implementation of logical and physical controls. Logical and physical controls are manifestations of administrative controls. Administrative controls are of paramount importance.

Logical Logical controls (also called technical controls) use software and data to monitor and control access to information and computing systems. For example: passwords, network and host-based firewalls, network intrusion detection systems, access control lists, and data encryption are logical controls.

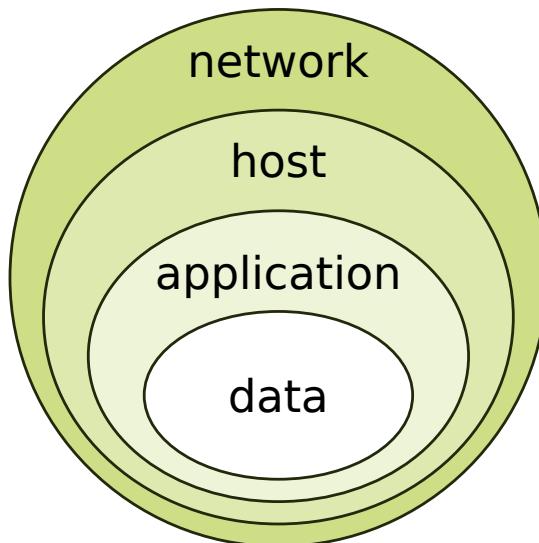
An important logical control that is frequently overlooked is the **principle of least privilege**. The principle of least privilege requires that an individual, program or system process is not granted any more access privileges than are necessary to perform the task. A blatant example of the failure to adhere to the principle of least privilege is logging into Windows as user Administrator to read email and surf the web. Violations of this principle can also occur when an individual collects additional access privileges over time. This happens when employees' job duties change, or they are promoted to a new position, or they

transfer to another department. The access privileges required by their new duties are frequently added onto their already existing access privileges which may no longer be necessary or appropriate.

Physical Physical controls monitor and control the environment of the work place and computing facilities. They also monitor and control access to and from such facilities. For example: doors, locks, heating and air conditioning, smoke and fire alarms, fire suppression systems, cameras, barricades, fencing, security guards, cable locks, etc. Separating the network and workplace into functional areas are also physical controls.

An important physical control that is frequently overlooked is the **separation of duties**. Separation of duties ensures that an individual can not complete a critical task by himself. For example: an employee who submits a request for reimbursement should not also be able to authorize payment or print the check. An applications programmer should not also be the server administrator or the database administrator – these roles and responsibilities must be separated from one another.^[24]

Defense in depth



The onion model of defense in depth

Information security must protect information throughout the life span of the information, from the initial creation of the information on through to the final disposal of the information. The information must be protected while in motion and while at rest. During its lifetime, information may pass through many different information processing systems and through many different parts of information processing systems. There are many different ways the information and information systems can be threatened. To fully protect the information during its lifetime, each component of the information processing

system must have its own protection mechanisms. The building up, layering on and overlapping of security measures is called defense in depth. The strength of any system is no greater than its weakest link. Using a defense in depth strategy, should one defensive measure fail there are other defensive measures in place that continue to provide protection.

Recall the earlier discussion about administrative controls, logical controls, and physical controls. The three types of controls can be used to form the basis upon which to build a defense-in-depth strategy. With this approach, defense-in-depth can be conceptualized as three distinct layers or planes laid one on top of the other. Additional insight into defense-in-depth can be gained by thinking of it as forming the layers of an onion, with data at the core of the onion, people the next outer layer of the onion, and **network security**, host-based security and **application security** forming the outermost layers of the onion. Both perspectives are equally valid and each provides valuable insight into the implementation of a good defense-in-depth strategy.

Security classification for information

An important aspect of information security and risk management is recognizing the value of information and defining appropriate procedures and protection requirements for the information. Not all information is equal and so not all information requires the same degree of protection. This requires information to be assigned a security classification.

The first step in information classification is to identify a member of senior management as the owner of the particular information to be classified. Next, develop a classification policy. The policy should describe the different classification labels, define the criteria for information to be assigned a particular label, and list the required security controls for each classification.

Some factors that influence which classification information should be assigned include how much value that information has to the organization, how old the information is and whether or not the information has become obsolete. Laws and other regulatory requirements are also important considerations when classifying information.

The Business Model for Information Security enables security professionals to examine security from systems perspective, creating an environment where security can be managed holistically, allowing actual risks to be addressed.

The type of information security classification labels selected and used will depend on the nature of the organization, with examples being:

- In the business sector, labels such as: **Public, Sensitive, Private, Confidential**.

- In the government sector, labels such as: **Unclassified, Unofficial, Protected, Confidential, Secret, Top Secret** and their non-English equivalents.
- In cross-sectoral formations, the **Traffic Light Protocol**, which consists of: **White, Green, Amber, and Red**.

All employees in the organization, as well as business partners, must be trained on the classification schema and understand the required security controls and handling procedures for each classification. The classification of a particular information asset that has been assigned should be reviewed periodically to ensure the classification is still appropriate for the information and to ensure the security controls required by the classification are in place and are followed in their right procedures.

Access control

Access to protected information must be restricted to people who are authorized to access the information. The computer programs, and in many cases the computers that process the information, must also be authorized. This requires that mechanisms be in place to control the access to protected information. The sophistication of the access control mechanisms should be in parity with the value of the information being protected – the more sensitive or valuable the information the stronger the control mechanisms need to be. The foundation on which access control mechanisms are built start with identification and authentication.

Access control is generally considered in three steps: Identification, Authentication, and Authorization.

Identification **Identification** is an assertion of who someone is or what something is. If a person makes the statement “Hello, my name is **John Doe**” they are making a claim of who they are. However, their claim may or may not be true. Before John Doe can be granted access to protected information it will be necessary to verify that the person claiming to be John Doe really is John Doe. Typically the claim is in the form of a username. By entering that username you are claiming “I am the person the username belongs to”.

Authentication **Authentication** is the act of verifying a claim of identity. When John Doe goes into a bank to make a withdrawal, he tells the bank teller he is John Doe—a claim of identity. The bank teller asks to see a photo ID, so he hands the teller his **driver's license**. The bank teller checks the license to make sure it has John Doe printed on it and compares the photograph on the license against the person claiming to be John Doe. If the photo and name match the person, then the teller has authenticated that John Doe is who he claimed to be. Simi-

larly by entering the correct password, the user is providing evidence that they are the person the username belongs to.

There are three different types of information that can be used for authentication:

- Something you know: things such as a PIN, a password, or your mother’s **maiden name**.
- Something you have: a driver’s license or a magnetic **swipe card**.
- Something you are: **biometrics**, including palm prints, fingerprints, voice prints and **retina (eye)** scans.

Strong authentication requires providing more than one type of authentication information (two-factor authentication). The **username** is the most common form of identification on computer systems today and the password is the most common form of authentication. Usernames and passwords have served their purpose but in our modern world they are no longer adequate. Usernames and passwords are slowly being replaced with more sophisticated authentication mechanisms.

Authorization After a person, program or computer has successfully been identified and authenticated then it must be determined what informational resources they are permitted to access and what actions they will be allowed to perform (run, view, create, delete, or change). This is called **authorization**. Authorization to access information and other computing services begins with administrative policies and procedures. The policies prescribe what information and computing services can be accessed, by whom, and under what conditions. The access control mechanisms are then configured to enforce these policies. Different computing systems are equipped with different kinds of access control mechanisms—some may even offer a choice of different access control mechanisms. The access control mechanism a system offers will be based upon one of three approaches to access control or it may be derived from a combination of the three approaches.

The **non-discretionary** approach consolidates all access control under a centralized administration. The access to information and other resources is usually based on the individuals function (role) in the organization or the tasks the individual must perform. The **discretionary approach** gives the creator or owner of the information resource the ability to control access to those resources. In the **Mandatory access control approach**, access is granted or denied basing upon the security classification assigned to the information resource.

Examples of common access control mechanisms in use today include role-based access control available in many advanced database management systems—simple

file permissions provided in the UNIX and Windows operating systems, Group Policy Objects provided in Windows network systems, Kerberos, RADIUS, TACACS, and the simple access lists used in many firewalls and routers.

To be effective, policies and other security controls must be enforceable and upheld. Effective policies ensure that people are held **accountable** for their actions. All failed and successful authentication attempts must be logged, and all access to information must leave some type of audit trail.

Also, **need-to-know principle** needs to be in affect when talking about access control. Need-to-know principle gives access rights to a person to perform their job functions. This principle is used in the government, when dealing with difference clearances. Even though two employees in different departments have a top-secret clearance, they must have a need-to-know in order for information to be exchanged. Within the need-to-know principle, network administrators grant the employee least amount privileges to prevent employees access and doing more than what they are supposed to. Need-to-know helps to enforce the confidentiality-integrity-availability (C-I-A) triad. Need-to-know directly impacts the confidential area of the triad.

Cryptography

Main article: [Cryptography](#)

Information security uses **cryptography** to transform usable information into a form that renders it unusable by anyone other than an authorized user; this process is called **encryption**. Information that has been encrypted (rendered unusable) can be transformed back into its original usable form by an authorized user, who possesses the **cryptographic key**, through the process of decryption. Cryptography is used in information security to protect information from unauthorized or accidental disclosure while the **information** is in transit (either electronically or physically) and while information is in storage.

Cryptography provides information security with other useful applications as well including improved authentication methods, message digests, digital signatures, **non-repudiation**, and encrypted network communications. Older less secure applications such as telnet and ftp are slowly being replaced with more secure applications such as **ssh** that use encrypted network communications. Wireless communications can be encrypted using protocols such as **WPA/WPA2** or the older (and less secure) **WEP**. Wired communications (such as **ITU-T G.hn**) are secured using **AES** for encryption and **X.1035** for authentication and key exchange. Software applications such as **GnuPG** or **PGP** can be used to encrypt data files and Email.

Cryptography can introduce security problems when it is not implemented correctly. Cryptographic solutions need to be implemented using industry accepted solutions that have undergone rigorous peer review by independent experts in cryptography. The **length** and **strength** of the encryption key is also an important consideration. A key that is **weak** or too short will produce weak encryption. The keys used for encryption and decryption must be protected with the same degree of rigor as any other confidential information. They must be protected from unauthorized disclosure and destruction and they must be available when needed. **Public key infrastructure (PKI)** solutions address many of the problems that surround **key management**.

2.3.7 Process

The terms **reasonable and prudent person**, **due care** and **due diligence** have been used in the fields of Finance, Securities, and Law for many years. In recent years these terms have found their way into the fields of computing and information security. U.S.A. **Federal Sentencing Guidelines** now make it possible to hold corporate officers liable for failing to exercise due care and due diligence in the management of their information systems.

In the business world, stockholders, customers, business partners and governments have the expectation that corporate officers will run the business in accordance with accepted business practices and in compliance with laws and other regulatory requirements. This is often described as the “reasonable and prudent person” rule. A prudent person takes due care to ensure that everything necessary is done to operate the business by sound business principles and in a legal ethical manner. A prudent person is also diligent (mindful, attentive, and ongoing) in their due care of the business.

In the field of Information Security, Harris^[25] offers the following definitions of **due care** and **due diligence**:

“Due care are steps that are taken to show that a company has taken responsibility for the activities that take place within the corporation and has taken the necessary steps to help protect the company, its resources, and employees.”
And, [Due diligence are the] “continual activities that make sure the protection mechanisms are continually maintained and operational.”

Attention should be made to two important points in these definitions. First, in due care, steps are taken to **show** - this means that the steps can be verified, measured, or even produce tangible artifacts. Second, in due diligence, there are **continual activities** - this means that people are actually doing things to monitor and maintain the protection mechanisms, and these activities are ongoing.

Security governance

See also: [Information Security Governance](#)

The Software Engineering Institute at Carnegie Mellon University, in a publication titled “Governing for Enterprise Security (GES)”, defines characteristics of effective security governance. These include:

- An enterprise-wide issue
- Leaders are accountable
- Viewed as a business requirement
- Risk-based
- Roles, responsibilities, and segregation of duties defined
- Addressed and enforced in policy
- Adequate resources committed
- Staff aware and trained
- A development life cycle requirement
- Planned, managed, measurable, and measured
- Reviewed and audited

Incident response plans

Main article: [Computer security incident management](#)

1 to 3 paragraphs (non technical) that discuss:

- Selecting team members
- Define roles, responsibilities and lines of authority
- Define a security incident
- Define a reportable incident
- Training
- Detection
- Classification
- Escalation
- Containment
- Eradication
- Documentation

Change management

Main article: [Change Management \(ITSM\)](#)

Change management is a formal process for directing and controlling alterations to the information processing environment. This includes alterations to desktop computers, the network, servers and software. The objectives of change management are to reduce the risks posed by changes to the information processing environment and improve the stability and reliability of the processing environment as changes are made. It is not the objective of change management to prevent or hinder necessary changes from being implemented.

Any change to the information processing environment introduces an element of risk. Even apparently simple changes can have unexpected effects. One of Management's many responsibilities is the management of risk. Change management is a tool for managing the risks introduced by changes to the information processing environment. Part of the change management process ensures that changes are not implemented at inopportune times when they may disrupt critical business processes or interfere with other changes being implemented.

Not every change needs to be managed. Some kinds of changes are a part of the everyday routine of information processing and adhere to a predefined procedure, which reduces the overall level of risk to the processing environment. Creating a new user account or deploying a new desktop computer are examples of changes that do not generally require change management. However, relocating user file shares, or upgrading the Email server pose a much higher level of risk to the processing environment and are not a normal everyday activity. The critical first steps in change management are (a) defining change (and communicating that definition) and (b) defining the scope of the change system.

Change management is usually overseen by a Change Review Board composed of representatives from key business areas, security, networking, systems administrators, Database administration, applications development, desktop support and the help desk. The tasks of the Change Review Board can be facilitated with the use of automated work flow application. The responsibility of the Change Review Board is to ensure the organizations documented change management procedures are followed. The change management process is as follows:

- **Requested:** Anyone can request a change. The person making the change request may or may not be the same person that performs the analysis or implements the change. When a request for change is received, it may undergo a preliminary review to determine if the requested change is compatible with the organizations **business model** and practices, and to determine the amount of resources needed to implement the change.

- **Approved:** Management runs the business and controls the allocation of resources therefore, Management must approve requests for changes and assign a priority for every change. Management might choose to reject a change request if the change is not compatible with the business model, industry standards or best practices. Management might also choose to reject a change request if the change requires more resources than can be allocated for the change.
- **Planned:** Planning a change involves discovering the scope and impact of the proposed change; analyzing the complexity of the change; allocation of resources and, developing, testing and documenting both implementation and backout plans. Need to define the criteria on which a decision to back out will be made.
- **Tested:** Every change must be tested in a safe test environment, which closely reflects the actual production environment, before the change is applied to the production environment. The backout plan must also be tested.
- **Scheduled:** Part of the change review board's responsibility is to assist in the scheduling of changes by reviewing the proposed implementation date for potential conflicts with other scheduled changes or critical business activities.
- **Communicated:** Once a change has been scheduled it must be communicated. The communication is to give others the opportunity to remind the change review board about other changes or critical business activities that might have been overlooked when scheduling the change. The communication also serves to make the Help Desk and users aware that a change is about to occur. Another responsibility of the change review board is to ensure that scheduled changes have been properly communicated to those who will be affected by the change or otherwise have an interest in the change.
- **Implemented:** At the appointed date and time, the changes must be implemented. Part of the planning process was to develop an implementation plan, testing plan and, a back out plan. If the implementation of the change should fail or, the post implementation testing fails or, other "drop dead" criteria have been met, the back out plan should be implemented.
- **Documented:** All changes must be documented. The documentation includes the initial request for change, its approval, the priority assigned to it, the implementation, testing and back out plans, the results of the change review board critique, the

date/time the change was implemented, who implemented it, and whether the change was implemented successfully, failed or postponed.

- **Post change review:** The change review board should hold a post implementation review of changes. It is particularly important to review failed and backed out changes. The review board should try to understand the problems that were encountered, and look for areas for improvement.

Change management procedures that are simple to follow and easy to use can greatly reduce the overall risks created when changes are made to the information processing environment. Good change management procedures improve the overall quality and success of changes as they are implemented. This is accomplished through planning, peer review, documentation and communication.

[ISO/IEC 20000, The Visible OPS Handbook: Implementing ITIL in 4 Practical and Auditable Steps^{\[26\]}](#) (Full book summary), [\[27\]](#) and [Information Technology Infrastructure Library](#) all provide valuable guidance on implementing an efficient and effective change management program information security.

2.3.8 Business continuity

Business continuity is the mechanism by which an organization continues to operate its critical business units, during planned or unplanned disruptions that affect normal business operations, by invoking planned and managed procedures.

Not only is business continuity simply about the business, but it also an IT system and process. Today disasters or disruptions to business are a reality. Whether the disaster is natural or man-made, it affects normal life and so business. Therefore, planning is important.

The planning is merely getting better prepared to face it, knowing fully well that the best plans may fail. Planning helps to reduce cost of recovery, operational overheads and most importantly sail through some smaller ones effortlessly.

For businesses to create effective plans they need to focus upon the following key questions. Most of these are common knowledge, and anyone can do a BCP.

1. Should a disaster strike, what are the first few things that I should do? Should I call people to find if they are OK or call up the bank to figure out my money is safe? This is Emergency Response. Emergency Response services help take the first hit when the disaster strikes and if the disaster is serious enough the Emergency Response teams need to quickly get a Crisis Management team in place.
2. What parts of my business should I recover first? The one that brings me most money or the one where

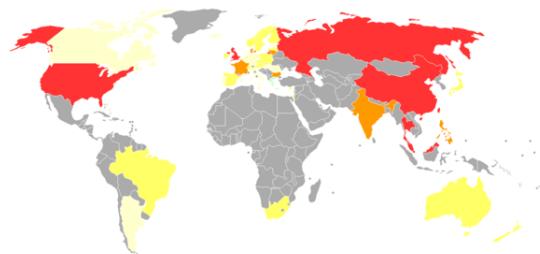
I spend the most, or the one that will ensure I shall be able to get sustained future growth? The identified sections are the critical business units. There is no magic bullet here, no one answer satisfies all. Businesses need to find answers that meet business requirements.

3. How soon should I target to recover my critical business units? In BCP technical jargon, this is called Recovery Time Objective, or RTO. This objective will define what costs the business will need to spend to recover from a disruption. For example, it is cheaper to recover a business in 1 day than in 1 hour.
4. What all do I need to recover the business? IT, machinery, records...food, water, people...So many aspects to dwell upon. The cost factor becomes clearer now...Business leaders need to drive business continuity. Hold on. My IT manager spent \$200000 last month and created a DRP (Disaster Recovery Plan), whatever happened to that? a DRP is about continuing an IT system, and is one of the sections of a comprehensive Business Continuity Plan. Look below for more on this.
5. And where do I recover my business from... Will the business center give me space to work, or would it be flooded by many people queuing up for the same reasons that I am.
6. But once I do recover from the disaster and work in reduced production capacity since my main operational sites are unavailable, how long can this go on. How long can I do without my original sites, systems, people? this defines the amount of business resilience a business may have.
7. Now that I know how to recover my business. How do I make sure my plan works? Most BCP pundits would recommend testing the plan at least once a year, reviewing it for adequacy and rewriting or updating the plans either annually or when businesses change.

Disaster recovery planning

While a business continuity plan (BCP) takes a broad approach to dealing with organizational-wide effects of a disaster, a disaster recovery plan (DRP), which is a subset of the business continuity plan, is instead focused on taking the necessary steps to resume normal business operations as quickly as possible. A disaster recovery plan is executed immediately after the disaster occurs and details what steps are to be taken in order to recover critical information technology infrastructure.^[28] Disaster recovery planning includes establishing a planning group, performing risk assessment, establishing priorities, developing recovery strategies, preparing inventories and documentation of the plan, developing verification criteria and procedure, and lastly implementing the plan.^[29]

2.3.9 Laws and regulations



Privacy International 2007 privacy ranking

green: Protections and safeguards

red: Endemic surveillance societies

Below is a partial listing of European, United Kingdom, Canadian and US governmental laws and regulations that have, or will have, a significant effect on data processing and information security. Important industry sector regulations have also been included when they have a significant impact on information security.

- UK Data Protection Act 1998 makes new provisions for the regulation of the processing of information relating to individuals, including the obtaining, holding, use or disclosure of such information. The European Union Data Protection Directive (EUDPD) requires that all EU member must adopt national regulations to standardize the protection of data privacy for citizens throughout the EU.
- The Computer Misuse Act 1990 is an Act of the UK Parliament making computer crime (e.g. hacking) a criminal offence. The Act has become a model upon which several other countries including Canada and the Republic of Ireland have drawn inspiration when subsequently drafting their own information security laws.
- EU Data Retention laws requires Internet service providers and phone companies to keep data on every electronic message sent and phone call made for between six months and two years.
- The Family Educational Rights and Privacy Act (FERPA) (20 U.S.C. § 1232 g; 34 CFR Part 99) is a US Federal law that protects the privacy of student education records. The law applies to all schools that receive funds under an applicable program of the U.S. Department of Education. Generally, schools must have written permission from the parent or eligible student in order to release any information from a student's education record.
- Federal Financial Institutions Examination Council's (FFIEC) security guidelines for auditors specifies requirements for online banking security.
- Health Insurance Portability and Accountability Act (HIPAA) of 1996 requires the adoption of national

standards for electronic health care transactions and national identifiers for providers, health insurance plans, and employers. And, it requires health care providers, insurance providers and employers to safeguard the security and privacy of health data.

- **Gramm–Leach–Bliley Act** of 1999 (GLBA), also known as the Financial Services Modernization Act of 1999, protects the privacy and security of private financial information that financial institutions collect, hold, and process.
- **Sarbanes–Oxley Act of 2002 (SOX)**. Section 404 of the act requires publicly traded companies to assess the effectiveness of their internal controls for financial reporting in annual reports they submit at the end of each fiscal year. Chief information officers are responsible for the security, accuracy and the reliability of the systems that manage and report the financial data. The act also requires publicly traded companies to engage independent auditors who must attest to, and report on, the validity of their assessments.
- **Payment Card Industry Data Security Standard (PCI DSS)** establishes comprehensive requirements for enhancing payment account data security. It was developed by the founding payment brands of the PCI Security Standards Council, including **American Express**, **Discover Financial Services**, **JCB**, **MasterCard Worldwide** and **Visa International**, to help facilitate the broad adoption of consistent **data security** measures on a global basis. The PCI DSS is a multifaceted security standard that includes requirements for security management, policies, procedures, **network architecture**, software design and other critical protective measures.
- State **security breach notification laws** (California and many others) require businesses, nonprofits, and state institutions to notify consumers when unencrypted “personal information” may have been compromised, lost, or stolen.
- **Personal Information Protection and Electronics Document Act (PIPEDA)** – An Act to support and promote electronic commerce by protecting personal information that is collected, used or disclosed in certain circumstances, by providing for the use of electronic means to communicate or record information or transactions and by amending the **Canada Evidence Act**, the Statutory Instruments Act and the Statute Revision Act.
- **Hellenic Authority for Communication Security and Privacy (ADAE)** (Law 165/2011) - The Greek Law establishes and describes the minimum Information Security controls that should be deployed by every company which provides electronic communication networks and/or services in Greece in order to protect customers’ Confidentiality. These include both

managerial and technical controls (i.e. log records should be stored for two years).

- Hellenic Authority for Communication Security and Privacy (ADAE) (Law 205/2013)- The latest Greek Law published by ADAE concentrates around the protection of the Integrity and Availability of the services and data offered by the Greek Telecommunication Companies. The new Law forces Telcos and associated companies to build, deploy and test appropriate Business Continuity Plans and redundant infrastructures.

2.3.10 Information Security Culture

Employee's behavior has a big impact to information security in organizations. Cultural concept can help different segments of the organization to concern about the information security within the organization. "Exploring the Relationship between Organizational Culture and Information Security Culture" provides the following definition of information security culture: "ISC is the totality of patterns of behavior in an organization that contribute to the protection of information of all kinds."^[30]

Information security culture needs to be improved continuously. In "Information Security Culture from Analysis to Change", authors commented, "It's a never ending process, a cycle of evaluation and change or maintenance." To manage the information security culture, five steps should be taken: Pre-evaluation, strategic planning, operative planning, implementation, and post-evaluation.^[31]

- Pre-Evaluation: to identify the awareness of information security within employees and to analysis current security policy.
- Strategic Planning: to come up a better awareness-program, we need to set clear targets. Clustering people is helpful to achieve it.
- Operative Planning: we can set a good security culture based on internal communication, management-buy-in, and security awareness and training program.^[31]
- Implementation: four stages should be used to implement the information security culture. They are commitment of the management, communication with organizational members, courses for all organizational members, and commitment of the employees.^[31]

2.3.11 Sources of standards

Main article: **Cyber Security Standards**

International Organization for Standardization (ISO) is a consortium of national standards institutes from 157

countries, coordinated through a secretariat in Geneva, Switzerland. ISO is the world's largest developer of standards. ISO 15443: "Information technology - Security techniques - A framework for IT security assurance", ISO/IEC 27002: "Information technology - Security techniques - Code of practice for information security management", ISO-20000: "Information technology - Service management", and ISO/IEC 27001: "Information technology - Security techniques - Information security management systems - Requirements" are of particular interest to information security professionals.

The US National Institute of Standards and Technology (NIST) is a non-regulatory federal agency within the U.S. Department of Commerce. The NIST Computer Security Division develops standards, metrics, tests and validation programs as well as publishes standards and guidelines to increase secure IT planning, implementation, management and operation. NIST is also the custodian of the US Federal Information Processing Standard publications (FIPS).

The Internet Society is a professional membership society with more than 100 organization and over 20,000 individual members in over 180 countries. It provides leadership in addressing issues that confront the future of the Internet, and is the organization home for the groups responsible for Internet infrastructure standards, including the Internet Engineering Task Force (IETF) and the Internet Architecture Board (IAB). The ISOC hosts the Requests for Comments (RFCs) which includes the Official Internet Protocol Standards and the RFC-2196 Site Security Handbook.

The Information Security Forum is a global nonprofit organization of several hundred leading organizations in financial services, manufacturing, telecommunications, consumer goods, government, and other areas. It undertakes research into information security practices and offers advice in its biannual Standard of Good Practice and more detailed advisories for members.

The Institute of Information Security Professionals (IISP) is an independent, non-profit body governed by its members, with the principal objective of advancing the professionalism of information security practitioners and thereby the professionalism of the industry as a whole. The Institute developed the IISP Skills Framework®. This framework describes the range of competencies expected of Information Security and Information Assurance Professionals in the effective performance of their roles. It was developed through collaboration between both private and public sector organisations and world-renowned academics and security leaders.

The German Federal Office for Information Security (in German *Bundesamt für Sicherheit in der Informationstechnik (BSI)*) BSI-Standards 100-1 to 100-4 are a set of recommendations including "methods, processes, procedures, approaches and measures relating to information security".^[32] The BSI-Standard 100-2 *IT-Grundschutz*

Methodology describes how an information security management can be implemented and operated. The Standard includes a very specific guide, the **IT Baseline Protection Catalogs** (also known as *IT-Grundschutz Catalogs*). Before 2005 the catalogs were formerly known as "**IT Baseline Protection Manual**". The Catalogs are a collection of documents useful for detecting and combating security-relevant weak points in the IT environment (IT cluster). The collection encompasses as of September 2013 over 4.400 pages with the introduction and catalogs. The *IT-Grundschutz* approach is aligned with the ISO/IEC 2700x family.

At the European Telecommunications Standards Institute a catalog of Information security indicators have been standardized by the Industrial Specification Group (ISG) ISI.

2.3.12 Scholars working in the field

2.3.13 See also

2.3.14 Further reading

- Anderson, K., "IT Security Professionals Must Evolve for Changing Market", *SC Magazine*, October 12, 2006.
- Aceituno, V., "On Information Security Paradigms", *ISSA Journal*, September 2005.
- Dhillon, G., *Principles of Information Systems Security: text and cases*, John Wiley & Sons, 2007.
- Easttom, C., *Computer Security Fundamentals (2nd Edition)* Pearson Press, 2011.
- Lambo, T., "ISO/IEC 27001: The future of infosec certification", *ISSA Journal*, November 2006.

Bibliography

- Allen, Julia H. (2001). *The CERT Guide to System and Network Security Practices*. Boston, MA: Addison-Wesley. ISBN 0-201-73723-X.
- Krutz, Ronald L.; Russell Dean Vines (2003). *The CISSP Prep Guide* (Gold ed.). Indianapolis, IN: Wiley. ISBN 0-471-26802-X.
- Layton, Timothy P. (2007). *Information Security: Design, Implementation, Measurement, and Compliance*. Boca Raton, FL: Auerbach publications. ISBN 978-0-8493-7087-8.
- McNab, Chris (2004). *Network Security Assessment*. Sebastopol, CA: O'Reilly. ISBN 0-596-00611-X.
- Peltier, Thomas R. (2001). *Information Security Risk Analysis*. Boca Raton, FL: Auerbach publications. ISBN 0-8493-0880-1.

- Peltier, Thomas R. (2002). *Information Security Policies, Procedures, and Standards: guidelines for effective information security management.* Boca Raton, FL: Auerbach publications. ISBN 0-8493-1137-3.
 - White, Gregory (2003). *All-in-one Security+ Certification Exam Guide.* Emeryville, CA: McGraw-Hill/Osborne. ISBN 0-07-222633-1.
 - Dhillon, Gurpreet (2007). *Principles of Information Systems Security: text and cases.* NY: John Wiley & Sons. ISBN 978-0-471-45056-6.
- [13] “Engineering Principles for Information Technology Security” (PDF). csrc.nist.gov.
- [14] “oecd.org” (PDF). Archived from the original (PDF) on May 16, 2011. Retrieved 2014-01-17.
- [15] “NIST Special Publication 800-27 Rev A” (PDF). csrc.nist.gov.
- [16] Y. Cherdantseva and J. Hilton, “A Reference Model of Information Assurance & Security,” Availability, Reliability and Security (ARES), 2013 Eighth International Conference on , vol. , no., pp.546-555, IEEE, doi: 10.1109/ARES.2013.72, 2–6 September 2013.
- [17] Boritz, J. Efrim. “IS Practitioners’ Views on Core Concepts of Information Integrity”. *International Journal of Accounting Information Systems*. Elsevier. Retrieved 12 August 2011.
- [18] Loukas, G.; Oke, G. (September 2010) [August 2009]. “Protection Against Denial of Service Attacks: A Survey” (PDF). *Comput. J.* **53** (7): 1020–1037. doi:10.1093/comjnl/bxp078.
- [19] ISACA (2006). *CISA Review Manual 2006*. Information Systems Audit and Control Association. p. 85. ISBN 1-933284-15-3.
- [20] Spagnoletti, Paolo; Resca A. (2008). “The duality of Information Security Management: fighting against predictable and unpredictable threats”. *Journal of Information System Security* **4** (3): 46–62.
- [21] Kiountouzis, E.A.; Kokolakis, S.A. *Information systems security: facing the information society of the 21st century.* London: Chapman & Hall, Ltd. ISBN 0-412-78120-4.
- [22] “NIST SP 800-30 Risk Management Guide for Information Technology Systems” (PDF). Retrieved 2014-01-17.
- [23]
- [24] “Segregation of Duties Control matrix”. ISACA. 2008. Archived from the original on 3 July 2011. Retrieved 2008-09-30.
- [25] Shon Harris (2003). *All-in-one CISSP Certification Exam Guide* (2nd ed.). Emeryville, California: McGraw-Hill/Osborne. ISBN 0-07-222966-7.
- [26] itpi.org Archived December 10, 2013 at the Wayback Machine
- [27] “wikisummaries.org”. wikisummaries.org. Retrieved 2014-01-17.
- [28] Harris, Shon (2008). *All-in-one CISSP Certification Exam Guide* (4th ed.). New York, NY: McGraw-Hill. ISBN 978-0-07-149786-2.
- [29] “The Disaster Recovery Plan”. Sans Institute. Retrieved 7 February 2012.
- [30] Lim, Joo S., et al. “Exploring the Relationship between Organizational Culture and Information Security Culture.” Australian Information Security Management Conference.

2.3.15 Notes and references

- [1] 44 U.S.C. § 3542(b)(1)
- [2] Gordon, Lawrence; Loeb, Martin (November 2002). “The Economics of Information Security Investment”. *ACM Transactions on Information and System Security* **5** (4): 438–457. doi:10.1145/581271.581274.
- [3] Cherdantseva Y. and Hilton J.: “Information Security and Information Assurance. The Discussion about the Meaning, Scope and Goals”. In: *Organizational, Legal, and Technological Dimensions of Information System Administrator*. Almeida F., Portela, I. (eds.). IGI Global Publishing. (2013)
- [4] ISO/IEC 27000:2009 (E). (2009). Information technology - Security techniques - Information security management systems - Overview and vocabulary. ISO/IEC.
- [5] Committee on National Security Systems: National Information Assurance (IA) Glossary, CNSS Instruction No. 4009, 26 April 2010.
- [6] ISACA. (2008). Glossary of terms, 2008. Retrieved from <http://www.isaca.org/Knowledge-Center/Documents/Glossary/glossary.pdf>
- [7] Pipkin, D. (2000). *Information security: Protecting the global enterprise*. New York: Hewlett-Packard Company.
- [8] B., McDermott, E., & Geer, D. (2001). Information security is information risk management. In Proceedings of the 2001 Workshop on New Security Paradigms NSPW ‘01, (pp. 97 – 104). ACM. doi:10.1145/508171.508187
- [9] Anderson, J. M. (2003). “Why we need a new definition of information security”. *Computers & Security*, 22(4), 308–313. doi:10.1016/S0167-4048(03)00407-3.
- [10] Venter, H. S., & Eloff, J. H. P. (2003). “A taxonomy for information security technologies”. *Computers & Security*, 22(4), 299–307. doi:10.1016/S0167-4048(03)00406-1.
- [11] [https://www.isc2.org/uploadedFiles/\(ISC\)2_Public_Content/2013%20Global%20Information%20Security%20Workforce%20Study%20Feb%202013.pdf](https://www.isc2.org/uploadedFiles/(ISC)2_Public_Content/2013%20Global%20Information%20Security%20Workforce%20Study%20Feb%202013.pdf)
- [12] Perrin, Chad. “The CIA Triad”. Retrieved 31 May 2012.

- [31] Schlienger, Thomas, and Stephanie Teufel. "Information security culture—from analysis to change." *South African Computer Journal* 31 (2003): 46-52.
- [32] "BSI-Standards". <https://www.bsi.bund.de>. BSI. Retrieved 29 November 2013.

2.3.16 External links

- DoD IA Policy Chart on the DoD Information Assurance Technology Analysis Center web site.
- patterns & practices Security Engineering Explained
- Open Security Architecture- Controls and patterns to secure IT systems
- An Introduction to Information Security
- IWS - Information Security Chapter
- Ross Anderson's book "Security Engineering"
- English translation of the Greek Regulation (165/2011)

2.4 Network security

Network security consists of the policies adopted to prevent and monitor authorized 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.

2.4.1 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]

2.4.2 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

2.4.3 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

2.4.4 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

2.4.5 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 Abeler, Cisco Press, Sep. 26, 2006.
- *Securing Your Business with Cisco ASA and PIX Firewalls*, Greg Abeler, 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.

2.4.6 External links

- Cyber Security Network
- Definition of Network Security
- Network Security - Comprehensive Security from Layer 2 to Layer 8
- Cisco IT Case Studies about Security and VPN
- Definition of Network Security
- OpenLearn - Network Security

2.5 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.

2.5.1 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 information 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]

2.5.2 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 it 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 TMSI 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](#)

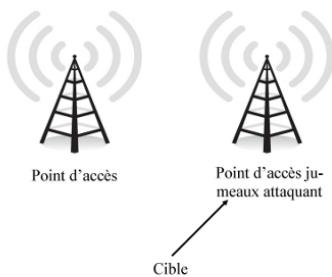
An attacker can try to eavesdrop on Wi-Fi communica-

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]



Access Point twins

tions 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

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.

2.5.3 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.

2.5.4 Attacks based on hardware vulnerabilities

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]

2.5.5 Password cracking

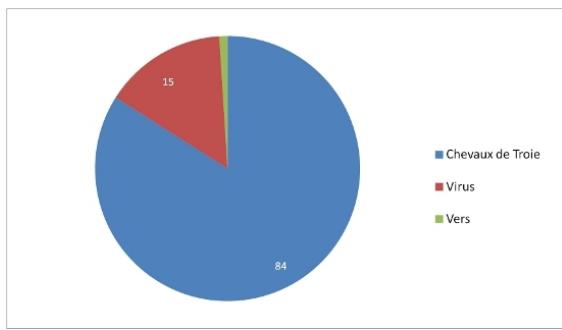
In 2010, researcher from the University of Pennsylvania investigated the possibility of [cracking](#) a device's password through a [smudge](#) attack (literally imaging the finger smudges on the screen to discern the user's password).^[25] The researchers were able to discern the device password up to 68% of the time under certain conditions.^[25]

2.5.6 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.^[26] 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.

^[27]



Types of malware based on their number of smartphones in 2009

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.^[28]

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 acquaintances. 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:^[29]

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:^[30]

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:^[31]

- 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](#)

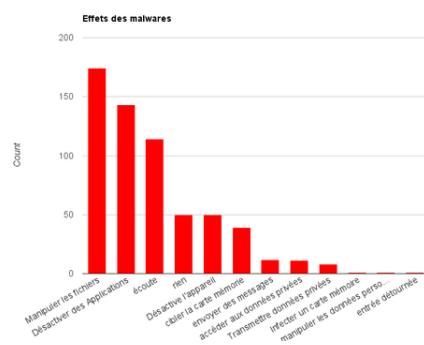
- **Cabir** (also known as **Caribe**, **SymbOS/Cabir**, **Symbian/Cabir** and **EPOC.cabir**) is the name of a computer worm developed in 2004 that is designed to infect mobile phones running Symbian OS. It is believed to be the first computer worm that can infect mobile phones
- **Commwarrior**, found 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 “Red-Browser” 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.^[32]
- **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.^[33] 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.

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.^[27]



Effects of Malware

We can see from the graph that at least 50 malwares exhibit no negative behavior, except their ability to spread.^[27]

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.

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.

2.5.7 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 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**. It is theoretically impossible for a computer to solve such a test, and therefore suspicious activities may be subject to approval or denial by the user.^[50]

Biometric identification Another method to use is biometrics.^[51] 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.^[52]

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.^[53]

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, communicated, 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.^{[54][55]}

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][56]}

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.^[57]

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.^{[58][59]}

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.^[57]

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.^[60]

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.^{[61][56][62]}

Be careful Protection of a user's phone through simple gestures and precautions, such as locking the smart-

phone 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.^{[63][64]}

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.^[65]

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.^[66]

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.^[67]

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.^[55]

Next Generation of mobile security

There is expected to be four mobile environments that will make up the security framework:^[68]

Rich operating system In this category will fall traditional Mobile OS like Android, iOS, Symbian OS or Windows Phone. They will provide the traditional functionality 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.

2.5.8 See also

- Telephone tapping
- Phone hacking
- Browser security
- Computer security
- Mobile virus
- Wireless Public Key Infrastructure (WPKI)
- Wireless security
- Mobile secure gateway

2.5.9 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 & Stärder 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 & Dallet 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] Guo, Wang & Zhu 2004, p. 2.
- [27] Schmidt et al. 2009a, p. 3.
- [28] 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*.
- [29] Becher 2009, p. 87.
- [30] Becher 2009, p. 88.
- [31] Mickens & Noble 2005, p. 1.
- [32] Raboin 2009, p. 272.
- [33] Töyssy & Helenius 2006, p. 114.
- [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] Becher et al. 2011, p. 109.
- [51] Thirumathyam & Derawi 2010, p. 1.
- [52] Schmidt, Schmidt & Clausen 2008, p. 7-12.
- [53] Becher 2009, p. 126.
- [54] Becher et al. 2011, p. 101.
- [55] Ruff 2011, p. 11.
- [56] Hogben & Dekker 2010, p. 45.
- [57] Becher 2009, p. 13.
- [58] Becher 2009, p. 34.
- [59] Ruff 2011, p. 7.
- [60] Hogben & Dekker 2010, p. 46-48.
- [61] Ruff 2011, p. 7-8.
- [62] Shabtai et al. 2009, p. 8-9.
- [63] Hogben & Dekker 2010, p. 43.
- [64] Hogben & Dekker 2010, p. 47.
- [65] Hogben & Dekker 2010, p. 43-45.
- [66] 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.
- [67] Becher 2009, p. 40.
- [68] [http://www.insidesecure.com/Markets-solutions/
Payment-and-Mobile-Banking/Mobile-Security](http://www.insidesecure.com/Markets-solutions/Payment-and-Mobile-Banking/Mobile-Security)
- 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.
- Dagon, David; Martin, Tom; Starger, 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.

2.5.10 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.

- 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).
- 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).
- 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.
- Siemens (2010). "Series M Siemens SMS DoS Vulnerability".

2.5.11 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.
- 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.
- 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*

Websites

- European Telecommunications Standards Institute (2011). "3GPP Confidentiality and Integrity Algorithms & UEA1 UIA1". Archived from the original on 12 May 2012.

Smartphone systems. 8th Annual Communication Networks and Services Research Conference. doi:10.1109/CNSR.2010.24. ISBN 978-1-4244-6248-3.

2.5.12 External links

- How Applications Lead your mobile to be Hacked - Ujjwal Sahay
- Android iOS Mobile Security Review

Chapter 3

Threats

3.1 Cybercrime

Computer crime, or **cybercrime**, is any crime that involves a computer and a network.^[1] The computer may have been used in the commission of a crime, or it may be the target.^[2] Dr. Debarati Halder and Dr. K. Jaishankar (2011) define Cybercrimes as: "Offences that are committed against individuals or groups of individuals with a criminal motive to intentionally harm the reputation of the victim or cause physical or mental harm, or loss, to the victim directly or indirectly, using modern telecommunication networks such as Internet (Chat rooms, emails, notice boards and groups) and mobile phones (SMS/MMS)".^[3] Such crimes may threaten a nation's security and financial health. Issues surrounding these types of crimes have become high-profile, particularly those surrounding hacking, copyright infringement, child pornography, and child grooming. There are also problems of privacy when confidential information is intercepted or disclosed, lawfully or otherwise. Dr. Debarati Halder and Dr. K. Jaishankar (2011) further define cybercrime from the perspective of gender and defined 'cybercrime against women' as "Crimes targeted against women with a motive to intentionally harm the victim psychologically and physically, using modern telecommunication networks such as internet and mobile phones".^[3]

Internationally, both governmental and non-state actors engage in cybercrimes, including espionage, financial theft, and other cross-border crimes. Activity crossing international borders and involving the interests of at least one nation state is sometimes referred to as **cyberwarfare**. The international legal system is attempting to hold actors accountable for their actions through the International Criminal Court.^[4]

A report (sponsored by McAfee) estimates that the annual damage to the global economy is at \$445 billion,^[5] however, a Microsoft report shows that such survey-based estimates are "hopelessly flawed" and exaggerate the true losses by orders of magnitude.^[6] Approximately \$1.5 billion was lost in 2012 to online credit and debit card fraud in the US.^[7]

3.1.1 Classification

Computer crime encompasses a broad range of activities.

Fraud and financial crimes

Main article: [Internet fraud](#)

Computer fraud is any dishonest misrepresentation of fact intended to let another to do or refrain from doing something which causes loss. In this context, the fraud will result in obtaining a benefit by:

- Altering in an unauthorized way. This requires little technical expertise and is common form of theft by employees altering the data before entry or entering false data, or by entering unauthorized instructions or using unauthorized processes;
- Altering, destroying, suppressing, or stealing output, usually to conceal unauthorized transactions. This is difficult to detect;
- Altering or deleting stored data;

Other forms of fraud may be facilitated using computer systems, including bank fraud, carding, identity theft, extortion, and theft of classified information.

A variety of [internet scams](#), many based on [phishing](#) and [social engineering](#), target consumers and businesses.

Cyberterrorism

Main article: [Cyberterrorism](#)

Government officials and [Information Technology](#) security specialists have documented a significant increase in Internet problems and server scans since early 2001. But there is a growing concern among federal officials that such intrusions are part of an organized effort by **cyberterrorists**, foreign intelligence services, or other groups to map potential security holes in critical systems. A cyberterrorist is someone who intimidates or coerces a

government or organization to advance his or her political or social objectives by launching a computer-based attack against computers, networks, or the information stored on them.

Cyberterrorism in general, can be defined as an act of terrorism committed through the use of cyberspace or computer resources (Parker 1983). As such, a simple propaganda in the Internet, that there will be bomb attacks during the holidays can be considered cyberterrorism. As well there are also hacking activities directed towards individuals, families, organized by groups within networks, tending to cause fear among people, demonstrate power, collecting information relevant for ruining peoples' lives, robberies, blackmailing etc.^[8]

Cyberextortion

Main article: Extortion

Cyberextortion occurs when a website, e-mail server, or computer system is subjected to or threatened with repeated denial of service or other attacks by malicious hackers. These hackers demand money in return for promising to stop the attacks and to offer "protection". According to the Federal Bureau of Investigation, cyberextortionists are increasingly attacking corporate websites and networks, crippling their ability to operate and demanding payments to restore their service. More than 20 cases are reported each month to the FBI and many go unreported in order to keep the victim's name out of the public domain. Perpetrators typically use a distributed denial-of-service attack.^[9]

An example of cyberextortion was the attack on Sony Pictures of 2014.^[10]

Cyberwarfare



Sailors analyze, detect and defensively respond to unauthorized activity within U.S. Navy information systems and computer networks

Main article: Cyberwarfare

The U.S. Department of Defense (DoD) notes that the cyberspace has emerged as a national-level concern through several recent events of geo-strategic significance. Among those are included, the attack on Estonia's infrastructure in 2007, allegedly by Russian hackers. "In August 2008, Russia again allegedly conducted cyberattacks, this time in a coordinated and synchronized kinetic and non-kinetic campaign against the country of Georgia. Fearing that such attacks may become the norm in future warfare among nation-states, the concept of cyberspace operations impacts and will be adapted by warfighting military commanders in the future.^[11]

Computer as a target

These crimes are committed by a selected group of criminals. Unlike crimes using the computer as a tool, these crimes require the technical knowledge of the perpetrators. These crimes are relatively new, having been in existence for only as long as computers have - which explains how unprepared society and the world in general is towards combating these crimes. There are numerous crimes of this nature committed daily on the internet:

Crimes that primarily target computer networks or devices include:

- Computer viruses
- Denial-of-service attacks
- Malware (malicious code)

Computer as a tool

Main articles: Internet fraud, Spamming, Phishing and Carding (fraud)

When the individual is the main target of cybercrime, the computer can be considered as the tool rather than the target. These crimes generally involve less technical expertise. Human weaknesses are generally exploited. The damage dealt is largely psychological and intangible, making legal action against the variants more difficult. These are the crimes which have existed for centuries in the offline world. Scams, theft, and the likes have existed even before the development in high-tech equipment. The same criminal has simply been given a tool which increases his potential pool of victims and makes him all the harder to trace and apprehend.^[12]

Crimes that use computer networks or devices to advance other ends include:

- Fraud and identity theft (although this increasingly uses malware, hacking and/or phishing, making it an example of both "computer as target" and "computer as tool" crime)

- Information warfare
- Phishing scams
- Spam
- Propagation of illegal obscene or offensive content, including harassment and threats

The unsolicited sending of bulk email for commercial purposes (spam) is unlawful in some jurisdictions.

Phishing is mostly propagated via email. Phishing emails may contain links to other websites that are affected by malware.^[13] Or, they may contain links to fake online banking or other websites used to steal private account information.

Obscene or offensive content The content of websites and other electronic communications may be distasteful, obscene or offensive for a variety of reasons. In some instances these communications may be legal.

The extent to which these communications are unlawful varies greatly between countries, and even within nations. It is a sensitive area in which the courts can become involved in arbitrating between groups with strong beliefs.

One area of Internet pornography that has been the target of the strongest efforts at curtailment is child pornography.

Harassment Main articles: Cyberbullying, Online predator, Cyberstalking, Hate Crime and Internet troll

Whereas content may be offensive in a non-specific way, harassment directs obscenities and derogatory comments at specific individuals focusing for example on gender, race, religion, nationality, sexual orientation. This often occurs in chat rooms, through newsgroups, and by sending hate e-mail to interested parties. Harassment targeting women and children in the internet also includes revenge pornography. Dr. Debarati Halder and Dr. K. Jaisankar (2013) defined online revenge pornography as “an act whereby the perpetrator satisfies his anger and frustration for a broken relationship through publicizing false, sexually provocative portrayal of his/her victim, by misusing the information that he may have known naturally, and that he may have stored in his personal computer, or may have been conveyed to his electronic device by the victim herself, or may have been stored in the device with the consent of the victim herself; and which may essentially have been done to publicly defame the victim.”^{[14][15]}

There are instances where committing a crime using a computer can lead to an enhanced sentence. For example, in the case of *United States v. Neil Scott Kramer*, Kramer was served an enhanced sentence according to the U.S. Sentencing Guidelines Manual §2G1.3(b)(3)^[16]

for his use of a cell phone to “persuade, induce, entice, coerce, or facilitate the travel of, the minor to engage in prohibited sexual conduct.” Kramer argued that this claim was insufficient because his charge included persuading through a computer device and his cellular phone technically is not a computer. Although Kramer tried to argue this point, U.S. Sentencing Guidelines Manual states that the term computer “means an electronic, magnetic, optical, electrochemical, or other high speed data processing device performing logical, arithmetic, or storage functions, and includes any data storage facility or communications facility directly related to or operating in conjunction with such device.”^[17]

Connecticut was the first state to pass a statute making it a criminal offense to harass someone by computer. Michigan, Arizona, and Virginia and South Carolina^[18] have also passed laws banning harassment by electronic means.^{[19][20]}

Harassment as defined in the U.S. computer statutes is typically distinct from cyberbullying, in that the former usually relates to a person’s “use a computer or computer network to communicate obscene, vulgar, profane, lewd, lascivious, or indecent language, or make any suggestion or proposal of an obscene nature, or threaten any illegal or immoral act,” while the latter need not involve anything of a sexual nature.

Although freedom of speech is protected by law in most democratic societies (in the US this is done by the First Amendment), it does not include all types of speech. In fact spoken or written “true threat” speech/text is criminalized because of “intent to harm or intimidate”, that also applies for online or any type of network related threats in written text or speech.^[21] The US Supreme Court definition of “true threat” is “statements where the speaker means to communicate a serious expression of an intent to commit an act of unlawful violence to a particular individual or group”.^[21]

Drug trafficking Main article: Darknet market

“Drug traffickers are increasingly taking advantage of the Internet” according to cyber authorities and personnel.” to sell their illegal substances through encrypted e-mail and other Internet Technology. Some drug traffickers arrange deals at internet cafes, use courier Web sites to track illegal packages of pills, and swap recipes for amphetamines in restricted-access chat rooms. The dark web site Silk Road was a major online marketplace for drugs before it was shut down by law enforcement (then reopened under new management, and then shut down by law enforcement again).

The rise in Internet drug trades could also be attributed to the lack of face-to-face communication. These virtual exchanges allow more intimidated individuals to more comfortably purchase illegal drugs. The sketchy effects that

are often associated with drug trades are severely minimized and the filtering process that comes with physical interaction fades away.

3.1.2 Documented cases

One of the highest profiled banking computer crime occurred during a course of three years beginning in 1970. The chief teller at the Park Avenue branch of New York's Union Dime Savings Bank embezzled over \$1.5 million from hundreds of accounts.^[22]

A hacking group called MOD (Masters of Deception), allegedly stole passwords and technical data from Pacific Bell, Nynex, and other telephone companies as well as several big credit agencies and two major universities. The damage caused was extensive, one company, Southwestern Bell suffered losses of \$370,000 alone.^[22]

In 1983, a nineteen-year-old UCLA student used his PC to break into a Defense Department international communications system.^[22]

Between 1995 and 1998 the News corp satellite pay to view encrypted SKY-TV service was hacked several times during an ongoing technological arms race between a pan-European hacking group and News corp. The original motivation of the hackers was to watch Star Trek re-runs in Germany; which was something which News corp did not have the copyright to allow.^[23]

On 26 March 1999, the Melissa worm infected a document on a victim's computer, then automatically sent that document and a copy of the virus spread via e-mail to other people.

In February 2000, an individual going by the alias of MafiaBoy began a series denial-of-service attacks against high profile websites, including Yahoo!, Amazon.com, Dell, Inc., E*TRADE, eBay, and CNN. About fifty computers at Stanford University, and also computers at the University of California at Santa Barbara, were amongst the zombie computers sending pings in DDoS attacks. On 3 August 2000, Canadian federal prosecutors charged MafiaBoy with 54 counts of illegal access to computers, plus a total of ten counts of mischief to data for his attacks.

The Russian Business Network (RBN) was registered as an internet site in 2006. Initially, much of its activity was legitimate. But apparently the founders soon discovered that it was more profitable to host illegitimate activities and started hiring its services to criminals. The RBN has been described by VeriSign as "the baddest of the bad".^[24] It offers web hosting services and internet access to all kinds of criminal and objectionable activities, with an individual activities earning up to \$150 million in one year. It specialized in and in some cases monopolized personal identity theft for resale. It is the originator of MPack and an alleged operator of the now defunct Storm botnet.

On 2 March 2010, Spanish investigators arrested 3 in infection of over 13 million computers around the world. The "botnet" of infected computers included PCs inside more than half of the Fortune 1000 companies and more than 40 major banks, according to investigators.

In August 2010 the international investigation Operation Delego, operating under the aegis of the Department of Homeland Security, shut down the international pedophile ring Dreamboard. The website had approximately 600 members, and may have distributed up to 123 terabytes of child pornography (roughly equivalent to 16,000 DVDs). To date this is the single largest U.S. prosecution of an international child pornography ring; 52 arrests were made worldwide.^[25]

On March 1, 2011 at Lassiter High School, two students were accused of impersonation of a staff member via cybercrime, but both claimed they were uninvolved. The offense was made a felony in the Cobb County School District two months after the impersonation had happened. Shortly afterwards, the head of the LHS School Board said "The teacher just wouldn't do this at all". The case ended on May 9, and no evidence was found.

In June 2012 LinkedIn and eHarmony were attacked, compromising 65 million password hashes. 30,000 passwords were cracked and 1.5 million eHarmony passwords were posted online.^[26]

December 2012 Wells Fargo website experienced a denial of service attack. Potentially compromising 70 million customers and 8.5 million active viewers. Other banks thought to be compromised: Bank of America, J. P. Morgan U.S. Bank, and PNC Financial Services.^[27]

In January 2012 Zappos.com experienced a security breach after as many as 24 million customers' credit card numbers, personal information, billing and shipping addresses had been compromised.^[28]

April 23, 2013 saw the Associated Press' Twitter account's hacking to release a hoax tweet about fictional attacks in the White House that left President Obama injured.^[29] This erroneous tweet resulted in a brief plunge of 130 points from the Dow Jones Industrial Average, removal of \$136 billion from S&P 500 index,^[30] and the temporary suspension of their Twitter account. The Dow Jones later restored its session gains.

3.1.3 Combating computer crime

Diffusion of Cybercrime

The broad diffusion of cybercriminal activities is an issue in computer crimes detection and prosecution. According to Jean-Loup Richet (Research Fellow at ESSEC ISIS), technical expertise and accessibility no longer act as barriers to entry into cybercrime.^[31] Indeed, hacking is much less complex than it was a few years ago, as hacking communities have greatly diffused their knowledge

through the Internet. Blogs and communities have hugely contributed to information sharing: beginners could benefit from older hackers' knowledge and advice. Furthermore, Hacking is cheaper than ever: before the cloud computing era, in order to spam or scam one needed a dedicated server, skills in server management, network configuration and maintenance, knowledge of Internet service provider standards, etc. By comparison, a mail software-as-a-service is a scalable, inexpensive, bulk, and transactional e-mail-sending service for marketing purposes and could be easily set up for spam.^[32] Jean-Loup Richet explains that cloud computing could be helpful for a cybercriminal as a way to leverage his attack - brute-forcing a password, improve the reach of a botnet, or facilitating a spamming campaign.^[33]

Investigation

A computer can be a source of evidence (see digital forensics). Even where a computer is not directly used for criminal purposes, it may contain records of value to criminal investigators in the form of a logfile. In most countries Internet Service Providers are required, by law, to keep their logfiles for a predetermined amount of time. For example; a European wide directive^[34] (applicable to all EU member states) states that all E-mail traffic should be retained for a minimum of 12 months.

Legislation

Due to easily exploitable laws, cybercriminals use developing countries in order to evade detection and prosecution from law enforcement. In developing countries, such as the Philippines, laws against cybercrime are weak or sometimes nonexistent. These weak laws allow cybercriminals to strike from international borders and remain undetected. Even when identified, these criminals avoid being punished or extradited to a country, such as the United States, that has developed laws that allow for prosecution. While this proves difficult in some cases, agencies, such as the FBI, have used deception and subterfuge to catch criminals. For example, two Russian hackers had been evading the FBI for some time. The FBI set up a fake computing company based in Seattle, Washington. They proceeded to lure the two Russian men into the United States by offering them work with this company. Upon completion of the interview, the suspects were arrested outside of the building. Clever tricks like this are sometimes a necessary part of catching cybercriminals when weak legislation makes it impossible otherwise.^[35]

President Barack Obama released in an executive order in April 2015 to combat cybercrime. The executive order allows the United States to freeze assets of convicted cybercriminals and block their economic activity within the United States. This is some of the first solid legislation that combats cybercrime in this way.^[36]

Penalties

Penalties for computer related crimes in New York State can range from a fine and a short period of jail time for a Class A misdemeanor such as unauthorized use of a computer up to computer tampering in the first degree which is a Class C felony and can carry 3 to 15 years in prison.^{[37][38]}

However, some hackers have been hired as information security experts by private companies due to their inside knowledge of computer crime, a phenomenon which theoretically could create perverse incentives. A possible counter to this is for courts to ban convicted hackers from using the internet or computers, even after they have been released from prison – though as computers and the internet become more and more central to everyday life, this type of punishment may be viewed as more and more harsh and draconian. However, nuanced approaches have been developed that manage cyberoffender behavior without resorting to total computer and/or Internet bans.^[39] These approaches involve restricting individuals to specific devices which are subject to computer monitoring and/or computer searches by probation and/or parole officers.^[40]

3.1.4 See also

- Computer trespass
- Cyber-
- Cyberbullying
- Cyberheist
- Cyberterrorism
- Economic and Industrial Espionage
- Federal Bureau of Investigation (FBI)
- Hacking
- Immigration and Customs Enforcement (ICE)
- Internet homicide
- Internet stalking
- Internet suicide
- Internet War
- INTERPOL
- Legal aspects of computing
- List of computer criminals
- Metasploit Project

- Online predator
- Organized crime
- Penetration test
- Personal Jurisdiction over International Defendants in US Courts
- Police National E-Crime Unit
- Protected computer
- Techno-thriller
- United States Secret Service
- White collar crime

3.1.5 References

- [1] Moore, R. (2005) "Cyber crime: Investigating High-Technology Computer Crime," Cleveland, Mississippi: Anderson Publishing.
- [2] Warren G. Kruse, Jay G. Heiser (2002). *Computer forensics: incident response essentials*. Addison-Wesley. p. 392. ISBN 0-201-70719-5.
- [3]
 - Halder, D., & Jaishankar, K. (2011) Cyber crime and the Victimization of Women: Laws, Rights, and Regulations. Hershey, PA, USA: IGI Global. ISBN 978-1-60960-830-9
- [4] "Cyber Warfare And The Crime Of Aggression: The Need For Individual Accountability On Tomorrow'S Battlefield". Law.duke.edu. Retrieved 2011-11-10.
- [5] "Cyber crime costs global economy \$445 billion a year: report". Reuters. 2014-06-09. Retrieved 2014-06-17.
- [6] "Sex, Lies and Cybercrime Surveys" (PDF). Microsoft. 2011-06-15. Retrieved 2015-03-11.
- [7] "#Cybercrime— what are the costs to victims - North Denver News". *North Denver News*. Retrieved 16 May 2015.
- [8] "Future Crimes". Retrieved 8 March 2015.
- [9] <http://www.ere-security.ca/PDF/Cyberextortion%20by%20DoS,%20Risk%20Magazine%20June%202006.pdf>
- [10] Mohanta, Abhijit (6 December 2014). "Latest Sony Pictures Breach : A Deadly Cyber Extortion". Retrieved 20 September 2015.
- [11] War is War? The utility of cyberspace operations in the contemporary operational environment
- [12] "Cyber Crime definition".
- [13] "Save browsing". google.
- [14]
 - Halder, D., & Jaishankar, K. (2013) Revenge Porn by Teens in the United States and India: A Socio-legal Analysis. *International Annals of Criminology*, 51(1-2), 85-111.
- [15] "Revenge Porn by Teens in the United States and India: A Socio-Legal Analysis". Retrieved 16 May 2015.
- [16] "2011 U.S. Sentencing Guidelines Manual § 2G1.3(b)(3)".
- [17] "United States of America v. Neil Scott Kramer". Retrieved 2013-10-23.
- [18] "South Carolina". Retrieved 16 May 2015.
- [19] Archived April 10, 2008 at the Wayback Machine
- [20] "Section 18.2-152.7:1". *Code of Virginia*. Legislative Information System of Virginia. Retrieved 2008-11-27.
- [21] Susan W. Brenner, *Cybercrime: Criminal Threats from Cyberspace*, ABC-CLIO, 2010, pp. 91
- [22] Weitzer, Ronald (2003). *Current Controversies in Criminology*. Upper Saddle River, New Jersey: Pearson Education Press. p. 150.
- [23] David Mann And Mike Sutton (2011-11-06). ">>>Netcrime". Bjc.oxfordjournals.org. Retrieved 2011-11-10.
- [24] "A walk on the dark side". The Economist. 2007-09-30.
- [25] "DHS: Secretary Napolitano and Attorney General Holder Announce Largest U.S. Prosecution of International Criminal Network Organized to Sexually Exploit Children". Dhs.gov. Retrieved 2011-11-10.
- [26] Salvador Rodriguez (June 6, 2012). "Like LinkedIn, eHarmony is hacked; 1.5 million passwords stolen". *Los Angeles Times*.
- [27] Rick Rothacker (Oct 12, 2012). "Cyber attacks against Wells Fargo "significant," handled well: CFO". *Reuters*.
- [28] DAVID K. LI (January 17, 2012). "Zappos cyber attack". *New York Post*.
- [29] "AP Twitter Hack Falsely Claims Explosions at White House". Samantha Murphy. April 23, 2013. Retrieved April 23, 2013.
- [30] "Fake Tweet Erasing \$136 Billion Shows Markets Need Humans". Bloomberg. April 23, 2013. Retrieved April 23, 2013.
- [31] Richet, Jean-Loup (2013). "From Young Hackers to Crackers". *International Journal of Technology and Human Interaction* 9 (1).
- [32] Richet, Jean-Loup (2011). "Adoption of deviant behavior and cybercrime 'Know how' diffusion". *York Deviancy Conference*.
- [33] Richet, Jean-Loup (2012). "How to Become a Black Hat Hacker? An Exploratory Study of Barriers to Entry Into Cybercrime.". *17th AIM Symposium*.
- [34] Data Retention (EC Directive) Regulations SI 2007/2199
- [35] Kshetri, Nir. "Diffusion and Effects of Cyber Crime in Developing Countries".

- [36] Northam, Jackie. "U.S. Creates First Sanctions Program Against Cybercriminals".
- [37] Kenniff, Raiser. "New York Internet Crimes Laws".
- [38] Computer fraud charges in New York. May 2011. Bukh Law Firm, PC - 14 Wall St, New York NY 10005 - (212) 729-1632. New York computer fraud lawyer
- [39] "Managing the Risks Posed by Offender Computer Use - Perspectives" (PDF). December 2011.
- [40] Bowker, Art (2012). *The Cybercrime Handbook for Community Corrections: Managing Risk in the 21st Century*. Springfield: Thomas. ISBN 9780398087289.
- Parker D (1983) *Fighting Computer Crime*, U.S.: Charles Scribner's Sons.
- Pattavina, A. (ed) *Information Technology and the Criminal Justice System*, Thousand Oaks, CA: Sage.
- Paul Taylor. *Hackers: Crime in the Digital Sublime* (November 3, 1999 ed.). Routledge; 1 edition. p. 200. ISBN 0-415-18072-4.
- Robertson, J. (2010, March 2). Authorities bust 3 in infection of 13m computers. Retrieved March 26, 2010, from Boston News: Boston.com
- Walden, I. (2007) *Computer Crimes and Digital Investigations*, Oxford: Oxford University Press.
- Rolón, Darío N. Control, vigilancia y respuesta penal en el ciberespacio, Latin American's New Security Thinking, Clacso, 2014, pp. 167/182
- Richet, J.L. (2013) From Young Hackers to Crackers, *International Journal of Technology and Human Interaction (IJTHI)*, 9(3), 53-62.
- Wall, D.S. (2007) *Cybercrimes: The transformation of crime in the information age*, Cambridge: Polity.
- Williams, M. (2006) *Virtually Criminal: Crime, Deviance and Regulation Online*, Routledge, London.
- Yar, M. (2006) *Cybercrime and Society*, London: Sage.

3.1.6 Further reading

- Balkin, J., Grimmelmann, J., Katz, E., Kozlowski, N., Wagman, S. & Zarsky, T. (2006) (eds) *Cybercrime: Digital Cops in a Networked Environment*, New York University Press, New York.
- Bowker, Art (2012) "The Cybercrime Handbook for Community Corrections: Managing Risk in the 21st Century" Charles C. Thomas Publishers, Ltd. Springfield.
- Brenner, S. (2007) *Law in an Era of Smart Technology*, Oxford: Oxford University Press
- Csonka P. (2000) Internet Crime; the Draft council of Europe convention on cyber-crime: A response to the challenge of crime in the age of the internet? *Computer Law & Security Report* Vol.16 no.5.
- Easttom C. (2010) *Computer Crime Investigation and the Law*
- Fafinski, S. (2009) *Computer Misuse: Response, regulation and the law* Cullompton: Willan
- Glenny, Misha, *DarkMarket : cyberthieves, cyber-cops, and you*, New York, NY : Alfred A. Knopf, 2011. ISBN 978-0-307-59293-4
- Grabosky, P. (2006) *Electronic Crime*, New Jersey: Prentice Hall
- Halder, D., & Jaishankar, K. (2011) *Cyber crime and the Victimization of Women: Laws, Rights, and Regulations*. Hershey, PA, USA: IGI Global. ISBN 978-1-60960-830-9
- Jaishankar, K. (Ed.) (2011). *Cyber Criminology: Exploring Internet Crimes and Criminal behavior*. Boca Raton, FL, USA: CRC Press, Taylor and Francis Group.
- McQuade, S. (2006) *Understanding and Managing Cybercrime*, Boston: Allyn & Bacon.
- McQuade, S. (ed) (2009) *The Encyclopedia of Cybercrime*, Westport, CT: Greenwood Press.
- Centre for Cyber Victim Counselling (CCVC)
- The American Society of Digital Forensics & eDiscovery - Cybercrime Information
- A Guide to Computer Crime from legal.practitioner.com
- *International Journal of Cyber Criminology*
- Virtual Forum Against Cybercrime
- High Technology Crime Investigation Association
- Computer Crime Research Center
- CyberCrime Asia Research Center - Information about computer crime, Internet fraud and CyberTerrorism in Asia
- Information and Research Center for Cybercrime Germany

Government resources

- Cybercrime.gov from the United States Department of Justice
- National Institute of Justice Electronic Crime Program from the United States Department of Justice
- FBI Cyber Investigators home page
- US Secret Service Computer Fraud
- Australian High Tech Crime Centre

Indian raw agency research and analysis wing

3.2 Vulnerability (computing)

In computer security, a **vulnerability** is a weakness which allows an attacker to reduce a system's information assurance. Vulnerability is the intersection of three elements: a system susceptibility or flaw, attacker access to the flaw, and attacker capability to exploit the flaw.^[1] To exploit a vulnerability, an attacker must have at least one applicable tool or technique that can connect to a system weakness. In this frame, vulnerability is also known as the attack surface.

Vulnerability management is the cyclical practice of identifying, classifying, remediating, and mitigating vulnerabilities.^[2] This practice generally refers to software vulnerabilities in computing systems.

A security risk may be classified as a vulnerability. The use of vulnerability with the same meaning of risk can lead to confusion. The risk is tied to the potential of a significant loss. Then there are vulnerabilities without risk: for example when the affected asset has no value. A vulnerability with one or more known instances of working and fully implemented attacks is classified as an exploitable vulnerability — a vulnerability for which an exploit exists. The **window of vulnerability** is the time from when the security hole was introduced or manifested in deployed software, to when access was removed, a security fix was available/deployed, or the attacker was disabled—see zero-day attack.

Security bug (security defect) is a narrower concept: there are vulnerabilities that are not related to software: hardware, site, personnel vulnerabilities are examples of vulnerabilities that are not software security bugs.

Constructs in programming languages that are difficult to use properly can be a large source of vulnerabilities.

3.2.1 Definitions

ISO 27005 defines **vulnerability** as:^[3]

A weakness of an asset or group of assets that can be exploited by one or more threats

where an *asset* is anything that has value to the organization, its business operations and their continuity, including information resources that support the organization's mission^[4]

IETF RFC 2828 define **vulnerability** as:^[5]

A flaw or weakness in a system's design, implementation, or operation and management that could be exploited to violate the system's security policy

The Committee on National Security Systems of United States of America defined **vulnerability** in CNSS Instruction No. 4009 dated 26 April 2010 National Information Assurance Glossary:^[6]

Vulnerability — Weakness in an IS, system security procedures, internal controls, or implementation that could be exploited

Many NIST publications define **vulnerability** in IT context in different publications: FISMApedia^[7] term^[8] provide a list. Between them SP 800-30,^[9] give a broader one:

A flaw or weakness in system security procedures, design, implementation, or internal controls that could be exercised (accidentally triggered or intentionally exploited) and result in a security breach or a violation of the system's security policy.

ENISA defines **vulnerability** in^[10] as:

The existence of a weakness, design, or implementation error that can lead to an unexpected, undesirable event [G.11] compromising the security of the computer system, network, application, or protocol involved.(ITSEC)

The Open Group defines **vulnerability** in^[11] as:

The probability that threat capability exceeds the ability to resist the threat.

Factor Analysis of Information Risk (FAIR) defines **vulnerability** as:^[12]

The probability that an asset will be unable to resist the actions of a threat agent

According FAIR vulnerability is related to Control Strength, i.e. the strength of a control as compared to a standard measure of force and the threat Capabilities, i.e. the probable level of force that a threat agent is capable of applying against an asset.

ISACA defines **vulnerability** in Risk It framework as:

A weakness in design, implementation, operation or internal control

Data and Computer Security: Dictionary of standards concepts and terms, authors Dennis Longley and Michael Shain, Stockton Press, ISBN 0-935859-17-9, defines **vulnerability** as:

- 1) In computer security, a weakness in automated systems security procedures, administrative controls, Internet controls, etc., that could be exploited by a threat to gain unauthorized access to information or to disrupt critical processing.
- 2) In computer security, a weakness in the physical layout, organization, procedures, personnel, management, administration, hardware or software that may be exploited to cause harm to the ADP system or activity.
- 3) In computer security, any weakness or flaw existing in a system. The attack or harmful event, or the opportunity available to a threat agent to mount that attack.

Matt Bishop and Dave Bailey^[13] give the following definition of computer **vulnerability**:

A computer system is composed of states describing the current configuration of the entities that make up the computer system. The system computes through the application of state transitions that change the state of the system. All states reachable from a given initial state using a set of state transitions fall into the class of authorized or unauthorized, as defined by a security policy. In this paper, the definitions of these classes and transitions is considered axiomatic. A vulnerable state is an authorized state from which an unauthorized state can be reached using authorized state transitions. A compromised state is the state so reached. An attack is a sequence of authorized state transitions which end in a compromised state. By definition, an attack begins in a vulnerable state. A vulnerability is a characterization of a vulnerable state which distinguishes it from all non-vulnerable states. If generic, the vulnerability may characterize many vulnerable states; if specific, it may characterize only one...

National Information Assurance Training and Education Center defines **vulnerability**: ^{[14][15]}

A weakness in automated system security procedures, administrative controls, internal controls, and so forth, that could be exploited by a threat to gain unauthorized access to information or disrupt critical processing. 2. A weakness in system security procedures, hardware design, internal controls, etc. , which could be exploited to gain unauthorized access to classified or sensitive information. 3. A weakness in the physical layout, organization, procedures, personnel, management, administration, hardware, or software that may be exploited to cause harm to the ADP system or activity. The presence of a vulnerability does not in itself cause harm; a vulnerability is merely a condition or set of conditions that may allow the ADP system or activity to be harmed by an attack. 4. An assertion primarily concerning entities of the internal environment (assets); we say that an asset (or class of assets) is vulnerable (in some way, possibly involving an agent or collection of agents); we write: $V(i,e)$ where: e may be an empty set. 5. Susceptibility to various threats. 6. A set of properties of a specific internal entity that, in union with a set of properties of a specific external entity, implies a risk. 7. The characteristics of a system which cause it to suffer a definite degradation (incapability to perform the designated mission) as a result of having been subjected to a certain level of effects in an unnatural (manmade) hostile environment.

3.2.2 Vulnerability and risk factor models

A resource (either physical or logical) may have one or more vulnerabilities that can be exploited by a threat agent in a threat action. The result can potentially compromise the **confidentiality**, **integrity** or **availability** of resources (not necessarily the vulnerable one) belonging to an organization and/or others parties involved (customers, suppliers).

The so-called **CIA triad** is the basis of **Information Security**.

An attack can be *active* when it attempts to alter system resources or affect their operation, compromising integrity or availability. A "passive attack" attempts to learn or make use of information from the system but does not affect system resources, compromising confidentiality.^[5]

OWASP (see figure) depicts the same phenomenon in slightly different terms: a threat agent through an attack vector exploits a weakness (vulnerability) of the system and the related security controls, causing a technical impact on an IT resource (asset) connected to a business impact.

The overall picture represents the **risk factors** of the risk scenario.^[16]



OWASP: relationship between threat agent and business impact

3.2.3 Information security management system

A set of policies concerned with information security management, the **information security management system (ISMS)**, has been developed to manage, according to **Risk management** principles, the **countermeasures** in order to ensure the security strategy is set up following the rules and regulations applicable in a country. These countermeasures are also called **Security controls**, but when applied to the transmission of information they are called **security services**.^[17]

3.2.4 Classification

Vulnerabilities are classified according to the asset class they are related to:^[3]

- hardware
 - susceptibility to humidity
 - susceptibility to dust
 - susceptibility to soiling
 - susceptibility to unprotected storage
- software
 - insufficient testing
 - lack of **audit trail**
- network
 - unprotected **communication lines**
 - insecure **network architecture**
- personnel
 - inadequate **recruiting process**
 - inadequate **security awareness**
- site
 - area subject to flood
 - unreliable power source
- organizational
 - lack of regular audits
 - lack of continuity plans
 - lack of security

3.2.5 Causes

- Complexity: Large, complex systems increase the probability of flaws and unintended access points^[18]
- Familiarity: Using common, well-known code, software, operating systems, and/or hardware increases the probability an attacker has or can find the knowledge and tools to exploit the flaw^[19]
- Connectivity: More physical connections, privileges, ports, protocols, and services and time each of those are accessible increase vulnerability^[12]
- Password management flaws: The computer user uses **weak passwords** that could be discovered by brute force. The computer user stores the password on the computer where a program can access it. Users re-use passwords between many programs and websites.^[18]
- Fundamental operating system design flaws: The operating system designer chooses to enforce sub-optimal policies on user/program management. For example operating systems with policies such as default **permit** grant every program and every user full access to the entire computer.^[18] This operating system flaw allows viruses and malware to execute commands on behalf of the administrator.^[20]
- Internet Website Browsing: Some internet websites may contain harmful **Spyware** or **Adware** that can be installed automatically on the computer systems. After visiting those websites, the computer systems become infected and personal information will be collected and passed on to third party individuals.^[21]
- Software bugs: The programmer leaves an exploitable bug in a software program. The software bug may allow an attacker to misuse an application.^[18]
- Unchecked user input: The program assumes that all user input is safe. Programs that do not check user input can allow unintended direct execution of commands or SQL statements (known as **Buffer overflows**, **SQL injection** or other non-validated inputs).^[18]
- Not learning from past mistakes:^{[22][23]} for example most vulnerabilities discovered in **IPv4** protocol software were discovered in the new **IPv6** implementations.^[24]

The research has shown that the most vulnerable point in most information systems is the human user, operator, designer, or other human:^[25] so humans should be considered in their different roles as asset, threat, information resources. Social engineering is an increasing security concern.

3.2.6 Vulnerability consequences

The impact of a security breach can be very high. The fact that IT managers, or upper management, can (easily) know that IT systems and applications have vulnerabilities and do not perform any action to manage the IT risk is seen as a misconduct in most legislations. Privacy law forces managers to act to reduce the impact or likelihood of that security risk. Information technology security audit is a way to let other independent people certify that the IT environment is managed properly and lessen the responsibilities, at least having demonstrated the good faith. Penetration test is a form of verification of the weakness and countermeasures adopted by an organization: a White hat hacker tries to attack an organization's information technology assets, to find out how easy or difficult it is to compromise the IT security.^[26] The proper way to professionally manage the IT risk is to adopt an Information Security Management System, such as ISO/IEC 27002 or Risk IT and follow them, according to the security strategy set forth by the upper management.^[17]

One of the key concept of information security is the principle of **defence in depth**: i.e. to set up a multilayer defence system that can:

- prevent the exploit
- detect and intercept the attack
- find out the threat agents and prosecute them

Intrusion detection system is an example of a class of systems used to detect attacks.

Physical security is a set of measures to protect physically the information asset: if somebody can get physical access to the information asset, it is quite easy to make resources unavailable to its legitimate users.

Some sets of criteria to be satisfied by a computer, its operating system and applications in order to meet a good security level have been developed: ITSEC and Common criteria are two examples.

3.2.7 Vulnerability disclosure

Responsible disclosure (many now refer to it as 'coordinated disclosure' because the first is a biased word) of vulnerabilities is a topic of great debate. As reported by The Tech Herald in August 2010, "Google, Microsoft, TippingPoint, and Rapid7 have recently issued guidelines and statements addressing how they will deal with disclosure going forward."^[27]

A responsible disclosure first alerts the affected vendors confidentially before alerting CERT two weeks later, which grants the vendors another 45 day grace period before publishing a security advisory.

Full disclosure is done when all the details of vulnerability is publicized, perhaps with the intent to put pressure on the software or procedure authors to find a fix urgently.

Well respected authors have published books on vulnerabilities and how to exploit them: **Hacking: The Art of Exploitation Second Edition** is a good example.

Security researchers catering to the needs of the cyberwarfare or cybercrime industry have stated that this approach does not provide them with adequate income for their efforts.^[28] Instead, they offer their exploits privately to enable Zero day attacks.

The never ending effort to find new vulnerabilities and to fix them is called Computer insecurity.

In January 2014 when Google revealed a Microsoft vulnerability before Microsoft released a patch to fix it, a Microsoft representative called for coordinated practices among software companies in revealing disclosures.^[29]

Vulnerability inventory

Mitre Corporation maintains a list of disclosed vulnerabilities in a system called **Common Vulnerabilities and Exposures**, where vulnerability are classified (scored) using **Common Vulnerability Scoring System (CVSS)**.

OWASP collects a list of potential vulnerabilities in order to prevent system designers and programmers from inserting vulnerabilities into the software.^[30]

3.2.8 Vulnerability disclosure date

The time of disclosure of a vulnerability is defined differently in the security community and industry. It is most commonly referred to as "a kind of public disclosure of security information by a certain party". Usually, vulnerability information is discussed on a mailing list or published on a security web site and results in a security advisory afterward.

The **time of disclosure** is the first date a security vulnerability is described on a channel where the disclosed information on the vulnerability has to fulfill the following requirement:

- The information is freely available to the public
- The vulnerability information is published by a trusted and independent channel/source
- The vulnerability has undergone analysis by experts such that risk rating information is included upon disclosure

3.2.9 Identifying and removing vulnerabilities

Many software tools exist that can aid in the discovery (and sometimes removal) of vulnerabilities in a computer system. Though these tools can provide an auditor with a good overview of possible vulnerabilities present, they can not replace human judgment. Relying solely on scanners will yield false positives and a limited-scope view of the problems present in the system.

Vulnerabilities have been found in every major operating system including Windows, Mac OS, various forms of Unix and Linux, OpenVMS, and others. The only way to reduce the chance of a vulnerability being used against a system is through constant vigilance, including careful system maintenance (e.g. applying software patches), best practices in deployment (e.g. the use of firewalls and access controls) and auditing (both during development and throughout the deployment lifecycle).

3.2.10 Examples of vulnerabilities

Vulnerabilities are related to:

- physical environment of the system
- the personnel
- management
- administration procedures and security measures within the organization
- business operation and service delivery
- hardware
- software
- communication equipment and facilities
- and their combinations.

It is evident that a pure technical approach cannot even protect physical assets: one should have administrative procedure to let maintenance personnel to enter the facilities and people with adequate knowledge of the procedures, motivated to follow it with proper care. See Social engineering (security).

Four examples of vulnerability exploits:

- an attacker finds and uses an overflow weakness to install malware to export sensitive data;
- an attacker convinces a user to open an email message with attached malware;
- an insider copies a hardened, encrypted program onto a thumb drive and cracks it at home;
- a flood damages one's computer systems installed at ground floor.

Software vulnerabilities

Common types of software flaws that lead to vulnerabilities include:

- Memory safety violations, such as:
 - Buffer overflows and over-reads
 - Dangling pointers
- Input validation errors, such as:
 - Format string attacks
 - SQL injection
 - Code injection
 - E-mail injection
 - Directory traversal
 - Cross-site scripting in web applications
 - HTTP header injection
 - HTTP response splitting
- Race conditions, such as:
 - Time-of-check-to-time-of-use bugs
 - Symlink races
- Privilege-confusion bugs, such as:
 - Cross-site request forgery in web applications
 - Clickjacking
 - FTP bounce attack
- Privilege escalation
- User interface failures, such as:
 - Warning fatigue^[31] or user conditioning.
 - Blaming the Victim Prompting a user to make a security decision without giving the user enough information to answer it^[32]
 - Race Conditions^{[33][34]}

Some set of coding guidelines have been developed and a large number of static code analysers has been used to verify that the code follows the guidelines.

3.2.11 See also

- Browser security
- Computer emergency response team
- Information security
- Internet security
- Mobile security
- Vulnerability scanner

3.2.12 References

- [1] “The Three Tenets of Cyber Security”. U.S. Air Force Software Protection Initiative. Retrieved 2009-12-15.
- [2] Foreman, P: *Vulnerability Management*, page 1. Taylor & Francis Group, 2010. ISBN 978-1-4398-0150-5
- [3] ISO/IEC, “Information technology -- Security techniques--Information security risk management” ISO/IEC FIDIS 27005:2008
- [4] British Standard Institute, Information technology -- Security techniques -- Management of information and communications technology security -- Part 1: Concepts and models for information and communications technology security management BS ISO/IEC 13335-1-2004
- [5] Internet Engineering Task Force RFC 2828 Internet Security Glossary
- [6] CNSS Instruction No. 4009 dated 26 April 2010
- [7] “FISMApedia”. *fismapedia.org*.
- [8] “Term:Vulnerability”. *fismapedia.org*.
- [9] NIST SP 800-30 Risk Management Guide for Information Technology Systems
- [10] “Glossary”. *eropa.eu*.
- [11] Technical Standard Risk Taxonomy ISBN 1-931624-77-1 Document Number: C081 Published by The Open Group, January 2009.
- [12] “An Introduction to Factor Analysis of Information Risk (FAIR)”, Risk Management Insight LLC, November 2006;
- [13] Matt Bishop and Dave Bailey. A Critical Analysis of Vulnerability Taxonomies. Technical Report CSE-96-11, Department of Computer Science at the University of California at Davis, September 1996
- [14] Schou, Corey (1996). Handbook of INFOSEC Terms, Version 2.0. CD-ROM (Idaho State University & Information Systems Security Organization)
- [15] NIATEC Glossary
- [16] ISACA THE RISK IT FRAMEWORK (registration required)
- [17] Wright, Joe; Harmening, Jim (2009). “15”. In Vacca, John. *Computer and Information Security Handbook*. Morgan Kaufmann Publications. Elsevier Inc. p. 257. ISBN 978-0-12-374354-1.
- [18] Kakareka, Almantas (2009). “23”. In Vacca, John. *Computer and Information Security Handbook*. Morgan Kaufmann Publications. Elsevier Inc. p. 393. ISBN 978-0-12-374354-1.
- [19] Krsul, Ivan (April 15, 1997). “Technical Report CSD-TR-97-026”. The COAST Laboratory Department of Computer Sciences, Purdue University. CiteSeerX: 10.1.1.26.5435.
- [20] “The Six Dumbest Ideas in Computer Security”. *ranum.com*.
- [21] “The Web Application Security Consortium / Web Application Security Statistics”. *webappsec.org*.
- [22] Ross Anderson. Why Cryptosystems Fail. Technical report, University Computer Laboratory, Cambridge, January 1994.
- [23] Neil Schlager. When Technology Fails: Significant Technological Disasters, Accidents, and Failures of the Twentieth Century. Gale Research Inc., 1994.
- [24] Hacking: The Art of Exploitation Second Edition
- [25] Kiountouzis, E. A.; Kokolakis, S. A. *Information systems security: facing the information society of the 21st century*. London: Chapman & Hall, Ltd. ISBN 0-412-78120-4.
- [26] Bavisi, Sanjay (2009). “22”. In Vacca, John. *Computer and Information Security Handbook*. Morgan Kaufmann Publications. Elsevier Inc. p. 375. ISBN 978-0-12-374354-1.
- [27] “The new era of vulnerability disclosure - a brief chat with HD Moore”. *The Tech Herald*.
- [28] “Browse - Content - SecurityStreet”. *rapid7.com*.
- [29] Betz, Chris (11 Jan 2015). “A Call for Better Coordinated Vulnerability Disclosure - MSRC - Site Home - TechNet Blogs”. *blogs.technet.com*. Retrieved 12 January 2015.
- [30] “Category:Vulnerability”. *owasp.org*.
- [31] “Warning Fatigue”. *freedom-to-tinker.com*.
- [32]
- [33] “Jesse Ruderman » Race conditions in security dialogs”. *squarefree.com*.
- [34] “lcamtuf’s blog”. *lcamtuf.blogspot.com*.

3.2.13 External links

- Security advisories links from the Open Directory http://www.dmoz.org/Computers/Security/Advisories_and_Patches/

3.3 Eavesdropping

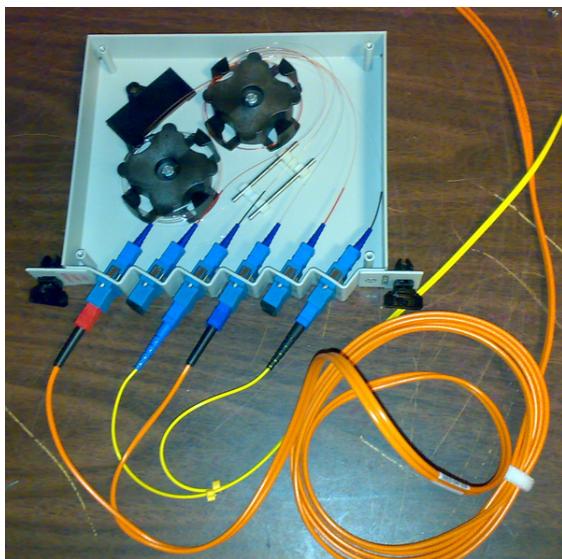
Eavesdropping is secretly listening to the private conversation of others without their consent, as defined by Black’s Law Dictionary.^[1] This is commonly thought to be **unethical** and there is an old **adage** that “eavesdroppers seldom hear anything good of themselves... eavesdroppers always try to listen to matters that concern them.”^[2]



Cardinals in the Vatican, by Henri Adolphe Laissement, 1895



"Belly-buster" hand-crank audio drill, used during the late 1950s and early 1960s to drill holes into masonry for implanting audio devices



A fiber-optic splitter, a modern method of eavesdropping.

3.3.1 Etymology

The verb *eavesdrop* was originally a back-formation of the noun *eavesdropper* ("a person who eavesdrops") which was formed from the unrelated noun *eavesdrop* ("the dripping of water from the eaves of a house; the

ground on which such water falls"). An eavesdropper was one who stood at the eavesdrop (where the water fell, i.e., near the house) so as to overhear what was said inside.^{[3][4]}

3.3.2 Techniques

Eavesdropping can also be done over telephone lines (wiretapping), email, instant messaging, and other methods of communication considered private. (If a message is publicly broadcast, witnessing it is not considered eavesdropping.) VoIP communications software is also vulnerable to electronic eavesdropping via infections such as trojans.

3.3.3 References

- [1] Garner, p. 550
- [2] Ronald R. Kline (2000). *Consumers in the Country*. Baltimore, Md.: Johns Hopkins Univ. Press. p. 46.
- [3] *Shorter Oxford English Dictionary* (6th ed.), Oxford University Press, 2007, ISBN 978-0-19-920687-2
- [4] "eavesdrop". Online Etymology Dictionary.

3.3.4 See also

- Computer surveillance
- ECHELON
- Espionage
- Fiber tapping
- *Katz v. United States* (1967)
- Keystroke logging
- Magic (cryptography)
- Man-in-the-middle attack
- Mass surveillance
- NSA warrantless surveillance controversy (December 2005 – 2006)
- Opportunistic encryption
- Privacy
- Secure communication
- Surveillance
- Telephone tapping
- Ultra

3.3.5 External links

- The dictionary definition of [eavesdropping](#) at Wiktionary
- Media related to [Eavesdropping](#) at Wikimedia Commons

3.4 Exploit (computer security)

An **exploit** (from the English verb *to exploit*, meaning “using something to one’s own advantage”) is a piece of software, a chunk of data, or a sequence of commands that takes advantage of a [bug](#) or [vulnerability](#) in order to cause unintended or unanticipated behavior to occur on computer software, hardware, or something electronic (usually computerized). Such behavior frequently includes things like gaining control of a computer system, allowing privilege escalation, or a denial-of-service attack.

3.4.1 Classification

There are several methods of classifying exploits. The most common is by how the exploit contacts the vulnerable software. A *remote exploit* works over a network and exploits the security vulnerability without any prior access to the vulnerable system. A *local exploit* requires prior access to the vulnerable system and usually increases the privileges of the person running the exploit past those granted by the system administrator. Exploits against client applications also exist, usually consisting of modified servers that send an exploit if accessed with a client application. Exploits against client applications may also require some interaction with the user and thus may be used in combination with the social engineering method. Another classification is by the action against the vulnerable system; unauthorized data access, arbitrary code execution, and denial of service are examples. Many exploits are designed to provide superuser-level access to a computer system. However, it is also possible to use several exploits, first to gain low-level access, then to escalate privileges repeatedly until one reaches root. Normally a single exploit can only take advantage of a specific software vulnerability. Often, when an exploit is published, the vulnerability is fixed through a patch and the exploit becomes obsolete until newer versions of the software become available. This is the reason why some [black hat hackers](#) do not publish their exploits but keep them private to themselves or other hackers. Such exploits are referred to as *zero day exploits* and to obtain access to such exploits is the primary desire of unskilled attackers, often nicknamed script kiddies.^[1]

Types

Exploits are commonly categorized and named by these criteria:

- The type of vulnerability they exploit (See [vulnerabilities](#) for a list)
- Whether they need to be run on the same machine as the program that has the vulnerability (local) or can be run on one machine to attack a program running on another machine (remote).
- The result of running the exploit (EoP, DoS, Spoofing, etc.)

Pivoting

Pivoting refers to a method used by [penetration testers](#) that uses the compromised system to attack other systems on the same network to avoid restrictions such as firewall configurations, which may prohibit direct access to all machines. For example, if an attacker compromises a web server on a corporate network, the attacker can then use the compromised web server to attack other systems on the network. These types of attacks are often called multi-layered attacks. Pivoting is also known as *island hopping*.

Pivoting can further be distinguished into [proxy](#) pivoting and [VPN](#) pivoting:

- **Proxy pivoting** generally describes the practice of channeling traffic through a compromised target using a proxy payload on the machine and launching attacks from the computer.^[2] This type of pivoting is restricted to certain [TCP](#) and [UDP](#) ports that are supported by the proxy.
- **VPN pivoting** enables the attacker to create an encrypted layer to tunnel into the compromised machine to route any network traffic through that target machine, for example, to run a vulnerability scan on the internal network through the compromised machine, effectively giving the attacker full network access as if they were behind the firewall.

Typically, the proxy or VPN applications enabling pivoting are executed on the target computer as the [payload](#) ([software](#)) of an exploit.

3.4.2 See also

- Computer security
- Computer virus
- Crimeware

- Exploit kit
- Hacking: The Art of Exploitation (second edition)
- IT risk
- Metasploit
- Shellcode
- w3af

3.4.3 References

- [1] Whitman, Michael (2012). "Chapter 2: The Need for Security". Principles of Information Security, Fourth Edition. Boston, Mass: Course Technology. p. 53.
- [2] Metasploit Framework Pivoting, Digital Bond: Metasploit Basics – Part 3: Pivoting and Interfaces
- Kahsari Alhadi, Milad. Metasploit Penetration Tester's Guide - Persian, ISBN 978-600-7026-62-5

3.5 Trojan horse (computing)

For other uses, see [Trojan horse \(disambiguation\)](#).

A **Trojan horse**, or **Trojan**, in computing is any malicious computer program which misrepresents itself as 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.^{[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] 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.^[7]

3.5.1 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.
- Spread malware across the network.
- Spy 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, surveillance 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]

3.5.2 Notable examples

- Netbus Advance System Care
- Subseven or Sub7
- Back Orifice
- Beast
- Zeus
- Flashback Trojan (Trojan BackDoor.Flashback)
- ZeroAccess
- Koobface
- Vundo
- Trojan:Win32/Meredrop

3.5.3 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

3.5.4 References

- Carnegie Mellon University (1999): "CERT Advisory CA-1999-02 Trojan Horses", ЎЦ
- [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 Spy-WareLoop.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 - Tech-world.com”. News.techworld.com. Retrieved 2014-01-26.
- [15] BitDefender.com Malware and Spam Survey
- [16] Datta, Ganesh. “What are Trojans?”. SecurAid.

3.5.5 External links

- Trojan Horses at DMOZ

3.6 Computer virus

Not to be confused with **Worm** (software) or **Trojan Horse** (computing).

```

0 00 00-6D 73 62 6C msbl
0 6A 75-73 74 20 77 ast.exe I just w
9 20 4C-4F 56 45 20 ant to say LOVE
0 62 69-6C 6C 79 20 YOU SAN!! billy
0 64 6F-20 79 6F 75 gates why do you
3 20 70-6F 73 73 69 make this possi
0 20 6D-61 6B 69 6E ble ? Stop makin
E 64 20-66 69 78 20 g money and fix
7 61 72-65 21 21 00 your software!!
0 00 00-7F 00 00 00 à ö► H à
0 00 00-01 00 01 00 ã ã à à à
0 00 00-00 00 00 46 á@ L F
C C9 11-9F E8 08 00 ♦Jèèù-ñññ
0 00 03-10 00 00 00 ►H à à à
3 00 00-01 00 04 00 p à à à à

```

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

A **computer virus** is a malware program that, when executed, **replicates** by inserting copies of itself (possibly modified) into other computer programs, data **files**, or the boot sector of the hard drive; when this replication succeeds, the affected areas are then said to be “infected”.^{[1][2][3][4]} Viruses often perform some type of harmful activity on infected hosts, such as stealing **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**,^{[5][6][7]} employing a variety of mechanisms to infect new hosts,^[8] and often using complex anti-detection/stealth strategies to evade antivirus software.^{[9][10][11][12]} 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**.^[13]

Computer viruses currently cause billions of dollars’ worth of economic damage each year,^[14] due to causing systems 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.^[15] 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.^[16]

3.6.1 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.^[17]

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.^[18]

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.^[19] The diversity of software systems on a network limits the destructive potential of viruses and malware.^[20] 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.^{[5][6][7][21]}

Only a few major viruses have hit Macs in the last years.^{[22][23]} The difference in virus vulnerability between Macs and Windows is a chief selling point, one that **Apple** uses in their **Get a Mac** advertising.^[24]

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”.^[25] 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.^[26]

3.6.2 Infection targets and replication techniques

Computer viruses infect a variety of different subsystems on their hosts.^[27] 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).^{[28][29]}

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).^{[30][31][32]}

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**.^{[33][34]}

Boot sector viruses

Boot sector viruses specifically target the **boot sector/Master Boot Record**^[35] (MBR) of the host’s hard drive or removable storage media (**flash drives**, **floppy disks**, etc.).^{[28][36][37]}

3.6.3 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.^[38]

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.^[39]

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.

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. Direct access to files without using the Windows OS is undocumented. 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.^[40]

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**.^{[41][42]}

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.

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 **cryptographic key** in cleartext.^[43] 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.

An old, but compact, encryption involves **XORing** each byte in a virus 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.

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.^{[44][45]} 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.^[46]

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^[47]). 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.^{[48][49]}

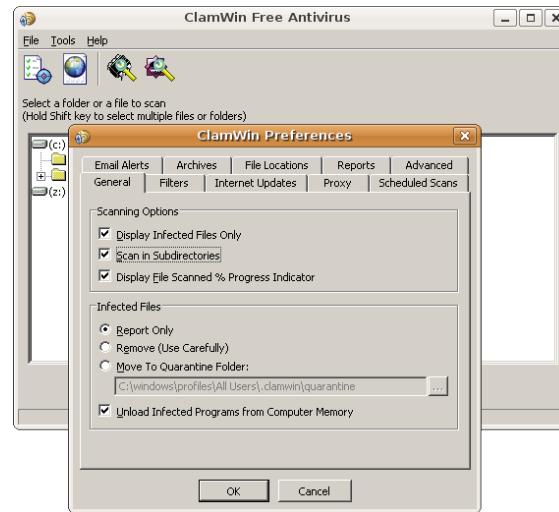
3.6.4 Countermeasures

See also: [Vulnerability to malware](#), [Anti-malware strategies](#), and [Browser hardening](#)

Antivirus software

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^[50] and Android.^[51]

Examples of Microsoft Windows anti virus and anti-malware software include the optional Microsoft Secu-



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

rity Essentials^[52] (for Windows XP, Vista and Windows 7) for real-time protection, the Windows Malicious Software Removal Tool^[53] (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).^[54] Additionally, several capable antivirus software programs are available for free download from the Internet (usually restricted to non-commercial use).^[55] Some such free programs are almost as good as commercial competitors.^[56] Common security vulnerabilities are assigned [CVE IDs](#) and listed in the US National Vulnerability Database. Secunia PSI^[57] 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.^[58] Certain browsers flag sites that have been reported to Google and that have been confirmed as hosting malware by Google.^{[59][60]}

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 iden-

tify 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").^[61]

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).

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.^{[62][63]}

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.^{[64][65]} Additionally, several capable antivirus software programs are available for free download from the Internet (usually restricted to non-commercial use).^[66] 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.^[67] 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.^{[68][69]}

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.

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.

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.

3.6.5 Historical development

See also: Timeline of notable computer viruses and worms

Early academic work on self-replicating programs

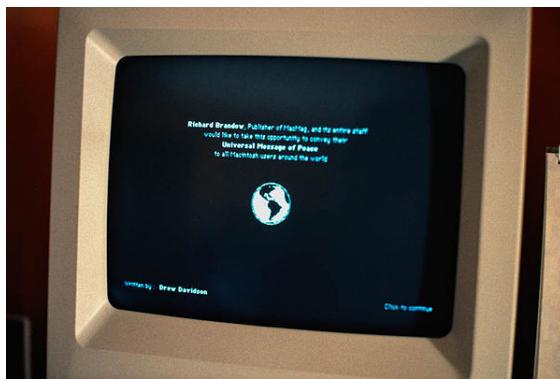
The first academic work on the theory of self-replicating computer programs^[70] was done in 1949 by John von Neumann who gave lectures at the University of Illinois about the "Theory and Organization of Complicated Automata". The work of von Neumann was later published as the "Theory of self-reproducing automata". In his essay von Neumann described how a computer program could be designed to reproduce itself.^[71] 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.^[72]

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).^[73] 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**.^[74] 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.^[75] Creeper was an experimental self-replicating program written by Bob Thomas at **BBN Technologies** in 1971.^[76] Creeper used the ARPANET to infect DEC PDP-10 computers running the **TENEX** operating system.^[77] 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.^[78]

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.^[79] Written in 1981 by **Richard Skrenta**, it attached itself to the **Apple DOS 3.3** operating system and spread via floppy disk.^{[79][80]} 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”.^[81] 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.^[82] **Fred Cohen's** theoretical **compression virus**^[83] 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.^[2]

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.^[84]

The first **IBM PC** virus in the wild was a boot sector virus dubbed **(c)Brain**,^[85] created in 1986 by the Farooq Alvi Brothers in **Lahore, Pakistan**, reportedly to deter piracy of the software they had written.^[86]

The first virus to specifically target Microsoft Windows, **WinVir** was discovered in April 1992, two years after the release of **Windows 3.0**. 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).^[87]

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.^[88]

The first social networking virus, **Win32.5-0-1**, Was created by **Matt Larose** on August 15, 2001.^[89] 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.

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. 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.^[90]

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.

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”.^[91]

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.^[92]

Viruses that spread using cross-site scripting were first reported in 2002,^[93] and were academically demonstrated in 2005.^[94] There have been multiple instances of the cross-site scripting viruses in the wild, exploiting websites such as MySpace and Yahoo!.

3.6.6 See also

- Comparison of computer viruses
- Botnet
- Computer insecurity
- Crimeware
- Cryptovirology

- Keystroke logging
- Multipartite virus
- Spam (electronic)
- Virus hoax
- Windows 7 File Recovery
- Windows Action Center (Security Center)

3.6.7 References

- [1] Aycock, John (2006). *Computer Viruses and Malware*. Springer. p. 14. ISBN 978-0-387-30236-2.
- [2] <http://vx.netlux.org/lib/aas10.html>
- [3] “Alan Solomon ‘All About Viruses’ (VX heavens)”. Web.archive.org. 2011-06-14. Retrieved 2014-07-17.
- [4] 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.
- [5] Mookhey, K.K. et al. (2005). *Linux: Security, Audit and Control Features*. ISACA. p. 128. ISBN 9781893209787.
- [6] Toxen, Bob (2003). *Real World Linux Security: Intrusion Prevention, Detection, and Recovery*. Prentice Hall Professional. p. 365. ISBN 9780130464569.
- [7] Noyes, Katherine (Aug 3, 2010). “Why Linux Is More Secure Than Windows”. PCWorld.
- [8] Skoudis, Edward (2004). “Infection mechanisms and targets”. *Malware: Fighting Malicious Code*. Prentice Hall Professional. pp. 31–48. ISBN 9780131014053.
- [9] Aycock, John (2006). *Computer Viruses and Malware*. Springer. p. 27. ISBN 978-0-387-30236-2.
- [10] Ludwig, Mark A. (1996). *The Little Black Book of Computer Viruses: Volume 1, The Basic Technologies*. pp. 16–17. ISBN 0-929408-02-0.
- [11] Harley, David et al. (2001). *Viruses Revealed*. McGraw-Hill. p. 6. ISBN 0-07-222818-0.
- [12] Filiol, Eric (2005). *Computer viruses:from theory to applications*. Springer. p. 8. ISBN 978-2-287-23939-7.
- [13] Bell, David J. et al, eds. (2004). “Virus”. *Cyberculture: The Key Concepts*. Routledge. p. 154. ISBN 9780203647059.
- [14] “Viruses that can cost you”.
- [15] Granneman, Scott. “Linux vs. Windows Viruses”. The Register. Retrieved September 4, 2015.

- [16] Kaspersky, Eugene (November 21, 2005). “The contemporary antivirus industry and its problems”. SecureLight.
- [17] “Virus Basics”. US-CERT.
- [18] “Virus Notice: Network Associates’ AVERT Discovers First Virus That Can Infect JPEG Files, Assigns Low-Profiled Risk”. Retrieved 2002-06-13.
- [19] “Operating system market share”. *netmarketshare.com*. Retrieved 2015-05-16.
- [20] This is analogous to how genetic diversity in a population decreases the chance of a single disease wiping out a population
- [21] Raggi, Emilio et al. (2011). *Beginning Ubuntu Linux*. Apress. p. 148. ISBN 9781430236276.
- [22] “History of Macintosh Viruses — TopTenREVIEWS”. Mac-antivirus-software-review.toptenreviews.com. Retrieved 2014-07-17.
- [23] There are, however, a variety of Trojans and other malware that exist for OS X, along with a variety of security vulnerabilities that can be exploited by attackers to gain unauthorized access to Mac systems. See for example: Sutter, John D. (22 April 2009). “Experts: Malicious program targets Macs”. CNN.com. Retrieved 24 April 2009.; “Trojan virus tricks Apple Mac users to steal passwords”. *The Daily Telegraph* (London). 2012-02-26. and “Malware Evolution: Mac OS X Vulnerabilities 2005–2006”. Kaspersky Lab. 2006-07-24. Retrieved August 19, 2006.
- [24] “Get a Mac”. Apple. Retrieved 2012-07-15.
- [25] “McAfee discovers first Linux virus” (Press release). McAfee, via Axel Boldt. 5 February 1997.
- [26] Boldt, Axel (19 January 2000). “Bliss, a Linux ‘virus’”.
- [27] 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.
- [28] Avoine, Gildas et al. (2007). *Computer System Security: Basic Concepts and Solved Exercises*. EPFL Press / CRC Press. pp. 21–22. ISBN 9781420046205.
- [29] Brain, Marshall; Fenton, Wesley. “How Computer Viruses Work”. HowStuffWorks.com. Retrieved 16 June 2013.
- [30] Grimes, Roger (2001). *Malicious Mobile Code: Virus Protection for Windows*. O'Reilly. pp. 37–38. ISBN 9781565926820.
- [31] Salomon, David (2006). *Foundations of Computer Security*. Springer. pp. 47–48. ISBN 9781846283413.
- [32] Polk, William T. (1995). *Antivirus Tools and Techniques for Computer Systems*. William Andrew (Elsevier). p. 4. ISBN 9780815513643.
- [33] Grimes, Roger (2001). “Macro Viruses”. *Malicious Mobile Code: Virus Protection for Windows*. O'Reilly. ISBN 9781565926820.
- [34] Aycock, John (2006). *Computer Viruses and Malware*. Springer. p. 89. ISBN 9780387341880.
- [35] “What is boot sector virus?”. Retrieved 2015-10-16.
- [36] Anonymous (2003). *Maximum Security*. Sams Publishing. pp. 331–333. ISBN 9780672324598.
- [37] Skoudis, Edward (2004). “Infection mechanisms and targets”. *Malware: Fighting Malicious Code*. Prentice Hall Professional. pp. 37–38. ISBN 9780131014053.
- [38] editor-in-chief, Béla G. Lipták, (2002). *Instrument engineers' handbook*. (3rd ed.). Boca Raton: CRC Press. p. 874. ISBN 9781439863442. Retrieved September 4, 2015.
- [39] “Computer Virus Strategies and Detection Methods” (PDF). Retrieved 2 September 2008.
- [40] Szor, Peter (2005). *The Art of Computer Virus Research and Defense*. Boston: Addison-Wesley. p. 285. ISBN 0-321-30454-3.
- [41] Fox-Brewster, Thomas. “Netflix Is Dumping Anti-Virus, Presages Death Of An Industry”. Forbes. Retrieved September 4, 2015.
- [42] “How Anti-Virus Software Works”. Stanford University. Retrieved September 4, 2015.
- [43] Bishop, Matt (2003). *Computer Security: Art and Science*. Addison-Wesley Professional. p. 620. ISBN 9780201440997.
- [44] Kizza, Joseph M. (2009). *Guide to Computer Network Security*. Springer. p. 341. ISBN 9781848009165.
- [45] Eilam, Eldad (2011). *Reversing: Secrets of Reverse Engineering*. John Wiley & Sons. p. 216. ISBN 9781118079768.
- [46] “Virus Bulletin : Glossary – Polymorphic virus”. Virusbtn.com. 2009-10-01. Retrieved 2010-08-27.
- [47] 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
- [48] Perriot, Fredrick; Peter Ferrie; Peter Szor (May 2002). “Striking Similarities” (PDF). Retrieved September 9, 2007.
- [49] “Virus Bulletin : Glossary — Metamorphic virus”. Virusbtn.com. Retrieved 2010-08-27.
- [50] “Detailed test reports—(Windows) home user”. AV-Test.org.
- [51] “Detailed test reports — Android mobile devices”. AV-Test.org.

- [52] “Microsoft Security Essentials”. Retrieved June 21, 2012.
- [53] “Malicious Software Removal Tool”. Retrieved June 21, 2012.
- [54] “Windows Defender”. Retrieved June 21, 2012.
- [55] Rubenking, Neil J. (Feb 17, 2012). “The Best Free Antivirus for 2012”. pcmag.com.
- [56] Rubenking, Neil J. (Jan 10, 2013). “The Best Antivirus for 2013”. pcmag.com.
- [57] Rubenking, Neil J. “Secunia Personal Software Inspector 3.0 Review & Rating”. PCMag.com. Retrieved 2013-01-19.
- [58] “10 Step Guide to Protect Against Viruses”. GrnLight.net. Retrieved 23 May 2014.
- [59] “Google Safe Browsing”.
- [60] “Report malicious software (URL) to Google”.
- [61] 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.
- [62] “W32.Gammima.AG”. Symantec. Retrieved 2014-07-17.
- [63] Category: Computer Articles. “Viruses! In! Space!” . GrnLight.net. Retrieved 2014-07-17.
- [64] “VirusTotal.com (a subsidiary of Google)”.
- [65] “VirScan.org”.
- [66] Rubenking, Neil J. “The Best Free Antivirus for 2014”. pcmag.com.
- [67] “Microsoft Safety Scanner”.
- [68] “Virus removal -Help”. Retrieved 2015-01-31.
- [69] “W32.Gammima.AG Removal — Removing Help”. Symantec. 2007-08-27. Retrieved 2014-07-17.
- [70] The term “computer virus” was not used at that time.
- [71] von Neumann, John (1966). “Theory of Self-Reproducing Automata” (PDF). *Essays on Cellular Automata* (University of Illinois Press): 66–87. Retrieved June 10, 2010.
- [72] Éric Filiol, *Computer viruses: from theory to applications, Volume 1*, Birkhäuser, 2005, pp. 19–38 ISBN 2-287-23939-1.
- [73] Risak, Veith (1972), “Selbstreproduzierende Automaten mit minimaler Informationsübertragung”, *Zeitschrift für Maschinenbau und Elektrotechnik*
- [74] Kraus, Jürgen (February 1980), *Selbstreproduktion bei Programmen* (PDF)
- [75] “Virus list”. Retrieved 2008-02-07.
- [76] Thomas Chen, Jean-Marc Robert (2004). “The Evolution of Viruses and Worms”. Retrieved 2009-02-16.
- [77] Parikka, Jussi (2007). *Digital Contagions: A Media Archaeology of Computer Viruses*. New York: Peter Lang. p. 50. ISBN 978-0-8204-8837-0.
- [78] Russell, Deborah & Gangemi, G.T. (1991). *Computer Security Basics*. O'Reilly. p. 86. ISBN 0-937175-71-4.
- [79] Anick Jesdanun (1 September 2007). “School prank starts 25 years of security woes”. CNBC. Retrieved April 12, 2013.
- [80] “The anniversary of a nuisance”.
- [81] Cohen, Fred (1984), *Computer Viruses – Theory and Experiments*
- [82] Cohen, Fred, *An Undetectable Computer Virus*, 1987, IBM
- [83] Burger, Ralph, 1991. *Computer Viruses and Data Protection*, pp. 19–20
- [84] 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. ISSN 0163-6006.
- [85] “Boot sector virus repair”. Antivirus.about.com. 2010-06-10. Retrieved 2010-08-27.
- [86] “Amjad Farooq Alvi Inventor of first PC Virus post by Zaghram”. YouTube. Retrieved 2010-08-27.
- [87] Grimes, Roger (2001). *Malicious Mobile Code: Virus Protection for Windows*. O'Reilly. pp. 99–100. ISBN 9781565926820.
- [88] “SCA virus”. Virus Test Center, University of Hamburg. 1990-06-05. Retrieved 2014-01-14.
- [89] <http://5-0-1.webs.com>
- [90] “What is a Computer Virus?”. Actlab.utexas.edu. 1996-03-31. Retrieved 2010-08-27.
- [91] Vesselin Bontchev. “Macro Virus Identification Problems”. *FRISK Software International*.
- [92] “Facebook 'photo virus' spreads via email.”. Retrieved 2014-04-28.
- [93] Berend-Jan Wever. “XSS bug in hotmail login page”. Retrieved 2014-04-07.
- [94] Wade Alcorn. “The Cross-site Scripting Virus”. bind-shell.net. Retrieved 2015-10-13.

3.6.8 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.
- 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.

3.6.9 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

3.7 Virus hoax

A **computer virus hoax** is a message warning the recipients of a non-existent computer virus threat. The message is usually a **chain e-mail** that tells the recipients to forward it to everyone they know.

3.7.1 Identification

Most hoaxes are sensational in nature and easily identified by the fact that they indicate that the virus will do nearly impossible things, like blow up the recipient's computer and set it on fire, or less sensational, delete everything on the user's computer. They often include fake announcements claimed to originate from reputable computer organizations together with mainstream news media. These bogus sources are quoted in order to give the hoax more credibility. Typically, the warnings use emotive language, stress the urgent nature of the threat and encourage readers to forward the message to other people as soon as possible.

Virus hoaxes are usually harmless and accomplish nothing more than annoying people who identify it as a hoax and waste the time of people who forward the message. Nevertheless, a number of hoaxes have warned users that vital system files are viruses and encourage the user to delete the file, possibly damaging the system. Examples

of this type include the **jdbgmgr.exe** virus hoax and the **SULFNBK.EXE** hoax.^{[1][2]}

Some consider virus hoaxes and other chain e-mails to be a computer worm in and of themselves. They replicate by social engineering—exploiting users' concern, ignorance, and disinclination to investigate before acting.

Hoaxes are distinct from computer pranks, which are harmless programs that perform unwanted and annoying actions on a computer, such as randomly moving the mouse, turning the screen display upside down, etc.

3.7.2 Action

Anti-virus specialists agree that recipients should delete virus hoaxes when they receive them, instead of forwarding them.^{[3][4]}

McAfee says:

We are advising users who receive the email to delete it and DO NOT pass it on as this is how an email HOAX propagates.^[3]

F-Secure recommends:

Do not forward hoax messages.

Hoax warnings are typically scare alerts started by malicious people – and passed on by innocent individuals that think they are helping the community by spreading the warning.

Corporate users can get rid of the hoax problem by simply setting a strict company guideline: End users must not forward virus alarms. Ever. It's not the job of an end user anyway. If such message is received, end users could forward it to the IT department but not to anyone else.^[4]

3.7.3 Comparison of computer virus hoaxes

This list is incomplete; you can help by expanding it.

3.7.4 Telephone scam

Main article: **Technical support scam**

A telephone scam, operated from call centres based in Kolkata, India, has been active since 2008. The victim is quoted his or her name and address, and is told: "I'm calling for Microsoft (or an entity that sounds like it is connected to Microsoft, such as the "Windows Service Center" or "Windows Technical Department"). We've had a

report from your internet service provider of serious virus problems from your Windows computer.” The victim is then directed to open the Windows event viewer, which displays apparently critical warnings, and is directed to a website to download an application to allow the scammer to control his or her computer remotely. The caller supposedly fixes the problems and demands a fee for the service. However, the process usually enables malware to be downloaded to the victim’s computer.^[12]

3.7.5 Parodies

The virus hoax has become part of the culture of the twenty-first century and the gullibility of novice computer users convinced to delete files on the basis of hoaxes has been parodied in several popular jokes and songs.

One such parody is “Weird Al” Yankovic’s song “Virus Alert” from the album *Straight Outta Lynwood*. The song makes fun of the exaggerated claims that are made in virus hoaxes, such as legally changing your name or opening a rift in time and space.^[13]

Another parody of virus hoaxes is the *honor system virus* which has been circulated under the name Amish Computer Virus, manual virus, the Blond Computer Virus, the Irish Computer Virus, the Syrian Computer Virus, the Norwegian Computer Virus, Newbie Virus, the Unix Computer Virus, the Mac OS 9 virus, Discount virus and many others. This joke email claims to be authored by the Amish or other similar low-technology populations who have no computers, programming skills or electricity to create viruses and thus ask you to delete your own hard drive contents manually after forwarding the message to your friends.^[14]

The Tuxissa virus is another parody of the virus hoax, based on the concept of the *Melissa virus*, but with its aim of installing Linux on the victim’s computer without the owner’s permission. The story says that it was spread via e-mail, contained in a message titled “Important Message About Windows Security”. It was supposed to first spread the virus to other computers, then download a stripped-down version of *Slackware* and uncompress it onto the hard disk. The Windows Registry is finally deleted and the boot options changed. Then the virus removes itself when it reboots the computer at the end, with the user facing the Linux login prompt and all his Windows security problems solved for him.^[15]

3.7.6 See also

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

3.7.7 References

- [1] Mikkelsen, Barbara and David P. (January 2008). “JDBGMGR.EXE Virus”. Retrieved 2011-08-08.
- [2] Mikkelsen, Barbara and David P. (January 2008). “SULFNBK.EXE Virus”. Retrieved 2011-08-08.
- [3] McAfee, Inc (December 2003). “Virus Profile: A Virtual Card For You Hoax”. Retrieved 2011-08-08.
- [4] F-Secure Corporation (2009). “Hoax Warnings”. Retrieved 2012-06-14.
- [5] Gutierrez, Ralph (July 2001). “Antichrist Hoax”. Retrieved 2011-08-08.
- [6] “Black Muslim in the White House”. *snopes.com*. 20 June 2013. Retrieved 17 January 2014.
- [7] Mikkelsen, Barbara and David P. (January 2008). “Budweiser Frogs Virus”. Retrieved 2011-08-08.
- [8] Jones, Les (December 1998). “Good Times Virus Hoax Frequently Asked Questions”. Retrieved 2011-08-08.
- [9] Christensen, Brett M. (2008). “Olympic Torch Invitation Virus Hoax”. Retrieved 2011-08-08.
- [10] Koris, George (2002-01-15). “Life is beautiful Hoax”. *Symantec.com*. Symantec. Retrieved 2011-08-08.
- [11] Symantec Corporation (February 2007). “FROGAPULT, ELFOWL, Y2KGAME Virus Hoax”. Retrieved 2011-08-08.
- [12] Charles Arthur (18 July 2010). “Virus phone scam being run from call centres in India”. Guardian News and Media Limited. Retrieved 1 May 2012.
- [13] “Weird Al unleashes his new album with a Bill Plympton Video DON’T DOWNLOAD THIS SONG!!!”. *Ain’t It Cool News*. 2006-09-11. Retrieved 2011-08-08.
- [14] Pearson, Karl (May 2000). “Humor: For a good time....”. Retrieved 2011-08-08.
- [15] Baughn, James (March 1999). “Attack of the Tuxissa Virus”. Retrieved 2009-04-17.

3.7.8 External links

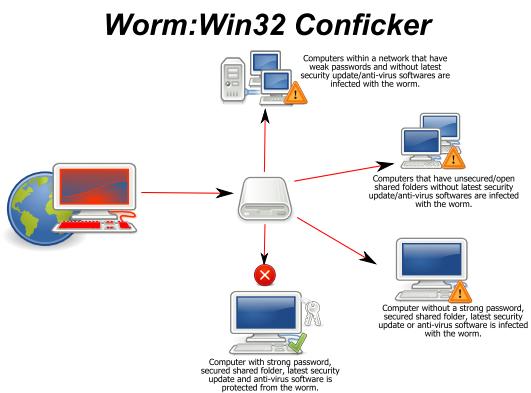
- McAfee virus hoaxes
- Symantec Threat Explorer
- Trend Micro Virus Encyclopedia
- Sector Zero/Virtual Card for You Email

```

0 00 00-6D 73 62 6C msbl
0 6A 75-73 74 20 77 ast.exe I just w
9 20 4C-4F 56 45 20 ant to say LOVE
0 62 69-6C 6C 79 20 YOU SAN!! billy
0 64 6F-20 79 6F 75 gates why do you
3 20 70-6F 73 73 69 make this possi
0 20 6D-61 6B 69 6E ble ? Stop makin
E 64 20-66 69 78 20 g money and fix
7 61 72-65 21 21 00 your software!!
0 00 00-7F 00 00 00 ♫ ♪ H ♫
0 00 00-01 00 01 00 ♫ ♪ H ♫ ♫
0 00 00-00 00 00 46 á@ ♫ ♪ H ♫
C C9 11-9F E8 08 00 ♫ ♪ Jéèù-ñ4 fñ
0 00 03-10 00 00 00 ♫ ♪ H ♫ ♫
3 00 00-01 00 04 00 ♫ ♪ H ♫ ♫

```

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



Spread of Conficker worm

3.8 Computer worm

This article is about malware. For the data storage device, see Write Once Read Many. For other uses, see [worm \(disambiguation\)](#).

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

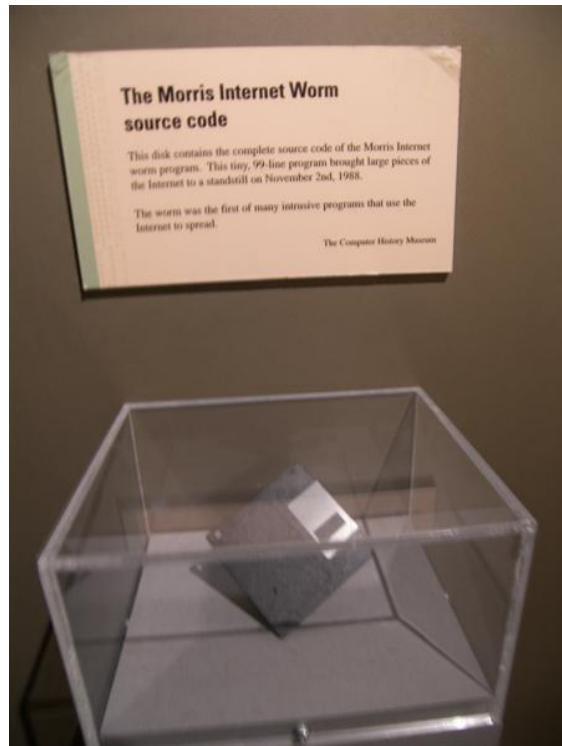
Many worms that have been created are designed only to spread, and do not attempt to change the systems they pass through. However, as the **Morris worm** and **Mydoom** showed, even these “payload free” worms can cause major disruption by increasing network traffic and other unintended effects. A “payload” 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 with threatened 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]

3.8.1 History



Morris worm source code disk at the Computer History Museum

The actual term “worm” was first used in John Brunner’s 1975 novel, *The Shockwave Rider*. In that novel, Nicholas 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]

3.8.2 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, 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.^{[16][17]} In addition, machine learning techniques can be used to detect new worms, by analyzing the behavior of the suspected computer.^[18]

Mitigation techniques

- **ACLs** in routers and switches
- **Packet-filters**
- **TCP Wrapper/libwrap** enabled network service daemons
- **Nullrouting**

3.8.3 Worms with good intent

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.^[19] 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.^[20]

3.8.4 See also

- Computer and network surveillance
- Computer virus
- Helpful worm
- Spam
- Timeline of notable computer viruses and worms
- Trojan horse (computing)
- XSS Worm

3.8.5 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] 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.
- [17] “A New Way to Protect Computer Networks from Internet Worms”. *Newswise*. Retrieved July 5, 2011.
- [18] Moskovich 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
- [19] “Virus alert about the Nachi worm”. Microsoft.
- [20] 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.

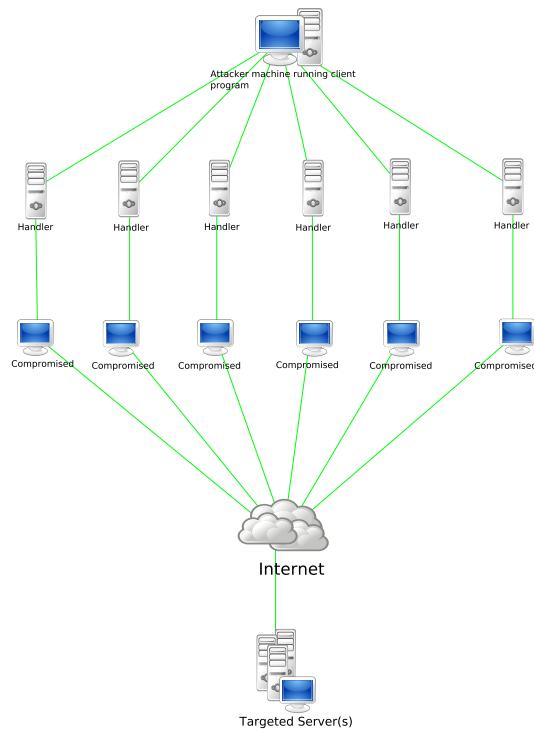
3.8.6 External links

- Malware Guide – Guide for understanding, removing and preventing worm infections on Vername.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.

3.9 Denial-of-service attack

“DoS” redirects here. For other uses, see **DoS** (disambiguation).

In computing, a **denial-of-service (DoS) attack** is an



DDoS Stacheldraht attack diagram.

attempt to make a machine or network resource unavailable to its intended users, such as to temporarily or indefinitely interrupt or suspend services of a host connected to the Internet. A **distributed denial-of-service (DDoS)** is where the attack source is more than one—and often thousands of—unique IP addresses.

Criminal perpetrators of DoS attacks often target sites or services hosted on high-profile web servers such as banks, credit card payment gateways; but motives of revenge, blackmail^{[1][2]} or activism^[3] can be behind other attacks.

3.9.1 Symptoms

The United States Computer Emergency Readiness Team (US-CERT) defines symptoms of denial-of-service attacks to include:

- Unusually slow network performance (opening files or accessing web sites)
- Unavailability of a particular web site
- Inability to access any web site

- Dramatic increase in the number of spam emails received—(this type of DoS attack is considered an e-mail bomb)^[4]
- Disconnection of a wireless or wired internet connection
- Long term denial of access to the web or any internet services

Denial-of-service attacks can also lead to problems in the network 'branches' around the actual computer being attacked. For example, the **bandwidth** of a router between the Internet and a **LAN** may be consumed by an attack, compromising not only the intended computer, but also the entire network or other computers on the LAN.^[5]

If the attack is conducted on a sufficiently large scale, entire geographical regions of Internet connectivity can be compromised without the attacker's knowledge or intent by incorrectly configured or flimsy network infrastructure equipment.

3.9.2 Attack techniques

A denial-of-service attack is characterized by an explicit attempt by attackers to prevent legitimate users of a service from using that service. There are two general forms of DoS attacks: those that crash services and those that flood services.

The most serious attacks are distributed^[6] and in many or most cases involve forging of IP sender addresses (**IP address spoofing**) so that the location of the attacking machines cannot easily be identified, nor can filtering be done based on the source address.

Internet Control Message Protocol (ICMP) flood

See also: Smurf attack, Ping flood and Ping of death

A smurf attack relies on misconfigured network devices that allow packets to be sent to all computer hosts on a particular network via the **broadcast address** of the network, rather than a specific machine. The attacker will send large numbers of **IP** packets with the source address faked to appear to be the address of the victim. The network's bandwidth is quickly used up, preventing legitimate packets from getting through to their destination.^[7]

Ping flood is based on sending the victim an overwhelming number of **ping** packets, usually using the "ping" command from **Unix-like** hosts (the -t flag on **Windows** systems is much less capable of overwhelming a target, also the -l (size) flag does not allow sent packet size greater than 65500 in Windows). It is very simple to launch, the primary requirement being access to greater bandwidth than the victim.

Ping of death is based on sending the victim a malformed ping packet, which will lead to a system crash on a vulnerable system.

(S)SYN flood

See also: SYN flood

A **SYN** flood occurs when a host sends a flood of TCP/SYN packets, often with a forged sender address. Each of these packets are handled like a connection request, causing the server to spawn a **half-open connection**, by sending back a TCP/SYN-ACK packet (Acknowledge), and waiting for a packet in response from the sender address (response to the ACK Packet). However, because the sender address is forged, the response never comes. These half-open connections saturate the number of available connections the server can make, keeping it from responding to legitimate requests until after the attack ends.^[8]

Teardrop attacks

A teardrop attack involves sending mangled IP fragments with overlapping, over-sized payloads to the target machine. This can crash various operating systems because of a bug in their **TCP/IP fragmentation re-assembly code**.^[9] Windows 3.1x, Windows 95 and Windows NT operating systems, as well as versions of **Linux** prior to versions 2.0.32 and 2.1.63 are vulnerable to this attack.

(Although in September 2009, a vulnerability in **Windows Vista** was referred to as a "teardrop attack", this targeted **SMB2** which is a higher layer than the TCP packets that teardrop used).^{[10][11]}

Peer-to-peer attacks

Main article: Direct Connect (protocol) § Direct Connect used for DDoS attacks

Attackers have found a way to exploit a number of bugs in **peer-to-peer** servers to initiate DDoS attacks. The most aggressive of these peer-to-peer-DDoS attacks exploits **DC++**. With peer-to-peer there is no botnet and the attacker does not have to communicate with the clients it subverts. Instead, the attacker acts as a "puppet master," instructing clients of large peer-to-peer file sharing hubs to disconnect from their peer-to-peer network and to connect to the victim's website instead.^{[12][13][14]}

Permanent denial-of-service attacks

Permanent denial-of-service (PDoS), also known loosely as **phlashing**,^[15] is an attack that damages a system so

badly that it requires replacement or reinstallation of hardware.^[16] Unlike the distributed denial-of-service attack, a PDoS attack exploits security flaws which allow remote administration on the management interfaces of the victim's hardware, such as routers, printers, or other networking hardware. The attacker uses these vulnerabilities to replace a device's **firmware** with a modified, corrupt, or defective firmware image—a process which when done legitimately is known as *flashing*. This therefore "bricks" the device, rendering it unusable for its original purpose until it can be repaired or replaced.

The PDoS is a pure hardware targeted attack which can be much faster and requires fewer resources than using a botnet or a root/vserver in a DDoS attack. Because of these features, and the potential and high probability of security exploits on Network Enabled Embedded Devices (NEEDs), this technique has come to the attention of numerous hacker communities.

PhlashDance is a tool created by Rich Smith (an employee of Hewlett-Packard's Systems Security Lab) used to detect and demonstrate PDoS vulnerabilities at the 2008 EUSecWest Applied Security Conference in London.^[17]

Application-layer floods

Various DoS-causing **exploits** such as buffer overflow can cause server-running software to get confused and fill the disk space or consume all available memory or CPU time.

Other kinds of DoS rely primarily on brute force, flooding the target with an overwhelming flux of packets, over-saturating its connection bandwidth or depleting the target's system resources. Bandwidth-saturating floods rely on the attacker having higher bandwidth available than the victim; a common way of achieving this today is via distributed denial-of-service, employing a botnet. Another target of DDoS attacks may be to produce added costs for the application operator, when the latter uses resources based on **Cloud Computing**. In this case normally application used resources are tied to a needed Quality of Service level (e.g. responses should be less than 200 ms) and this rule is usually linked to automated software (e.g. Amazon CloudWatch^[18]) to raise more virtual resources from the provider in order to meet the defined QoS levels for the increased requests. The main incentive behind such attacks may be to drive the application owner to raise the elasticity levels in order to handle the increased application traffic, in order to cause financial losses or force them to become less competitive. Other floods may use specific packet types or connection requests to saturate finite resources by, for example, occupying the maximum number of open connections or filling the victim's disk space with logs.

A "banana attack" is another particular type of DoS. It involves redirecting outgoing messages from the client back onto the client, preventing outside access, as well as flood-

ing the client with the sent packets.

An attacker with shell-level access to a victim's computer may slow it until it is unusable or crash it by using a fork bomb.

A kind of application-level DoS attack is **XDoS** (or XML DoS) which can be controlled by modern web application firewalls (WAFs).

Nuke

A **Nuke** is an old denial-of-service attack against computer networks consisting of fragmented or otherwise invalid ICMP packets sent to the target, achieved by using a modified ping utility to repeatedly send this corrupt data, thus slowing down the affected computer until it comes to a complete stop.

A specific example of a nuke attack that gained some prominence is the **WinNuke**, which exploited the vulnerability in the **NetBIOS** handler in **Windows 95**. A string of out-of-band data was sent to **TCP** port 139 of the victim's machine, causing it to lock up and display a **Blue Screen of Death** (BSOD).

HTTP POST DoS attack

First discovered in 2009, the HTTP POST attack sends a complete, legitimate **HTTP POST** header, which includes a 'Content-Length' field to specify the size of the message body to follow. However, the attacker then proceeds to send the actual message body at an extremely slow rate (e.g. 1 byte/110 seconds). Due to the entire message being correct and complete, the target server will attempt to obey the 'Content-Length' field in the header, and wait for the entire body of the message to be transmitted, hence slowing it down.^[19] Further combined with the fact that **Apache** will, by default, accept requests up to 2GB in size, this attack can be particularly powerful. HTTP POST attacks are difficult to differentiate from legitimate connections, and are therefore able to bypass some protection systems. **OWASP**, an open source web application security project, has released a testing tool to test the security of servers against this type of attack.

R-U-Dead-Yet? (RUDY)

This attack targets web applications by starvation of available sessions on the web server. Much like **Slowloris**, RUDY keeps sessions at halt using never-ending POST transmissions and sending an arbitrarily large content-length header value.

Slow Read attack

Slow Read attack sends legitimate application layer requests but reads responses very slowly, thus trying to

exhaust the server's connection pool. Slow reading is achieved by advertising a very small number for the TCP Receive Window size and at the same time by emptying clients' TCP receive buffer slowly. That naturally ensures a very low data flow rate.

Distributed attack

A distributed denial-of-service (DDoS) attack occurs when multiple systems flood the bandwidth or resources of a targeted system, usually one or more web servers.^[6] Such an attack is often the result of multiple compromised systems (for example a botnet) flooding the targeted system with traffic. A botnet is a network of zombie computers programmed to receive commands without the owners' knowledge.^[20] When a server is overloaded with connections, new connections can no longer be accepted. The major advantages to an attacker of using a distributed denial-of-service attack are that multiple machines can generate more attack traffic than one machine, multiple attack machines are harder to turn off than one attack machine, and that the behavior of each attack machine can be stealthier, making it harder to track and shut down. These attacker advantages cause challenges for defense mechanisms. For example, merely purchasing more incoming bandwidth than the current volume of the attack might not help, because the attacker might be able to simply add more attack machines. This after all will end up completely crashing a website for periods of time.

Malware can carry DDoS attack mechanisms; one of the better-known examples of this was MyDoom. Its DoS mechanism was triggered on a specific date and time. This type of DDoS involved hardcoding the target IP address prior to release of the malware and no further interaction was necessary to launch the attack.

A system may also be compromised with a trojan, allowing the attacker to download a zombie agent, or the trojan may contain one. Attackers can also break into systems using automated tools that exploit flaws in programs that listen for connections from remote hosts. This scenario primarily concerns systems acting as servers on the web. Stacheldraht is a classic example of a DDoS tool. It utilizes a layered structure where the attacker uses a client program to connect to handlers, which are compromised systems that issue commands to the zombie agents, which in turn facilitate the DDoS attack. Agents are compromised via the handlers by the attacker, using automated routines to exploit vulnerabilities in programs that accept remote connections running on the targeted remote hosts. Each handler can control up to a thousand agents.^[21] In some cases a machine may become part of a DDoS attack with the owner's consent, for example, in Operation Payback, organized by the group Anonymous. These attacks can use different types of internet packets such as: TCP, UDP, ICMP etc.

These collections of systems compromisers are known as

botnets / rootservers. DDoS tools like Stacheldraht still use classic DoS attack methods centered on IP spoofing and amplification like smurf attacks and fraggle attacks (these are also known as bandwidth consumption attacks). SYN floods (also known as resource starvation attacks) may also be used. Newer tools can use DNS servers for DoS purposes. Unlike MyDoom's DDoS mechanism, botnets can be turned against any IP address. Script kiddies use them to deny the availability of well known websites to legitimate users.^[22] More sophisticated attackers use DDoS tools for the purposes of extortion – even against their business rivals.^[23]

Simple attacks such as SYN floods may appear with a wide range of source IP addresses, giving the appearance of a well distributed DoS. These flood attacks do not require completion of the TCP three way handshake and attempt to exhaust the destination SYN queue or the server bandwidth. Because the source IP addresses can be trivially spoofed, an attack could come from a limited set of sources, or may even originate from a single host. Stack enhancements such as syn cookies may be effective mitigation against SYN queue flooding, however complete bandwidth exhaustion may require involvement.

If an attacker mounts an attack from a single host it would be classified as a DoS attack. In fact, any attack against availability would be classed as a denial-of-service attack. On the other hand, if an attacker uses many systems to simultaneously launch attacks against a remote host, this would be classified as a DDoS attack.

UK's GCHQ has tools built for DDOS, named PREATORS FACE and ROLLING THUNDER.^[24]

See also: DDoS mitigation

Reflected / spoofed attack

A distributed denial-of-service attack may involve sending forged requests of some type to a very large number of computers that will reply to the requests. Using Internet Protocol address spoofing, the source address is set to that of the targeted victim, which means all the replies will go to (and flood) the target. (This reflected attack form is sometimes called a "DRDOS".^[25])

ICMP Echo Request attacks (Smurf Attack) can be considered one form of reflected attack, as the flooding host(s) send Echo Requests to the broadcast addresses of mis-configured networks, thereby enticing hosts to send Echo Reply packets to the victim. Some early DDoS programs implemented a distributed form of this attack.

Many services can be exploited to act as reflectors, some harder to block than others.^[26] US-CERT have observed that different services implies in different amplification factors, as you can see below:^[27]

DNS amplification attacks involve a new mechanism that increased the amplification effect, using a much larger

list of DNS servers than seen earlier.^{[30][31]} SNMP and NTP can also be exploited as reflector in an amplification attack.

Telephony denial-of-service (TDoS)

Voice over IP has made abusive origination of large numbers of telephone voice calls inexpensive and readily automated while permitting call origins to be misrepresented through caller ID spoofing.

According to the US Federal Bureau of Investigation, telephony denial-of-service (TDoS) has appeared as part of various fraudulent schemes:

- A scammer contacts the victim's banker or broker, impersonating the victim to request a funds transfer. The banker's attempt to contact the victim for verification of the transfer fails as the victim's telephone lines are being flooded with thousands of bogus calls, rendering the victim unreachable.^[32]
- A scammer contacts consumers with a bogus claim to collect an outstanding **payday loan** for thousands of dollars. When the consumer objects, the scammer retaliates by flooding the victim's employer with thousands of automated calls. In some cases, displayed caller ID is spoofed to impersonate police or law enforcement agencies.^[33]
- A scammer contacts consumers with a bogus debt collection demand and threatens to send police; when the victim balks, the scammer floods local police numbers with calls on which caller ID is spoofed to display the victim's number. Police soon arrive at the victim's residence attempting to find the origin of the calls.

Telephony denial-of-service can exist even without Internet telephony. In the 2002 New Hampshire Senate election phone jamming scandal, telemarketers were used to flood political opponents with spurious calls to jam phone banks on election day. Widespread publication of a number can also flood it with enough calls to render it unusable, as happened with multiple +1-area code—867-5309 subscribers inundated by hundreds of misdialed calls daily in response to a popular song 867-5309/Jenny.

TDoS differs from other telephone harassment (such as **prank calls** and **obscene phone calls**) by the number of calls originated; by occupying lines continuously with repeated automated calls, the victim is prevented from making or receiving both routine and emergency telephone calls.

Related exploits include **SMS flooding** attacks and **black fax** or **fax loop** transmission.

Denial-of-service Level II

The goal of DoS L2 (possibly DDoS) attack is to cause a launching of a defense mechanism which blocks the network segment from which the attack originated. In case of distributed attack or IP header modification (that depends on the kind of security behavior) it will fully block the attacked network from the Internet, but without system crash.

Advanced Persistent DoS (APDoS)

An APDoS is more likely to be perpetrated by an advanced persistent threat (APT): actors who are well resourced, exceptionally skilled and have access to substantial commercial grade computer resources and capacity. APDoS attacks represent a clear and emerging threat needing specialised monitoring and incident response services and the defensive capabilities of specialised DDoS mitigation service providers. This type of attack involves massive network layer DDoS attacks through to focused application layer (HTTP) floods, followed by repeated (at varying intervals) SQLI and XSS attacks. Typically, the perpetrators can simultaneously use from 2 to 5 attack vectors involving up to several tens of millions of requests per second, often accompanied by large SYN floods that can not only attack the victim but also any service provider implementing any sort of managed DDoS mitigation capability. These attacks can persist for several weeks- the longest continuous period noted so far lasted 38 days. This APDoS attack involved approximately 50+ petabits (51,000+ terabits) of malicious traffic. Attackers in this scenario may (or often will) tactically switch between several targets to create a diversion to evade defensive DDoS countermeasures but all the while eventually concentrating the main thrust of the attack onto a single victim. In this scenario, threat actors with continuous access to several very powerful network resources are capable of sustaining a prolonged campaign generating enormous levels of un-amplified DDoS traffic.

APDoS attacks are characterised by:

- Advanced reconnaissance (pre-attack OSINT and extensive decoyed scanning crafted to evade detection over long periods)
- Tactical execution (attack with a primary and secondary victims but focus is on Primary)
- Explicit motivation (a calculated end game/goal target)
- Large computing capacity (access to substantial computer power and network bandwidth resources)
- Simultaneous multi-threaded ISO layer attacks (sophisticated tools operating at layers 3 through 7)

- Persistence over extended periods (utilising all the above into a concerted, well managed attack across a range of targets^[34])

DDoS extortion

In 2015, DDoS botnets such as DD4BC grew in prominence, taking aim at financial institutions.^[35] Cyber-extortionists typically begin with a low-level attack and a warning that a larger attack will be carried out if a ransom is not paid in Bitcoin.^[36] Security experts recommend targeted websites to not pay the ransom.^[37]

3.9.3 Attack tools

A wide array of programs are used to launch DoS-attacks. In cases such as MyDoom the tools are embedded in malware, and launch their attacks without the knowledge of the system owner. **Stacheldraht** is a classic example of a DDoS tool. It utilizes a layered structure where the attacker uses a client program to connect to handlers, which are compromised systems that issue commands to the zombie agents, which in turn facilitate the DDoS attack. Agents are compromised via the handlers by the attacker, using automated routines to exploit vulnerabilities in programs that accept remote connections running on the targeted remote hosts. Each handler can control up to a thousand agents.^[21]

In other cases a machine may become part of a DDoS attack with the owner's consent, for example, in Operation Payback, organized by the group Anonymous. The LOIC has typically been used in this way.

UK's GCHQ has tools built for DDOS, named PREATORS FACE and ROLLING THUNDER.^[24]

3.9.4 Defense techniques

Defensive responses to denial-of-service attacks typically involve the use of a combination of attack detection, traffic classification and response tools, aiming to block traffic that they identify as illegitimate and allow traffic that they identify as legitimate.^[38]

A list of prevention and response tools is provided below:

Firewalls

In the case of a simple attack, a firewall could have a simple rule added to deny all incoming traffic from the attackers, based on protocols, ports or the originating IP addresses.

More complex attacks will however be hard to block with simple rules: for example, if there is an ongoing attack

on port 80 (web service), it is not possible to drop all incoming traffic on this port because doing so will prevent the server from serving legitimate traffic.^[39] Additionally, firewalls may be too deep in the network hierarchy, with routers being adversely affected before the traffic gets to the firewall.

Switches

Most switches have some rate-limiting and **ACL** capability. Some switches provide automatic and/or system-wide rate limiting, traffic shaping, delayed binding (TCP splicing), deep packet inspection and Bogon filtering (bogus IP filtering) to detect and remediate denial-of-service attacks through automatic rate filtering and WAN Link failover and balancing.

These schemes will work as long as the DoS attacks can be prevented by using them. For example, SYN flood can be prevented using delayed binding or TCP splicing. Similarly content based DoS may be prevented using deep packet inspection. Attacks originating from dark addresses or going to dark addresses can be prevented using bogon filtering. Automatic rate filtering can work as long as set rate-thresholds have been set correctly and granularly. Wan-link failover will work as long as both links have DoS/DDoS prevention mechanism.

Routers

Similar to switches, routers have some rate-limiting and **ACL** capability. They, too, are manually set. Most routers can be easily overwhelmed under a DoS attack. **Cisco IOS** has optional features that can reduce the impact of flooding.^[40]

Application front end hardware

Application front end hardware is intelligent hardware placed on the network before traffic reaches the servers. It can be used on networks in conjunction with routers and switches. Application front end hardware analyzes data packets as they enter the system, and then identifies them as priority, regular, or dangerous. There are more than 25 bandwidth management vendors.

Application level Key Completion Indicators

In order to meet the case of application level DDoS attacks against Cloud based applications, approaches may be based on an application layer analysis, to indicate whether an incoming traffic bulk is legitimate or not and thus enable the triggering of elasticity decisions without the economical implications of a DDoS attack. These approaches mainly rely on an identified path of value inside the application and monitor the macroscopic progress of

the requests in this path, towards the final generation of profit, through markers denoted as Key Completion Indicators^[41]

IPS based prevention

Intrusion-prevention systems (IPS) are effective if the attacks have signatures associated with them. However, the trend among the attacks is to have legitimate content but bad intent. Intrusion-prevention systems which work on content recognition cannot block behavior-based DoS attacks.

An **ASIC** based IPS may detect and block denial-of-service attacks because they have the processing power and the granularity to analyze the attacks and act like a circuit breaker in an automated way.

A rate-based IPS (RBIPS) must analyze traffic granularly and continuously monitor the traffic pattern and determine if there is traffic anomaly. It must let the legitimate traffic flow while blocking the DoS attack traffic.^[42]

DDS based defense

More focused on the problem than IPS, a DoS Defense System (DDS) can block connection-based DoS attacks and those with legitimate content but bad intent. A DDS can also address both protocol attacks (such as Teardrop and Ping of death) and rate-based attacks (such as ICMP floods and SYN floods).

Blackholing and sinkholing

With blackholing, all the traffic to the attacked DNS or IP address is sent to a “black hole” (null interface or a non-existent server). To be more efficient and avoid affecting network connectivity, it can be managed by the ISP.^[43]

Sinkholing routes traffic to a valid IP address which analyzes traffic and rejects bad packets. Sinkholing is not efficient for most severe attacks.

Clean pipes

All traffic is passed through a “cleaning center” or a “scrubbing center” via various methods such as proxies, tunnels or even direct circuits, which separates “bad” traffic (DDoS and also other common internet attacks) and only sends good traffic beyond to the server. The provider needs central connectivity to the Internet to manage this kind of service unless they happen to be located within the same facility as the “cleaning center” or “scrubbing center”.^[44]

Examples of providers of this service:

- Radware^[45]

- A10 Networks
- Arbor Networks^[46]
- AT&T^[47]
- F5 Networks^[48]
- Incapsula^[49]
- Neustar Inc^[50]
- Prolexic Technologies^[51]
- Sprint^[52]
- Staminus Communications^{[53][54]}
- Tata Communications^[55]
- Verisign^[56]
- Verizon^{[57][58]}

3.9.5 Unintentional denial-of-service

An Unintentional denial-of-service can occur when a system ends up denied, not due to a deliberate attack by a single individual or group of individuals, but simply due to a sudden enormous spike in popularity. This can happen when an extremely popular website posts a prominent link to a second, less well-prepared site, for example, as part of a news story. The result is that a significant proportion of the primary site’s regular users – potentially hundreds of thousands of people – click that link in the space of a few hours, having the same effect on the target website as a DDoS attack. A VIPDoS is the same, but specifically when the link was posted by a celebrity.

When **Michael Jackson** died in 2009, websites such as Google and Twitter slowed down or even crashed.^[59] Many sites’ servers thought the requests were from a virus or spyware trying to cause a denial-of-service attack, warning users that their queries looked like “automated requests from a computer virus or spyware application”.^[60]

News sites and link sites – sites whose primary function is to provide links to interesting content elsewhere on the Internet – are most likely to cause this phenomenon. The canonical example is the **Slashdot effect** when receiving traffic from **Slashdot**. Sites such as **Reddit**, **Digg**, the **Drudge Report**, **Fark**, **Something Awful**, and the web-comic **Penny Arcade** have their own corresponding “effects”, known as “the Reddit hug of death”, “the Digg effect”, being “drudged”, “farking”, “goonrushing” and “wanging”; respectively.

Routers have also been known to create unintentional DoS attacks, as both **D-Link** and **Netgear** routers have created **NTP vandalism** by flooding NTP servers without respecting the restrictions of client types or geographical limitations.

Similar unintentional denials-of-service can also occur via other media, e.g. when a URL is mentioned on television. If a server is being indexed by Google or another search engine during peak periods of activity, or does not have a lot of available bandwidth while being indexed, it can also experience the effects of a DoS attack.

Legal action has been taken in at least one such case. In 2006, Universal Tube & Rollform Equipment Corporation sued YouTube: massive numbers of would-be youtube.com users accidentally typed the tube company's URL, utube.com. As a result, the tube company ended up having to spend large amounts of money on upgrading their bandwidth.^[61] The company appears to have taken advantage of the situation, with utube.com now containing ads for advertisement revenue.

In March 2014, after Malaysia Airlines Flight 370 went missing, DigitalGlobe launched a crowdsourcing service on which users could help search for the missing jet in satellite images. The response overwhelmed the company's servers.^[62]

An unintentional denial-of-service may also result from a prescheduled event created by the website itself. This could be caused when a server provides some service at a specific time. This might be a university website setting the grades to be available where it will result in many more login requests at that time than any other.

3.9.6 Side effects of attacks

Backscatter

See also: Backscatter (email) and Internet background noise

In computer network security, backscatter is a side-effect of a spoofed denial-of-service attack. In this kind of attack, the attacker spoofs (or forges) the source address in IP packets sent to the victim. In general, the victim machine cannot distinguish between the spoofed packets and legitimate packets, so the victim responds to the spoofed packets as it normally would. These response packets are known as backscatter.^[63]

If the attacker is spoofing source addresses randomly, the backscatter response packets from the victim will be sent back to random destinations. This effect can be used by network telescopes as indirect evidence of such attacks.

The term “backscatter analysis” refers to observing backscatter packets arriving at a statistically significant portion of the IP address space to determine characteristics of DoS attacks and victims.

3.9.7 Legality

See also: Computer crime

Many jurisdictions have laws under which denial-of-service attacks are illegal.

- In the US, denial-of-service attacks may be considered a federal crime under the Computer Fraud and Abuse Act with penalties that include years of imprisonment.^[64] The Computer Crime and Intellectual Property Section of the US Department of Justice handles cases of (D)DoS.
- The United Kingdom is unusual in that it specifically outlawed denial-of-service attacks and set a maximum penalty of 10 years in prison with the Police and Justice Act 2006, which amended Section 3 of the Computer Misuse Act 1990.^[65]

On January 7, 2013, Anonymous posted a petition on the whitehouse.gov site asking that DDoS be recognized as a legal form of protest similar to the Occupy protests.^{[66][67]}

3.9.8 See also

Columns:

- Application layer DDoS attack
- Billion laughs
- DDoS mitigation
- DoSnet
- Hit-and-run DDoS
- Industrial espionage
- Intrusion detection system
- LAND
- Low Orbit Ion Cannon
- Network intrusion detection system
- ReDoS
- SlowDroid
- UDP Unicorn
- Virtual sit-in
- Wireless signal jammer

3.9.9 References

- [1] “Brand.com President Mike Zammuto Reveals Blackmail Attempt”. 5 March 2014. Archived from the original on 11 March 2014.
- [2] “Brand.com’s Mike Zammuto Discusses Meetup.com Extortion”. 5 March 2014. Archived from the original on 13 March 2014.
- [3] “The Philosophy of Anonymous”. Radicalphilosophy.com. 2010-12-17. Retrieved 2013-09-10.
- [4] McDowell, Mindi (November 4, 2009). “Cyber Security Tip ST04-015 - Understanding Denial-of-Service Attacks”. United States Computer Emergency Readiness Team. Archived from the original on 2013-11-04. Retrieved December 11, 2013.
- [5] Kumar, S. (2010). “Denial of Service Due to Direct and Indirect ARP Storm Attacks in LAN Environment*”. *Journal of Information Security* **01** (2): 88–80. doi:10.4236/jis.2010.12010. (PDF)
- [6] Taghavi Zargar, Saman (November 2013). “A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks” (PDF) **15** (4). IEEE COMMUNICATIONS SURVEYS & TUTORIALS. pp. 2046–2069. Retrieved 2014-03-07.
- [7] “Types of DDoS Attacks”. *Distributed Denial of Service Attacks(DDoS) Resources, Pervasive Technology Labs at Indiana University*. Advanced Networking Management Lab (ANML). December 3, 2009. Archived from the original on 2010-09-14. Retrieved December 11, 2013.
- [8] “RFC 4987 – TCP SYN Flooding Attacks and Common Mitigations”. Tools.ietf.org. August 2007. Retrieved 2011-12-02.
- [9] “CERT Advisory CA-1997-28 IP Denial-of-Service Attacks”. CERT. 1998. Retrieved July 18, 2014.
- [10] “Windows 7, Vista exposed to ‘teardrop attack’”. ZDNet. September 8, 2009. Retrieved 2013-12-11.
- [11] “Microsoft Security Advisory (975497): Vulnerabilities in SMB Could Allow Remote Code Execution”. Microsoft.com. September 8, 2009. Retrieved 2011-12-02.
- [12] Paul Sop (May 2007). “Prolexic Distributed Denial of Service Attack Alert”. Prolexic Technologies Inc. Prolexic Technologies Inc. Archived from the original on 2007-08-03. Retrieved 2007-08-22.
- [13] Robert Lemos (May 2007). “Peer-to-peer networks co-opted for DOS attacks”. SecurityFocus. Retrieved 2007-08-22.
- [14] Fredrik Ullner (May 2007). “Denying distributed attacks”. DC++: Just These Guys, Ya Know?. Retrieved 2007-08-22.
- [15] “Phlashing attack thrashes embedded systems”. *The Register*. Retrieved 2009-03-07. |first1= missing |last1= in Authors list (help)
- [16] Jackson Higgins, Kelly (May 19, 2008). “Permanent Denial-of-Service Attack Sabotages Hardware”. Dark Reading. Archived from the original on December 8, 2008.
- [17] “EUSecWest Applied Security Conference: London, U.K.”. EUSecWest. 2008. Archived from the original on 2009-02-01.
- [18] <http://aws.amazon.com/cloudwatch/>
- [19] “OWASP Plan - Strawman - Layer_7_DDOS.pdf” (PDF). Open Web Application Security Project. 18 March 2014. Retrieved 18 March 2014.
- [20] “Has Your Website Been Bitten By a Zombie?”. Cloudbric. 3 August 2015. Retrieved 15 September 2015.
- [21] Dittrich, David (December 31, 1999). “The “stacheldraht” distributed denial of service attack tool”. University of Washington. Retrieved 2013-12-11.
- [22] Boyle, Phillip (2000). “SANS Institute – Intrusion Detection FAQ: Distributed Denial of Service Attack Tools: n/a”. SANS Institute. Retrieved 2008-05-02.
- [23] Leyden, John (2004-09-23). “US credit card firm fights DDoS attack”. *The Register*. Retrieved 2011-12-02.
- [24] <https://firstlook.org/theintercept/2014/07/14/manipulating-online-polls-ways-british-spies-seek-control-internet/>
- [25] http://www.internetsociety.org/sites/default/files/01_5.pdf
- [26] Paxson, Vern (2001). “An Analysis of Using Reflectors for Distributed Denial-of-Service Attacks”. ICIR.org.
- [27] “Alert (TA14-017A) UDP-based Amplification Attacks”. US-CERT. July 8, 2014. Retrieved 2014-07-08.
- [28] van Rijswijk-Deij, Roland (2014). “DNSSEC and its potential for DDoS attacks - a comprehensive measurement study”. ACM Press.
- [29] Adamsky, Florian (2015). “P2P File-Sharing in Hell: Exploiting BitTorrent Vulnerabilities to Launch Distributed Reflective DoS Attacks”.
- [30] Vaughn, Randal; Evron, Gadi (2006). “DNS Amplification Attacks” (PDF). ISOTF. Archived from the original (PDF) on 2010-12-14.
- [31] “Alert (TA13-088A) DNS Amplification Attacks”. US-CERT. July 8, 2013. Retrieved 2013-07-17.
- [32] “FBI — Phony Phone Calls Distract Consumers from Genuine Theft”. FBI.gov. 2010-05-11. Retrieved 2013-09-10.
- [33] “Internet Crime Complaint Center’s (IC3) Scam Alerts January 7, 2013”. IC3.gov. 2013-01-07. Retrieved 2013-09-10.
- [34] <http://www.scmagazineuk.com/video-games-company-hit-by-38-day-ddos-attack/article/367329/>

- [35] "Who's Behind DDoS Attacks and How Can You Protect Your Website?". Cloudbric. 10 September 2015. Retrieved 15 September 2015.
- [36] Solon, Olivia (9 September 2015). "Cyber-Extortionists Targeting the Financial Sector Are Demanding Bitcoin Ransoms". Bloomberg. Retrieved 15 September 2015.
- [37] Greenberg, Adam (14 September 2015). "Akamai warns of increased activity from DDoS extortion group". SC Magazine. Retrieved 15 September 2015.
- [38] Loukas, G.; Oke, G. (September 2010) [August 2009]. "Protection Against Denial of Service Attacks: A Survey" (PDF). *Comput. J.* 53 (7): 1020–1037. doi:10.1093/comjnl/bxp078.
- [39] Froutan, Paul (June 24, 2004). "How to defend against DDoS attacks". *Computerworld*. Retrieved May 15, 2010.
- [40] Suzen, Mehmet. "Some IoT tips for Internet Service (Providers)" (PDF). Archived from the original (PDF) on 2008-09-12.
- [41] Kousiouris, George (2014). "KEY COMPLETION INDICATORS:minimizing the effect of DoS attacks on elastic Cloud-based applications based on application-level markov chain checkpoints". *CLOSER Conference*. Retrieved 2015-05-24.
- [42] Abante, Carl (March 2, 2013). "Relationship between Firewalls and Protection against DDoS". *Ecommerce Wisdom*. Retrieved 2013-05-24.
- [43] Patrikakis, C.; Masikos, M.; Zouraraki, O. (December 2004). "Distributed Denial of Service Attacks". *The Internet Protocol Journal* 7 (4): 13–35.
- [44] "DDoS Mitigation via Regional Cleaning Centers (Jan 2004)" (PDF). *SprintLabs.com*. Sprint ATL Research. Archived from the original (PDF) on 2008-09-21. Retrieved 2011-12-02.
- [45] "Defensepipe". *radware.com*. Retrieved November 2015.
- [46] "Clean Pipes DDoS Protection and Mitigation from Arbor Networks & Cisco". *ArborNetworks.com*. 8 August 2013.
- [47] "AT&T Internet Protect Distributed Denial of Service Defense" (PDF). *ATT.com* (Product brief). 16 October 2012.
- [48] "Silverline DDoS Protection service". *f5.com*. Retrieved March 2015.
- [49] "Infrastructure DDos Protection". *incapsula.com*. Retrieved June 2015.
- [50] "DDoS Protection". *Neustar.biz*. Retrieved November 2014.
- [51] "Spamhaus-style DDoS attacks: All the hackers are doing it". Darkreading.com. 3 June 2013. Retrieved 2014-01-07.
- [52] "Managed Security". *Sprint.com*. Retrieved January 2015.
- [53] "Staminus Establishes Presence in Telx's 60 Hudson Street Data Center in New York". 3 March 2014. Retrieved 2 July 2015.
- [54] "Staminus Hybrid DDoS Protection & Mitigation Security". *Staminus.net*. Retrieved 6 October 2015.
- [55] "DDoS Protection with Network Agnostic Option". *Tat-acomunications.com*. 7 September 2011.
- [56] "VeriSign Rolls Out DDoS Monitoring Service". Darkreading.com. 11 September 2009. Retrieved 2 December 2011.
- [57] "Security: Enforcement and Protection". *Verizon.com*. Retrieved January 2015.
- [58] "Verizon Digital Media Services Launches Cloud-Based Web Application Firewall That Increases Defenses Against Cyberattacks". *Verizon.com*. Retrieved January 2015.
- [59] Shiels, Maggie (2009-06-26). "Web slows after Jackson's death". *BBC News*.
- [60] "We're Sorry. Automated Query error". *Google Product Forums > Google Search Forum*. Google.com. October 20, 2009. Retrieved 2012-02-11.
- [61] "YouTube sued by sound-alike site". *BBC News*. 2006-11-02.
- [62] People Overload Website, Hoping To Help Search For Missing Jet | WNMU-FM
- [63] "Backscatter Analysis (2001)". *Animations* (video). Cooperative Association for Internet Data Analysis. Retrieved December 11, 2013.
- [64] "United States Code: Title 18,1030. Fraud and related activity in connection with computers | Government Printing Office". www.gpo.gov. 2002-10-25. Retrieved 2014-01-15.
- [65] "Computer Misuse Act 1990". *legislation.gov.uk — The National Archives, of UK*. 10 January 2008.
- [66] "Anonymous DDoS Petition: Group Calls On White House To Recognize Distributed Denial Of Service As Protest". HuffingtonPost.com. 2013-01-12.
- [67] "DDOS Attack: crime or virtual sit-in?". RT.com. YouTube.com. October 6, 2011.

3.9.10 Further reading

- Ethan Zuckerman, Hal Roberts, Ryan McGrady, Jillian York, John Palfrey (December 2011). "Distributed Denial of Service Attacks Against Independent Media and Human Rights Sites". The Berkman Center for Internet & Society at Harvard University. Archived from the original (PDF) on 2011-05-01. Retrieved 2011-03-02.
- "DDOS Public Media Reports". Harvard. Archived from the original on 2011-05-01.

- Christian Rossow (22 Feb 2014). “Amplification Hell: Revisiting Network Protocols for DDoS Abuse”. Internet Society. Retrieved 2015-01-13.

3.9.11 External links

- http://www.scmagazineuk.com/video-games-company-hit-by-38-day-ddos-attack/article/367329/* RFC 4732 Internet Denial-of-Service Considerations
- W3C The World Wide Web Security FAQ
- cert.org CERT's Guide to DoS attacks. (historic document)
- ATLAS Summary Report – Real-time global report of DDoS attacks.
- Report: Distributed Denial of Service Attacks Against Independent Media and Human Rights Sites Berkman Center for Internet and Society Report on DDoS

3.10 Malware



Beast, a Windows-based backdoor Trojan horse.

Malware, short for **malicious software**, is any software used to disrupt computer operations, gather sensitive information, or gain access to private computer systems.^[1] 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.^[2]

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

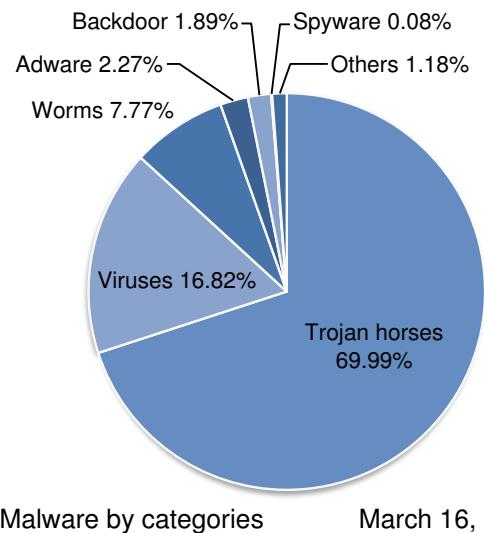
to extort payment (**CryptoLocker**). 'Malware' is an umbrella term used to refer to a variety of forms of hostile or intrusive software,^[3] including computer viruses, worms, trojan horses, ransomware, spyware, adware, scareware, and other malicious programs. It can take the form of executable code, scripts, active content, and other software.^[4] 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.^[5]

In law, malware is sometimes known as a **computer contaminant**, as in the legal codes of several U.S. states.^{[6][7]}

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.^[8]

Software such as anti-virus, anti-malware, and firewalls are used to protect against activity identified as malicious, and to recover from attacks.^[9]

3.10.1 Purposes



Malware by categories on 16 March 2011.

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.^{[10][11]}

Malware is sometimes used broadly against government or corporate websites to gather guarded information,^[12] 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 against by various types of firewall, anti-virus software, and network hardware).^[13]

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.^[14] Infected "zombie computers" are used to send email spam, to host contraband data such as child pornography,^[15] or to engage in distributed denial-of-service attacks as a form of extortion.^[16]

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 packaged together with user-installed software, such as peer-to-peer applications.^[17]

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.^[18]

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.Distrack) and Saudi Aramco (August 2012).^{[19][20]}

3.10.2 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."^[21] According to F-Secure,

"As much malware [was] produced in 2007 as in the previous 20 years altogether."^[22] Malware's most common pathway from criminals to users is through the **Internet**: primarily by e-mail and the **World Wide Web**.^[23]

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.^[24] 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.^[25]

On 29 March 2010, **Symantec** Corporation named Shaoxing, China, as the world's malware capital.^[26] 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.^[27]

A 2014 study found that malware is being increasingly aimed at mobile devices such as smartphones as they increase in popularity.^[28]

3.10.3 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.^[29]

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

These categories are not mutually exclusive, so malware may use multiple techniques.^[30] This section only applies to malware designed to operate undetected, not sabotage and ransomware.

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).^[31]

Trojan horses

Main article: Trojan horse (computing)

For a malicious program to accomplish its goals, it must be able to run without being detected, shut down, or deleted. When a malicious program is disguised as something normal or desirable, users may unwittingly install it. This is the technique of the *Trojan horse* or *trojan*. In broad terms, a Trojan horse is any program that invites the user to run it, concealing harmful or malicious executable code of any description. The code may take effect immediately and can lead to many undesirable effects, such as encrypting the user's files or downloading and implementing further malicious functionality.

In the case of some spyware, adware, etc. the supplier may require the user to acknowledge or accept its installation, describing its behavior in loose terms that may easily be misunderstood or ignored, with the intention of deceiving the user into installing it without the supplier technically in breach of the law.

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.^[32]

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.^[33]

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,^[34] 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.^[35] Backdoors may be installed by Trojan horses, worms, implants, or other methods.^{[36][37]}

Evasion

Since the beginning of 2015, a sizable portion of malware utilizes a combination of many techniques designed to avoid detection and analysis.^[38]

- The most common evasion technique is when the malware evades analysis and detection by fingerprinting the environment when executed.^[39]
- 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.^[40]
- 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.

3.10.5 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^[41]), or in vulnerable versions of browser plugins such as Adobe Flash Player, Adobe Acrobat or Reader, or Java (see Java SE critical security issues).^{[42][43]} Sometimes even installing new versions of such plugins does not automatically uninstall old versions. Security advisories from plug-in providers announce security-related updates.^[44] Common vulnerabilities are assigned [CVE IDs](#) and listed in the US National Vulnerability Database. Secunia PSI^[45] 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.^[46] 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.^[46]

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.^[46] 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 overprivileged 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 priv-

ilege 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.^[47] 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.

3.10.6 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.^[48]

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^[49] (for Windows XP, Vista, and Windows 7) for real-time protection, the Windows Malicious Software Removal Tool^[50] (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).^[51] Additionally, several capable antivirus software programs are available for free download from the Internet (usually restricted to non-commercial use).^[52] Tests found some free programs to be competitive with commercial ones.^[52] 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,^[53] and then using system tools or Microsoft Safety Scanner.^[54]

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.^{[55][56][57][58]} 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, information can be transmitted in unrecognized ways; in December 2013 researchers in Germany showed one way that an apparent air gap can be defeated.^[59]

Later in 2015, “BitWhisper”, a Covert Signaling Channel between Air-Gapped Computers using Thermal Manipulations was introduced. “BitWhisper” supports bidirectional communication and requires no additional dedicated peripheral hardware.^{[60][61]}

3.10.7 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.^[62]

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.^[63]

Another term, PUP, which stands for *Potentially Unwanted Program* (or PUA *Potentially Unwanted Application*),^[64] 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.^[65]

3.10.8 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.^[66] 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^[67] and the L10n worm^[68] 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.^[69]

3.10.9 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.^[70] 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.^[71]

3.10.10 See also

- Browser hijacking
- Category:Web security exploits
- Comparison of antivirus software
- Computer insecurity
- Cyber spying
- Identity theft
- Industrial espionage
- Malvertising
- Phishing
- Riskware
- Security in Web applications
- Social engineering (security)
- Targeted threat
- Typosquatting
- Web server overload causes

3.10.11 References

- [1] "Malware definition". techterms.com. Retrieved 27 September 2015.
- [2] "What is badware?". StopBadware. Retrieved 18 February 2015.
- [3] "Defining Malware: FAQ". technet.microsoft.com. Retrieved 10 September 2009.
- [4] "An Undirected Attack Against Critical Infrastructure" (PDF). United States Computer Emergency Readiness Team(Us-cert.gov). Retrieved 28 September 2014.
- [5] "Evolution of Malware-Malware Trends". Microsoft Security Intelligence Report-Featured Articles. Microsoft.com. Retrieved 28 April 2013.
- [6] "Virus/Contaminant/Destructive Transmission Statutes by State". National Conference of State Legislatures. 2012-02-14. Retrieved 26 August 2013.
- [7] "§ 18.2-152.4:1 Penalty for Computer Contamination" (PDF). Joint Commission on Technology and Science. Retrieved 17 September 2010.
- [8] Russinovich, Mark (2005-10-31). "Sony, Rootkits and Digital Rights Management Gone Too Far". *Mark's Blog*. Microsoft MSDN. Retrieved 2009-07-29.
- [9] "Protect Your Computer from Malware". OnGuardOnline.gov. Retrieved 26 August 2013.
- [10] "Malware". FEDERAL TRADE COMMISSION- CONSUMER INFORMATION. Retrieved 27 March 2014.
- [11] Hernandez, Pedro. "Microsoft Vows to Combat Government Cyber-Spying". eWeek. Retrieved 15 December 2013.
- [12] Kovacs, Eduard. "MiniDuke Malware Used Against European Government Organizations". Softpedia. Retrieved 27 February 2013.
- [13] "South Korea network attack 'a computer virus'". BBC. Retrieved 20 March 2013.
- [14] "Malware Revolution: A Change in Target". March 2007.
- [15] "Child Porn: Malware's Ultimate Evil". November 2009.
- [16] PC World – Zombie PCs: Silent, Growing Threat.
- [17] "Peer To Peer Information". NORTH CAROLINA STATE UNIVERSITY. Retrieved 25 March 2011.
- [18] "Another way Microsoft is disrupting the malware ecosystem". Retrieved 18 February 2015.
- [19] "Shamoon is latest malware to target energy sector". Retrieved 18 February 2015.
- [20] "Computer-killing malware used in Sony attack a wake-up call". Retrieved 18 February 2015.
- [21] "Symantec Internet Security Threat Report: Trends for July–December 2007 (Executive Summary)" (PDF) XIII. Symantec Corp. April 2008. p. 29. Retrieved 11 May 2008.
- [22] "F-Secure Reports Amount of Malware Grew by 100% during 2007" (Press release). F-Secure Corporation. 4 December 2007. Retrieved 11 December 2007.
- [23] "F-Secure Quarterly Security Wrap-up for the first quarter of 2008". F-Secure. 31 March 2008. Retrieved 25 April 2008.
- [24] "Continuing Business with Malware Infected Customers". Gunter Ollmann. October 2008.
- [25] "New Research Shows Remote Users Expose Companies to Cybercrime". Webroot. April 2013.
- [26] "Symantec names Shaoxing, China as world's malware capital". Engadget. Retrieved 15 April 2010.
- [27] Rooney, Ben (2011-05-23). "Malware Is Posing Increasing Danger". Wall Street Journal.
- [28] 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*.
- [29] "computer virus – Encyclopedia Britannica". Britannica.com. Retrieved 28 April 2013.
- [30] All about Malware and Information Privacy

- [31] "What are viruses, worms, and Trojan horses?". *Indiana University*. The Trustees of Indiana University. Retrieved 23 February 2015.
- [32] McDowell, Mindi. "Understanding Hidden Threats: Rootkits and Botnets". US-CERT. Retrieved 6 February 2013.
- [33] "Catb.org". Catb.org. Retrieved 15 April 2010.
- [34] Vincentas (11 July 2013). "Malware in Spy-WareLoop.com". Spyware Loop. Retrieved 28 July 2013.
- [35] Staff, SPIEGEL. "Inside TAO: Documents Reveal Top NSA Hacking Unit". SPIEGEL. Retrieved 23 January 2014.
- [36] Edwards, John. "Top Zombie, Trojan Horse and Bot Threats". IT Security. Retrieved 25 September 2007.
- [37] Appelbaum, Jacob. "Shopping for Spy Gear:Catalog Advertises NSA Toolbox". SPIEGEL. Retrieved 29 December 2013.
- [38] Evasive malware
- [39] 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.
- [40] The Four Most Common Evasive Techniques Used by Malware. April 27, 2015.
- [41] "Global Web Browser... Security Trends" (PDF). Kaspersky lab. November 2012.
- [42] Rashid, Fahmida Y. (27 November 2012). "Updated Browsers Still Vulnerable to Attack if Plugins Are Outdated". pcmag.com.
- [43] Danchev, Dancho (18 August 2011). "Kaspersky: 12 different vulnerabilities detected on every PC". pcmag.com.
- [44] "Adobe Security bulletins and advisories". Adobe.com. Retrieved 19 January 2013.
- [45] Rubenking, Neil J. "Secunia Personal Software Inspector 3.0 Review & Rating". PCMag.com. Retrieved 19 January 2013.
- [46] "USB devices spreading viruses". CNET. CBS Interactive. Retrieved 18 February 2015.
- [47] "LNCS 3786 – Key Factors Influencing Worm Infection", U. Kanlayasiri, 2006, web (PDF): SL40-PDF.
- [48] "How Antivirus Software Works?". Retrieved 2015-10-16.
- [49] "Microsoft Security Essentials". Microsoft. Retrieved 21 June 2012.
- [50] "Malicious Software Removal Tool". Microsoft. Retrieved 21 June 2012.
- [51] "Windows Defender". Microsoft. Retrieved 21 June 2012.
- [52] Rubenking, Neil J. (8 January 2014). "The Best Free Antivirus for 2014". pcmag.com.
- [53] "How do I remove a computer virus?". Microsoft. Retrieved 26 August 2013.
- [54] "Microsoft Safety Scanner". Microsoft. Retrieved 26 August 2013.
- [55] "An example of a website vulnerability scanner". Unmaskparasites.com. Retrieved 19 January 2013.
- [56] "Redleg's File Viewer. Used to check a webpage for malicious redirects or malicious HTML coding". Awsnap.info. Retrieved 19 January 2013.
- [57] "Example Google.com Safe Browsing Diagnostic page". Google.com. Retrieved 19 January 2013.
- [58] "Safe Browsing (Google Online Security Blog)". Retrieved 21 June 2012.
- [59] Hanspach, Michael; Goetz, Michael (November 2013). "On Covert Acoustical Mesh Networks in Air". *Journal of Communications*. doi:10.12720/jcm.8.11.758-767.
- [60] Guri, Mordechai; Monitz, Matan; Mirski, Yisroel; Elovici, Yuval (April 2015). "BitWhisper: Covert Signaling Channel between Air-Gapped Computers using Thermal Manipulations". arXiv (C) IEEE.
- [61] Guri, Mordechai; Monitz, Matan; Mirski, Yisroel; Elovici, Yuval (March 2015). "BitWhisper: The Heat is on the Air-Gap". BGU Cyber Security Labs.
- [62] Vincentas (11 July 2013). "Grayware in Spy-WareLoop.com". Spyware Loop. Retrieved 28 July 2013.
- [63] "Threat Encyclopedia – Generic Grayware". Trend Micro. Retrieved 27 November 2012.
- [64] "Rating the best anti-malware solutions". ArsTechnica. Retrieved 28 January 2014.
- [65] "PUP Criteria". malwarebytes.org. Retrieved 13 February 2015.
- [66] William A Hendric (4 September 2014). "Computer Virus history". The Register. Retrieved 29 March 2015.
- [67] Nick Farrell (20 February 2006). "Linux worm targets PHP flaw". The Register. Retrieved 19 May 2010.
- [68] John Leyden (28 March 2001). "Highly destructive Linux worm mutating". The Register. Retrieved 19 May 2010.
- [69] "Aggressive net bug makes history". BBC News. 3 February 2003. Retrieved 19 May 2010.
- [70] 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.
- [71] Fred Cohen, "Computer Viruses", PhD Thesis, University of Southern California, ASP Press, 1988.

3.10.12 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

3.11 Payload (computing)

Payload in computing (sometimes referred to as the actual or body data) is the cargo of a **data transmission**. It is the part of the transmitted **data** which is the fundamental purpose of the transmission, to the exclusion of information sent with it (such as **headers** or **metadata**, sometimes referred to as overhead data) solely to facilitate delivery.^{[1][2]}

In computer security, payload refers to the part of malware which performs a malicious action.^[3] In the analysis of malicious software such as worms, viruses and Trojans, it refers to the software's harmful results. Examples of payloads include data destruction, messages with insulting text or spurious e-mail messages sent to a large number of people.

In summary, payload refers to the actual intended message in a transmission.

3.11.1 References

- [1] "Payload definition". Pcmag.com. 1994-12-01. Retrieved 2012-02-07.
- [2] "Payload definition". Techterms.com. Retrieved 2012-02-07.
- [3] "Payload definition". Securityfocus.com. Retrieved 2012-02-07.

3.12 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.

3.12.1 History

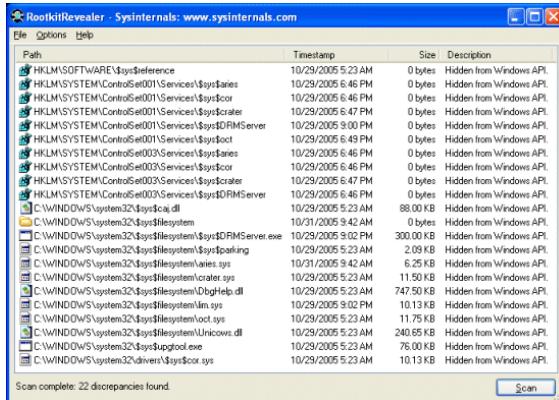
The term *rootkit* or *root kit* originally referred to a maliciously modified set of administrative tools for a Unix-like operating system that granted "root" access.^[3] If an intruder could replace the standard administrative tools on a system with a rootkit, the intruder could obtain root access 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 modifica-

tions 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.

3.12.2 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, which can be very inconvenient even to those who did legitimately purchase it.

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, bypass-

ing 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).

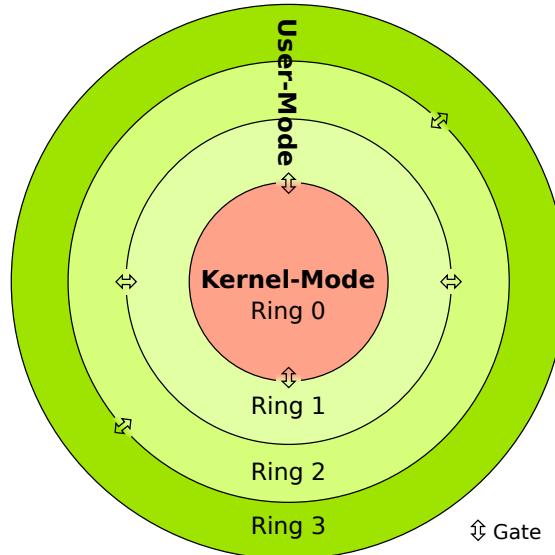
In some instances, rootkits provide desired functionality, and may be installed intentionally on behalf of the computer user:

- 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]

3.12.3 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]



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

User mode

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.^[33] 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 his or her 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 undetectable spying and redirection by management or hackers who might gain control.

3.12.4 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.

3.12.5 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 Zeppo,[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 pro-

grams. 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

```

Terminal — bash — bash (ttyp1) — 96x50 — #1

Your system contains some unknown version numbers. Please run Rootkit Hunter
with the --update parameter or fill in the contact form (www.rootkit.nl)

Security advisories
* Check: Groups and Accounts
  Searching for /etc/passwd...
  Checking users with UID '0' (root)... [ Found ] [ OK ]
* Check: SSH
  Searching for sshd_config...
  Found /etc/sshd_config
  Checking for allowed root login... Watch out! Root login possible. Possible risk!
  Info: Hint: See logfile for more information about this issue [ Warning (SSH v1 allowed) ]
  Checking for allowed protocols...
* Check: Events and Logging
  Search for syslog configuration...
  Checking for running syslog slave...
  Checking for logging to remote system...
  Info: Install... [ OK ] [ OK ] [ OK (remote logging) ] 8127.8.0.1:32376
[Press <ENTER> to continue]

----- Scan results -----
MD5
MD5 compared: 0
Incorrect MD5 checksums: 0

File scan
Scanned files: 342
Possible infected files: 0

Application scan
Vulnerable applications: 0

Scanning took 129 seconds

Do you have some problems, undetected rootkits, false positives, ideas
or suggestions?
Please e-mail me by filling in the contact form (http://www.rootkit.nl)

```

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.

3.12.6 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]

3.12.7 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.

3.12.8 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.

3.12.9 See also

- Hacker con
- Host-based intrusion detection system
- Man-in-the-middle attack
- The Rootkit Arsenal: Escape and Evasion in the Dark Corners of the System

3.12.10 Notes

- [1] The process name of Sysinternals RootkitRevealer was targeted by malware; in an attempt to counter this counter-measure, 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.

3.12.11 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. Archived from the original on 2012-09-11. Retrieved 2010-12-04.
- [11] “Spyware Detail: XCP.Sony.Rootkit”. Computer Associates. 2005-11-05. Archived from the original on 2012-09-21. Retrieved 2010-08-19.
- [12] Russinovich, Mark (2005-10-31). “Sony, Rootkits and Digital Rights Management Gone Too Far”. *TechNet Blogs*. Microsoft. Archived from the original on 2012-07-07. Retrieved 2010-08-16.
- [13] “Sony’s long-term rootkit CD woes”. BBC News. 2005-11-21. Archived from the original on 2012-07-15. Retrieved 2008-09-15.
- [14] Felton, Ed (2005-11-15). “Sony’s Web-Based Uninstaller Opens a Big Security Hole; Sony to Recall Discs”. Archived from the original on 2012-09-05.
- [15] Knight, Will (2005-11-11). “Sony BMG sued over cloaking software on music CD”. *New Scientist* (Sutton, UK: Reed Business Information). Archived from the original on 2012-09-21. 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”. Archived from the original on 2012-09-21.
- [18] Russinovich, Mark (June 2005). “Unearthing Root Kits”. Windows IT Pro. Archived from the original on 2012-09-18. Retrieved 2010-12-16.
- [19] “World of Warcraft Hackers Using Sony BMG Rootkit”. The Register. 2005-11-04. Archived from the original on 2012-09-17. 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*. Sys-Internals. 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). Archived from the original on 2012-09-12.
- [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). Archived from the original on 2012-07-14. Retrieved 2010-11-21.
- [29] Shevchenko, Alisa (2008-09-01). "Rootkit Evolution". *Help Net Security*. Help Net Security. p. 2. Archived from the original on 2012-09-03.
- [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. Archived from the original on 2012-09-11. Retrieved 2010-11-13.
- [32] Butler, James; Sparks, Sherri (2005-11-03). "Windows Rootkits of 2005, Part One". *Symantec Connect*. Symantec. Archived from the original on 2012-09-12. Retrieved 2010-11-12.
- [33] Burdach, Mariusz (2004-11-17). "Detecting Rootkits And Kernel-level Compromises In Linux". Symantec. Archived from the original on 2012-09-13. 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. Archived from the original on 2012-05-30. Retrieved 2008-07-06.
- [36] Soeder, Derek; Permeh, Ryan (2007-05-09). "Bootroot". eEye Digital Security. Archived from the original on 2012-09-21. Retrieved 2010-11-23.
- [37] Schneier, Bruce (2009-10-23). "'Evil Maid' Attacks on Encrypted Hard Drives". Archived from the original on 2012-09-11. 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. Archived from the original on 2012-09-21. Retrieved 2009-11-07.
- [41] Goodin, Dan (2010-11-16). "World's Most Advanced Rootkit Penetrates 64-bit Windows". The Register. Archived from the original on 2012-09-21. 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](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. Archived from the original on 2012-09-12. 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. Archived from the original on 2012-09-21. Retrieved 2009-03-25.
- [56] Sacco, Anibal; Ortéga, Alfredo (2009-06-01). “Persistent BIOS Infection: The Early Bird Catches the Worm”. *Phrack* 66 (7). Archived from the original on 2012-07-17. 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. Archived from the original on 2012-05-27.
- [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”. Retrieved 13 Sep 2012.
- [64] “Zeppo”. SourceForge. 18 July 2009. Archived from the original on 2012-07-19. Retrieved 8 August 2011.
- [65] Cogswell, Bryce; Russinovich, Mark (2006-11-01). “RootkitRevealer v1.71”. Microsoft. Archived from the original on 2012-06-04. Retrieved 2010-11-13.
- [66] “Sophos Anti-Rootkit”. Sophos. Archived from the original on 2012-09-21. Retrieved 8 August 2011.
- [67] “BlackLight”. F-Secure. Archived from the original on 2012-09-21. Retrieved 8 August 2011.
- [68] “Radix Anti-Rootkit”. usec.at. Archived from the original on 2012-09-21. Retrieved 8 August 2011.
- [69] “GMER”. Archived from the original on 2012-08-02. 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. Archived from the original on 2012-09-21. Retrieved 2010-08-19.
- [72] “Restart Issues After Installing MS10-015”. Microsoft. 2010-02-11. Archived from the original on 2012-07-07. Retrieved 2010-10-05.
- [73] “Strider GhostBuster Rootkit Detection”. Microsoft Research. 2010-01-28. Archived from the original on 2012-07-29. Retrieved 2010-08-14.
- [74] “Signing and Checking Code with Authenticode”. Microsoft. Archived from the original on 2012-09-21. 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. Archived from the original on 2012-07-20. 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 Reveal vs. Hacker Defender”. Archived from the original on 2012-07-13.
- [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. Archived from the original on 2012-09-21.
- [81] Hultquist, Steve (2007-04-30). “Rootkits: The next big enterprise threat?”. *InfoWorld* (IDG). Archived from the original on 2012-09-21. Retrieved 2010-11-21.
- [82] “Security Watch: Rootkits for fun and profit”. CNET Reviews. 2007-01-19. Archived from the original on 2012-07-18. Retrieved 2009-04-07.
- [83] Bort, Julie (2007-09-29). “Six ways to fight back against botnets”. PCWorld. San Francisco: PCWorld Communications. Archived from the original on 2012-09-07. Retrieved 2009-04-07.

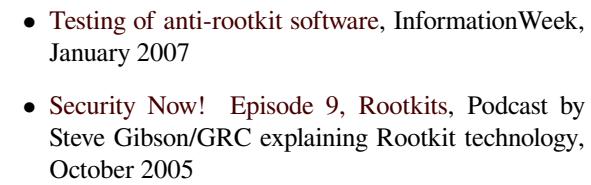
- [84] Hoang, Mimi (2006-11-02). "Handling Today's Tough Security Threats: Rootkits". *Symantec Connect*. Symantec. Archived from the original on 2012-09-21. Retrieved 2010-11-21.
 - [85] Danseglio, Mike; Bailey, Tony (2005-10-06). "Rootkits: The Obscure Hacker Attack". Microsoft. Archived from the original on 2012-09-21.
 - [86] Messmer, Ellen (2006-08-26). "Experts Divided Over Rootkit Detection and Removal". *NetworkWorld.com* (Framingham, Mass.: IDG). Archived from the original on 2012-09-03. 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.

3.12.12 Further reading

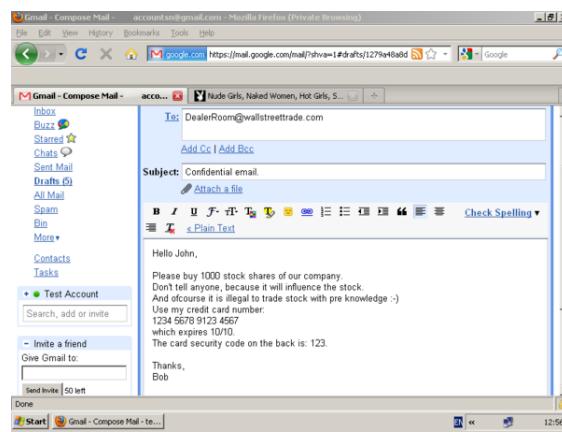
- Blunden, Bill (2009). *The Rootkit Arsenal: Escape and Evasion in the Dark Corners of the System*. Wordware. ISBN 978-1-59822-061-2.
 - Hoglund, Greg; Butler, James (2005). *Rootkits: Subverting the Windows Kernel*. Addison-Wesley Professional. ISBN 0-321-29431-9.
 - Grampp, F. T.; Morris, Robert H., 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.

3.12.13 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



3.13 Keystroke logging

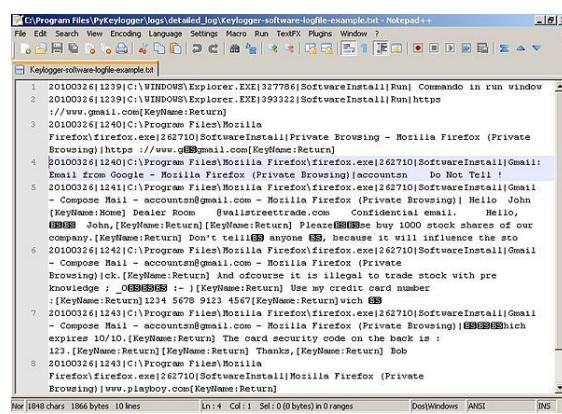


A keylogger example of a screencapture, which holds potentially confidential and private information. The image below holds the corresponding keylogger text result.

Keystroke logging, often referred to as **keylogging** or **keyboard capturing**, is the action of recording (or logging) the keys struck on a **keyboard**, typically in a covert manner so that the person using the keyboard is unaware that their actions are being monitored.^[1] It has uses in the study of human–computer interaction. There are numerous keylogging methods, ranging from hardware and software-based approaches to acoustic analysis.

3.13.1 Application

Software-based keyloggers



A logfile from a software-based keylogger, based on the screen-capture above.

These are computer programs designed to work on the target computer's software.^[2] Keyloggers are used in IT organizations to troubleshoot technical problems with computers and business networks. Other legal uses include family or business people using them to monitor the network usage without their users' direct knowledge. However, malicious individuals may 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 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 starts intercepting keystrokes that pass through the kernel. This method is difficult both to write and to combat. Such keyloggers reside at the **kernel level** and are thus difficult to detect, especially for user-mode applications who don't have root access. They are frequently implemented as **rootkits** that subvert the operating system kernel and gain unauthorized access to the hardware, making 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 for 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. These happen when the user finishes filling in a form and submits it usually by clicking a button or hitting enter. This records form data before it is passed over the Internet.
- **Memory injection based:** Memory Injection (**MitB**)-based keyloggers alter memory tables associated with the browser and other system functions to perform their logging functions. By patching the

memory tables or injecting directly into memory, this technique can be used by malware authors who are looking 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 need to be thwarted somehow by the keylogger.

- **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 the locally recorded data from a remote location. Remote communication may be achieved using one of these methods:

- 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 to be accessed.

Most of these 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 has become an established research method to study writing processes.^{[6][7]} Different programs have been developed to collect online process data of writing activities,^[8] including **Inputlog**, **Scriptlog**, and **Translog**.

In terms of legitimate uses, Keystroke logging can be a suitable research instrument in a number of writing contexts. These include studies on cognitive writing processes, description 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 in research specifically on writing, 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:

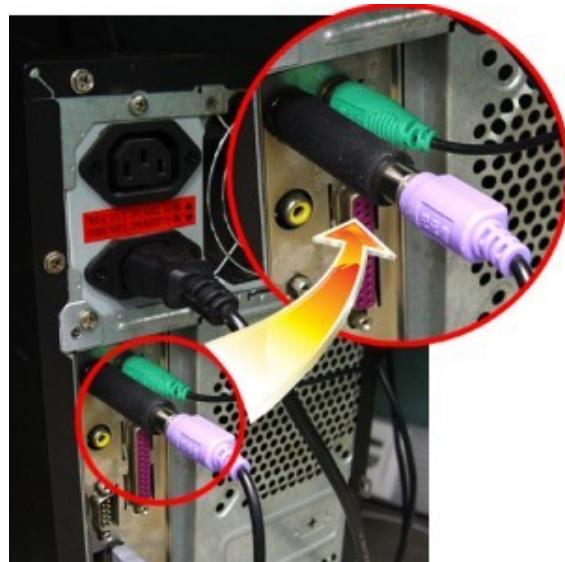
- Clipboard logging. Anything that has been copied to the **clipboard** can be captured by the program.
- Screen logging. Screenshots are taken in order to capture graphics-based information. Applications with screen logging abilities may take screenshots of the whole screen, just one application or even just around the mouse cursor. They may take these screenshots periodically or in response to user behaviours (for example, when a user has clicked the mouse). A practical application used by some keyloggers with this screen logging ability is to take small screenshots around where a mouse has just clicked; these defeat 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.

Main article: [Hardware keylogger](#)



A connected hardware-based 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 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.

- 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.^[13]
- 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.^[14] A fairly long recording (1000 or more keystrokes) is required so that a big enough sample is collected.^[15]
- 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.^[16] 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.^[17] 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.^{[18][19]}
- 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.^[20] 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.^[21] Similar techniques have also been shown to be effective at capturing keystrokes on touchscreen keyboards^{[22][23][24]} while in some cases, in combination with gyroscope.^{[25][26]}

3.13.2 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.^[27] 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.^{[28][29]} They installed the bugs in **Selectric II** and **Selectric III** electric typewriters.^[30]

Soviet embassies used manual typewriters, rather than electric typewriters, for classified information—apparently because they are immune to such bugs.^[30] As of 2013, Russian special services still use typewriters.^{[29][31][32]}

3.13.3 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.

Trojan

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.^[33] ^[34] 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**. 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.^[33]

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**.^[35] 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.^[36]

3.13.4 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.

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^[37] 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.^[38] 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 software

Keystroke interference software is also available.^[39] 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^[40] 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 “asdfsdf”. Then, these dummies could be selected with the mouse, and the next character from the password “e” is typed, which replaces the dummies “asdfsdf”.

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.

3.13.5 See also

- Anti keylogger
- Black-bag cryptanalysis
- Computer surveillance
- Digital footprint
- Hardware keylogger
- Reverse connection
- Spyware
- Trojan horse
- Virtual keyboard

3.13.6 References

- [1] “Keylogger”. Oxford dictionaries.
- [2] “What is a Keylogger?”. PC Tools.
- [3] “The Evolution of Malicious IRC Bots” (PDF). Symantec. 2005-11-26. pp. 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 attacks against x86 pre-boot authentication software)” (PDF). Iviz Technosolutions. Retrieved 2008-09-23.
- [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 Spy-WareLoop.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.
- [12] “Keylogger Removal”. Keylogger Removal. SpyReveal Anti Keylogger. Retrieved 25 April 2011.
- [13] Jeremy Kirk (2008-12-16). “Tampered Credit Card Terminals”. IDG News Service. Retrieved 2009-04-19.
- [14] Andrew Kelly (2010-09-10). “Cracking Passwords using Keyboard Acoustics and Language Modeling” (PDF).
- [15] Sarah Young (14 September 2005). “Researchers recover typed text using audio recording of keystrokes”. UC Berkeley NewsCenter.
- [16] “Remote monitoring uncovered by American techno activists”. ZDNet. 2000-10-26. Retrieved 2008-09-23.
- [17] Martin Vuagnoux and Sylvain Pasini (2009-06-01). “Compromising Electromagnetic Emanations of Wired and Wireless Keyboards”. Lausanne: Security and Cryptography Laboratory (LASEC).
- [18] “ATM camera”. snopes.com. Retrieved 2009-04-19.
- [19] 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.
- [20] 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.
- [21] “iPhone Accelerometer Could Spy on Computer Keystrokes”. Wired. 19 October 2011. Retrieved August 25, 2014.

- [22] 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.
- [23] 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.
- [24] 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.
- [25] 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.
- [26] 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.
- [27] “The Security Digest Archives”. Retrieved 2009-11-22.
- [28] “Soviet Spies Bugged World’s First Electronic Typewriters”. *qccglobal.com*.
- [29] Geoffrey Ingersoll. “Russia Turns To Typewriters To Protect Against Cyber Espionage”. 2013.
- [30] Sharon A. Maneki. “Learning from the Enemy: The GUNMAN Project”. 2012.
- [31] Agence France-Presse, Associated Press. “Wanted: 20 electric typewriters for Russia to avoid leaks”. *inquirer.net*.
- [32] Anna Arutunyan. “Russian security agency to buy typewriters to avoid surveillance”.
- [33] 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.
- [34] 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.
- [35] 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.
- [36] John Leyden (2002-08-16). “Russians accuse FBI Agent of Hacking”. The Register.
- [37] Austin Modine (2008-10-10). “Organized crime tampers with European card swipe devices”. The Register. Retrieved 2009-04-18.
- [38] Scott Dunn (2009-09-10). “Prevent keyloggers from grabbing your passwords”. Windows Secrets. Retrieved 2014-05-10.
- [39] Christopher Ciabarra (2009-06-10). “Anti Keylogger”. Networkintercept.com.
- [40] 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.

3.13.7 External links

- Keyloggers at DMOZ

Chapter 4

Defenses

4.1 Access control



A sailor allows a driver to enter a military base.

In the fields of **physical security** and **information security**, **access control** is the selective restriction of access to a place or other resource.^[1] The act of *accessing* may mean consuming, entering, or using. Permission to access a resource is called *authorization*.

Locks and login credentials are two analogous mechanisms of access control.

4.1.1 Physical security

Main article: Physical security

Geographical access control may be enforced by personnel (e.g., **border guard**, bouncer, **ticket checker**), or with a device such as a **turnstile**. There may be **fences** to avoid circumventing this access control. An alternative of access control in the strict sense (physically controlling access itself) is a system of checking authorized presence, see e.g. **Ticket controller** (transportation). A variant is **exit control**, e.g. of a shop (checkout) or a country.

The term **access control** refers to the practice of restricting entrance to a property, a building, or a room to authorized persons. Physical access control can be achieved by a human (a guard, bouncer, or receptionist), through mechanical means such as locks and keys, or through technological means such as access control systems like the **mantrap**. Within these environments, phys-



Space Saver Drop Arm Optical Turnstiles Manufactured by Fabtronusa



Underground entrance to the New York City Subway system

ical key management may also be employed as a means of further managing and monitoring access to mechanically keyed areas or access to certain small assets.

Physical access control is a matter of who, where, and when. An access control system determines who is allowed to enter or exit, where they are allowed to exit or enter, and when they are allowed to enter or exit. Historically, this was partially accomplished through keys and locks. When a door is locked, only someone with a key can enter through the door, depending on how the lock is configured. Mechanical locks and keys do not allow



Physical security access control with a hand geometry scanner



Example of fob based access control using an ACT reader

restriction of the key holder to specific times or dates. Mechanical locks and keys do not provide records of the key used on any specific door, and the keys can be easily copied or transferred to an unauthorized person. When a mechanical key is lost or the key holder is no longer authorized to use the protected area, the locks must be re-keyed.

Electronic access control uses computers to solve the limitations of mechanical locks and keys. A wide range of credentials can be used to replace mechanical keys. The electronic access control system grants access based on the credential presented. When access is granted, the door is unlocked for a predetermined time and the transaction is recorded. When access is refused, the door remains locked and the attempted access is recorded. The system will also monitor the door and alarm if the door is forced open or held open too long after being unlocked.

Access control system operation

When a credential is presented to a reader, the reader sends the credential's information, usually a number, to a control panel, a highly reliable processor. The control panel compares the credential's number to an access control list, grants or denies the presented request, and sends a transaction log to a database. When access is denied based on the access control list, the door remains locked. If there is a match between the credential and the access control list, the control panel operates a relay that in turn unlocks the door. The control panel also ignores a door open signal to prevent an alarm. Often the reader provides feedback, such as a flashing red LED for an access denied and a flashing green LED for an access granted.

The above description illustrates a single factor transaction. Credentials can be passed around, thus subverting the access control list. For example, Alice has access rights to the server room, but Bob does not. Alice either gives Bob her credential, or Bob takes it; he now has access to the server room. To prevent this, **two-factor authentication** can be used. In a two factor transaction, the presented credential and a second factor are needed for access to be granted; another factor can be a PIN, a second credential, operator intervention, or a biometric input.

There are three types (factors) of authenticating information:^[2]

- something the user knows, e.g. a password, passphrase or PIN
- something the user has, such as smart card or a key fob
- something the user is, such as fingerprint, verified by biometric measurement

Passwords are a common means of verifying a user's identity before access is given to information systems. In addition, a fourth factor of authentication is now recognized: someone you know, whereby another person who knows you can provide a human element of authentication in situations where systems have been set up to allow for such scenarios. For example, a user may have their password, but have forgotten their smart card. In such a

scenario, if the user is known to designated cohorts, the cohorts may provide their smart card and password, in combination with the extant factor of the user in question, and thus provide two factors for the user with the missing credential, giving three factors overall to allow access.

Credential

A credential is a physical/tangible object, a piece of knowledge, or a facet of a person's physical being, that enables an individual access to a given physical facility or computer-based information system. Typically, credentials can be something a person knows (such as a number or PIN), something they have (such as an access badge), something they are (such as a biometric feature) or some combination of these items. This is known as **multi-factor authentication**. The typical credential is an access card or key-fob, and newer software can also turn users' smartphones into access devices.^[3]

There are many card technologies including magnetic stripe, bar code, Wiegand, 125 kHz proximity, 26-bit card-swipe, contact smart cards, and contactless smart cards. Also available are key-fobs, which are more compact than ID cards, and attach to a key ring. Biometric technologies include fingerprint, facial recognition, iris recognition, retinal scan, voice, and hand geometry.^[4] The built-in biometric technologies found on newer smartphones can also be used as credentials in conjunction with access software running on mobile devices.^[5] In addition to older more traditional card access technologies, newer technologies such as **Near field communication (NFC)** and **Bluetooth low energy** also have potential to communicate user credentials to readers for system or building access.^{[6][7][8]}

Access control system components

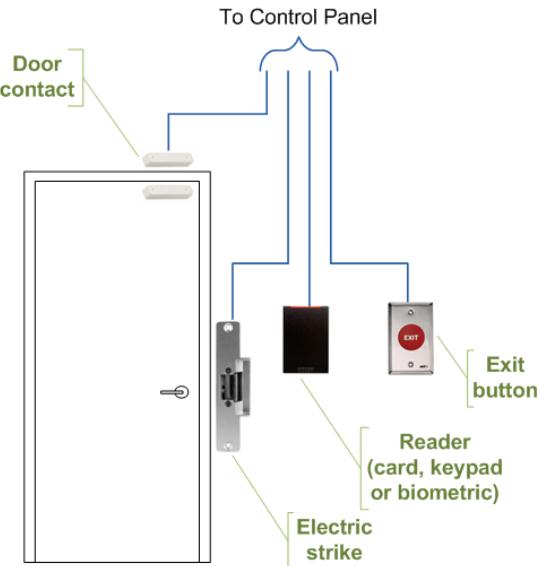


control system components can be found in major cities such as New York City

An access control point, which can be a door, turnstile, parking gate, elevator, or other physical barrier, where granting access can be electronically controlled. Typically, the access point is a door. An electronic access

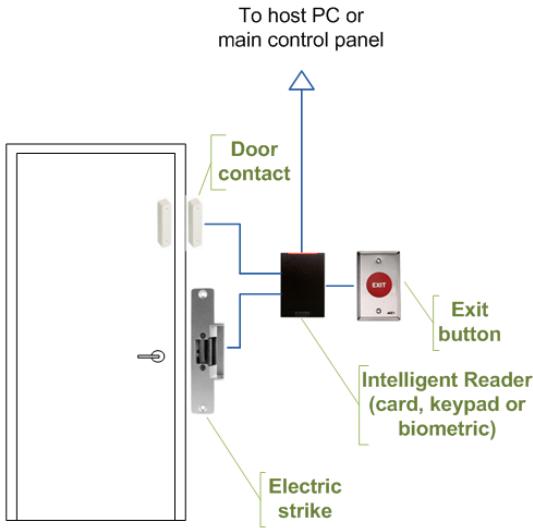
control door can contain several elements. At its most basic, there is a stand-alone electric lock. The lock is unlocked by an operator with a switch. To automate this, operator intervention is replaced by a reader. The reader could be a keypad where a code is entered, it could be a **card reader**, or it could be a biometric reader. Readers do not usually make an access decision, but send a card number to an access control panel that verifies the number against an access list. To monitor the door position a magnetic door switch can be used. In concept, the door switch is not unlike those on refrigerators or car doors. Generally only entry is controlled, and exit is uncontrolled. In cases where exit is also controlled, a second reader is used on the opposite side of the door. In cases where exit is not controlled, free exit, a device called a request-to-exit (REX) is used. Request-to-exit devices can be a push-button or a motion detector. When the button is pushed, or the motion detector detects motion at the door, the door alarm is temporarily ignored while the door is opened. Exiting a door without having to electrically unlock the door is called mechanical free egress. This is an important safety feature. In cases where the lock must be electrically unlocked on exit, the request-to-exit device also unlocks the door.

Access control topology



Typical access control door wiring

Access control decisions are made by comparing the credential to an access control list. This look-up can be done by a host or server, by an access control panel, or by a reader. The development of access control systems has seen a steady push of the look-up out from a central host to the edge of the system, or the reader. The predominant topology circa 2009 is hub and spoke with a control panel as the hub, and the readers as the spokes. The look-up and control functions are by the control panel. The



Access control door wiring when using intelligent readers

spokes communicate through a serial connection; usually RS-485. Some manufacturers are pushing the decision making to the edge by placing a controller at the door. The controllers are IP enabled, and connect to a host and database using standard networks.

Types of readers

Access control readers may be classified by the functions they are able to perform:

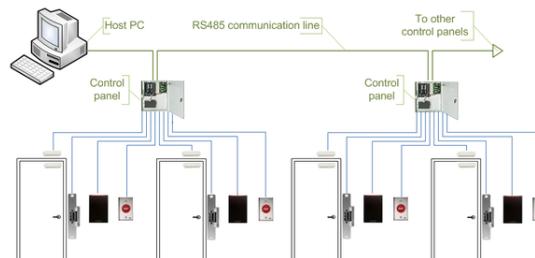
- Basic (non-intelligent) readers: simply read card number or PIN, and forward it to a control panel. In case of biometric identification, such readers output the ID number of a user. Typically, **Wiegand protocol** is used for transmitting data to the control panel, but other options such as RS-232, RS-485 and Clock/Data are not uncommon. This is the most popular type of access control readers. Examples of such readers are RF Tiny by RFLOGICS, ProxPoint by HID, and P300 by Farpointe Data.
- Semi-intelligent readers: have all inputs and outputs necessary to control door hardware (lock, door contact, exit button), but do not make any access decisions. When a user presents a card or enters a PIN, the reader sends information to the main controller, and waits for its response. If the connection to the main controller is interrupted, such readers stop working, or function in a degraded mode. Usually semi-intelligent readers are connected to a control panel via an RS-485 bus. Examples of such readers are InfoProx Lite IPL200 by CEM Systems, and AP-510 by Apollo.
- Intelligent readers: have all inputs and outputs necessary to control door hardware; they also

have memory and processing power necessary to make access decisions independently. Like semi-intelligent readers, they are connected to a control panel via an RS-485 bus. The control panel sends configuration updates, and retrieves events from the readers. Examples of such readers could be InfoProx IPO200 by CEM Systems, and AP-500 by Apollo. There is also a new generation of intelligent readers referred to as "IP readers". Systems with IP readers usually do not have traditional control panels, and readers communicate directly to a PC that acts as a host. Examples of such readers are Foxtech FX-50UX, FX-632 Fingerprint Reader/Controller Access Control System PowerNet IP Reader by Isonas Security Systems,^[9] ID 11 by Solus (has a built in webservice to make it user friendly), Edge ER40 reader by HID Global, LogLock and UNiLOCK by ASPiSYS Ltd, BioEntry Plus reader by Suprema Inc., and 4G V-Station by Bioscrypt Inc.

Some readers may have additional features such as an LCD and function buttons for data collection purposes (i.e. clock-in/clock-out events for attendance reports), camera/speaker/microphone for intercom, and smart card read/write support.

Access control readers may also be classified by their type of identification technology.

Access control system topologies



Access control system using serial controllers

1. Serial controllers. Controllers are connected to a host PC via a serial RS-485 communication line (or via 20mA current loop in some older systems). External RS-232/485 converters or internal RS-485 cards have to be installed, as standard PCs do not have RS-485 communication ports.

Advantages:

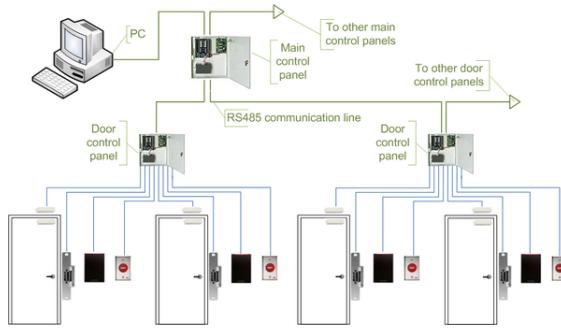
- RS-485 standard allows long cable runs, up to 4000 feet (1200 m)
- Relatively short response time. The maximum number of devices on an RS-485 line is limited to 32,

which means that the host can frequently request status updates from each device, and display events almost in real time.

- High reliability and security as the communication line is not shared with any other systems.

Disadvantages:

- RS-485 does not allow Star-type wiring unless splitters are used
- RS-485 is not well suited for transferring large amounts of data (i.e. configuration and users). The highest possible throughput is 115.2 kbit/sec, but in most system it is downgraded to 56.2 kbit/sec, or less, to increase reliability.
- RS-485 does not allow the host PC to communicate with several controllers connected to the same port simultaneously. Therefore, in large systems, transfers of configuration, and users to controllers may take a very long time, interfering with normal operations.
- Controllers cannot initiate communication in case of an alarm. The host PC acts as a master on the RS-485 communication line, and controllers have to wait until they are polled.
- Special serial switches are required, in order to build a redundant host PC setup.
- Separate RS-485 lines have to be installed, instead of using an already existing network infrastructure.
- Cable that meets RS-485 standards is significantly more expensive than regular Category 5 UTP network cable.
- Operation of the system is highly dependent on the host PC. In the case that the host PC fails, events from controllers are not retrieved, and functions that require interaction between controllers (i.e. anti-passback) stop working.



Access control system using serial main and sub-controllers

2. Serial main and sub-controllers. All door hardware is connected directly to sub-controllers (a.k.a. door controllers

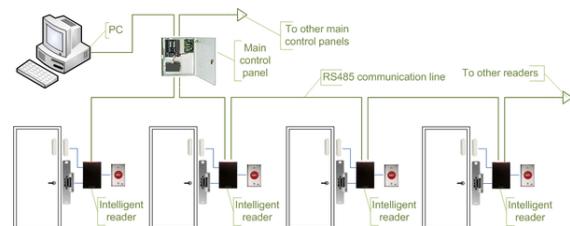
or door interfaces). Sub-controllers usually do not make access decisions, and instead forward all requests to the main controllers. Main controllers usually support from 16 to 32 sub-controllers.

Advantages:

- Work load on the host PC is significantly reduced, because it only needs to communicate with a few main controllers.
- The overall cost of the system is lower, as sub-controllers are usually simple and inexpensive devices.
- All other advantages listed in the first paragraph apply.

Disadvantages:

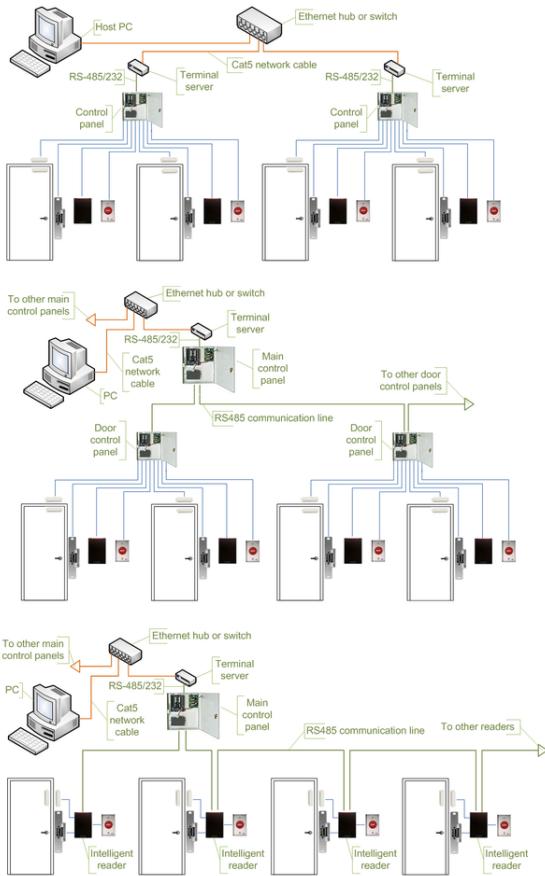
- Operation of the system is highly dependent on main controllers. In case one of the main controllers fails, events from its sub-controllers are not retrieved, and functions that require interaction between sub-controllers (i.e. anti-passback) stop working.
- Some models of sub-controllers (usually lower cost) have not the memory or processing power to make access decisions independently. If the main controller fails, sub-controllers change to degraded mode in which doors are either completely locked or unlocked, and no events are recorded. Such sub-controllers should be avoided, or used only in areas that do not require high security.
- Main controllers tend to be expensive, therefore such a topology is not very well suited for systems with multiple remote locations that have only a few doors.
- All other RS-485-related disadvantages listed in the first paragraph apply.



Access control system using serial main controller and intelligent readers

3. Serial main controllers & intelligent readers. All door hardware is connected directly to intelligent or semi-intelligent readers. Readers usually do not make access decisions, and forward all requests to the main controller. Only if the connection to the main controller is unavailable, will the readers use their internal database to make

access decisions and record events. Semi-intelligent reader that have no database and cannot function without the main controller should be used only in areas that do not require high security. Main controllers usually support from 16 to 64 readers. All advantages and disadvantages are the same as the ones listed in the second paragraph.



Access control systems using serial controllers and terminal servers

4. Serial controllers with terminal servers. In spite of the rapid development and increasing use of computer networks, access control manufacturers remained conservative, and did not rush to introduce network-enabled products. When pressed for solutions with network connectivity, many chose the option requiring less efforts: addition of a **terminal server**, a device that converts serial data for transmission via LAN or WAN.

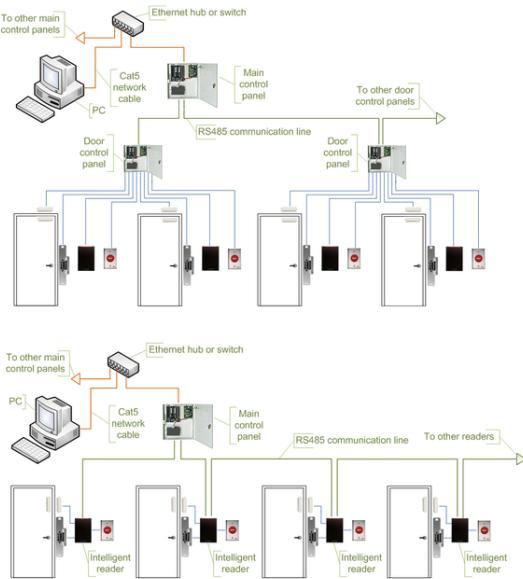
Advantages:

- Allows utilizing the existing network infrastructure for connecting separate segments of the system.
- Provides a convenient solution in cases when the installation of an RS-485 line would be difficult or impossible.

Disadvantages:

- Increases complexity of the system.
- Creates additional work for installers: usually terminal servers have to be configured independently, and not through the interface of the access control software.
- Serial communication link between the controller and the terminal server acts as a bottleneck: even though the data between the host PC and the terminal server travels at the 10/100/1000Mbit/sec network speed, it must slow down to the serial speed of 112.5 kbit/sec or less. There are also additional delays introduced in the process of conversion between serial and network data.

All the RS-485-related advantages and disadvantages also apply.

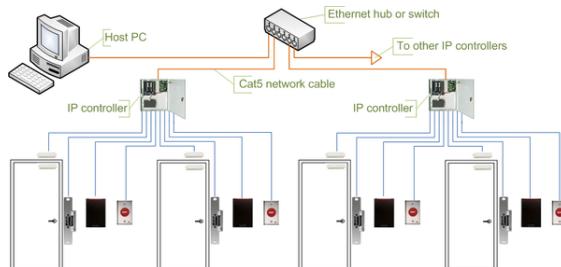


Access control system using network-enabled main controllers

5. Network-enabled main controllers. The topology is nearly the same as described in the second and third paragraphs. The same advantages and disadvantages apply, but the on-board network interface offers a couple of valuable improvements. Transmission of configuration and user data to the main controllers is faster, and may be done in parallel. This makes the system more responsive, and does not interrupt normal operations. No special hardware is required in order to achieve redundant host PC setup: in the case that the primary host PC fails, the secondary host PC may start polling network controllers. The disadvantages introduced by terminal servers (listed in the fourth paragraph) are also eliminated.

6. IP controllers. Controllers are connected to a host PC via Ethernet LAN or WAN.

Advantages:



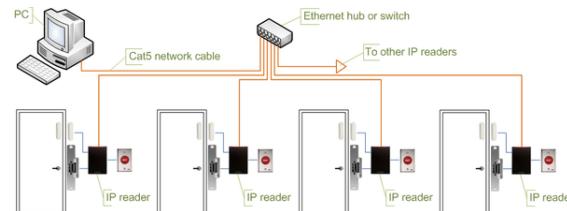
Access control system using IP controllers

- An existing network infrastructure is fully utilized, and there is no need to install new communication lines.
- There are no limitations regarding the number of controllers (as the 32 per line in cases of RS-485).
- Special RS-485 installation, termination, grounding and troubleshooting knowledge is not required.
- Communication with the controllers may be done at the full network speed, which is important if transferring a lot of data (databases with thousands of users, possibly including biometric records).
- In case of an alarm, controllers may initiate connection to the host PC. This ability is important in large systems, because it serves to reduce network traffic caused by unnecessary polling.
- Simplifies installation of systems consisting of multiple sites that are separated by large distances. A basic Internet link is sufficient to establish connections to the remote locations.
- Wide selection of standard network equipment is available to provide connectivity in various situations (fiber, wireless, VPN, dual path, PoE)

Disadvantages:

- The system becomes susceptible to network related problems, such as delays in case of heavy traffic and network equipment failures.
- Access controllers and workstations may become accessible to hackers if the network of the organization is not well protected. This threat may be eliminated by physically separating the access control network from the network of the organization. Also it should be noted that most IP controllers utilize either Linux platform or proprietary operating systems, which makes them more difficult to hack. Industry standard data encryption is also used.
- Maximum distance from a hub or a switch to the controller (if using a copper cable) is 100 meters (330 ft).

- Operation of the system is dependent on the host PC. In case the host PC fails, events from controllers are not retrieved and functions that require interaction between controllers (i.e. anti-passback) stop working. Some controllers, however, have a peer-to-peer communication option in order to reduce dependency on the host PC.



Access control system using IP readers

7. IP readers.

Readers are connected to a host PC via Ethernet LAN or WAN.

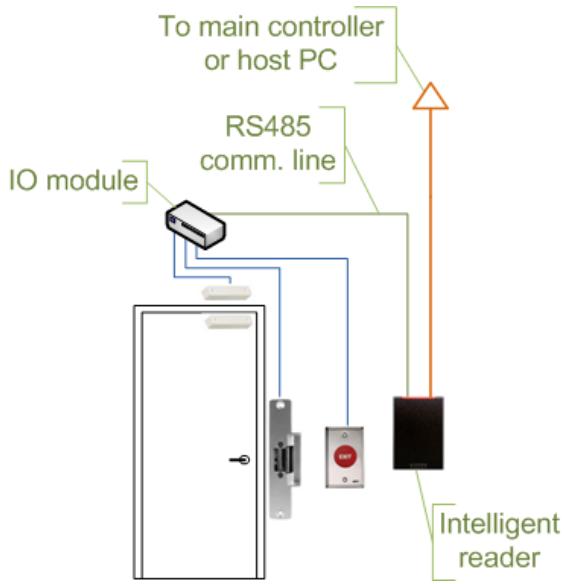
Advantages:

- Most IP readers are PoE capable. This feature makes it very easy to provide battery backed power to the entire system, including the locks and various types of detectors (if used).
- IP readers eliminate the need for controller enclosures.
- There is no wasted capacity when using IP readers (e.g. a 4-door controller would have 25% of unused capacity if it was controlling only 3 doors).
- IP reader systems scale easily: there is no need to install new main or sub-controllers.
- Failure of one IP reader does not affect any other readers in the system.

Disadvantages:

- In order to be used in high-security areas, IP readers require special input/output modules to eliminate the possibility of intrusion by accessing lock and/or exit button wiring. Not all IP reader manufacturers have such modules available.
- Being more sophisticated than basic readers, IP readers are also more expensive and sensitive, therefore they should not be installed outdoors in areas with harsh weather conditions, or high probability of vandalism, unless specifically designed for exterior installation. A few manufacturers make such models.

The advantages and disadvantages of IP controllers apply to the IP readers as well.



Access control door wiring when using intelligent readers and IO module

Security risks

The most common security risk of intrusion through an access control system is by simply following a legitimate user through a door, and this is referred to as “tailgating”. Often the legitimate user will hold the door for the intruder. This risk can be minimized through security awareness training of the user population, or more active means such as turnstiles. In very high security applications this risk is minimized by using a sally port, sometimes called a security vestibule or mantrap, where operator intervention is required presumably to assure valid identification.

The second most common risk is from levering a door open. This is relatively difficult on properly secured doors with strikes or high holding force maglocks. The lever could be as small as a screwdriver or big as a crow bar. Fully implemented access control systems include forced door monitoring alarms. These vary in effectiveness, usually failing from high false positive alarms, poor database configuration, or lack of active intrusion monitoring.

Similar to levering is crashing through cheap partition walls. In shared tenant spaces the divisional wall is a vulnerability. A vulnerability along the same lines is the breaking of sidelights.

Spoofing locking hardware is fairly simple and more elegant than levering. A strong magnet can operate the solenoid controlling bolts in electric locking hardware. Motor locks, more prevalent in Europe than in the US, are also susceptible to this attack using a doughnut shaped magnet. It is also possible to manipulate the power to the lock either by removing or adding current, although most Access Control systems incorporate battery back-up systems and the locks are almost always located on the secure

side of the door.

Access cards themselves have proven vulnerable to sophisticated attacks. Enterprising hackers have built portable readers that capture the card number from a user's proximity card. The hacker simply walks by the user, reads the card, and then presents the number to a reader securing the door. This is possible because card numbers are sent in the clear, no encryption being used. To counter this, dual authentication methods, such as a card plus a PIN, should always be used.

Many access control credentials unique serial numbers are programmed in sequential order during manufacturing. Known as a sequential attack, if an intruder has a credential once used in the system they can simply increment or decrement the serial number until they find a credential that is currently authorized in the system. Ordering credentials with random unique serial numbers is recommended to counter this threat.^[10]

Finally, most electric locking hardware still have mechanical keys as a fail-over. Mechanical key locks are vulnerable to bumping.

The need-to-know principle The need to know principle can be enforced with user access controls and authorization procedures and its objective is to ensure that only authorized individuals gain access to information or systems necessary to undertake their duties. See Principle of least privilege.

4.1.2 Computer security

Further information: [Computer access control](#)

In computer security, general access control includes authorization, authentication, access approval, and audit. A more narrow definition of access control would cover only access approval, whereby the system makes a decision to grant or reject an access request from an already authenticated subject, based on what the subject is authorized to access. Authentication and access control are often combined into a single operation, so that access is approved based on successful authentication, or based on an anonymous access token. Authentication methods and tokens include **passwords**, biometric scans, physical keys, electronic keys and devices, hidden paths, social barriers, and monitoring by humans and automated systems.

In any access-control model, the entities that can perform actions on the system are called *subjects*, and the entities representing resources to which access may need to be controlled are called *objects* (see also [Access Control Matrix](#)). Subjects and objects should both be considered as software entities, rather than as human users: any human users can only have an effect on the system via the software entities that they control.

Although some systems equate subjects with *user IDs*, so that all processes started by a user by default have the same authority, this level of control is not fine-grained enough to satisfy the principle of least privilege, and arguably is responsible for the prevalence of malware in such systems (see computer insecurity).

In some models, for example the object-capability model, any software entity can potentially act as both subject and object.

As of 2014, access-control models tend to fall into one of two classes: those based on capabilities and those based on access control lists (ACLs).

- In a capability-based model, holding an unforgettable reference or *capability* to an object provides access to the object (roughly analogous to how possession of one's house key grants one access to one's house); access is conveyed to another party by transmitting such a capability over a secure channel
- In an ACL-based model, a subject's access to an object depends on whether its identity appears on a list associated with the object (roughly analogous to how a bouncer at a private party would check an ID to see if a name appears on the guest list); access is conveyed by editing the list. (Different ACL systems have a variety of different conventions regarding who or what is responsible for editing the list and how it is edited.)

Both capability-based and ACL-based models have mechanisms to allow access rights to be granted to all members of a *group* of subjects (often the group is itself modeled as a subject).

Access control systems provide the essential services of *authorization, identification and authentication (I&A), access approval, and accountability* where:

- authorization specifies what a subject can do
- identification and authentication ensure that only legitimate subjects can log on to a system
- access approval grants access during operations, by association of users with the resources that they are allowed to access, based on the authorization policy
- accountability identifies what a subject (or all subjects associated with a user) did

4.1.3 Access Control

Access to accounts can be enforced through many types of controls.^[11]

1. Mandatory Access Control (MAC)

In MAC, users do not have much freedom to determine who has access to their files. For example,

security clearance of users and classification of data (as confidential, secret or top secret) are used as security labels to define the level of trust.

2. Discretionary Access Control (DAC)

In DAC, the data owner determines who can access specific resources. For example, a system administrator may create a hierarchy of files to be accessed based on certain permissions.

3. Role-Based Access Control (RBAC)

RBAC allows access based on the job title. For example, a human resources specialist should not have permissions to create network accounts; this should be a role reserved for network administrators.

4. Rule-Based Access Control

An example of this would be only allowing students to use the labs during a certain time of the day.

5. Identity-Based Access Control

Using this network administrators can more effectively manage activity and access based on individual needs.^[12]

6. Organization-Based Access control (OrBAC)

OrBAC model allows the policy designer to define a security policy independently of the implementation^[13]

7. Responsibility Based Access control

Information is accessed based on the responsibilities assigned to an actor or a business role^[14]

8. Attribute-based Access Control

A new access control paradigm whereby access rights are granted to users through the use of policies which combine attributes together

4.1.4 Telecommunication

In telecommunication, the term *access control* is defined in U.S. Federal Standard 1037C^[15] with the following meanings:

1. A service feature or technique used to permit or deny use of the components of a communication system.
2. A technique used to define or restrict the rights of individuals or application programs to obtain data from, or place data onto, a storage device.
3. The definition or restriction of the rights of individuals or application programs to obtain data from, or place data into, a storage device.
4. The process of limiting access to the resources of an AIS (Automated Information System) to authorized users, programs, processes, or other systems.

5. That function performed by the resource controller that allocates system resources to satisfy user requests.

This definition depends on several other technical terms from Federal Standard 1037C.

4.1.5 Public policy

In public policy, access control to restrict access to systems ("authorization") or to track or monitor behavior within systems ("accountability") is an implementation feature of using trusted systems for security or social control.

4.1.6 See also

- Security, Security engineering, Security lighting, Security Management, Security policy
- Alarm devices, Alarm management, Burglar alarm
- Door security, Lock picking, Lock smithing, Electronic lock, Safe, Safe-cracking, Bank vault
- Card reader, Common Access Card, Magnetic stripe card, Proximity card, Smart card, Swipe card, Optical turnstile, Access badge
- Identity management, ID Card, OpenID, IP Controller, IP reader
- Key management, Key cards
- Computer security, Logical security, Htaccess, Wiegand effect, XACML, Credential, Dual Unit
- Fingerprint scanner, Photo identification, Biometrics
- Physical Security Information Management - PSIM
- Physical Security Professional
- Prison, Razor wire, Mantrap
- Castle, Fortification

4.1.7 References

- [1] RFC 4949
- [2] Federal Financial Institutions Examination Council (2008). "Authentication in an Internet Banking Environment" (PDF). Retrieved 2009-12-31.
- [3] "MicroStrategy's office of the future includes mobile identity and cybersecurity". Washington Post. 2014-04-14. Retrieved 2014-03-30.
- [4] biometric access control technology overview

- [5] "iPhone 5S: A Biometrics Turning Point?". BankInfoSecurity.com. 2013-09-16. Retrieved 2014-03-30.
- [6] "NFC access control: cool and coming, but not close". Security Systems News. 2013-09-25. Retrieved 2014-03-30.
- [7] "Ditch Those Tacky Key Chains: Easy Access with EC Key". Wireless Design and Development. 2012-06-11. Retrieved 2014-03-31.
- [8] "Kisi And KeyMe, Two Smart Phone Apps, Might Make House Keys Obsolete". *The Huffington Post*. The Huffington Post. Retrieved 2 September 2015.
- [9] isonas.com
- [10] [http://www.clonemykey.com/blog/
smart-access-control-polices-for-residential-commercial-buildings/](http://www.clonemykey.com/blog/smart-access-control-polices-for-residential-commercial-buildings/)
- [11] [http://www.evollution.com/media_resources/
cybersecurity-access-control/](http://www.evollution.com/media_resources/cybersecurity-access-control/)
- [12] [http://patft.uspto.gov/netacgi/nph-Parser?Sect1=
PTO2&Sect2=HITOFF&p=1&u=%2Fnetacgi%
2FPTO%2Fsearch-bool.html&r=1&f=G&l=50&co1=
AND&d=PTXT&s1=8,984,620.PN.&OS=PN/8,984,
620&RS=PN/8,984,620](http://patft.uspto.gov/netacgi/nph-Parser?Sect1=PTO2&Sect2=HITOFF&p=1&u=%2Fnetacgi%2FPTO%2Fsearch-bool.html&r=1&f=G&l=50&co1=AND&d=PTXT&s1=8,984,620.PN.&OS=PN/8,984,620&RS=PN/8,984,620)
- [13] <http://orbac.org>
- [14] http://ceur-ws.org/Vol-599/BUISTAL2010_Paper5.pdf
- [15] <http://www.its.blrdoc.gov/fs-1037/other/a.pdf>
- U.S. Federal Standard 1037C
- U.S. MIL-STD-188
- U.S. National Information Systems Security Glossary
- Harris, Shon, All-in-one CISSP Exam Guide, 6th Edition, McGraw Hill Osborne, Emeryville, California, 2012.
- "Integrated Security Systems Design" - Butterworth/Heinemann - 2007 - Thomas L. Norman, CPP/PSP/CSC Author
- Government Open Source Access Control—Next Generation (GOSAC-N)
- NIST.gov - Computer Security Division - Computer Security Resource Center - ATTRIBUTE BASED ACCESS CONTROL (ABAC) - OVERVIEW

4.1.8 External links

- eXtensible Access Control Markup Language. An OASIS standard language/model for access control. Also XACML.

4.2 Application security

Application security encompasses measures taken throughout the code's life-cycle to prevent gaps in the security policy of an application or the underlying system (vulnerabilities) through flaws in the design, development, deployment, upgrade, or maintenance of the application.

Applications only control the kind of resources granted to them, and not *which* resources are granted to them. They, in turn, determine the use of these resources by users of the application through application security.

4.2.1 Methodology

According to the patterns & practices *Improving Web Application Security* book, a principle-based approach for application security includes:^[1]

- Knowing the threats.
- Securing the network, host and application..
- Incorporating security into your software development process

Note that this approach is technology / platform independent. It is focused on principles, patterns, and practices.

4.2.2 Threats, attacks, vulnerabilities, and countermeasures

According to the patterns & practices *Improving Web Application Security* book, the following terms are relevant to application security:^[1]

- **Asset.** A resource of value such as the data in a database or on the file system, or a system resource.
- **Threat.** Anything that can exploit a vulnerability and obtain, damage, or destroy an asset.
- **Vulnerability.** A weakness or gap in security program that can be exploited by threats to gain unauthorized access to an asset.
- **Attack** (or exploit). An action taken to harm an asset.
- **Countermeasure.** A safeguard that addresses a threat and mitigates risk.

4.2.3 Application threats / attacks

According to the patterns & practices *Improving Web Application Security* book, the following are classes of common application security threats / attacks:^[1]

4.2.4 Mobile application security

Main article: [Mobile security](#)

The proportion of mobile devices providing open platform functionality is expected to continue to increase in future. The openness of these platforms offers significant opportunities to all parts of the mobile eco-system by delivering the ability for flexible program and service delivery options that may be installed, removed or refreshed multiple times in line with the user's needs and requirements. However, with openness comes responsibility and unrestricted access to mobile resources and APIs by applications of unknown or untrusted origin could result in damage to the user, the device, the network or all of these, if not managed by suitable security architectures and network precautions. Application security is provided in some form on most open OS mobile devices (Symbian OS,^[2] Microsoft, BREW, etc.). Industry groups have also created recommendations including the GSM Association and Open Mobile Terminal Platform (OMTP).^[3]

There are several strategies to enhance mobile application security including

- Application white listing
- Ensuring transport layer security
- Strong authentication and authorization
- Encryption of data when written to memory
- Sandboxing of applications
- Granting application access on a per-API level
- Processes tied to a user ID
- Predefined interactions between the mobile application and the OS
- Requiring user input for privileged/elevated access
- Proper session handling

4.2.5 Security testing for applications

Security testing techniques scour for vulnerabilities or security holes in applications. These vulnerabilities leave applications open to **exploitation**. Ideally, security testing is implemented throughout the entire **software development life cycle** (SDLC) so that vulnerabilities may be addressed in a timely and thorough manner. Unfortunately, testing is often conducted as an afterthought at the end of the development cycle.

Vulnerability scanners, and more specifically web application scanners, otherwise known as penetration testing tools (i.e. ethical hacking tools) have been historically

used by security organizations within corporations and security consultants to automate the security testing of http request/responses; however, this is not a substitute for the need for actual source code review. Physical code reviews of an application's source code can be accomplished manually or in an automated fashion. Given the common size of individual programs (often 500,000 lines of code or more), the human brain can not execute a comprehensive data flow analysis needed in order to completely check all circuitous paths of an application program to find vulnerability points. The human brain is suited more for filtering, interrupting and reporting the outputs of automated source code analysis tools available commercially versus trying to trace every possible path through a compiled code base to find the root cause level vulnerabilities.

The two types of automated tools associated with application vulnerability detection (application vulnerability scanners) are Penetration Testing Tools (often categorized as **Black Box Testing Tools**) and static code analysis tools (often categorized as **White Box Testing Tools**).

According to Gartner Research,^[4] "...next-generation modern Web and mobile applications requires a combination of SAST and DAST techniques, and new interactive application security testing (IAST) approaches have emerged that combine static and dynamic techniques to improve testing...". Because IAST combines SAST and DAST techniques, the results are highly actionable, can be linked to the specific line of code, and can be recorded for replay later for developers.

Banking and large **E-Commerce** corporations have been the very early adopter customer profile for these types of tools. It is commonly held within these firms that both Black Box testing and White Box testing tools are needed in the pursuit of application security. Typically sited, Black Box testing (meaning Penetration Testing tools) are ethical hacking tools used to attack the application surface to expose vulnerabilities suspended within the source code hierarchy. Penetration testing tools are executed on the already deployed application. White Box testing (meaning Source Code Analysis tools) are used by either the application security groups or application development groups. Typically introduced into a company through the application security organization, the White Box tools complement the Black Box testing tools in that they give specific visibility into the specific root vulnerabilities within the source code in advance of the source code being deployed. Vulnerabilities identified with White Box testing and Black Box testing are typically in accordance with the **OWASP** taxonomy for software coding errors. White Box testing vendors have recently introduced dynamic versions of their source code analysis methods; which operates on deployed applications. Given that the White Box testing tools have dynamic versions similar to the Black Box testing tools, both tools can be correlated in the same software error detection paradigm ensuring full application protection to the

client company.

The advances in professional **Malware** targeted at the Internet customers of online organizations has seen a change in Web application design requirements since 2007. It is generally assumed that a sizable percentage of Internet users will be compromised through **malware** and that any data coming from their infected host may be tainted. Therefore, application security has begun to manifest more advanced anti-fraud and heuristic detection systems in the back-office, rather than within the client-side or Web server code.^[5]

4.2.6 Security certifications

There are a number of certifications available for security professionals to demonstrate their knowledge in the subject matter (e.g. **Certified Information Systems Security Professional**, **Certified Information Security Manager**, etc.), however the usefulness of security certifications and certifications in general typically receives mixed reviews by experienced professionals.

4.2.7 Security standards and regulations

- Sarbanes-Oxley Act (SOX)
- Health Insurance Portability and Accountability Act (HIPAA)
- IEEE P1074
- ISO/IEC 7064:2003 *Information technology -- Security techniques -- Check character systems*
- ISO/IEC 9796-2:2002 *Information technology -- Security techniques -- Digital signature schemes giving message recovery -- Part 2: Integer factorization based mechanisms*
- ISO/IEC 9796-3:2006 *Information technology -- Security techniques -- Digital signature schemes giving message recovery -- Part 3: Discrete logarithm based mechanisms*
- ISO/IEC 9797-1:1999 *Information technology -- Security techniques -- Message Authentication Codes (MACs) -- Part 1: Mechanisms using a block cipher*
- ISO/IEC 9797-2:2002 *Information technology -- Security techniques -- Message Authentication Codes (MACs) -- Part 2: Mechanisms using a dedicated hash-function*
- ISO/IEC 9798-1:1997 *Information technology -- Security techniques -- Entity authentication -- Part 1: General*

- ISO/IEC 9798-2:1999 *Information technology -- Security techniques -- Entity authentication -- Part 2: Mechanisms using symmetric encipherment algorithms*
- ISO/IEC 9798-3:1998 *Information technology -- Security techniques -- Entity authentication -- Part 3: Mechanisms using digital signature techniques*
- ISO/IEC 9798-4:1999 *Information technology -- Security techniques -- Entity authentication -- Part 4: Mechanisms using a cryptographic check function*
- ISO/IEC 9798-5:2004 *Information technology -- Security techniques -- Entity authentication -- Part 5: Mechanisms using zero-knowledge techniques*
- ISO/IEC 9798-6:2005 *Information technology -- Security techniques -- Entity authentication -- Part 6: Mechanisms using manual data transfer*
- ISO/IEC 14888-1:1998 *Information technology - - Security techniques -- Digital signatures with appendix -- Part 1: General*
- ISO/IEC 14888-2:1999 *Information technology - - Security techniques -- Digital signatures with appendix -- Part 2: Identity-based mechanisms*
- ISO/IEC 14888-3:2006 *Information technology - - Security techniques -- Digital signatures with appendix -- Part 3: Discrete logarithm based mechanisms*
- ISO/IEC 27001:2005 and ISO/IEC 27001:2013 *Information technology -- Security techniques -- Information security management systems -- Requirements*
- ISO/IEC 27002:2005 *Information technology -- Security techniques -- Code of practice for information security management*
- ISO/IEC 24762:2008 *Information technology -- Security techniques -- Guidelines for information and communications technology disaster recovery services* - **now withdrawn.**
- ISO/IEC 27006:2007 *Information technology -- Security techniques -- Requirements for bodies providing audit and certification of information security management systems*
- ISO/IEC 27031:2011 *Information technology -- Security techniques -- Guidelines for ICT readiness for Business Continuity*
- ISO/IEC 27034-1:2011 *Information technology — Security techniques — Application security -- Part 1: Overview and concepts*
- ISO/IEC TR 24772:2013 *Information technology — Programming languages — Guidance to avoiding vulnerabilities in programming languages through language selection and use*
- Gramm-Leach-Bliley Act
- PCI Data Security Standardized (PCI DSS)

4.2.8 See also

- Countermeasure
- Data security
- Database security
- HERAS-AF
- Information security
- Trustworthy Computing Security Development Lifecycle
- Web application
- Web application framework
- XACML

4.2.9 References

- [1] Improving Web Application Security: Threats and Countermeasures, published by Microsoft Corporation.
- [2] “Platform Security Concepts”, Simon Higginson.
- [3] Application Security Framework, Open Mobile Terminal Platform
- [4] http://www.gartner.com/technology/reprints.do?id=1-1GT3BKT&ct=130702&st=sb&mkt_tok=3RkMMJWWfF9wsRokvazAZKXonjHpfX76%252B4qX6WyIMI%252F0ER3fOvrPUfGji4CTsRml%252BSLDwEYGJlv6SgFTbnFMbprzbgPUhA%253D
- [5] “Continuing Business with Malware Infected Customers”. Gunter Ollmann. October 2008.

4.2.10 External links

- The Web Application Security Consortium
- The Microsoft Security Development Lifecycle (SDL)
- patterns & practices Security Guidance for Applications
- Advantages of an integrated security solution for HTML and XML

- patterns & practices Application Security Methodology
- Understanding the Windows Mobile Security Model, Windows Mobile Security]
- Network Security Testing

4.3 Antivirus software

“Antivirus” redirects here. For the antiviral medication, see [Antiviral drug](#).

Antivirus or **anti-virus** software (often abbreviated



ClamTk, an open source antivirus based on the ClamAV antivirus engine, originally developed by Tomasz Kojm in 2001.

as **AV**), sometimes known as **anti-malware** software, is computer software used to prevent, detect and remove malicious software.

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.^[1] 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), botnets DDoS attacks.^[2]

4.3.1 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*”,^[3] the first known computer virus appeared in 1971 and was dubbed the “Creeper virus”.^[4] This computer virus infected Digital Equipment Corporation’s (DEC) PDP-10 mainframe computers running the TENEX operating system.^{[5][6]}

The Creeper virus was eventually deleted by a program created by Ray Tomlinson and known as “The Reaper”.^[7] 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.^{[7][8][9]}

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.^{[10][11][12]}

In 1983, the term “computer virus” was coined by Fred Cohen in one of the first ever published academic papers on computer viruses.^[13] 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.”^[14] (note that a more recent, and precise, definition of computer virus has been given by the Hungarian security researcher Péter Szőr: “a code that recursively replicates a possibly evolved copy of itself”^{[15][16]})

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.^{[17][18]} 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.

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.^[19]

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.^{[20][21]}

In 1987, Andreas Lüning and Kai Figge founded G Data

Software and released their first antivirus product for the Atari ST platform.^[22] Later in the same year, also the *Ultimate Virus Killer (UVK) 2000* antivirus was released.^[23]

In 1987, in USA, John McAfee founded the McAfee company (now part of Intel Security^[24]) and, at the end of that year, he released the first version of *VirusScan*.^[25] In the meanwhile, in Slovakia, Peter Paško and Miroslav Trnka created the first version of *NOD32* antivirus (albeit they established ESET only in 1992).

In 1987, Fred Cohen wrote that **there is no algorithm that can perfectly detect all possible computer viruses.**^[26]

The first antivirus signatures were simply hashes of the entire files or sequences of bytes that represented the particular malware.

Finally, in the end of 1987, the first two heuristic antivirus utilities were released: *FluShot Plus* by Ross Greenberg^{[27][28][29]} and *Anti4us* by Erwin Lanting.^{[30][31]}

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 an **heuristic engine** which resembles the ones used nowadays was **F-PROT** in 1991.^[32] 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 Safe-ground** 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 *VI* (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**). At the end of the year, in the USA, Ross M. Greenberg

released his second antivirus program, called *VirexPC*.

Also in 1988, a mailing list named **VIRUS-L**^[33] 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).^[33]

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 USA, Symantec (founded by Gary Hendrix in 1982) launched its first *Symantec antivirus for Macintosh* (SAM).^{[34][35]} 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.^[36]

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). 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).^[37] Finally, in the end of the year, **Trend Micro** released its first antivirus software, named *PC-Cillin*.

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.^[38] 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.^{[39][40]}

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.^[41]

In 1991, the European Institute for Computer Antivirus Research (EICAR) was founded to further antivirus research and improve development of antivirus software.^{[42][43]}

In 1992, in Russia, Igor Danilov released the first version of *SpiderWeb*, which later became Dr. Web.^[44]

In 1994, AV-TEST reported that there were 28,613 unique malware samples (based on MD5) in their database.^[45]

Over time other companies were founded. In 1996, in Romania, Bitdefender was founded and released the first version of *Anti-Virus eXpert* (AVX).^[46] In 1997, in Russia, Eugene Kaspersky and Natalia Kaspersky co-founded security firm Kaspersky Lab.^[47]

In 1996, there was also the first “in the wild” Linux virus, known as “Staog”.^[48]

In 1999, AV-TEST reported that there were 98,428 unique malware samples (based on MD5) in their database.^[45]

2000-2005 period

In 2000, Rainer Link and Howard Fuhs started the first open source antivirus engine, called *OpenAntivirus Project*.^[49]

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,^[50] which in turn was acquired by Cisco Systems in 2013.^[51]

In 2002, in United Kingdom, Morten Lund and Theis Søndergaard co-founded the antivirus firm BullGuard.^[52]

In 2005, AV-TEST reported that there were 333,425 unique malware samples (based on MD5) in their database.^[45]

2005 to present

As always-on broadband connections became the norm, and more and more viruses were released, it became essential to update antivirus more and more frequently. Even then, a new zero-day or next-generation malware could become widespread before antivirus firms released an update to protect against it.

In 2007, AV-TEST reported a number of 5,490,960 new unique malware samples (based on MD5) only for that year.^[45] In 2012 and 2013, antivirus firms reported a new malware samples range from 300.000 to over 500.000 per

day.^{[53][54]}

Slowly, in order to catch up with the malware productions, antivirus firms have moved to more and more complex algorithms.

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 detections 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.^[55]
- The possibility of embedding executable objects inside otherwise non-executable file formats can make opening those files a risk.^[56]
- 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.^[57]

In 2005, F-Secure was the first security firm that developed an Anti-Rootkit technology, called *BlackLight*.

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.^[58]

In November 2009, Panda Security unveiled its first Cloud-based antivirus technology, the first commercial CloudAV ever released. A year after, Sophos also added to its host-based antivirus product a Cloud-based one. In the following years, many other antivirus firms have added a CloudAV to their security products (see Comparison of antivirus software for a complete overview).

In 2011, AVG introduced a similar cloud service, called Protective Cloud Technology.^[59]

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.^{[60][61]}

4.3.2 Identification methods

One of the few solid theoretical results in the study of computer viruses is Frederick B. Cohen’s 1987 demon-

stration that there is no algorithm that can perfectly detect all possible viruses.^[26] However, using different layer of defense, a good detection rate may be achieved.

There are several methods which antivirus engine can use to identify malware:

- **Signature-based detection:** is the most common method. To identify viruses and other malware, the antivirus engine compares the contents of a file to its database of known malware signatures.
- **Heuristic-based detection:** is generally used together with signature-based detection. It detects malware based on characteristics typically used in known malware code.
- **Behavioural-based detection:** is similar to heuristic-based detection and used also in **Intrusion Detection System**. The main difference is that, instead of characteristics hardcoded in the malware code itself, it is based on the behavioural fingerprint of the malware at run-time. Clearly, this technique is able to detect (known or unknown) malware only after they have started doing their malicious actions.
- **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.^[62] 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.^[63]
- **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.^{[64][65][66][67][68][69][70][71][72][73][74][75][76][77]}

Signature-based detection

Traditional antivirus software relies heavily upon signatures to identify malware.^[78]

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.^[79] When a particular file has to be

scanned, the antivirus engine compares the contents of the file with all the malware signatures in the signatures database. If the file matches one signature, then the engine knows which malware it is and which procedure has to be performed in order to clean the infection.

Signature-based detection technique can be very effective but, clearly, cannot defend against malware unless some of its samples have already been obtained, a proper signature generated and the antivirus product updated. Signature-based detection systems rely on the premise that, generally speaking, the more infective a malware is the faster arrives in the hands of security researchers. Thus, even if it does not guarantee perfection, it protects from the most widespread threats. However, this approach is not really effective against **zero-day** or **next-generation** malware, i.e. malware that has not been yet encountered/analysed.

As new malware are being created each day, the signature-based detection approach requires frequent updates of the signatures database. To assist the antivirus firms, the software may automatically upload new malware to the company or allow the user to manually do it, allowing the antivirus firms to dramatically shorten the life of those threats. Some antivirus products includes also advanced software to spot **zero-day** or **next-generation** malware.

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.^[80]

Heuristics

Some more sophisticated antivirus software uses **heuristic analysis** to identify new malware or variants of known malware.

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.^[81]

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*.^{[82][83]}

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.^[84] 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.^[85]

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.^[86]

4.3.3 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^[87] while **BitDefender** sends notifications to unsubscribe 30 days before the renewal.^[88] **Norton AntiVirus** also renews subscriptions automatically by default.^[89]

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**.^[90]

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.^[91] Recovering from such damage to critical software infrastructure incurs technical support costs and businesses can be forced to close whilst remedial action is undertaken.^{[92][93]} 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.^[94]

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.^[95] 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.^{[96][97]}

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.^[98]

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**.^[99]

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.^{[100][101]}

System and interoperability related issues

Running (the real-time protection of) multiple antivirus programs concurrently can degrade performance and create conflicts.^[102] However, using a concept called **multiscanning**, several companies (including **G Data**^[103] and **Microsoft**^[104]) have created applications which can run multiple engines concurrently.

It is sometimes necessary to temporarily disable virus protection when installing major updates such as Win-

dows Service Packs or updating graphics card drivers.^[105] 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.^{[106][107][108]}

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.^[109] Anti-virus software can impair the performance and stability of games running in the Steam platform.^[110]

Support issues also exist around antivirus application interoperability with common solutions like SSL VPN remote access and network access control products.^[111] 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.^[112] According to the *Zeus tracker* website the average detection rate for all variants of the well-known Zeus trojan is as low as 40%.^[113]

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.^[114]

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.^[115]

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.^[116]

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.^{[117][118]}

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 anti-virus applications to make sure that they are not detected before releasing them into the wild.^[119]

Some new viruses, particularly ransomware, use polymorphic code to avoid detection by virus scanners. Jerome Segura, a security analyst with ParetoLogic, explained:^[120]

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.^[121]

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^[122] 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.^{[123][124]} In such circumstances, damaged files can only be restored from existing backups or shadow copies (this is also true for ransomware^[125]); installed software that is damaged requires re-installation^[126] (however, see System File Checker).

Firmware issues

Active anti-virus software can interfere with a firmware update process.^[127] Any writeable firmware in the computer can be infected by malicious code.^[128] 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.^[129] Anti-virus software is not effective at protecting firmware and the **motherboard BIOS** from infection.^[130] 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.^{[131][132]}

4.3.4 Performance and other drawbacks

Antivirus software has some drawbacks, first of which that it can impact a computer's performance.^[133]

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).^[134]

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.^[135]

4.3.5 Alternative solutions

```
adam@a:~$ sudo freshclam
ClamAV update process started at Tue Aug 18 08:16:01 2009
main.cld is up to date (version: 51, sigs: 545035, f-level: 42, builder: svn)
daily.cld is up to date (version: 9709, sigs: 64749, f-level: 43, builder: ccrcd)
adam@a:~$ clamscan ~/Virus/setup.exe
/home/adam/Virus/setup.exe: Trojan.Fakeav-104 FOUND

----- SCAN SUMMARY -----
Known Viruses: 609137
Engine version: 0.95.2
Scanned directories: 0
Scanned files: 1
Infected files: 1
Data scanned: 0.02 MB
Data read: 0.02 MB (ratio 1.00:1)
Time: 2.443 sec (0 m 2 s)
adam@a:~$
```

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.^[136]

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.^[137]

Some examples of cloud anti-virus products are **Panda Cloud Antivirus** and **Immunet**.

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.^[138]

```
[ Rootkit Hunter version 1.3.6 ]
Checking rkHunter version...
This version : 1.3.6
Latest version: 1.3.8
Update available

Checking system commands...

Performing 'strings' command checks
  Checking 'strings' command [ OK ]

Performing 'shared libraries' checks
  Checking for preloading variables [ None found ]
  Checking for preloaded libraries [ None found ]
  Checking LD_LIBRARY_PATH variable [ Not found ]

Performing file properties checks
  Checking for prerequisites
    /bin/bash [ OK ]
    /bin/cat [ OK ]
    /bin/chmod [ OK ]
    /bin/chown [ OK ]
    /bin/cp [ OK ]
```

The command-line *rkhunter* scanner, an engine to scan for Linux rootkits. Here running the tool on Ubuntu.

Specialist tools

Virus removal tools are available to help remove stubborn infections or certain types of infection. Examples include Trend Micro's *Rootkit Buster*,^[139] and *rkhunter* for the detection of rootkits, Avira's *AntiVir Removal Tool*,^[140] *PCTools Threat Removal Tool*,^[141] and AVG's Anti-Virus Free 2011.^[142]

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*,^[140] *PCTools Alternate Operating System Scanner*,^[143] and *AVG Rescue CD*.^[144] The AVG Rescue CD software can also be installed onto a USB storage device, that is bootable on newer computers.^[144]

4.3.6 Usage and risks

According to an FBI survey, major businesses lose \$12 million annually dealing with virus incidents.^[145] 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.^[146] According to a sociological survey conducted by G Data Software in 2010 49% of women did not use any antivirus program at all.^[147]

4.3.7 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
- List of trojan horses
- Quarantine technology
- Sandbox (computer security)
- Timeline of notable computer viruses and worms
- Virus hoax

4.3.8 References

- [1] lifehacker: The Difference Between Antivirus and Anti-Malware (and Which to Use)
- [2] "What is antivirus software?". Microsoft.
- [3] John von Neumann: "Theory of self-reproducing automata" (1949)
- [4] Thomas Chen, Jean-Marc Robert (2004). "The Evolution of Viruses and Worms". Retrieved 2009-02-16.
- [5] From the first email to the first YouTube video: a definitive internet history. Tom Meltzer and Sarah Phillips. *The Guardian*. 23 October 2009
- [6] *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."
- [7] John Metcalf (2014). "Core War: Creeper & Reaper". Retrieved 2014-05-01.
- [8] Creeper - The Virus Encyclopedia
- [9] What was the First Antivirus Software?
- [10] "Elk Cloner". Retrieved 2010-12-10.
- [11] "Top 10 Computer Viruses: No. 10 - Elk Cloner". Retrieved 2010-12-10.
- [12] "List of Computer Viruses Developed in 1980s". Retrieved 2010-12-10.
- [13] Fred Cohen: "Computer Viruses – Theory and Experiments" (1983)
- [14] Fred Cohen 1988 "On the implications of Computer Viruses and Methods of Defense"
- [15] Péter Szőr: "The Art of Computer Virus Research and Defense" (2005)

- [16] VirusBulletin: "In memoriam: Péter Ször 1970-2013" (2013)
- [17] History of viruses
- [18] Leyden, John (January 19, 2006). "PC virus celebrates 20th birthday". *The Register*. Retrieved March 21, 2011.
- [19] Panda Security (April 2004). "(II) Evolution of computer viruses". Archived from the original on 2 August 2009. Retrieved 2009-06-20.
- [20] Kaspersky Lab Virus list
- [21] Wells, Joe (1996-08-30). "Virus timeline". IBM. Archived from the original on 4 June 2008. Retrieved 2008-06-06.
- [22] G Data Software AG (2011). "G Data presents security firsts at CeBIT 2010". Retrieved 22 August 2011.
- [23] Karsmakers, Richard (January 2010). "The ultimate Virus Killer UVK 2000". Retrieved 22 August 2011.
- [24] "McAfee Becomes Intel Security". McAfee Inc. Retrieved 15 January 2014.
- [25] Cavendish, Marshall (2007). *Inventors and Inventions, Volume 4*. Paul Bernabeo. p. 1033. ISBN 0761477675.
- [26] Cohen, Fred, An Undetectable Computer Virus (Archived), 1987, IBM
- [27] Patricia A. Yevics:"Flu Shot for Computer Viruses"
- [28] How friends help friends on the Internet: The Ross Greenberg Story
- [29] Anti-virus is 30 years old
- [30] A Brief History of Antivirus Software
- [31] Antivirus software history
- [32] <http://www.frisk.is/fyrirtaeki.html>
- [33] VIRUS-L mailing list archive
- [34] Symantec Softwares and Internet Security at PCM
- [35] SAM Identifies Virus-Infected Files, Repairs Applications, *InfoWorld*, May 22, 1989
- [36] SAM Update Lets Users Program for New Viruses, *InfoWorld*, Feb 19, 1990
- [37] TG Soft History
- [38] Skúlason and Bontchev: "Virus Naming Scheme" (1991)
- [39] "CARO Members". CARO. Retrieved 6 June 2011.
- [40] CAROids, Hamburg 2003
- [41] "F-Secure Weblog : News from the Lab". F-secure.com. Retrieved 2012-09-23.
- [42] "About EICAR". *EICAR official website*. Retrieved 28 October 2013.
- [43] 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.
- [44] "Dr. Web LTD Doctor Web / Dr. Web Reviews, Best AntiVirus Software Reviews, Review Centre". Reviewcentre.com. Retrieved 2014-02-17.
- [45] [In 1994, AV-Test.org reported 28,613 unique malware samples (based on MD5). "A Brief History of Malware; The First 25 Years"]
- [46] "BitDefender Product History".
- [47] "InfoWatch Management". InfoWatch. Retrieved 12 August 2013.
- [48] Linuxvirus
- [49]
- [50] "Sourcefire acquires ClamAV". ClamAV. 2007-09-17. Retrieved 2008-02-12.
- [51] "Cisco Completes Acquisition of Sourcefire". *cisco.com*. 2013-10-07. Retrieved 2014-06-18.
- [52] "(german) Interview with Morten Lund in Brandeins".
- [53] "The digital detective: Mikko Hypponen's war on malware is escalating." (March 2012, Wired)
- [54] James Lyne: "Everyday cybercrime – and what you can do about it" (February 2013, TED)
- [55] Szor 2005, pp. 66–67
- [56] "New virus travels in PDF files". 7 August 2001. Retrieved 2011-10-29.
- [57] Slipstick Systems (February 2009). "Protecting Microsoft Outlook against Viruses". Archived from the original on 2 June 2009. Retrieved 2009-06-18.
- [58] Jon Oberheide: "CloudAV: N-Version Antivirus in the Network Cloud" (2008, Usenix)
- [59] "TECHNOLOGY OVERVIEW". *AVG Security*. Retrieved 16 February 2015.
- [60] NetworkWorld, Ellen Messmer, August 19, 2014:"Startup offers up endpoint detection and response for behavior-based malware detection"
- [61] HSToday.US, Kylie Bull, June 19, 2014:"Bromium Research Reveals Insecurity In Existing Endpoint Malware Protection Deployments"
- [62] "Sandboxing against unknown zero day threats". Retrieved 2015-01-30.
- [63] Szor 2005, pp. 474–481
- [64] A Machine Learning Approach to Anti-virus System
- [65] Data Mining Methods for Malware Detection

- [66] Data mining and Machine Learning in Cybersecurity
- [67] Analysis of Machine learning Techniques Used in Behavior-Based Malware Detection
- [68] A survey of data mining techniques for malware detection using file features
- [69] Intelligent automatic malicious code signatures extraction
- [70] Malware Detection by Data Mining Techniques Based on Positionally Dependent Features
- [71] Data mining methods for detection of new malicious executables
- [72] IMDS: Intelligent Malware Detection System
- [73] Learning to Detect and Classify Malicious Executables in the Wild
- [74] Malware detection using statistical analysis of byte-level file content
- [75] An intelligent PE-malware detection system based on association mining
- [76] Malware detection based on mining API calls
- [77] “Andromaly”: a behavioral malware detection framework for android devices
- [78] Fox-Brewster, Thomas. “Netflix Is Dumping Anti-Virus, Presages Death Of An Industry”. Forbes. Retrieved September 4, 2015.
- [79] Automatic Malware Signature Generation
- [80] Szor 2005, pp. 252–288
- [81] “Generic detection”. Kaspersky. Retrieved 2013-07-11.
- [82] Symantec Corporation (February 2009). “Trojan.Vundo”. Archived from the original on 9 April 2009. Retrieved 2009-04-14.
- [83] Symantec Corporation (February 2007). “Trojan.Vundo.B”. Archived from the original on 27 April 2009. Retrieved 2009-04-14.
- [84] “Antivirus Research and Detection Techniques”. ExtremeTech. Archived from the original on 27 February 2009. Retrieved 2009-02-24.
- [85] Rootkit
- [86] Kaspersky Lab Technical Support Portal Archived 13 February 2011 at WebCite
- [87] Kelly, Michael (October 2006). “Buying Dangerously”. Retrieved 2009-11-29.
- [88] Bitdefender (2009). “Automatic Renewal”. Retrieved 2009-11-29.
- [89] Symantec (2014). “Norton Automatic Renewal Service FAQ”. Retrieved 2014-04-09.
- [90] SpywareWarrior (2007). “Rogue/Suspect Anti-Spyware Products & Web Sites”. Retrieved 2009-11-29.
- [91] Emil Protalinski (November 11, 2008). “AVG incorrectly flags user32.dll in Windows XP SP2/SP3”. Ars Technica. Retrieved 2011-02-24.
- [92] McAfee to compensate businesses for buggy update, retrieved 2 December 2010
- [93] Buggy McAfee update whacks Windows XP PCs, archived from the original on 13 January 2011, retrieved 2 December 2010
- [94] Aaron Tan (May 24, 2007). “Flawed Symantec update cripples Chinese PCs”. CNET Networks. Retrieved 2009-04-05.
- [95] 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.
- [96] “McAfee DAT 5958 Update Issues”. 21 April 2010. Archived from the original on 24 April 2010. Retrieved 22 April 2010.
- [97] “Botched McAfee update shutting down corporate XP machines worldwide”. 21 April 2010. Archived from the original on 22 April 2010. Retrieved 22 April 2010.
- [98] John Leyden (December 2, 2010). “Horror AVG update ballsup bricks Windows 7”. The Register. Retrieved 2010-12-02.
- [99] MSE false positive detection forces Google to update Chrome, retrieved 3 October 2011
- [100] Sophos Antivirus Detects Itself as Malware, Deletes Key Binaries, The Next Web, retrieved 5 March 2014
- [101] Shh/Updater-B false positive by Sophos anti-virus products, Sophos, retrieved 5 March 2014
- [102] Microsoft (January 2007). “Plus! 98: How to Remove McAfee VirusScan”. Archived from the original on 27 September 2014. Retrieved 2014-09-27.
- [103] Robert Vamosi (May 28, 2009). “G-Data Internet Security 2010”. PC World. Retrieved 2011-02-24.
- [104] Kelly Jackson Higgins (May 5, 2010). “New Microsoft Forefront Software Runs Five Antivirus Vendors’ Engines”. Darkreading. Retrieved 2011-02-24.
- [105] 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.
- [106] “Upgrading from Windows Vista to Windows 7”. Retrieved 24 March 2012. Mentioned within “Before you begin”.
- [107] “Upgrading to Microsoft Windows Vista recommended steps.”. Retrieved 24 March 2012.
- [108] “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”.
- [109] “Troubleshooting”. Retrieved 2011-02-17.

- [110] "Spyware, Adware, and Viruses Interfering with Steam". Retrieved 11 April 2013. Steam support page.
- [111] Field Notice: FN - 63204 - Cisco Clean Access has Interoperability issue with Symantec Anti-virus - delays Agent start-up
- [112] Dan Goodin (December 21, 2007). "Anti-virus protection gets worse". *Channel Register*. Retrieved 2011-02-24.
- [113]
- [114] Dan Illett (July 13, 2007). "Hacking poses threats to business". *Computer Weekly*. Retrieved 2009-11-15.
- [115] Tom Espiner (June 30, 2008). "Trend Micro: Antivirus industry lied for 20 years". *ZDNet*. Retrieved 2014-09-27.
- [116] 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.
- [117] Guidelines released for antivirus software tests
- [118] Harley, David (2011). *AVIEN Malware Defense Guide for the Enterprise*. Elsevier. p. 487. ISBN 9780080558660. Retrieved 2013-06-10.
- [119] Kotadia, Munir (July 2006). "Why popular antivirus apps 'do not work'". Retrieved 14 April 2010.
- [120] 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.
- [121] Researchers up evilness ante with GPU-assisted malware - Coming to a PC near you, by Dan Goodin
- [122] GIBSON RESEARCH CORPORATION SERIES: Security Now!
- [123] "Why F-PROT Antivirus fails to disinfect the virus on my computer?". Retrieved 2015-08-20.
- [124] "Actions to be performed on infected objects". Retrieved 2015-08-20.
- [125] "Cryptolocker Ransomware: What You Need To Know". Retrieved 2014-03-28.
- [126] "How Anti-Virus Software Works". Retrieved 2011-02-16.
- [127] "BT Home Hub Firmware Upgrade Procedure". Retrieved 2011-03-06.
- [128] "The 10 faces of computer malware". July 17, 2009. Retrieved 2011-03-06.
- [129] "New BIOS Virus Withstands HDD Wipes". 27 March 2009. Retrieved 2011-03-06.
- [130] "Phrack Inc. Persistent BIOS Infection". June 1, 2009. Archived from the original on 30 April 2011. Retrieved 2011-03-06.
- [131] "Turning USB peripherals into BadUSB". Retrieved 2014-10-11.
- [132] "Why the Security of USB Is Fundamentally Broken". 2014-07-31. Retrieved 2014-10-11.
- [133] "How Antivirus Software Can Slow Down Your Computer". Support.com Blog. Retrieved 2010-07-26.
- [134] "Softpedia Exclusive Interview: Avira 10". *Ionut Ilascu*. Softpedia. 14 April 2010. Retrieved 2011-09-11.
- [135] "Norton AntiVirus ignores malicious WMI instructions". *Munir Kotadia*. CBS Interactive. 21 October 2004. Retrieved 2009-04-05.
- [136] 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.
- [137] Jon Erickson (August 6, 2008). "Antivirus Software Heads for the Clouds". *Information Week*. Retrieved 2010-02-24.
- [138] Brian Krebs (March 9, 2007). "Online Anti-Virus Scans: A Free Second Opinion". *Washington Post*. Retrieved 2011-02-24.
- [139] Ryan Naraine (February 2, 2007). "Trend Micro ships free 'rootkit buster'". *ZDNet*. Retrieved 2011-02-24.
- [140] Neil J. Rubenking (March 26, 2010). "Avira AntiVir Personal 10". *PC Magazine*. Retrieved 2011-02-24.
- [141] Neil J. Rubenking (September 16, 2010). "PC Tools Spyware Doctor with AntiVirus 2011". *PC Magazine*. Retrieved 2011-02-24.
- [142] Neil J. Rubenking (October 4, 2010). "AVG Anti-Virus Free 2011". *PC Magazine*. Retrieved 2011-02-24.
- [143] Neil J. Rubenking (November 19, 2009). "PC Tools Internet Security 2010". *PC Magazine*. Retrieved 2011-02-24.
- [144] Carrie-Ann Skinner (March 25, 2010). "AVG Offers Free Emergency Boot CD". *PC World*. Retrieved 2011-02-24.
- [145] "FBI estimates major companies lose \$12m annually from viruses". 30 January 2007. Retrieved 20 February 2011.
- [146] Michael Kaiser (April 17, 2009). "Small and Medium Size Businesses are Vulnerable". *National Cyber Security Alliance*. Retrieved 2011-02-24.
- [147] "Nearly 50% of women don't use antivirus". SPAM-fighter.

4.3.9 Bibliography

- Szor, Peter (2005), *The Art of Computer Virus Research and Defense*, Addison-Wesley, ISBN 0-321-30454-3

4.3.10 External links

- Antivirus software at DMOZ

4.4 Secure coding

Securing coding is the practice of developing computer software in a way that guards against the accidental introduction of security vulnerabilities. Defects, bugs and logic flaws are consistently the primary cause of commonly exploited software vulnerabilities. Through the analysis of thousands of reported vulnerabilities, security professionals have discovered that most vulnerabilities stem from a relatively small number of common software programming errors. By identifying the insecure coding practices that lead to these errors and educating developers on secure alternatives, organizations can take proactive steps to help significantly reduce or eliminate vulnerabilities in software before deployment.

4.4.1 See also

- Defensive programming
- Secure input and output handling
- Security bug

4.4.2 References

- Taylor, Art; Brian Buege; Randy Layman (2006). *Hacking Exposed J2EE & Java*. McGraw-Hill Primis. p. 426. ISBN 0-390-59975-1.

4.4.3 External links

4.5 Secure by design

For the British building security initiative, see Secured by Design.

Secure by design, in software engineering, means that the software has been designed from the ground up to be secure. Malicious practices are taken for granted and care is taken to minimize impact when a security vulnerability is discovered or on invalid user input.

Generally, designs that work well do not rely on being secret. It is not mandatory, but proper security usually means that everyone is allowed to know and understand the design *because it is secure*. This has the advantage that many people are looking at the code, and this improves the odds that any flaws will be found sooner (Linus' law). Of course, attackers can also obtain the code, which makes it easier for them to find vulnerabilities as well.

Also, it is very important that everything works with the least amount of privileges possible (**principle of least privilege**). For example a Web server that runs as the administrative user (root or admin) can have the privilege to remove files and users that do not belong to itself.

Thus, a flaw in that program could put the entire system at risk. On the other hand, a Web server that runs inside an isolated environment and only has the privileges for required network and filesystem functions, cannot compromise the system it runs on unless the security around it is in itself also flawed.

4.5.1 Security by design in practice

Many things, especially input, should be distrusted by a secure design. A fault-tolerant program could even distrust its own internals.

Two examples of insecure design are allowing buffer overflows and format string vulnerabilities. The following C program demonstrates these flaws:

```
int main() { char a_chBuffer[100]; printf("What is your name?\n"); gets(a_chBuffer); printf("Hello, "); printf(a_chBuffer); printf("\n"); return 0; }
```

Because the `gets` function in the C standard library does not stop writing bytes into buffer until it reads a new-line character or EOF, typing more than 99 characters at the prompt constitutes a buffer overflow. Allocating 100 characters for buffer with the assumption that almost any given name from a user is no longer than 99 characters doesn't prevent the user from actually *typing* more than 99 characters. This can lead to arbitrary machine code execution.

The second flaw is that the program tries to print its input by passing it directly to the `printf` function. This function prints out its first argument, replacing conversion specifications (such as "%s", "%d", et cetera) sequentially with other arguments from its call stack as needed. Thus, if a malicious user entered "%d" instead of his name, the program would attempt to print out a non-existent integer value, and undefined behavior would occur.

A related mistake in Web programming is for an online script not to validate its parameters. For example, consider a script that fetches an article by taking a filename, which is then read by the script and parsed. Such a script might use the following hypothetical URL to retrieve an article about dog food:

`http://www.example.net/cgi-bin/article.sh?name=dogfood.html`

If the script has no input checking, instead trusting that the filename is always valid, a malicious user could forge a URL to retrieve configuration files from the webserver:

`http://www.example.net/cgi-bin/article.sh?name=../../../../etc/passwd`

Depending on the script, this may expose the `/etc/passwd` file, which on Unix-like systems contains (among others) user IDs, their login names, home directory paths and shells. (See SQL injection for a similar attack.)

4.5.2 Server/client architectures

In server/client architectures, the program at the other side may not be an authorised client and the client's server may not be an authorised server. Even when they are, a man-in-the-middle attack could compromise communications.

Often the easiest way to break the security of a client/server system is not to go head on to the security mechanisms, but instead to go around them. A man in the middle attack is a simple example of this, because you can use it to collect details to impersonate a user. Which is why it is important to consider encryption, hashing, and other security mechanisms in your design to ensure that information collected from a potential attacker won't allow access.

Another key feature to client-server security design is general good-coding practices. For example, following a known software design structure such as client and broker can help in designing a well-built structure with a solid foundation. Furthermore, if the software is to be modified in the future, it is even more important that it follows a logical foundation of separation between the client and server. This is because if a programmer comes in and cannot clearly understand the dynamics of the program they may end up adding or changing something that can add a security flaw. Even with the best design this is always a possibility, but the better standardized the design the less chance there is of this occurring.

4.5.3 See also

- Computer security
- Cyber security standards
- Hardening
- Multiple Independent Levels of Security
- Secure by default
- Security through obscurity
- Software Security Assurance

4.5.4 External links

- Secure Programming for Linux and Unix HOWTO
- Secure UNIX Programming FAQ
- Top 10 Secure Coding Practices

4.6 Security-focused operating system

This is a list of **operating systems** with a sharp **security** focus. Here, “security-focused” means that the project is devoted to increasing the security as a major goal. As such, something can be secure without being “security-focused.” For example, almost all of the operating systems mentioned here are faced with security bug fixes in their lifetime; however, they all strive to consistently approach all generic security flaws inherent in their design with new ideas in an attempt to create a secure computing environment. Security-focused does not mean **security-evaluated operating system**, which refers to operating systems that have achieved certification from an external security-auditing organization. An operating system that provides sufficient support for multilevel security and evidence of correctness to meet a particular set of government requirements is called a “**trusted operating system**”. The list is alphabetical and does not imply a ranking.

4.6.1 BSD

BSD is a family of **Unix** variants derived from a code base originating at the **University of California, Berkeley**. All derived BSD operating systems are released under the terms of a **BSD-style license**. There are several BSD variants, with only one being heavily focused on security.

OpenBSD

OpenBSD is an open source **BSD** operating system that is known to be concerned heavily with security. The project has completed rigorous manual reviews of the code and addressed issues most systems have not.^[1] OpenBSD also supplies an executable space protection scheme known as **W^X** (memory is writeable xor executable), as well as a **ProPolice** compiled executable base. OpenBSD became the first mainstream operating system to support partial **ASLR** and to activate it by default, ASLR support was completed in 2008 when it added support for **position-independent executable (PIE)** binaries.

See also: [OpenBSD security features](#)

TrustedBSD

TrustedBSD is a sub-project of **FreeBSD** designed to add trusted operating system extensions, targeting the **Common Criteria for Information Technology Security Evaluation** (see also **Orange Book**). Its main focuses are working on access control lists, event auditing, extended attributes, mandatory access controls, and fine-grained capabilities. Since access control lists are known to be

confronted with the **confused deputy problem**, capabilities are a different way to avoid this issue. As part of the TrustedBSD project, there is also a port of the NSA's FLASK/TE implementation to run on FreeBSD. Many of these trusted extensions have been integrated into the main FreeBSD branch starting at 5.x.

HardenedBSD

HardenedBSD is a forked project from FreeBSD,^[2] that brings low level security enhancements to the FreeBSD project, by aiming “to continuously implement kernel and userland hardening features, such as Address Space Layout Randomization (**ASLR**), mprotect hardening, Position Independent Executable (**PIE**) support, and PTrace hardening, among other features.”^{[3][4]} Together with TrustedBSD, the HardenedBSD project resembles Trusted Solaris, a precursor which provided further security enhancements to the Solaris operating system. These early enhancements found their way into security features across a number of different operating systems, mostly Unix-like ones.

4.6.2 Linux

The **Linux** kernel provides among other security features the **Linux Security Module (LSM)**, officially integrated with the mainline Linux kernel since 2003. However, there have been specialized distributions and projects which attempt to make Linux more secure in general or for particular scenarios.

Alpine Linux

Alpine Linux is a lightweight **musl** and **BusyBox**-based distribution. It uses **PaX** and **grsecurity** patches in the default kernel and compiles all packages with **stack-smashing** protection. Version 3.0 was released June 4, 2014.

Annvix

Annvix was originally forked from **Mandriva** to provide a security-focused server distribution that employs **ProPolice** protection, hardened configuration, and a small footprint. There were plans to include full support for the **RSBAC** mandatory access control system. However, Annvix is dormant, with the last version being released on December 30, 2007.

Debian

The “Securing Debian Manual”^[5] contains information for **Debian** administrators. Debian includes support for **SELinux**^[6] since version 5.0, as well as **AppArmor** and

Tomoyo. See also **Debian Security** information and policy.

EnGarde Secure Linux

EnGarde Secure Linux is a secure platform designed for servers. It has had a browser-based tool for **MAC** using SELinux since 2003. Additionally, it can be accompanied with Web, DNS, and email enterprise applications, specifically focusing on security without any unnecessary software. The community platform of EnGarde Secure Linux is the bleeding-edge version freely available for download.

Fedora

Fedora is a free, Red Hat-sponsored community-developed **Linux** distribution. Fedora is a mainstream distribution that makes significant efforts to improve security.^[7] As a consequence it has a fully integrated **SELinux MAC** and fine-grained executable memory permission system (**Exec Shield**) and all binaries compiled with **GCC**'s standard stack-smashing protection, as well as focusing on getting security updates into the system in a timely manner.

Hardened Gentoo

Hardened Gentoo is a sub-project of the **Gentoo Linux** project. Hardened Gentoo offers a **ProPolice** protected and position-independent executable base using exactly the same package tree as Gentoo. Executable space protection in Hardened Gentoo is handled by **PaX**. The Hardened Gentoo project is an extremely modular project, and also provides subprojects to integrate other **intrusion-detection** and **mandatory access control** systems into Gentoo. All of these can be optionally installed in any combination, with or without **PaX** and a **ProPolice** base.

Hardened Linux

Hardened Linux is a small distribution for firewalls, intrusion detection systems, VPN-gateways and authentication jobs that is still under heavy development. It includes **grsecurity**, **PaX** and **GCC** stack smashing protection.

Immunix

Immunix was a commercial distribution of Linux focused heavily on security. They supplied many systems of their own making, including **StackGuard**; cryptographic signing of executables; race condition patches; and format

string exploit guarding code. Immunix traditionally releases older versions of their distribution free for non-commercial use. The Immunix distribution itself is licensed under two licenses: The Immunix commercial and non-commercial licenses. Many tools within are GPL, however; as is the kernel.

Kali Linux

Kali Linux is a Debian-derived Linux distribution designed for digital forensics and penetration testing, formerly known as Backtrack.

Mempo

Mempo [8] is a privacy-focused derivative of Debian and aims to provide the “most secure and yet comfortable out-of-the-box Desktop and Server” operating system. Among others, Mempo also relies on the Freenet and Tor.

Openwall Project

Solar Designer's Openwall Project (Owl) was the first distribution to have a non-executable userspace stack, /tmp race condition protection and access control restrictions to /proc data, by way of a kernel patch. It also features a per-user tmp directory via the pam_mktemp PAM module, and supports Blowfish password encryption.

Parrot Security OS

Parrot Security OS is a Debian-based GNU/Linux distribution designed to perform security and penetration tests, do forensic analysis or act in anonymity. It uses the MATE Desktop Environment, Linux Kernel 3.16 or higher and it is available as a live lightweight installable ISO image for 32-bit, 64-bit and ARM processors with forensic options at boot, optimizations for programmers and new custom pentesting tools.

Pentoo Project

Pentoo Penetration Testing Overlay and Livecd is a live CD and Live USB designed for penetration testing and security assessment. Based on Gentoo Linux, Pentoo is provided both as 32-bit and 64-bit installable live cd. Pentoo is also available as an overlay for an existing Gentoo installation. It features packet injection patched wifi drivers, GPGPU cracking software, and lots of tools for penetration testing and security assessment. The Pentoo kernel includes grsecurity and PAX hardening and extra patches – with binaries compiled from a hardened toolchain with the latest nightly versions of some tools available.

Qubes OS

Qubes OS is a Linux distribution based around the Xen hypervisor that allows to group programs into a number of isolated sandboxes (virtual machines) to provide security. Windows for programs running within these sandboxes (“security domains”) can be color coded for easy recognition. The security domains are configurable, they can be transient (changes to the file system will not be preserved), and their network connection can be routed through special virtual machines (for example one that only provides Tor networking). The operating system provides secure mechanisms for copy and paste and for copying files between the security domains. [9]

Replicant

Replicant is a free and open-source operating system based on the Android mobile platform, which aims to replace all proprietary Android components with their free software counterparts. It is available for several smartphones and tablet computers. [10][11][12][13]

In March 2014, the Replicant project announced the discovery of a backdoor present in a wide range of Samsung Galaxy products that allows the baseband processor to read and write the device’s storage, [14][15] sometimes with normal user privileges and sometimes as the root user depending on device model. [16] It is unknown whether Samsung’s proprietary firmware for the radio chip can be remotely instructed to use these access features and whether the vulnerability was introduced with legitimate uses in mind.

Red Hat Enterprise Linux

Red Hat Enterprise Linux offers the same security benefits as Fedora with the additional support of back-porting security fixes to the released versions of the packages (particularly the kernel) so the sys-admin does not have to perform a significant (and risky) upgrade to get a security fix.

Subgraph OS

Currently still under development, Subgraph OS is designed to be difficult to attack. This is accomplished through system hardening and a proactive, ongoing focus on security and attack resistance. Subgraph OS also places emphasis on the integrity of installable software packages. It features Grsecurity/PaX, Linux Namespaces and Xpra for application containment, mandatory file system encryption, and resistance to cold boot attacks and ensures each application goes through a different circuit over the Tor network.

Tails (The Amnesic Incognito Live System)

Tails or The Amnesic Incognito Live System is a Linux distribution aimed at preserving privacy and anonymity. It is the next iteration of development on the Incognito Linux distribution. It is based on Debian, with all outgoing connections forced to go through Tor and optionally I2P. Direct (non-anonymous) connections are blocked. The system is designed to be booted as a live CD or USB and no trace (digital footprint) is left on local storage unless explicitly told to. The Tor Project has provided most of the financial support for development.

Ubuntu Privacy Remix

Ubuntu Privacy Remix is a distribution based on Ubuntu, focused on security.^[17]

Whonix (anonymous operating system)

Whonix is an anonymous general-purpose operating system based on VirtualBox, Debian GNU/Linux, Tor, and optionally I2P. By Whonix design, IP and DNS leaks are impossible. Not even malware as superuser can find out the user's real IP address or location, because Whonix consists of two (virtual) machines. One machine solely runs Tor and acts as a gateway, called Whonix-Gateway. The other machine, called Whonix-Workstation, is on a completely isolated network. Only connections through Tor are possible.

IprediaOS

IprediaOS is an operating system in which all connections go through I2P.^[18] This system is notable by his use of anonymous bittorrent download.^[19] But it is known as using out-of date I2P version, so users focusing on security must update I2P before use.^[20]

Liberté Linux

Liberté Linux is a secure, reliable, lightweight and easy to use Gentoo-based LiveUSB/SD/CD Linux distribution with the primary purpose of enabling anyone to communicate safely and covertly in hostile environments. The last official release was in 2012, and seems to have been abandoned at this point.^[21]

Security Onion

Security Onion is an Ubuntu-based distribution for intrusion detection and network security monitoring. In contains tools such as Snort, Suricata, Sguil, Bro and Xplico.^[22] ^[23] ^[24]

4.6.3 Solaris

Solaris is a Unix variant created by Sun Microsystems. Solaris itself is not inherently security-focused. The major portion of the Solaris source code has been released via the OpenSolaris project, mostly under the Common Development and Distribution License. Enhancements to OpenSolaris, both security related and others, are backported to the official Solaris when Sun certifies their quality.

Trusted Solaris

Trusted Solaris is a security-focused version of the Solaris Unix operating system. Aimed primarily at the government computing sector, Trusted Solaris adds detailed auditing of all tasks, pluggable authentication, mandatory access control, additional physical authentication devices, and fine-grained access control. Trusted Solaris is Common Criteria certified. (See and) The most recent version, Trusted Solaris 8 (released 2000), received the EAL4 certification level augmented by a number of protection profiles. Telnet was vulnerable to buffer overflow exploits until patched in April 2001.^[25]

Solaris 10 and trusted functionality

Trusted Solaris functionality has now been added to the mainstream version of Solaris. In the 11/06 update to Solaris 10, the *Solaris Trusted Extensions* feature adds mandatory access control and labelled security. Introduced in the same update, the *Secure by Default Networking* feature implements less services on by default compared to most previous releases which had most services enabled. RBAC, found in both mainstream Solaris and Trusted Solaris, dramatically lessens the need for using root directly by providing a way for fine grained control over various administrative tasks.

4.6.4 Microsoft Windows Server

Starting with Windows Server 2008, the server can run in “core” mode. In this mode of operation, the traditional graphical user interface is done away with, and replaced with a Windows command prompt. Roles and software for the server are then installed individually. This serves not only to lessen the strain on system resources produced by unwanted or unneeded applications, but also to reduce the overall “attack surface” of the operating system by virtue of excluding programs that may contain vulnerabilities.^[26]

4.6.5 Object-capability systems

These operating systems are all engineered around a different paradigm of security, object-capabilities, where in-

stead of having the system deciding if an access request should be granted (usually through one or several access control lists), the bundling of authority and designation makes it impossible to request anything not legitimate.

- KeyKOS
- EROS
- CapROS
- seL4
- Fiasco.OC

4.6.6 See also

- Operating system (section Security)
- Comparison of operating systems
- Capabilities and access control lists
- IX (operating system)
- OpenBSM
- Security-evaluated operating system
- Trusted operating system
- Security engineering
- Damn Vulnerable Linux

4.6.7 References

- [1] McIntire, Tim (8 August 2006). “Take a closer look at OpenBSD”. IBM. Retrieved 19 February 2015.
- [2] “HardenedBSD – Proactive Security Project”. <http://bsdmag.org>. HAKIN9 MEDIA SP. Z O.O. SP. K. Retrieved 12 June 2015.
- [3] Carlier, David (December 4, 2014). “Hardened BSD – A Proactive Security Project”. *BSD Magazine* (HAKIN9 MEDIA SP. Z O.O. SP. K.) **8** (10/2014 (63)). ISSN 1898-9144. Retrieved 12 June 2015.
- [4] Percival, Colin, Fabian Keil, delphij, mahrens, et. al. (12 June 2015). “HardenedBSD — sys/dev/xen/blkback blk-back.c, sys/dev/xen/blkfront blkfront.c block.h” (4443). secure.freshbsd.org. Retrieved 12 June 2015.
- [5] “Securing Debian Manual”. debian.org. Retrieved 19 April 2015.
- [6] “SELinux”. debian.org. Retrieved 19 April 2015.
- [7] “SELinux: бронежилет для корпоративного пингвина” [SELinux: bullet-proof vest for corporate penguin] (in Russian). 6 September 2011. Retrieved 26 October 2011.
- [8] Mempo
- [9] “Redirecting...”. qubes-os.org. Retrieved 19 April 2015.
- [10] “Overview - Replicant”. Redmine.replicant.us. Retrieved 2013-09-30.
- [11] Paul Kocialkowski (February 4, 2012). “WikiStart – Replicant”. Redmine.replicant.us. Retrieved 2013-09-30.
- [12] “Android and Users’ Freedom - GNU Project - Free Software Foundation”. Gnu.org. Retrieved 2013-09-30.
- [13] “About”. Replicant project. Retrieved 2013-09-30.
- [14] Don Reisinger (13 March 2014). “Samsung Galaxy devices may have backdoor to user data, developer says”. CNET. Retrieved 25 April 2014.
- [15] Michael Larabel (12 March 2014). “Replicant Developers Find Backdoor In Android Samsung Galaxy Devices”. *Phoronix*. Retrieved 25 April 2014.
- [16] Paul Kocialkowski. “Samsung Galaxy Back-door”. *Replicant Wiki*. Archived from the original on 6 April 2014. Retrieved 25 April 2014.
- [17] Grey One. “Linux Distributions Built For Anonymity”. *GreyCoder*. Retrieved 19 April 2015.
- [18] “Ipredia”. ipredia.org. Retrieved 19 April 2015.
- [19] “IprediaOS”. ipredia.org/os. Retrieved 19 May 2015.
- [20] “IprediaI2P-out-of-date”. reddit.com/r/i2p/comments/2hcblh/anyone_else_here_use_iprediaos/. Retrieved 19 May 2015.
- [21] “DE(E)SU - Liberté Linux”. dee.su. Retrieved 19 April 2015.
- [22] Security Onion (official website).
- [23] [<http://sourceforge.net/p/security-onion/wiki/Home/> Security Onion Wiki (on SourceForge)].
- [24] Security Onion Solutions: Tools (on GitHub).
- [25] “Sun Patch: Trusted Solaris 8 4/01: in.telnet patch”. 4 October 2002. Retrieved 13 August 2012. 4734086 in.telnetd vulnerable to buffer overflow ?? (Solaris bug 4483514)
- [26] “What is Server Core?”. Microsoft TechNet. Microsoft Corporation. Retrieved 17 October 2013.

4.6.8 External links

- OpenBSD
- Hardened Linux
- Qubes

4.7 Authentication

For other uses of the terms “authentication,” “authentic,” and “authenticity”, see [Authenticity](#).

Authentication (from Greek: αὐθεντικός *authentikos*, “real, genuine,” from αὐθέντης *authentes*, “author”) is the act of confirming the truth of an attribute of a single piece of data (a datum) claimed true by an entity. In contrast with [identification](#) which refers to the act of stating or otherwise indicating a claim purportedly attesting to a person or thing’s identity, authentication is the process of actually confirming that identity. It might involve confirming the identity of a person by validating their identity documents, verifying the validity of a Website with a [digital certificate](#), tracing the age of an artifact by [carbon dating](#), or ensuring that a product is what its packaging and labeling claim to be. In other words, authentication often involves verifying the validity of at least one form of identification.

4.7.1 Methods

Main article: [Provenance](#)

Authentication has relevance to multiple fields. In art, antiques, and [anthropology](#), a common problem is verifying that a given artifact was produced by a certain person or was produced in a certain place or period of history. In computer science, verifying a person’s identity is often required to secure access to confidential data or systems.

Authentication can be considered to be of three types:

The **first type** of authentication is accepting proof of identity given by a credible person who has first-hand evidence that the identity is genuine. When authentication is required of art or physical objects, this proof could be a friend, family member or colleague attesting to the item’s provenance, perhaps by having witnessed the item in its creator’s possession. With autographed sports memorabilia, this could involve someone attesting that they witnessed the object being signed. A vendor selling branded items implies authenticity, while he or she may not have evidence that every step in the supply chain was authenticated. Authority based trust relationships (centralized) drive the majority of secured internet communication through known public [certificate authorities](#); peer based trust (decentralized, [web of trust](#)) is used for personal services like email or files ([pretty good privacy](#), [GNU Privacy Guard](#)) and trust is established by known individuals signing each other’s keys (proof of identity) or at [Key signing parties](#), for example.

The **second type** of authentication is comparing the attributes of the object itself to what is known about objects of that origin. For example, an art expert might look for similarities in the style of painting, check the location

and form of a signature, or compare the object to an old photograph. An archaeologist might use carbon dating to verify the age of an artifact, do a chemical analysis of the materials used, or compare the style of construction or decoration to other artifacts of similar origin. The physics of sound and light, and comparison with a known physical environment, can be used to examine the authenticity of audio recordings, photographs, or videos. Documents can be verified as being created on ink or paper readily available at the time of the item’s implied creation.

Attribute comparison may be vulnerable to forgery. In general, it relies on the facts that creating a forgery indistinguishable from a genuine artifact requires expert knowledge, that mistakes are easily made, and that the amount of effort required to do so is considerably greater than the amount of profit that can be gained from the forgery.

In art and antiques, certificates are of great importance for authenticating an object of interest and value. Certificates can, however, also be forged, and the authentication of these poses a problem. For instance, the son of [Han van Meegeren](#), the well-known art-forger, forged the work of his father and provided a certificate for its provenance as well; see the article [Jacques van Meegeren](#).

Criminal and civil penalties for [fraud](#), [forgery](#), and [counterfeiting](#) can reduce the incentive for falsification, depending on the risk of getting caught.

[Currency](#) and other financial instruments commonly use this second type of authentication method. Bills, coins, and [cheques](#) incorporate hard-to-duplicate physical features, such as fine printing or engraving, distinctive feel, watermarks, and [holographic imagery](#), which are easy for trained receivers to verify.

The **third type** of authentication relies on documentation or other external affirmations. In criminal courts, the rules of evidence often require establishing the [chain of custody](#) of evidence presented. This can be accomplished through a written evidence log, or by testimony from the police detectives and forensics staff that handled it. Some antiques are accompanied by certificates attesting to their authenticity. Signed sports memorabilia is usually accompanied by a certificate of authenticity. These external records have their own problems of forgery and [perjury](#), and are also vulnerable to being separated from the artifact and lost.

In computer science, a user can be given access to secure systems based on user credentials that imply authenticity. A network administrator can give a user a password, or provide the user with a key card or other access device to allow system access. In this case, authenticity is implied but not guaranteed.

Consumer goods such as pharmaceuticals, perfume, fashion clothing can use all three forms of authentication to prevent counterfeit goods from taking advantage of a popular brand’s reputation (damaging the brand owner’s

sales and reputation). As mentioned above, having an item for sale in a reputable store implicitly attests to it being genuine, the first type of authentication. The second type of authentication might involve comparing the quality and craftsmanship of an item, such as an expensive handbag, to genuine articles. The third type of authentication could be the presence of a trademark on the item, which is a legally protected marking, or any other identifying feature which aids consumers in the identification of genuine brand-name goods. With software, companies have taken great steps to protect from counterfeiters, including adding holograms, security rings, security threads and color shifting ink.^[1]

4.7.2 Factors and identity

The ways in which someone may be authenticated fall into three categories, based on what are known as the factors of authentication: something the user *knows*, something the user *has*, and something the user *is*. Each **authentication factor** covers a range of elements used to authenticate or verify a person's identity prior to being granted access, approving a transaction request, signing a document or other work product, granting authority to others, and establishing a chain of authority.

Security research has determined that for a positive authentication, elements from at least two, and preferably all three, factors should be verified.^[2] The three factors (classes) and some of elements of each factor are:



This is a picture of the front (top) and back (bottom) of an ID Card.

- the **knowledge factors**: Something the user *knows* (e.g., a password, Partial Password, pass phrase, or

personal identification number (PIN), challenge response (the user must answer a question, or pattern), Security question

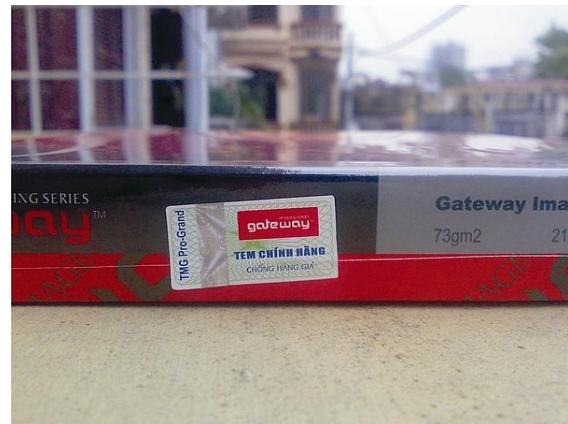
- the **ownership factors**: Something the user **has** (e.g., wrist band, ID card, security token, cell phone with built-in hardware token, software token, or cell phone holding a software token)
- the **inherence factors**: Something the user **is** or **does** (e.g., fingerprint, retinal pattern, DNA sequence (there are assorted definitions of what is sufficient), signature, face, voice, unique bio-electric signals, or other biometric identifier).

Two-factor authentication

Main article: Two-factor authentication

When elements representing two factors are required for authentication, the term *two-factor authentication* is applied — e.g. a bankcard (something the user **has**) and a PIN (something the user **knows**). Business networks may require users to provide a password (knowledge factor) and a pseudorandom number from a security token (ownership factor). Access to a very-high-security system might require a mantrap screening of height, weight, facial, and fingerprint checks (several inherence factor elements) plus a PIN and a day code (knowledge factor elements), but this is still a two-factor authentication.

4.7.3 Product authentication



A Security hologram label on an electronics box for authentication

Counterfeit products are often offered to consumers as being authentic. Counterfeit consumer goods such as electronics, music, apparel, and counterfeit medications have been sold as being legitimate. Efforts to control the supply chain and educate consumers help ensure that authentic products are sold and used. Even security printing

on packages, **labels**, and **nameplates**, however, is subject to counterfeiting.

A secure key storage device can be used for authentication in consumer electronics, network authentication, license management, supply chain management, etc. Generally the device to be authenticated needs some sort of wireless or wired digital connection to either a host system or a network. Nonetheless, the component being authenticated need not be electronic in nature as an authentication chip can be mechanically attached and read through a connector to the host e.g. an authenticated ink tank for use with a printer. For products and services that these Secure Coprocessors can be applied to, they can offer a solution that can be much more difficult to counterfeit than most other options while at the same time being more easily verified.

Packaging

Packaging and labeling can be engineered to help reduce the risks of **counterfeit consumer goods** or the theft and resale of products.^{[3][4]} Some package constructions are more difficult to copy and some have pilfer indicating seals. Counterfeit goods, unauthorized sales (diversion), material substitution and tampering can all be reduced with these anti-counterfeiting technologies. Packages may include authentication seals and use **security printing** to help indicate that the package and contents are not **counterfeit**; these too are subject to counterfeiting. Packages also can include anti-theft devices, such as dye-packs, **RFID** tags, or **electronic article surveillance**^[5] tags that can be activated or detected by devices at exit points and require specialized tools to deactivate. Anti-counterfeiting technologies that can be used with packaging include:

- **Taggant fingerprinting** - uniquely coded microscopic materials that are verified from a database
- Encrypted micro-particles - unpredictably placed markings (numbers, layers and colors) not visible to the human eye
- **Holograms** - graphics printed on seals, patches, foils or labels and used at point of sale for visual verification
- Micro-printing - second line authentication often used on currencies
- Serialized barcodes
- UV printing - marks only visible under UV light
- **Track and trace systems** - use codes to link products to database tracking system
- Water indicators - become visible when contacted with water

- DNA tracking - genes embedded onto labels that can be traced
- Color shifting ink or film - visible marks that switch colors or texture when tilted
- **Tamper evident** seals and tapes - destructible or graphically verifiable at point of sale
- 2d barcodes - data codes that can be tracked
- **RFID** chips

4.7.4 Information content

The authentication of information can pose special problems with electronic communication, such as vulnerability to **man-in-the-middle attacks**, whereby a third party taps into the communication stream, and poses as each of the two other communicating parties, in order to intercept information from each. Extra identity factors can be required to authenticate each party's identity.

Literary forgery can involve imitating the style of a famous author. If an original **manuscript**, typewritten text, or recording is available, then the medium itself (or its packaging — anything from a box to **e-mail headers**) can help prove or disprove the authenticity of the document.

However, text, audio, and video can be copied into new media, possibly leaving only the informational content itself to use in authentication.

Various systems have been invented to allow authors to provide a means for readers to reliably authenticate that a given message originated from or was relayed by them. These involve authentication factors like:

- A difficult-to-reproduce physical artifact, such as a seal, signature, **watermark**, special stationery, or **fingerprint**.
- A shared secret, such as a passphrase, in the content of the message.
- An **electronic signature**; **public-key infrastructure** is often used to cryptographically guarantee that a message has been signed by the holder of a particular private key.

The opposite problem is detection of **plagiarism**, where information from a different author is passed off as a person's own work. A common technique for proving plagiarism is the discovery of another copy of the same or very similar text, which has different attribution. In some cases, excessively high quality or a style mismatch may raise suspicion of plagiarism.

Factual verification

Determining the **truth** or factual accuracy of information in a message is generally considered a separate problem from authentication. A wide range of techniques, from detective work, to fact checking in journalism, to scientific experiment might be employed.

Video authentication

It is sometimes necessary to authenticate the veracity of video recordings used as evidence in judicial proceedings. Proper chain-of-custody records and secure storage facilities can help ensure the admissibility of digital or analog recordings by the Court.

Literacy & Literature authentication

In literacy, authentication is a readers' process of questioning the veracity of an aspect of literature and then verifying those questions via research. The fundamental question for authentication of literature is - Do you believe it? Related to that, an authentication project is therefore a reading and writing activity which students documents the relevant research process ^[6]). It builds students' critical literacy. The documentation materials for literature go beyond narrative texts and likely include informational texts, primary sources, and multimedia. The process typically involves both internet and hands-on library research. When authenticating historical fiction in particular, readers consider the extent that the major historical events, as well as the culture portrayed (e.g., the language, clothing, food, gender roles), are believable for the time period. ^[7].

4.7.5 History and state-of-the-art

Historically, fingerprints have been used as the most authoritative method of authentication, but recent court cases in the US and elsewhere have raised fundamental doubts about fingerprint reliability. Outside of the legal system as well, fingerprints have been shown to be easily spoofable, with British Telecom's top computer-security official noting that "few" fingerprint readers have not already been tricked by one spoof or another.^[8] Hybrid or two-tiered authentication methods offer a compelling solution, such as private keys encrypted by fingerprint inside of a USB device.

In a computer data context, cryptographic methods have been developed (*see* **digital signature** and **challenge-response authentication**) which are currently not spoofable **if and only if** the originator's key has not been compromised. That the originator (or anyone other than an attacker) knows (or doesn't know) about a compromise is irrelevant. It is not known whether these cryptographically based authentication methods are provably

secure, since unanticipated mathematical developments may make them vulnerable to attack in future. If that were to occur, it may call into question much of the authentication in the past. In particular, a digitally signed contract may be questioned when a new attack on the cryptography underlying the signature is discovered.

Strong authentication

The U.S. Government's National Information Assurance Glossary defines **strong authentication** as

layered authentication approach relying on two or more authenticators to establish the identity of an originator or receiver of information.

The above definition is consistent with that of the European Central Bank, as discussed in the **strong authentication** entry.

4.7.6 Authorization

Main article: **Authorization**

The process of **authorization** is distinct from that of au-



A soldier checks a driver's identification card before allowing her to enter a military base.

thentication. Whereas authentication is the process of verifying that "you are who you say you are", authorization is the process of verifying that "you are permitted to do what you are trying to do". Authorization thus presupposes authentication.

For example, a client showing proper identification credentials to a bank teller is asking to be authenticated that he really is the one whose identification he is showing. A client whose authentication request is approved becomes authorized to access the accounts of that account holder, but no others.

However note that if a stranger tries to access someone else's account with his own identification credentials,

the stranger's identification credentials will still be successfully authenticated because they are genuine and not counterfeit, however the stranger will not be successfully authorized to access the account, as the stranger's identification credentials had not been previously set to be eligible to access the account, even if valid (i.e. authentic).

Similarly when someone tries to log on a computer, they are usually first requested to identify themselves with a login name and support that with a password. Afterwards, this combination is checked against an existing login-password validity record to check if the combination is authentic. If so, the user becomes authenticated (i.e. the identification he supplied in step 1 is valid, or authentic). Finally, a set of pre-defined permissions and restrictions for that particular login name is assigned to this user, which completes the final step, authorization.

Even though authorization cannot occur without authentication, the former term is sometimes used to mean the combination of both.

To distinguish "authentication" from the closely related "authorization", the shorthand notations **A1** (authentication), **A2** (authorization) as well as **AuthN / AuthZ** (**AuthR**) or **Au / Az** are used in some communities.^[9]

Normally delegation was considered to be a part of authorization domain. Recently authentication is also used for various type of delegation tasks. Delegation in IT network is also a new but evolving field.^[10]

4.7.7 Access control

Main article: Access control

One familiar use of authentication and authorization is **access control**. A computer system that is supposed to be used only by those authorized must attempt to detect and exclude the unauthorized. Access to it is therefore usually controlled by insisting on an authentication procedure to establish with some degree of confidence the identity of the user, granting privileges established for that identity. One such procedure involves the usage of **Layer 8** which allows IT administrators to identify users, control Internet activity of users in the network, set user based policies and generate reports by username. Common examples of access control involving authentication include:

- Asking for photoID when a contractor first arrives at a house to perform work.
- Using **captcha** as a means of asserting that a user is a human being and not a computer program.
- By using One Time Password (OTP), received on a tele-network enabled device like mobile phone, as an authentication password/PIN
- A computer program using a blind credential to authenticate to another program

- Entering a country with a passport
- Logging in to a computer
- Using a confirmation E-mail to verify ownership of an e-mail address
- Using an Internet banking system
- Withdrawing cash from an ATM

In some cases, ease of access is balanced against the strictness of access checks. For example, the credit card network does not require a personal identification number for authentication of the claimed identity; and a small transaction usually does not even require a signature of the authenticated person for proof of authorization of the transaction. The security of the system is maintained by limiting distribution of credit card numbers, and by the threat of punishment for fraud.

Security experts argue that it is impossible to prove the identity of a computer user with absolute certainty. It is only possible to apply one or more tests which, if passed, have been previously declared to be sufficient to proceed. The problem is to determine which tests are sufficient, and many such are inadequate. Any given test can be spoofed one way or another, with varying degrees of difficulty.

Computer security experts are now also recognising that despite extensive efforts, as a business, research and network community, we still do not have a secure understanding of the requirements for authentication, in a range of circumstances. Lacking this understanding is a significant barrier to identifying optimum methods of authentication. major questions are:

- What is authentication for?
- Who benefits from authentication/who is disadvantaged by authentication failures?
- What disadvantages can effective authentication actually guard against?

4.7.8 See also

- Access Control Service
- Athens access and identity management
- Atomic Authorization
- Authentication OSID
- Authenticity in art
- Authorization
- Basic access authentication
- Biometrics

- CAPTCHA
- Chip Authentication Program
- Closed-loop authentication
- Diameter (protocol)
- Digital Identity
- EAP
- Electronic authentication
- Encrypted key exchange (EKE)
- Fingerprint Verification Competition
- Geolocation
- Global Trust Center
- HMAC
- Identification (information)
- Identity Assurance Framework
- Java Authentication and Authorization Service
- Kerberos
- Multi-factor authentication
- Needham-Schroeder protocol
- OAuth - an open standard for authorization
- OpenID Connect – an authentication method for the web
- OpenID – an authentication method for the web
- Provenance
- Public key cryptography
- RADIUS
- Recognition of human individuals
- Reliance authentication
- Secret sharing
- Secure remote password protocol (SRP)
- Secure Shell
- Security printing
- SQRL
- Strong authentication
- Tamper-evident
- TCP Wrapper
- Time-based authentication
- Two-factor authentication
- Woo Lam 92 (protocol)

4.7.9 References

- [1] <http://www.microsoft.com/en-us/howtotell/Software.aspx>
- [2] Federal Financial Institutions Examination Council (2008). “Authentication in an Internet Banking Environment” (PDF). Retrieved 2009-12-31.
- [3] Eliasson, C; Matousek (2007). “Noninvasive Authentication of Pharmaceutical Products through Packaging Using Spatially Offset Raman Spectroscopy”. *Analytical Chemistry* **79** (4): 1696–1701. doi:10.1021/ac062223z. PMID 17297975. Retrieved 9 Nov 2014.
- [4] Li, Ling (March 2013). [<http://www.sciencedirect.com/science/article/pii/S0007681312001668>] access-date=9 Nov 2014 “Technology designed to combat fakes in the global supply chain”] Check url=scheme (help). *Business Horizons* **56** (2): 167–177. doi:10.1016/j.bushor.2012.11.010.
- [5] How Anti-shoplifting Devices Work”, HowStuffWorks.com
- [6] Norton, D. E. (2004). The effective teaching of language arts. New York: Pearson/Merrill/Prentice Hall.
- [7] McGtigue, E.; Thornton, E.; Wiese, P. (2013). “Authentication Projects for Historical Fiction: Do you believe it?”. *The Reading Teacher* **66**: 495–505. doi:10.1002/trtr.1132.
- [8] The Register, UK; Dan Goodin; 30/3/08; *Get your German Interior Minister's fingerprint, here*. Compared to other solutions, “It's basically like leaving the password to your computer everywhere you go, without you being able to control it anymore,” one of the hackers comments.
- [9] <http://www.cloudave.com/472/authn-authz-and-gluecon/>
- [10] A mechanism for identity delegation at authentication level, N Ahmed, C Jensen - Identity and Privacy in the Internet Age - Springer 2009

4.7.10 External links

- " New NIST Publications Describe Standards for Identity Credentials and Authentication Systems"

4.8 Multi-factor authentication

Multi-factor authentication (MFA) is a method of computer access control which a user can pass by successfully presenting several separate authentication stages through credentials based on knowledge (something you know), possession (something you have), and inherence (something you are).^[1]

4.8.1 Authentication Factors

Knowledge factors

Knowledge factors are the most commonly used form of authentication. In this form, the user is required to prove knowledge of a secret in order to authenticate, for example a password.

A **password** is a secret word or string of characters that is used for user authentication. This is the most commonly used mechanism of authentication. Many multi-factor authentication techniques rely on password as one factor of authentication.^[2]

A **personal identification number (PIN)** is a secret numeric password and is typically used in ATMs. Credit and ATM cards do not contain the PIN or **CVV** on the magnetic stripe.^[3]

Secret questions such as “Where were you born?”, are also a knowledge factor.

Possession factors

Main article: Security token

Possession factors (“something only the user has”) have been used for authentication for centuries, in the form of a key to a lock. The basic principle is that the key embodies a secret which is shared between the lock and the key, and the same principle underlies possession factor authentication in computer systems.

Several methods are used as possession factors:

1.



RSA SecurID token, an example of a disconnected token generator.

Disconnected tokens Disconnected tokens have no connections to the client computer. They typically use a built-in screen to display the generated authentication data, which is manually typed in by the user.^[4]

Connected tokens Connected tokens are devices that are physically connected to the computer to be used, and transmit data automatically.^[5] There are a number of different types, including card readers, wireless tags and USB tokens.^[5]

Inherence factors

These are factors associated with the user, and are usually biometric methods, including fingerprint readers, retina scanners or voice recognition.^[6]

4.8.2 Legislation

United States

Regulation Details for authentication in the USA are defined with the Homeland Security Presidential Directive 12 (HSPD-12).^[7]

Existing authentication methodologies involve the explained three types of basic “factors”. Authentication methods that depend on more than one factor are more difficult to compromise than single-factor methods.^[8]

IT regulatory standards for access to Federal Government systems require the use of multi-factor authentication to access sensitive IT resources, for example when logging on to network devices to perform administrative tasks^[9] and when accessing any computer using a privileged login.^[10]

Guidance NIST Special Publication 800-63-2 discusses various forms of two-factor authentication and provides guidance on using them in business processes requiring different levels of assurance.^[11]

In 2005, the United States' Federal Financial Institutions Examination Council issued guidance for financial institutions recommending financial institutions conduct risk-based assessments, evaluate customer awareness programs, and develop security measures to reliably authenticate customers remotely accessing online financial services, officially recommending the use of authentication methods that depend on more than one factor (specifically, what a user knows, has, and is) to determine the user's identity.^[12] In response to the publication, numerous authentication vendors began improperly promoting challenge-questions, secret images, and other knowledge-based methods as “multi-factor” authentication. Due to the resulting confusion and widespread adoption of such methods, on August 15, 2006, the FFIEC published supplemental guidelines—which states that by definition, a “true” multi-factor authentication system must use distinct instances of the three factors of authentication it had defined, and not just use multiple instances of a single factor.^[13]

4.8.3 Security

According to proponents, multi-factor authentication could drastically reduce the incidence of online identity theft, and other online fraud, because the victim's password would no longer be enough to give a thief permanent access to their information. However, many multi-factor authentication approaches remain vulnerable to phishing,^[14] man-in-the-browser, and man-in-the-middle attacks.^[15]

4.8.4 Implementation considerations

Many multi-factor authentication products require users to deploy client software to make multi-factor authentication systems work. Some vendors have created separate installation packages for network login, Web access credentials and VPN connection credentials. For such products, there may be four or five different software packages to push down to the client PC in order to make use of the token or smart card. This translates to four or five packages on which version control has to be performed, and four or five packages to check for conflicts with business applications. If access can be operated using web pages, it is possible to limit the overheads outlined above to a single application. With other multi-factor authentication solutions, such as "virtual" tokens and some hardware token products, no software must be installed by end users.

Multi-factor authentication is not standardized. There are various implementations of it. Therefore, interoperability is an issue. There exist many processes and facets to consider in choosing, developing, testing, implementing and maintaining an end-to-end secure identity management system, inclusive of all relevant authentication mechanisms and their technologies: this context is considered the "Identity Lifecycle".^[16]

There are drawbacks to multi-factor authentication that are keeping many approaches from becoming widespread. Some consumers have difficulty keeping track of a hardware token or USB plug. Many consumers do not have the technical skills needed to install a client-side software certificate by themselves. Generally, multi-factor solutions require additional investment for implementation and costs for maintenance. Most hardware token-based systems are proprietary and some vendors even charge an annual fee per user. Deployment of hardware tokens is logically challenging. Hardware tokens may get damaged or lost and issuance of tokens in large industries such as banking or even within large enterprises needs to be managed. In addition to deployment costs, multi-factor authentication often carries significant additional support costs. A 2008 survey of over 120 U.S. credit unions by the *Credit Union Journal* reported on the support costs associated with two-factor authentication. In their report, software certificates and software toolbar approaches were reported to have the highest support

costs.

4.8.5 Examples

Several popular web services employ multi-factor authentication, usually as an optional feature that is deactivated by default.^[17]

- Two-factor authentication
- Many Internet services (among them: Google, Amazon AWS) use open Time-based One-time Password Algorithm (TOTP) to support multi-factor or two-factor authentication

4.8.6 See also

- Identity management
- Mutual authentication
- Reliance authentication
- Strong authentication

4.8.7 References

- [1] "Two-factor authentication: What you need to know (FAQ) - CNET". *CNET*. Retrieved 2015-10-31.
- [2] "Securevoy - what is 2 factor authentication?". Retrieved April 3, 2015.
- [3] "Information technology -- Identification cards -- Financial transaction cards". *ISO/IEC 7813:2006*.
- [4] de Borde, Duncan. "Two-factor authentication" (PDF). Archived from the original (PDF) on January 12, 2012.
- [5] van Tilborg, Henk C.A.; Jajodia, Sushil, eds. (2011). *Encyclopedia of Cryptography and Security, Volume 1*. Springer Science & Business Media. p. 1305. ISBN 9781441959058.
- [6] Biometrics for Identification and Authentication - Advice on Product Selection
- [7] US Security Directive as issued on August 12, 2007
- [8] "Frequently Asked Questions on FFIEC Guidance on Authentication in an Internet Banking Environment", August 15, 2006
- [9] "SANS Institute, Critical Control 10: Secure Configurations for Network Devices such as Firewalls, Routers, and Switches".
- [10] "SANS Institute, Critical Control 12: Controlled Use of Administrative Privileges".
- [11] "Electronic Authentication Guide" (PDF). *Special Publication 800-63-2*. NIST. 2013. Retrieved 2014-11-06.

- [12] “FFIEC Press Release”. 2005-10-12. Retrieved 2011-05-13.
- [13] FFIEC (2006-08-15). “Frequently Asked Questions on FFIEC Guidance on Authentication in an Internet Banking Environment” (PDF). Retrieved 2012-01-14.
- [14] Citibank Phish Spoofs 2-Factor Authentication (Brian Krebs, July 10, 2006)
- [15] The Failure of Two-Factor Authentication (Bruce Schneier, March 2005)
- [16] The Identity Lifecycle, Part 1 (Brent Williams, 2010)
- [17] GORDON, WHITSON (3 September 2012). “Two-Factor Authentication: The Big List Of Everywhere You Should Enable It Right Now”. *LifeHacker* (Australia). Retrieved 1 November 2012.

4.8.8 External links

- Attackers breached the servers of RSA and stole information that could be used to compromise the security of two-factor authentication tokens used by 40 million employees (register.com, 18 Mar 2011)
- Banks to Use Two-factor Authentication by End of 2006, (slashdot.org, 20 Oct 2005)
- List of commonly used websites and whether or not they support Two-Factor Authentication
- Microsoft to abandon passwords, Microsoft preparing to dump passwords in favour of two-factor authentication in forthcoming versions of Windows (vnunet.com, 14 Mar 2005)
- Why is two factor authentication needed for your corporate assets
- Multi Factor Visitor Authentication System

4.9 Two-factor authentication

Two-factor authentication (also known as **2FA** or **2-Step Verification**) is a technology patented in 1984^[1] that provides identification of users by means of the combination of two different components. These components may be something that the user knows, something that the user possesses or something that is inseparable from the user. A good example from everyday life is the withdrawing of money from a cash machine. Only the correct combination of a **bank card** (something that the user possesses) and a PIN (personal identification number, i.e. something that the user knows) allows the transaction to be carried out. 2FA is ineffective against modern threats,^[2] like ATM skimming, phishing, and malware etc. Two-factor authentication is a type of **multi-factor authentication**.

4.9.1 Components

The use of two-factor authentication to prove one's identity is based on the premise that an unauthorized actor is unlikely to be able to supply both factors required for access. If in an authentication attempt at least one of the components is missing or supplied incorrectly, the user's identity is not established with sufficient certainty and access to the asset (e.g., a building, or data) being protected by two-factor authentication then remains blocked. The authentication factors of a two-factor authentication scheme may include:

- some physical object in the possession of the user, such as a USB stick with a secret token, a bank card, a key, etc.
- some secret known to the user, such as a username, password, PIN, TAN, etc.
- some physical characteristic of the user (biometrics), such as a fingerprint, eye iris, voice, typing speed, pattern in key press intervals, etc.^[3]

4.9.2 Mobile phone two-factor authentication

The major drawback of authentication performed using something that the user possesses and one other factor is that the plastic token used (the USB stick, the bank card, the key or similar) must be carried around by the user at all times. And if this is stolen or lost, or if the user simply does not have it with him or her, access is impossible. There are also costs involved in procuring and subsequently replacing tokens of this kind. In addition, there are inherent conflicts and unavoidable **trade-offs** between usability and security.

Mobile phone two-factor authentication was developed to provide an alternative method that would avoid such issues. This approach uses mobile devices such as mobile phones and smartphones to serve as “something that the user possesses”. If users want to authenticate themselves, they can use their personal access license (i.e. something that only the individual user knows) plus a one-time-valid, dynamic passcode consisting of digits. The code can be sent to their mobile device by **SMS** or via a special app. An alternative method for authentication developed by Bookit (Finnish company), **iSMS**, embeds the authentication code into the headers (e.g. originating address) so that the user only needs to reply to the message in order to authorize and authenticate. The advantage of both of these methods is that there is no need for an additional, dedicated token, as users tend to carry their mobile devices around at all times anyway. Some professional two-factor authentication solutions also ensure that there is always a valid passcode available for users. If the user has already used a sequence of digits (passcode), this is automatically deleted and the system sends a new code to the

mobile device. And if the new code is not entered within a specified time limit, the system automatically replaces it. This ensures that no old, already used codes are left on mobile devices. For added security, it is possible to specify how many incorrect entries are permitted before the system blocks access.

Advantages of mobile phone two-factor authentication:

- No additional tokens are necessary because it uses mobile devices that are (usually) carried all the time.
- As they are constantly changed, dynamically generated passcodes are safer to use than fixed (static) login information.
- Depending on the solution, passcodes that have been used are automatically replaced in order to ensure that a valid code is always available; acute transmission/reception problems do not therefore prevent logins.
- The option to specify a maximum permitted number of incorrect entries reduces the risk of attacks by unauthorized persons.

Disadvantages of mobile phone two-factor authentication:

- The mobile phone must be carried by the user, charged, and kept in range of a cellular network whenever authentication might be necessary. If the phone is unable to display messages, access is often impossible without backup plans.
- The user must share their personal mobile number with the provider, reducing personal privacy and potentially allowing spam.
- Text messages to mobile phones using **SMS** are insecure and can be intercepted. The token can thus be stolen and used by third parties.^[4]
- Text messages may not be delivered instantly, adding additional delays to the authentication process.
- Modern smart phones are used both for browsing email and for receiving SMS. Email is usually always logged in. So if the phone is lost or stolen, all accounts for which the email is the key can be hacked as the phone can receive the second factor. So smart phones combine the two factors into one factor.

4.9.3 List of mobile phone two-factor authentication solution providers

- ASP.NET Microsoft Identity^[5]

- Twilio Authy
- ASPTOKEN
- Duo
- Google Authenticator
- FreeOTP
- Infobip

4.9.4 References

- [1] “Patent US4720860 - Method and apparatus for positively identifying an individual”. *google.com*. Retrieved 23 October 2015.
- [2] “The Failure of Two-Factor Authentication - Schneier on Security”. *schneier.com*. Retrieved 23 October 2015.
- [3] “What is 2FA?”. Retrieved 19 February 2015.
- [4] SSMS – A Secure SMS Messaging Protocol for the M-Payment Systems, Proceedings of the 13th IEEE Symposium on Computers and Communications (ISCC'08), pp. 700–705, July 2008 arXiv:1002.3171
- [5] “Tutorial: ASP.NET MVC 5 app with SMS and email Two-Factor Authentication”. Microsoft. 20 August 2015. Retrieved 2015-08-20.

4.9.5 See also

- Authentication
- Multi-factor authentication
- Strong authentication

4.10 Authorization (computer access control)

“Authorized” redirects here. For the 2007 Epsom Derby winner, see **Authorized (horse)**.

Authorization or **authorisation** is the function of specifying access rights to resources related to **information security** and **computer security** in general and to **access control** in particular. More formally, “to authorize” is to define an access policy. For example, **human resources** staff is normally authorized to access employee records and this policy is usually formalized as access control rules in a computer system. During operation, the system uses the access control rules to decide whether access requests from (**authenticated**) consumers shall be approved (**granted**) or disapproved (**rejected**). Resources include individual files or an item’s **data**, **computer programs**, **computer devices** and functionality provided by

computer applications. Examples of consumers are computer users, computer programs and other devices on the computer.

4.10.1 Overview

Access control in computer systems and networks rely on access policies. The access control process can be divided into the following two phases: 1) policy definition phase where access is authorized, and 2) policy enforcement phase where access requests are approved or disapproved. Authorization is thus the function of the policy definition phase which precedes the policy enforcement phase where access requests are approved or disapproved based on the previously defined authorizations.

Most modern, multi-user operating systems include access control and thereby rely on authorization. Access control also uses authentication to verify the identity of consumers. When a consumer tries to access a resource, the access control process checks that the consumer has been authorized to use that resource. Authorization is the responsibility of an authority, such as a department manager, within the application domain, but is often delegated to a custodian such as a system administrator. Authorizations are expressed as access policies in some types of “policy definition application”, e.g. in the form of an access control list or a capability, on the basis of the “principle of least privilege”: consumers should only be authorized to access whatever they need to do their jobs. Older and single user operating systems often had weak or non-existent authentication and access control systems.

“Anonymous consumers” or “guests”, are consumers that have not been required to authenticate. They often have limited authorization. On a distributed system, it is often desirable to grant access without requiring a unique identity. Familiar examples of access tokens include keys and tickets: they grant access without proving identity.

Trusted consumers are often authorized for unrestricted access to resources on a system, but must be authenticated so that the access control system can make the access approval decision. “Partially trusted” and guests will often have restricted authorization in order to protect resources against improper access and usage. The access policy in some operating systems, by default, grant all consumers full access to all resources. Others do the opposite, insisting that the administrator explicitly authorizes a consumer to use each resource.

Even when access is controlled through a combination of authentication and access control lists, the problems of maintaining the authorization data is not trivial, and often represents as much administrative burden as managing authentication credentials. It is often necessary to change or remove a user’s authorization: this is done by changing or deleting the corresponding access rules on the system. Using atomic authorization is an alternative to per-system authorization management, where a trusted

third party securely distributes authorization information.

4.10.2 Authorize v authorise

Both spellings are acceptable in UK and AU English. *Authorize* has entered English via medieval Latin, while *authorise* comes from old French. Latin *auctorizare* → French *autoriser* → English.^{[1][2]}

4.10.3 Related interpretations

Public policy

In public policy, authorization is a feature of trusted systems used for security or social control.

Banking

In banking, an authorization is a hold placed on a customer’s account when a purchase is made using a debit card or credit card.

Publishing

In publishing, sometimes public lectures and other freely available texts are published without the consent of the author. These are called unauthorized texts. An example is the 2002 *The Theory of Everything: The Origin and Fate of the Universe*, which was collected from Stephen Hawking’s lectures and published without his permission as per copyright law.

4.10.4 See also

- Security engineering
- Computer security
- Authentication
- Access control
- Kerberos (protocol)
- Operating system
- Authorization OSID
- Authorization hold
- Privilege escalation
- XACML

4.10.5 References

[1] *Shorter Oxford English Dictionary, Fifth Edition*.

[2] *The Australian Oxford Dictionary, Second Edition*.

4.11 Data-centric security

Data-centric security is an approach to security that emphasizes the security of the data itself rather than the security of networks, servers, or applications. Data-centric security is evolving rapidly as enterprises increasingly rely on digital information to run their business and big data projects become mainstream.^[1] ^[2] Data-centric security also allows organizations to overcome the disconnect between IT security technology and the objectives of business strategy by relating security services directly to the data they implicitly protect; a relationship that is often obscured by the presentation of security as an end in itself.^[3]

4.11.1 Key Concepts

Common processes in a data-centric security model include:^[4]

- Discover: the ability to inspect data storage areas at rest to detect sensitive information.
- Manage: the ability to define access policies that will determine if certain data is accessible, editable, or blocked from specific users, or locations.
- Protect: the ability to defend against data loss or unauthorized use of data and prevent sensitive data from being sent to unauthorized users or locations.
- Track: the constant monitoring of data usage to identify meaningful deviations from normal behavior that would point to possible malicious intent.

From a technical point of view, information(data)-centric security relies on the implementation of the following:^[5]

- Information (data) that is self-describing and defending.
- Policies and controls that account for business context.
- Information that remains protected as it moves in and out of applications and storage systems, and changing business context.
- Policies that work consistently through the different defensive layers and technologies implemented.

4.11.2 Technology

Data Access Controls and Policies: Data access control is the selective restriction of access to data. Accessing may mean viewing, editing, or using. Defining proper access controls requires to map out the information, where it resides, how important it is, who it is important to, how sensitive it is and then designing appropriate controls.^[6] Controls need to be able to match the most granular level of access a subject (the user) should have to a data element. Applied to relational databases, data access controls that support a data-centric security model should be able to define and control access at the table, field/column, cell and partial cell levels. In the case of complex enterprise data environments, the organization might need to enforce standards, and manage centrally the use of data protection techniques and security policies^[7]

Encryption

Main article: [Encryption](#)

Encryption is a proven data-centric technique to address the risk of data theft in smartphones, laptops, desktops and even servers, including the cloud. One limitation is that encryption becomes useless once a network intrusion has occurred and cybercriminals operate with stolen valid user credentials.^[8]

Data Masking

Main article: [Data masking](#)

Data Masking is the process of hiding specific data within a database table or cell to ensure that data security is maintained and that sensitive information is not exposed to unauthorized personnel. This may include masking the data from users, developers, third-party and outsourcing vendors, etc. Data masking can be achieved multiple ways: by duplicating data to eliminate the subset of the data that needs to be hidden or by obscuring the data dynamically as users perform requests.

4.11.3 Data-centric security and cloud computing

Cloud computing is an evolving paradigm with tremendous momentum, but its unique aspects exacerbate security and privacy challenges. Heterogeneity and diversity of cloud services and environments demand fine-grained access control policies and services that should be flexible enough to capture dynamic, context, or attribute- or credential-based access requirements and data protection.^[9]

4.11.4 See Also

- [Data masking](#)
- [Data security](#)
- [Defense in depth \(computing\)](#)
- [Information security](#)
- [Information security policies](#)

4.11.5 References

- [1] Gartner Group (2014). “Gartner Says Big Data Needs a Data-Centric Security Focus”.
- [2] SANS Institute (2015). “Data-Centric Security Needed to Protect Big Data Implementations”.
- [3] IEEE (2007). “Elevating the Discussion on Security Management: The Data Centric Paradigm”.

- [4] Wired Magazine (2014). “Information-Centric Security: Protect Your Data From the Inside-Out”.
- [5] Mogull, Rich (2014). “The Information-Centric Security Lifecycle” (PDF).
- [6] Federal News Radio (2015). “NASA Glenn becoming more data-centric across many fronts”.
- [7] IAAP community (2014). “Data-Centric Security: Reducing Risk at the Endpoints of the Organization”.
- [8] MIT Technology Review (2015). “Encryption Wouldn’t Have Stopped Anthem’s Data Breach”.
- [9] IEEE (2010). “Security and Privacy Challenges in Cloud Computing Environments” (PDF).

- In 1988, an employee at the NASA Ames Research Center in California sent a memo by email to his colleagues^[10] 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.^[11]

First generation: packet filters



Gufw firewall. The firewall shows its settings for incoming and outgoing traffic.

4.12.1 History

The term **firewall** originally referred to a wall intended to confine a fire or potential fire within a building.^[7] 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.^[8] The predecessors to firewalls for network security were the routers used in the late 1980s:^[9]

- Clifford Stoll's discovery of German spies tampering with his system^[9]
- Bill Cheswick's “Evening with Berferd” 1992, in which he set up a simple electronic “jail” to observe an attacker^[9]

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.^[12] 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.^[13]

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.^[14]

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.^[15] 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.^[16]

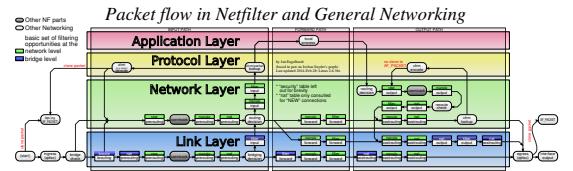
Second generation: “stateful” filters

Main article: Stateful firewall

From 1989–1990 three colleagues from AT&T Bell Laboratories, Dave Presotto, Janardan Sharma, and Kshiti Nigam, developed the second generation of firewalls, calling them Circuit-level gateways.^[17]

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.^[18] 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.^[19] 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.^[20]



Flow of network packets through Netfilter, a Linux kernel module

Third generation: application layer

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

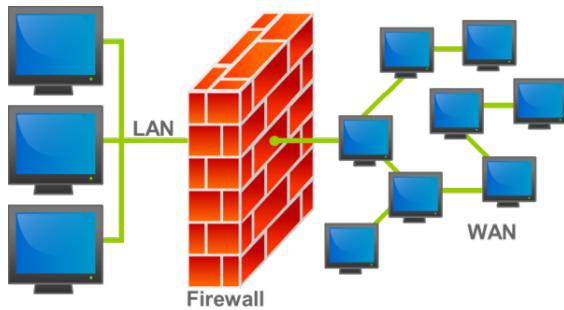
- Intrusion prevention systems (IPS)
- User identity integration (by binding user IDs to IP or MAC addresses for “reputation”); and/or
- Web Application Firewall (WAF). WAF attacks may be implemented in the tool “WAF Fingerprinting utilizing timing side channels” (WAFFle)^[21]

4.12.2 Types

There are different types of firewalls depending on where the communication is taking place, where the communication is intercepted and the state that is being traced.^[22]

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



An illustration of where a firewall would be located in a network.

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 sub-categories, **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.^[23]

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.^[24]

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. Hiding the addresses of protected devices has become an increasingly important defense against network reconnaissance.^[25]

4.12.3 See also

- Access control list
- Windows Firewall
- Next-Generation Firewall
- Bastion host
- Comparison of firewalls
- Computer security
- Distributed firewall
- Egress filtering
- End-to-end connectivity
- 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
- Personal firewall
- Screened-subnet firewall

- Unidirectional network
- Unified threat management
- Virtual firewall
- Vulnerability scanner

4.12.4 References

- [1] Boudriga, Noureddine (2010). *Security of mobile communications*. Boca Raton: CRC Press. pp. 32–33. ISBN 0849379423.
- [2] Oppiger, 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] Definition of Firewall, Check Point Resources
- [6] 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.
- [7] Canavan, John E. (2001). *Fundamentals of Network Security* (1st ed.). Boston, MA: Artech House. p. 212. ISBN 9781580531764.
- [8] Liska, Allan (Dec 10, 2014). *Building an Intelligence-Led Security Program*. Syngress. p. 3. ISBN 0128023708.
- [9] Ingham, Kenneth; Forrest, Stephanie (2002). “A History and Survey of Network Firewalls” (PDF). Retrieved 2011-11-25.
- [10] Firewalls by Dr.Talal Alkharobi
- [11] RFC 1135 The Helminthiasis of the Internet
- [12] Peltier, Justin; Peltier, Thomas R. (2007). *Complete Guide to CISM Certification*. Hoboken: CRC Press. p. 210. ISBN 9781420013252.
- [13] Ingham, Kenneth; Forrest, Stephanie (2002). “A History and Survey of Network Firewalls” (PDF). p. 4. Retrieved 2011-11-25.
- [14] TCP vs. UDP By Erik Rodriguez
- [15] William R. Cheswick, Steven M. Bellovin, Aviel D. Rubin (2003). “Google Books Link”. *Firewalls and Internet Security: repelling the wily hacker*
- [16] Aug 29, 2003 Virus may elude computer defenses by Charles Duhigg, Washington Post
- [17] *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.

- [18] Conway, Richard (204). *Code Hacking: A Developer's Guide to Network Security*. Hingham, Massachusetts: Charles River Media. p. 281. ISBN 1-58450-314-9.
- [19] 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.
- [20] 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.
- [21] "WAFFle: Fingerprinting Filter Rules of Web Application Firewalls". 2012.
- [22] "Firewalls". MemeBridge. Retrieved 13 June 2014.
- [23] "Software Firewalls: Made of Straw? Part 1 of 2". Symantec Connect Community. 2010-06-29. Retrieved 2014-03-28.
- [24] "Auto Sandboxing". Comodo Inc. Retrieved 2014-08-28.
- [25] "Advanced Security: Firewall". Microsoft. Retrieved 2014-08-28.

4.12.5 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

4.13 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 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 security environment (e.g. reconfiguring a firewall) or changing the attack’s content.^[1]

4.13.1 Terminology

- **Burglar Alert/Alarm:** A signal suggesting that a system has been or is being attacked.^[2]
- **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.^[3]
- **False Alarm Rate:** defined as the number of 'normal' patterns classified as attacks (False Positive) divided by the total number of 'normal' patterns.^[3]

ALERT TYPE:-

- **True Positive:** : Attack - Alert
- **False Positive:** : No attack - Alert
- **False Negative:** : Attack - No Alert
- **True Negative:** : No attack - No Alert
- **True Positive:** A legitimate attack which triggers an IDS to produce an alarm.^[2]
- **False Positive:** An event signaling an IDS to produce an alarm when no attack has taken place.^[2]
- **False Negative:** When no alarm is raised when an attack has taken place.^[2]
- **True Negative:** An event when no attack has taken place and no detection is made.

- **Noise:** Data or interference that can trigger a false positive or obscure a true positive.^[2]
- **Site policy:** Guidelines within an organization that control the rules and configurations of an IDS.
- **Site policy awareness:** An IDS's ability to dynamically change its rules and configurations in response to changing environmental activity.^[2]
- **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.^[2]
- **Alarm filtering:** The process of categorizing attack alerts produced from an IDS in order to distinguish false positives from actual attacks.^[2]
- **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.
- **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.
- **Clandestine user:** A person who acts as a supervisor and tries to use his privileges so as to avoid being captured.

4.13.2 HIDS and NIDS

Intrusion detection systems are of two main types, network based (NIDS) and host based (HIDS) intrusion detection systems.

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 and it analyses the Ethernet packet and applies it on the some rules to decide if it is an attack or not. Off-line NIDS deals with a stored data and pass it on a some process to decide if it is an attack or not.^[4]

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](#).

4.13.3 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 [IDPS](#) is commonly used where this can happen automatically or at the command of an operator; systems that both "detect (alert)" and "prevent".

4.13.4 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 pat-

terms (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**.

4.13.5 Statistical anomaly and signature-based IDSe

All Intrusion Detection Systems use one of two detection techniques:

Statistical anomaly-based IDS

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, what ports and devices generally connect to each other—and alert the administrator or user when traffic is detected which is anomalous, or significantly different, than the baseline. The issue is that it may raise a False Positive alarm for a legitimate use of bandwidth if the baselines are not intelligently configured.^[2]

Signature-based IDS

A signature based IDS will monitor packets on the network and compare them against a database of **signatures** or attributes from known malicious threats. This is similar to the way most antivirus software detects malware. The issue is that there will be a lag between a new threat being discovered in the wild and the signature for detecting that threat being applied to the IDS. During that lag time the IDS would be unable to detect the new threat.

4.13.6 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.^[5]
- 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.^[5]
- 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.^[5]

- For signature-based IDSe 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.^[2]
- 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.^[6]

4.13.7 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.

4.13.8 Development

One preliminary IDS concept consisted of a set of tools intended to help administrators review audit trails.^[7] 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.^[8] 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.^[9] 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).^[10]

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.^[11] Haystack was also developed in that year using statistics to reduce audit trails.^[12]

Wisdom & Sense (W&S) was a statistics-based anomaly detector developed in 1989 at the Los Alamos National Laboratory.^[13] 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.^[14] The Network Security Monitor (NSM) performed masking on access matrices for anomaly detection on a Sun-3/50 workstation.^[15] 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.^[16] Computer-

Watch at AT&T Bell Labs used statistics and rules for audit data reduction and intrusion detection.^[17]

Then, in 1991, researchers at the University of California, Davis created a prototype Distributed Intrusion Detection System (DIDS), which was also an expert system.^[18] 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.^[19] 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.^[20] Network Flight Recorder (NFR) in 1999 also used libpcap.^[21] 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.^[22]

The Audit Data Analysis and Mining (ADAM) IDS in 2001 used tcpdump to build profiles of rules for classifications.^[23]

In 2003, Dr. Yongguang Zhang and Dr. Wenke Lee argue for the importance of IDS in networks with mobile nodes.^[24]

4.13.9 See also

- Anomaly-based intrusion detection system
- Application protocol-based intrusion detection system (APIDS)
- Artificial immune system
- Autonomous Agents for Intrusion Detection
- DNS analytics
- Host-based intrusion detection system (HIDS)
- Intrusion prevention system (IPS)
- Protocol-based intrusion detection system (PIDS)
- Security Management
- Intrusion Detection Message Exchange Format (ID-MEF)

4.13.10 Free Intrusion Detection Systems

- ACARM-ng
- AIDE
- Bro NIDS
- Fail2ban

- OSSEC HIDS
- Prelude Hybrid IDS
- Samhain
- Snort
- Suricata

4.13.11 References

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).

- [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] nitin.; Mattord, verma (2008). *Principles of Information Security*. Course Technology. pp. 290–301. ISBN 978-1-4239-0177-8.
- [3] www-users.cs.york.ac.uk/~{}jac/PublishedPapers/AdhocNetsFinal.pdf
- [4] [1] 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.
- [5] 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.
- [6] <http://www.giac.org/paper/gsec/235/limitations-network-intrusion-detection/100739>
- [7] Anderson, James P., “Computer Security Threat Monitoring and Surveillance,” Washington, PA, James P. Anderson Co., 1980.
- [8] Denning, Dorothy E., “An Intrusion Detection Model,” Proceedings of the Seventh IEEE Symposium on Security and Privacy, May 1986, pages 119–131
- [9] 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.
- [10] Lunt, Teresa F., “Detecting Intruders in Computer Systems,” 1993 Conference on Auditing and Computer Technology, SRI International
- [11] Sebring, Michael M., and Whitehurst, R. Alan., “Expert Systems in Intrusion Detection: A Case Study,” The 11th National Computer Security Conference, October, 1988
- [12] Smaha, Stephen E., “Haystack: An Intrusion Detection System,” The Fourth Aerospace Computer Security Applications Conference, Orlando, FL, December, 1988
- [13] Vaccaro, H.S., and Liepins, G.E., “Detection of Anomalous Computer Session Activity,” The 1989 IEEE Symposium on Security and Privacy, May, 1989
- [14] 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
- [15] 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
- [16] 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
- [17] Dowell, Cheri, and Ramstedt, Paul, “The ComputerWatch Data Reduction Tool,” Proceedings of the 13th National Computer Security Conference, Washington, D.C., 1990
- [18] 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.
- [19] Jackson, Kathleen, DuBois, David H., and Stallings, Cathy A., “A Phased Approach to Network Intrusion Detection,” 14th National Computing Security Conference, 1991
- [20] Paxson, Vern, “Bro: A System for Detecting Network Intruders in Real-Time,” Proceedings of The 7th USENIX Security Symposium, San Antonio, TX, 1998
- [21] 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
- [22] 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
- [23] Barbara, Daniel, Couto, Julia, Jajodia, Sushil, Popack, 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
- [24] Intrusion Detection Techniques for Mobile Wireless Networks, ACM WINET 2003 <<http://www.cc.gatech.edu/~{}wenke/papers/winet03.pdf>>

4.13.12 Further reading

- 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". *Avinashilingam Institute for Home Science and Higher Education for Women* (6(4)).
- 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). pp. 554–570.

4.13.13 External links

- [Intrusion Detection Systems at DMOZ](#)

4.14 Intrusion prevention system

Intrusion prevention systems (IPS), also known as **intrusion detection and prevention systems (IDPS)**, are **network security** appliances that monitor network and/or system activities for malicious activity. The main functions of intrusion prevention systems are to identify malicious activity, log information about this activity, attempt to block/stop it, and report it.^[1]

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/block intrusions that are detected.^{[2][3]} More specifically, IPS can take such actions as sending an alarm, dropping the malicious packets, resetting the connection and/or blocking the traffic from the offending IP address.^[4] An IPS can also correct Cyclic Redundancy Check (CRC) errors, unfragment packet streams, prevent TCP sequencing issues, and clean up unwanted transport and network layer options.^{[2][5]}

4.14.1 Classifications

Intrusion prevention systems can be classified into four different types:^{[1][6]}

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.

4.14.2 Detection methods

The majority of intrusion prevention systems utilize one of three detection methods: signature-based, statistical anomaly-based, and stateful protocol analysis.^{[3][3][7]}

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**: A statistical anomaly-based IDS determines the normal network activity —like what sort of bandwidth is generally used, what protocols are used, what ports and devices generally connect to each other— and alerts the administrator or user when traffic is detected which is anomalous (not normal).
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.”^[3]

4.14.3 See also

- Deviations of Protocol States
- **Intrusion Detection System (IDS)**
- Denial-of-service attack
- Host-based intrusion detection (HIDS)
- Network intrusion detection system

- Real-time adaptive security
- Software-defined protection
- Bypass switch

4.14.4 References

- [1] "NIST – Guide to Intrusion Detection and Prevention Systems (IDPS)" (PDF). February 2007. Retrieved 2010-06-25.
- [2] Robert C. Newman (19 February 2009). *Computer Security: Protecting Digital Resources*. Jones & Bartlett Learning. pp. 273–. ISBN 978-0-7637-5994-0. Retrieved 25 June 2010.
- [3] Michael E. Whitman; Herbert J. Mattord (2009). *Principles of Information Security*. Cengage Learning EMEA. pp. 289–. ISBN 978-1-4239-0177-8. Retrieved 25 June 2010.
- [4] Tim Boyles (2010). *CCNA Security Study Guide: Exam 640-553*. John Wiley and Sons. pp. 249–. ISBN 978-0-470-52767-2. Retrieved 29 June 2010.
- [5] Harold F. Tipton; Micki Krause (2007). *Information Security Management Handbook*. CRC Press. pp. 1000–. ISBN 978-1-4200-1358-0. Retrieved 29 June 2010.
- [6] John R. Vacca (2010). *Managing Information Security*. Syngress. pp. 137–. ISBN 978-1-59749-533-2. Retrieved 29 June 2010.
- [7] 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. pp. 162–. ISBN 978-3-642-04341-3. Retrieved 29 June 2010.

4.14.5 External links

- 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”

4.15 Mobile secure gateway

Mobile secure gateway (MSG) is an industry term for the software or hardware appliance that provides secure communication between a mobile application and respective backend resources typically within a corporate network. It addresses challenges in the field of mobile security.

MSG is typically composed of two components - Client library and Gateway. The client is a library that is linked with the mobile application. It establishes secure connectivity to Gateway using cryptographic protocol typically **SSL/TLS**. This represents a secured channel used for communication between the mobile application and hosts. Gateway separates internal IT infrastructure from the Internet, allowing only an authorized client requests to reach a specific set of hosts inside restricted network.

4.15.1 Client Library

The Client library is linked with the corresponding mobile application, and that provides secure access via the Gateway to the set of Hosts. The Client library exposes public API to the mobile application, mimicking platform default **HTTP** client library. The application uses this API to communicate with the desired hosts in a secure way.

4.15.2 Gateway

Gateway is a server or **daemon** typically installed onto physical or virtual appliance placed into **DMZ**. Gateway public interface is exposed to the Internet (or other untrusted network) and accepts **TCP/IP** connections from mobile applications. It operates on **IPv4** and/or **IPv6** networks. Incoming client connections typically use **SSL/TLS** to provide security for the network communication and a mutual trust of communicating peers. Communication protocol is typically based on **SPDY** or **HTTP**.

4.15.3 Host

Gateway forwards requests from connected apps to a collection of configured hosts. These are typically **HTTP** or **HTTPS** servers or services within an **internal network**. The response from a host is sent back to the respective mobile app.

4.15.4 External links

- Mobile Secure Gateway description on TeskaLabs.
- Mobile Secure Gateway description on Symantec.

Chapter 5

Text and image sources, contributors, and licenses

5.1 Text

- **Computer security** *Source:* https://en.wikipedia.org/wiki/Computer_security?oldid=691237030 *Contributors:* Tobias Hoevekamp, Derek Ross, Tuxusuau, Brion VIBBER, Eloquence, Zardoz, Mav, Robert Merkel, The Anome, Stephen Gilbert, Taw, Arcade~enwiki, Graham Chapman, Dachshund, Arvindn, PierreAbbat, Fubar Obfusco, SimonP, Ben-Zin~enwiki, Ant, Ark~enwiki, Heron, Dwheeler, Chuq, Iorek~enwiki, Frecklefoot, Edward, Michael Hardy, Pnm, Kku, Ixfd64, Dcljr, Dori, Arpingstone, CesarB, Haakon, Ronz, Snoyes, Yaronf, Nikai, Smatty, Qwert, Mydogategodshat, Jengod, JidGom, Aarontay, Gingekerr, Taxman, Joy, Vaceituno, Khym Chanur, Pakaran, Robot, Yas~enwiki, Fredrik, ZimZalaBim, Rursus, Texture, KellyCoinGuy, 2501~enwiki, Hadal, Tobias Bergemann, David Gerard, Honta, Wolf530, Tom harrison, Dratman, Mike40033, Siroxo, C17GMaster, Matt Crypto, SWAdair, Bobblewik, Wmahan, Mu, Geni, Antandrus, Beland, Mak098765, CSTAR, GeoGreg, Marc Mongenet, Gcshoyru, Joyous!, Bluefoxy, Squash, Strbenjr, Mike Rosoft, Kmccoy, Monkeyman, Pyrop, Rich Farmbrough, Rhobite, Leibniz, FT2, Jesper Laisen, ArnoldReinhold, YUL89YYZ, Jwalden, Zarutian, MeltBanana, Sperling, Bender235, ZeroOne, Moa3333, JoeSmack, Danakil, Omniparius, Jensbn, El C, Joanjoc~enwiki, Marcok, Perspective, Spearhead, EurekaLott, Nigelj, Stesmo, Smalljim, Rvera~enwiki, Myria, Adrian~enwiki, Boredzo, ClementSeveillac, JohnnyDog, Poweroid, Alansohn, Quiggles, Arthena, Lightdarkness, Cdc, Mrhollybrain, Caesura, Gbeeker, Raraoul, Filx, Proton, M3tainfo, Suruena, HenkvD, 2mcm, Wikicaz, H2g2bob, Condor33~enwiki, Bsdlogical, Johntex, Dan100, Woohookitty, Daira Hopwood, Al E., Prashanthns, Zhen-Xjell, Palica, Kesla, Vininim, Graham87, Clapaucus, BD2412, Icey, Sjakkalle, Rjwilmsi, Seidenstud, Koavf, Guyd, DeadlyAssassin, Dookie~enwiki, Edggar, Oblivious, QuickFox, Kazrak, Dawson, Ligulem, Smtully, Aapo Laitinen, Ground Zero, RexNL, Alvincs, BMF81, JonathanFreed, Jmorgan, J.Ammon, DVdm, Bgwhite, Hall Monitor, Digitalme, Gwernol, FrankTobia, Elf guy, WaveLength, NTBot~enwiki, Alan216, StuffOfInterest, Foxygirltamara, Stephenb, Gaius Cornelius, Ptomes, Morphh, Salsb, Wimt, Bachrach44, Al-Mac, Irishguy, Albedo, Rmky87, Amcfreely, Deku-shrub, Romal, Peter Schmiedeskamp, Zuuuzz, Gorgonzilla, Papergrl, Arthur Rubin, Ka-Ping Yee, Juliano, GraemeL, Rlove, JoanneB, Whouk, NeilN, SkerHawx, SmackBot, Mmernex, Tripletmot, Reedy, KnowledgeOf-Self, TestPilot, Kosik, McGeddon, Stretch 135, Ccalvin, Manjunathbhattacharyya, Zyxw, Gilliam, Ohnoitsjamie, Skizzik, Lakshmin, Kurykh, Autarch, Snori, Miquonranger03, Deli nk, Jenny MacKinnon, Kungming2, Jonasyorg, Timothy Clemans, Frap, Ponnampalam, Nixeagle, KevM, JonHarder, Wine Guy, Cpt~enwiki, Krich, Bslede, Richard001, Stor stark7, Newtonlee, Doug Bell, Harryboyles, Kuru, Geoinline, Disavian, Robofish, Joffeloff, Kwestin, Mr. Lefty, Beetstra, Jadams76, Ehheh, Boxflux, Kvng, Chadnibal, Wfgiuliano, Dthvt, Ivan-Lanin, DavidHOzAu, Lcamtuf, CmdrObot, Tional, ShelfSkewed, Michael B. Trausch, Phatom87, Cydebot, Mblumber, Future Perfect at Sunrise, Blackjackmagic, UncleBubba, Gogo Dodo, Anonymi, Anthonyhcole, GRevolution824, Clovis Sangrail, SpK, Njan, Ebyabe, Thijss!bot, Epbr123, The Punk, Kpavery, Wistless, Oarchimondeo, RichardVeryard, EdJohnston, Druiloor, SusanLesch, I already forgot, Sheridbm, AntiVandalBot, Obiwankenobi, Shirt58, Marokwitz, Khodges, Ellenaz, Manionc, Chill doubt, Dmerrill, SecurityGuy, JAnDbot, Jimothytrotter, Barek, MER-C, The Transhumanist, Technologyvoices, Tqbf, Dave Nelson, Acroterion, Raanoo, VoABot II, Ukuser, JNW, Michi.bo, Szh~enwiki, Hubbardaie, Arcticific, Froid, JXS, AlephGamma, Rohasnagpal, Catgut, WhatamIdoing, Marzoog, Gerrardperrett, Thireus, Devmem, Dharmadhyaksha, DerHexer, JaGa, Reseacord, XandroZ, Gwern, SolitaryWolf, CliffC, =JeffH, Sjjupadhyay~enwiki, Bertix, Booker.ercu, J.delanoy, Gam2121, Maurice Carbonaro, Public Menace, Jesant13, Jreferee, JA.Davidson, Kata-laveno, Touisiau, Ansh1979, Toon05, Mufka, Largoplazo, Dubhe.sk, YoavD, Bonadea, Red Thrush, RJASE1, Cralar, Javeed Safai, ABF, Wiki-ay, Davidwr, Zifert, Crazypete101, Dictouray, Shanata, Haseo9999, Falcon8765, Pctechbytes, Sapphic, Donnymo, FutureDomain, Smith bruce, Kbrose, JonnyJD, Lxicm, Whitehatnetizen, Jargonexpert, SecurInfos~enwiki, Mi-crest, Immzw4, Sephiroth storm, Graceup, Yuxin19, Agilmore, JohnManuel, Flyer22 Reborn, Jojalozzo, Riya.agarwal, Corp Vision, Lightmouse, KathrynLybarger, Mscwriter, Soloxide, StaticGull, Capitalismojo, PabloStraub, Rinconsoleao, Denisarona, White Stealth, Ishisaka, WikipediaMarlith, Sfan00 IMG, Elassint, ClueBot, Shonharris, PipepBot, TransporterMan, Supertouch, Add32, Emantras, Tanglewood4, Mild Bill Hiccup, Nicieguyedc, Dkonytyko, Trivialist, Gordon Ecker, DragonBot, Dwcmsc, Excirial, Socrates2008, Dcampbell30, Moomoo987, Dr-Mx, Rblesky, DanielPharos, Versus22, HarrivBOT, Fathisules, Raysecurity, XLinkBot, BodhisattvaBot, Solinym, Skarebo, Wingfamily, WikiDao, MystBot, Dsimic, JimWalker67, Addbot, Non-dropframe, Cst17, MrOllie, Passport90, Favonian, Torla42, AgadaUrbanit, Tassedethe, Jarble, Ben Ben, Tartarus, Luckas-bot, Yobot, OrgasGirl, The Grumpy Hacker, Librsh, Cyanoa Crylate, Grammaton, THEN WHO WAS PHONE?, Dr Roots, Sweerek, AnomieBOT, JDavis680, Jim1138, Galoubet, Dwayne, Piano non troppo, AdjustShift, Rwhalb, Quantumseven, HRV, Vijay Varadharajan, Materialscientist, Aneah, Stationcall, ArthurBot, Cameron Scott, Intelati, Securitywiki, Hi878, Coolkidmoa, Zarcillo, Mark Schierbecker, Pradameinhoff, Amaury, George1997, Architectchamp, =Josh.Harris, Shadowjams, President of hittin' that ass, FrescoBot, Bingo-101a, Nageh, Ionutzmovie, Cudwin, Expertour, Intelligentsium, Pinethicket, I dream of horses, Edderso, Access-bb, Yahia.barie, RedBot, MastiBot, Wlalng123, Mentmic, Dac04, Banej, Codemaster32, Tjmannos, Nitesh13579, Lotje, Sumone10154, Arkelweis, Ntlhui, Aoidh, Endpointsecurity, Tbhotch, Jesse V., DARTH SIDIOUS 2, Ripchip Bot, Panda Madrid, DASHBot, Julie188, EmausBot,

Timtempleton, Dewritech, Active Banana, P@ddington, Susfele, Dolovis, Cosmoskramer, Alxndrpaz, AvicAWB, Bar-abban, Ocaasi, Solipsys, Tolly4bolly, Sharpie66, DennisIsMe, Veryfoolish, Geohac, ChuispastonBot, Pastore Italy, Tentontunic, Sepersann, Gadgad1973, Rocketrod1960, Jramio, ClueBot NG, AAriel42, Name Omitted, Enfcer, Iliketurtlesmeow, Widr, Helpful Pixie Bot, TechGeek70, Curb Chain, Calabe1992, BG19bot, Mollsieber, M0rphzone, Rubnum, Mohilekedar, Karlomagnus, IraChesterfield, Sburkeel, Zune0112, Ven- era Seyranyan, Wondervoll, Mihai.scridoresi, Jtlopez, Nfirdosian, Alessandra Napolitano, Wannabemodel, Keeper03, BattyBot, Popescu- calin, Arr4, Mrt3366, Cyberbot II, Khazar2, Peter A. Wolff, Soulparadox, Ilker Savas, BIG ISSUE LADY, Saturdayswiki, Dexbot, Jmitola, Mogism, Pete Mahen, Lugia2453, Doopbridge, Sbhalotra, SFK2, Arjungiri, Jamesx12345, ElinaSy, Patna01, Dr Dinosaur IV, Pdecalc- lus, Mbmpexpress, Idavies007, RaheemaHussain, Cyberlawjustin, Rkocher, MoHafesji, ResearcherQ, Westonbowden, Peter303x, Karinera, OccultZone, LukeJeremy, Robevans123, Chima4mani, ClyderRakker46, Jonathan lampe, Jppcap, Leejjung86, Azulfiqar, IrvingCarR, Nyashinski, Monkbot, Nitzy99, Carpalclip3, RicardoBanchez, Owais Khursheed, Oushee, 405Duke, BrettofMoore, Gr3yHatf001, Thetech- girl, ChamithN, Fimatic, Hchaudh3, AndrewKin, JRPolicy, Paeguy, HVanIderstine, Leemily, FormerPatchEditor, Pixelized frog, 3 of Diamonds, Johngot, Bmore84, Informationsystemgeeks, PardonTheComma, Nemoanon, JohnEvans79, Hatebott, Ajay Yankee, Nbyrd2000, Soderbounce, Tejalpatel, Matthews david, Tonybdistil, Shedletskyy, Prshnktiitk, Joanoc, ZacharyKeeton, RSTumber and Anonymous: 702

- **Internet security** *Source:* https://en.wikipedia.org/wiki/Internet_security?oldid=690923534 *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, Pearl, Smoothy, Wtmitchell, Jim Mikulak, Versageek, Bsdlogical, Johntex, Mindmatrix, Deltabeignet, Kevinkinnett, Willlangford, Vary, Gurch, Vec, Intgr, Chobot, DVdm, Bgwhite, Peterl, Elf guy, Wavelength, Argav, Alan216, Phantomsteve, Sarranduin, Akamad, Stephenb, Pseudomonas, Moe Epsilon, Amcfreely, HopeSeekr of xmule, S33k3r, KGasso, Th1rt3en, GraemeL, Thespian, DoriSmith, LSnK, Carlosguitar, Dracone, KnowledgeOfSelf, McGeddon, Jjalexand, Thumperward, N2f, Baa, Can't sleep, clown will eat me, Frap, Tcwolf, Nixeagle, JonHarder, Parent5446, Ianmacm, Mistress Selina Kyle, Ninnmu~enwiki, A. Parrot, Beetstra, Ehheh, A Clown in the Dark, Mere Mortal, Fleet- Command, CmdrObot, Equendil, Cydebot, Gogo Dodo, Tiger williams, ErrantX, Edupedro, Mojo Hand, John254, Dawnseeker2000, Obiwankenobi, Jerrypcj, SiobhanHansa, Raanoo, Io Katai, Gstroot, JohnLai, 28421u2232nfencenc, Gomm, Stephenhou0722, CliffC, Sjjupadhyay~enwiki, Lutz.hausmann, Trusilver, Jesant13, Dbiel, JudyJohn, WikiBone, Karrade, Sbunker, Wiki-ay, Philip Trueman, Zifert, Natg 19, Bryan.dollery, Dirkbb, Mr Nic, LittleBenW, Debog, Kbrose, SieBot, WereSpielChequers, Caltas, DaBler, Bananastalktome, Oxy- moron83, Jruderman, 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, Debsalmi, Kongr43gen, CanadianLinuxUser, Chzz, Yobot, QueenCake, AnomieBOT, DemocraticLuntz, 1exec1, JDavis680, AdjustShift, Bestija11, Ulric1313, Materialscientist, Waltzmoore66, ArthurBot, Itbuddy, LilHelpa, TheAMmollusc, Mgfl2rw, Nasnema, Craftymin- ion, BritishWatcher, S0aasdf2sf, Seldridge99, Fjollig arme, FrescoBot, Nageh, Chevymontecarlo, D'ohBot, A little insignificant, Ham- burgerRadio, Singaporesuperboy, DrilBot, Smeago, Jusses2, In3go, Piandcompany, Jujutacular, Cnwiliams, Intsafetycenter, Train2104, Lotje, Vrenator, TBloemink, DARTH SIDIOUS 2, VernoWhitney, EmausBot, WikitanvirBot, Timtempleton, Andyyype, Zollerriia, K6ka, MathMaven, 30, Riittaajo, Fæ, Zloyvolsheb, Enikuo, Rameez-NJITWILL, W163, Sbmeirow, Donner60, Autoerrant, Tyros1972, Chuis- pastonBot, Humannetwork, Makeet, 28bot, ClueBot NG, Snocman, Cntras, Widr, Karl 334, Helpful Pixie Bot, Electriccatfish2, Tito- dutta, Jeraphine Gryphon, BG19bot, Mleoking, Muneeb2000, Altaïr, Valentinocourt, Cin316, RroccoMaroc, BKoteska, MeanMotherJr, Motta.allo, Sakmaz, Dexbot, بونـا بشـپرـی, Corn cheese, Palmbeachguy, The Mol Man, Tentinator, Wuerzele, Matty.007, PCRepair- Guy, Gaodong, Mickel1982, Chenthilvelmurugana, SiavooshPayandehAzad, Thisara1990, BrettofMoore, Joewalter, Greenmow, XamieA, Mathewherz, AmberHeney and Anonymous: 305
- **Cyberwarfare** *Source:* <https://en.wikipedia.org/wiki/Cyberwarfare?oldid=690126460> *Contributors:* The Anome, Edward, Paul A, Conti, Tpbradbury, Jredmond, Donreed, Nurg, Ancheta Wis, Tom harrison, Gracefool, Utcursh, Slowking Man, Beland, AndrewKeenanRichardson, Orange Goblin, Openfly, Rich Farmbrough, Wkiacc, Kzzl, Pk2000, TheMile, Devil Master, Bobo192, Giraffedata, Obradovic Goran, YDZ, Dominic, H2g2bob, Johntex, Bobrayner, Richard Arthur Norton (1958-), Woohookitty, Mindmatrix, Rbcwa, JeffUK, Plrk, Paxsim- ius, BD2412, Rjwilmsi, Friejose, Qqqqqq, Nihiltres, Vsion, Lmatt, Chobot, Benlisquare, Bgwhite, Wavelength, RussBot, WritersCramp, ONEder Boy, Neil.steiner, Neurotoxic, Deku-shrub, Eclipsed, Kal-El, Intershark, Gregzeng, Arthur Rubin, Geoffrey.landis, Katieh5584, SmackBot, Mmernex, Rtc, McGeddon, C.Fred, Chris the speller, Bluebot, H2ppyme, Snori, Mordantkitten, Frap, Radagast83, Abmac, Uriel-238, Daveschroeder, Ohconfucius, WngLdr34, Mathiasrex, Robofish, NYCJosh, Hvn0413, Hu12, Levineps, Iridescent, Kencf0618, Joseph Solis in Australia, Octane, Quodfui, CmdrObot, Ale jr, Zarex, TVC 15, Neelix, NathanDahlin, Old port, Bobblehead, Hcobb, Nick Number, Alphachimpbot, NByz, Res2216firestar, Turgidson, MER-C, SiobhanHansa, Geniac, Magioladitis, Bongwarrior, Flayer, Buckshot06, XMog, Esanchez7587, Bytecount, Hbent, Atulsnischal, Sjjupadhyay~enwiki, CommonsDelinker, KTo288, Billy Pilgrim, Teh dave, Maurice Carbonaro, Public Menace, Hodja Nasreddin, Octopus-Hands, Andareed, Tukkek, Hillock65, Jevansen, TopGun, Tkgd2007, Ashcroftgm, Fences and windows, Pmedema, WazzaMan, Saibod, LeaveSleaves, Rsnbrgr, Staka, Falcon8765, B.L.A.Z.E, Oth, Kevinfromhk, Boulainvilliers, Kbrose, Moonriddengirl, Rockstone35, BStarky, Flyer22 Reborn, Avaarga, WannabeAmatureHistorian, Oxymoron83, Skinny87, Jericho347, Dillard421, RJ CG, Deciwill, Martarius, Chessy999, Niceguyedc, Trivialist, MasterXC, Gordon Ecker, Nymf, Bylax2005, Rhododendrites, Arjayay, Light show, DumZiBoT, Kgcoleman, Emdarraj, XLinkBot, Avoided, Scostigan, Jaanusele, Addbot, AkhtaBot, StereotypicalWizard, Spy-Ops, Green Squares, Blaylockjam10, Golf2232, 5 albert square, Myk60640, Lightbot, Jarble, Goodmanjoon, Suwa, আশীর্বাদ মন্তব্য, Luckas-bot, Yobot, Tohd8BohaiithuGh1, Bruce404, Waqashsn, Millinski, AnomieBOT, Apollo1758, FeelSunny, 1exec1, Bsimmmons666, Jim1138, Mbeimcik, Ulric1313, Materialscientist, Aneah, ArthurBot, Xqbot, Sionus, Capricorn42, Dondiw, DJWolfy, Alexander Mclean, Wdl1961, BritishWatcher, Miguelito Vieira, Irwick, Peace2keeper, Crzer07, Pradameinhoff, What9999999333, Howsa12, Gordonrox24, Vihelik, Tanagram, FrescoBot, Sidna, Stanszyk, Remdarraj, Jersey92, Deadhorseflogging, HamburgerRadio, Pinethicket, I dream of horses, RedBot, Rochdalehornet, Full-date unlinking bot, Xeworlebi, Utility Monster, NFSReloaded, Jonkerz, Lotje, Collegeofgolf, Oliver H, Remiked, Woodlot, Ifmantis, RjwimsiBot, TjBot, VernoWhitney, Lopifalko, Techhead7890, 09lamin, EmausBot, Ghostofnemo, SoAuthentic, Hirsutism, Kaiser1935, Rail88, GoingBatty, P@ddington, Cybercitizen123, Cowfman, Wikipelli, 7th sojourn, Unmoured, Sabres87, ZéroBot, Prayerfortheworld, EneMsty12, Cymru.lass, Eri- anna, Quantumor, Quite vivid blur, GermanJoe, Joesolo13, Pastore Italy, Boundto, Zabanio, Cat10001a, Paddingtonbaer, Pymansorl, ClueBot NG, Antrim Kate, Satellizer, Rawkage, BrekekekexKoaxKoax, Rezabot, Helpful Pixie Bot, Dukes08, BG19bot, M0rphzone, Pa- ganinip, Donzae, Utkal Ranjan Sahoo, Cyfyann, Jassy pal, Jayadevp13, Cyberbot II, Khazar2, Jabotito48, Mogism, Freshjane0, Dileepp89, Hellowns, Pahlevun, SFK2, ThatIguy77, Dhruvpubby, Kennethgeers, Madtee, Tentinator, Reziebear, TheJoeAlt, Newuser2013, Rmj1, Wikiuser13, Reacher1989, Beastlymac, CFLaherty, Fixture, CrowJU2, Monkbot, Cybersecurity101, Arsonhussy99, Betyd, S166865h, ICPSGWU, ChamithN, Kashifs294, TheCoffeeAddict, Rubbish computer, LCEwald, BlakeTS, Prinsezinde, Nøkkenbuer, Alainsfeir2, RoadWarrior445 and Anonymous: 236
- **Information security** *Source:* https://en.wikipedia.org/wiki/Information_security?oldid=691319625 *Contributors:* Tuxisuu, The Anome, Khendon, William Avery, Ant, Hephaestos, Edward, Madvenu, JohnOwens, Michael Hardy, SebastianHelm, Ronz, Angela, Marteau, Nikai,

Andrewman327, Itai, LMB, ZeWrestler, Chrisbrown, Joy, Vaceituno, Kimvais, Saqib (usurped)~enwiki, Chealer, Fredrik, Rursus, Adhemar, HaeB, Centrx, Inkling, Peruvianllama, Wikibob, Matt Crypto, Edcolins, Utcursch, Andycjp, LiDaobing, Sfoskett, Sam Hocevar, Xinconnu, Imjustmatthew, Ohka-, Klemen Kocjancic, Mike Rosoft, Rich Farmbrough, Rhobite, Wrp103, Sperling, Pnevares, Violetriga, Danakil, MBisanz, El C, Spearhead, Bobo192, John Vandenberg, Wiki-Ed, Giraffedata, Grutness, Guy Harris, Davebailey, Revmachine21, Sutch, Wtmitchell, Suruenia, Amorymeltzer, RainbowOfLight, SteinbDJ, Chris Brown, Kbolina, Bobrayner, Nuno Tavares, Woohookitty, Mindmatrix, Tabletop, SDC, Btyner, Mandarax, Graham87, BD2412, Qwerty, Foulter, Rjwilmsi, Ian the younger, The wub, Aapo Laitinen, JdforresterBot, RexNL, Intgr, Lmatt, DVdm, GroupOne, YurikBot, Borgx, RobotE, RussBot, Stephenb, Gaius Cornelius, Ptomes, Rsrikanth05, AIMac, Irishguy, DeadEyeArrow, Mitte, Mcicogni, Bill.martin, Awillcox, Closedmouth, GraemeL, Rearden9, Ajuk, Whouk, Tom Morris, SmackBot, Glavin, Grye, CommodoCast, Bburton, Mauls, Ohnoitsjamie, JorgePeixoto, Chris the speller, Chris Ulbrich, MK8, Wikipediatrix, Kungming2, DHN-bot-enwiki, Colonies Chris, CelebritySecurity, JonHarder, Addshore, Jdlambert, Richard001, Mistress Selina Kyle, KizzoGirl, Kuru, Noah Salzman, Kvng, Hu12, DouglasCalvert, Iridescent, Xsmith, Wilcho, IvanLanin, Michter, CmdrObot, Ale jrb, Alexconlin, Gfragkos, Mike0131, Robwhitcher, John Yesberg, Puntarenus, UncleBubba, Riker2000, Tenbergen, Alucard (Dr.), Dancter, Nunewsco, Cscryborg, Ebpr123, Wmasterj, Dawnseeker2000, Jplususer, AntiVandalBot, Widefox, Obiwankenobi, Seaphoto, Rossj81, Eddy Scott, Esmond.pitt, MikeLynch, JAnDbot, MER-C, Tqbf, Ishikawa Minoru, Dekimasu, Ged fi, Think outside the box, Richard Bartholomew, Nyttend, Aka042, Morphus063, AlephGamma, Brwriter2, Thireus, Gomm, Kozmando, JaGa, CliffC, Jim.henderson, Jstaryuk, Ranman45, Anaxial, CommonsDelinker, Fellwalker57, Ali, Uncle Dick, Public Menace, INFOSECFORCE, BrWriter2006, Adamdaley, Evb-wiki, Yakheart, Cralar, Soliloquial, Emergentchaos, NoticeBored, WideClyde, HansWDaniel, Sa ashok, Hap-pyroze, UriBraun, Aclyon, OKAMI-InfoSec, Liko81, Shanata, Falcon8765, Enviroboy, Truestate, Cartermichael, Rgalexander, Finnrrind, Amatriain, Rlendog, Derekcslater, Gerakibot, MI-crest, MerileeNC, Sephiroth storm, Alanpc1, JohnManuel, Flyer22 Reborn, JCLately, ObscurO, Corp Vision, Vbscript2, Kieraf~enwiki, Denisarona, Martarius, ClueBot, Shonharris, Kl4m, Hatcher, Vinodvmenon, The Thing That Should Not Be, Donalcampbell, Plastikspork, Little saturn, Mild Bill Hiccup, PolarYukon, V1ntage318, Ktr101, Eekster, Sun Creator, WalterGR, Colemannick, Prokopenya Viktor, DanielPharos, Aitias, Scgilardi, Portal60, Bearsona, XLinkBot, FTsafe, Scottdim-mick, Wingfamily, Cupids wings, Addbot, Nitinbhogan, Stuartfost, PatrickFlaherty, Open4Ever-enwiki, Leszek Jaćzuk, MrOllie, Jmil-gram, Gpdhillon2, Sbilly, Ramfoss, Calm reason, SasiSasi, Jarble, Friedy-peach, Sfoak, Luckas-bot, Yobot, Fraggles81, Idumont, Chlm001, Evans1982, Canoru, Sweerek, Stevedaily, AnomieBOT, Cdschuett, DemocraticLuntz, Jim1138, Piano non troppo, Elieb001, Apau98, Quantumseven, Materialscientist, Citation bot, Aneah, Eumolpo, Stationcall, RealityApologist, LithonHelpa, Delmundo.aver ganzado, Fancy steve, Grantgw, GrouchoBot, Favertama, Aron.j.j, Prunesqualer, Mattg82, Kernel.package, Mathonius, Choibg, TarseeRota, Shadow-jams, Umbrussels, Breadtkt, Thehelpfulbot, Cascadeuse, Srandaman, FrescoBot, Nageh, Itus15q4user, Ionutzmovie, Nainawalli, Haeinous, Raed abu farha, JIK1975, Infinitesteps, Abductive, Shawnse, Cpaidhrin, Feldermouse, Hoo man, MastiBot, Mdjang, Hnguyen322, FoxBot, Prowriter16, Lotje, Himugk, Brian the Editor, DARTH SIDIOUS 2, Mean as custard, RjwilmsiBot, Docpd, VernoWhitney, Alexandru47, DRAGON BOOSTER, Ioliver2, MeS2135, EmausBot, John of Reading, WikitanvirBot, ایمپاریسیون, Timtempleton, Abyss of enchantment, Akjar13, Dkosutic, Inthenet, GoingBatty, Cmlloyd1969, K6ka, Mz7, Vanished user wijn4oj3t8hsafclkhwd2l4tg4, Man-asprakash79, Mattatpredictive, Kilopi, TyA, Dcressb, 111Alpha, Donner60, Alexttcn, Puffin, Pastore Italy, GrayFullbuster, Bzwas, Jramio, ClueBot NG, Satellizer, MosaicSecurity, Sonakshi87, Itsecparidis, Asukite, Widr, Helpful Pixie Bot, Strike Eagle, Wbm1058, BG19bot, Vdmerwe.johann, JaredPoepelman, Jobin RV, Saurav kashyap, Richu jose, BattyBot, BKD-Banker\$, Pratyaa Ghosh, MSS. Selina Kyle, Dtruongwhs, Cyberbot II, ChrisGualtieri, Pspagnoletti, Scaar123, BooleanMaybe, Dexbot, Codename Lisa, Mogism, 2edsphere, RazrRekr201, Patrick.bausemer, Pete Mahen, Lugia2453, Aperks1811, Anotherderelict, Phamnhatkhanh, Param21, Zeromadcowz, She-bang42, Pdecaculus, Rakomwolvesbane, Kolbeinnk, Tentinator, Postonm, Tsarihan, Pkleimr, Mldisch, Reenaii, Sparkau, Malmoe7, Rons corner, Sukhamoyjana, Ittybittytittykitty, Wikilubina, Desire cilow, Saectar, Kate Lynn, Amgauna, Monkbot, Neilcame9978', Greedo8, S166865h, Hannasnow, ScienceGuard, Amatimoltwki, 405Duke, Nitinkathuria80, Thetechgirl, TranquilHope, Manish9050, Raaj.ajm, Datasecurity, Meletisb, Samkel95, Manikaran.singh, Godfarther48, Supdiop, Informationsecurityprofessional, Informationsystemgeeks, DarkSocrates, Latosh Boris, RSTumber and Anonymous: 499

- **Network security** Source: https://en.wikipedia.org/wiki/Network_security?oldid=688686720 Contributors: Ed Brey, ZimZalaBim, Mike Rosoft, Discospinster, Wrp103, Bobo192, Nsaa, Arthena, Woohookitty, Mindmatrix, Aarghdaark, BD2412, Kbdank71, Seraphimblade, Nivix, Ewlyahoocom, Intgr, Chobot, DVdm, Bgwhite, FrankTobia, Epabthit, 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, Iri-descent, Shoeofdeath, Dthvt, Cbrown1023, Leujohn, E smith2000, MaxEnt, Sutekh.destroyer, Shandon, Ayzmo, Thij's!bot, Ebpr123, TheFearow, JustAGal, Michael A. White, Dawnseeker2000, AntiVandalBot, Obiwankenobi, Seaphoto, Marokwitz, Prolog, Ellenaz, AndreasWittenstein, Res2216firestar, Andonic, PhilKnight, Tqbf, Raanoo, VoABot II, AuburnPilot, AlephGamma, Rohasnagpal, Stephan-chou0722, STBot, CliffC, Jonathan Hall, R'n'B, Obscurans, Pharaoh of the Wizards, Jesant13, Ginsengbomb, Smyle.dips, Electric-Valkyrie, DH85868993, Ken g6, Vanished user 39948282, CardinalDan, Wiki-ay, Philip Trueman, Mdeshon, M2petite, GidsR, Fallon8765, NHRHS2010, SieBot, Liamoshan, RJaguar3, Quest for Truth, Flyer22 Reborn, Mscwriter, Dodger67, Hariva, ManOnPipes, ClueBot, Arcitsol, Donalcampbell, Drmies, Mild Bill Hiccup, Boing! said Zebedee, Krmashall, Somno, Jswd, Eekster, Leonard^Bloom, Radiosband, Rhododendrites, Tnspartan, Ranjithsutari, Johnuniq, SoxBot III, SF007, Stickee, Nepenthes, BaroloLover, Mitch Ames, ErkinBatu, Nakamura36, Sindbad72, HexaChord, Sweeper tamonten, Addbot, Willking1979, Dmsynx, Ronjhones, TutterMouse, Jncraton, CanadianLinuxUser, T38291, Cst17, MrOllie, Download, Тиверополник, Tide rolls, Ghalleen, Bodies3819, Batman2472, CaliMan88, Rogger.Ferguson, Sgarchik, Pheebalicious, AnomieBOT, JDavis680, Dwayne, Piano non troppo, AdjustShift, Rwhalb, Ulric1313, Materi-alscientist, Aneah, Stationcall, ArthurBot, Xtremejames183, TheAMmollusc, Simsianity, Impakti, FreshBreak, Tedzdog, Pradameinhoff, SassoBot, SCARECROW, Seldridge99, Jcj04864, E0steven, Coffeerules9999, Citation bot 1, Farazv, FoxBot, Lotje, Vrenator, Dmuellen-berg, Reach Out to the Truth, RjwilmsiBot, Sundar.ciet, Ripchip Bot, Tremaster, Happyisenoough, Timtempleton, 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, Eduardofeld, W.D., Mdann52, Mrt3366, Yy.sujin, YFdyh-bot, Prelude after noon, Jags707, SimonWiseman, Codename Lisa, Patrick.bausemer, Pete Mahen, Vanischenu from public computers, Phamnhatkhanh, Csteinb, Jakec, Ahoora62, 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 and Anonymous: 301
- **Mobile security** Source: https://en.wikipedia.org/wiki/Mobile_security?oldid=691419590 Contributors: Dreamyshade, Edward, Ronz, Phil Boswell, Bearcat, Giraffedata, Ceyockey, Feezo, Mindmatrix, Mattisha, Tabletop, Rjwilmsi, Ahunt, Wavelength, Gilliam, Deli nk, Kwestin, Shritwod, Drogers.uk, Dawnseeker2000, Visik, Obiwankenobi, Magioladitis, JamesWatson, Cesarth~enwiki, Jesant13, Katharineamy, Bonadea, SylviaStanley, Flyer22 Reborn, Capitalismojo, Josang, Socrates2008, Boleyn, Addbot, Yobot, WikiDan61, AnomieBOT, Rubinbot, Mark Schierbecker, 78.26, 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, ChrisGaultieri, MikeD-NJITWILL, Dexbot, Rickbacl, Luggia2453, Securevoicecalling, Gtangil, Mbqt31, Tsarian, Dannyruthe, Monkbot, RicardoHeinz, Dsprc, Nitin93flanker, Alesteska, Ujjwal Sahay, Anarchyte, Marieswiss12, Cmason1234, Lisahamilton90, Muhammad mobeen83, Matthews david, Wikijagon and Anonymous: 44

- **Cybercrime** *Source:* <https://en.wikipedia.org/wiki/Cybercrime?oldid=689916576> *Contributors:* Damian Yerrick, Frecklefoot, Edward, D, Ixfd64, Sannse, Dori, Ihcoyc, Ronz, Jebba, Darkwind, Andrewa, Julesd, Andres, Kaihsu, GCarty, Ww, Greenrd, Zoicon5, Katana0182, Robbot, ZimZalaBim, Lowellian, Desmay, UtherSRG, Alan Lifting, Everyking, Edcolins, Utcursch, Antandrus, Jorm, Beland, Reagle, Joyous!, Ta bu shi da yu, DanielCD, Discospinster, Rich Farmbrough, ArnoldReinhold, Atchom, MarkS, Elwikipedista-enwiki, Narcisse, Cmdrjameson, Elipongo, WikiLeon, Vishnu vijay, Timmywimmy, ADM, Zachlipton, Alansohn, Arthena, Snowolf, Wtmitchell, L33th4x0rguy, TaintedMustard, Harej, RainbowOfLight, H2g2bob, BlastOBetter42, Y0u, Woohookitty, Mindmatrix, Wikikrlsc, Prashanthns, BD2412, Galwhaa, Josh Parris, Rjwilmsi, Bill37212, Bruce Lee, Bhadani, Amelio Vázquez, Rabreu, Nivix, Gurch, Tieno007-enwiki, Czar, Alphachimp, David91, Bgwhite, Wavelength, Phantomsteve, SpuriousQ, IanManka, Akamad, Stephenb, Markjx, NawlinWiki, Welsh, Renata3, Deku-shrub, FoolsWar, Lippard, Zzuuzz, Gtdp, Rurik, CWenger, Tom Morris, Sardanaphalus, Crystallina, SmackBot, Reedy, Stifle, Canthusus, Nil Einne, Gilliam, Skizzik, Jrkagan, Kurykh, JDCMAN, Dimonicquo, Silly rabbit, Octahedron80, Miharad, Tim Pierce, ConMan, Expugilist, Savidan, RolandR, FlyHigh, Prehistoricmaster2, Kuru, Ocee, Shadowlynk, Joffeloff, Kirkconnell, Barrycarlyon, Beetstra, Invisifan, Hu12, MikeWazowski, Iridescent, Kencf0618, CapitalR, Sim8183, Tawkerbot2, Dlohcierekim, CmdrObot, Ale jrb, JohnCD, Penbat, MrFish, Equendil, Anthonycole, DumbBOT, ErrantX, Heathniederee, Epbr123, Mojo Hand, Vertium, Esemono, The Legendary Ranger, Dzubint, I already forgot, AntiVandalBot, Oducado, QuiteUnusual, Paste, Joe Schmedley, Oddity-, Wayiran, Gilliantayloryoung, JAnDbot, Dustin gayler, Levitica, SiobhanHansa, VoABot II, Maheshkumaryadav, Joellee, Kiwimandy, Edper castro, DerHexer, JaGa, Mahnol, Cocytus, MartinBot, Lordmyx, Jeannealcid, Jim.henderson, Rhlitonjua, Psychoair, Keith D, Jerry teps, Bemsor, Nixonmahilum, Tgeairn, JonBurrows, Jmm6f488, Kemiv, Semaja, Reno911, Boxmoor, Neon white, NYCRuss, Vanillagorillas, Tokyogirl79, Turner70, HiLo48, DadaNeem, Olegwiki, Druss666uk, Ja 62, Funandtrvl, Metallicaguy007, VolkovBot, Philip Trueman, MissionInn.Jim, Technopat, Sparkzy, Helper, Jose figueredo, Sankalpravid, Qxz, Anna Lincoln, The3stars, Tpk5010, Snowbot, Jlh, Milan Keršlager, Billinghurst, Enigmaman, Falcon8765, Justmeherenow, Noncompliant one, Cool110110, DeanC81, Yintan, LeadSong-Dog, Flyer22 Reborn, Jojalozzo, Iestynpugh, Oxymoron83, Harry-enwiki, Techman224, Manway, Millstream3, AMbot, Mr. Stradivarius, Barry Jameson, Denisarona, Jons63, Elassint, ClueBot, Kai-Hendrik, Binksternet, The Thing That Should Not Be, Jotag14, Taroaldo, Tomas e, Chris.tripledot, CounterVandalismBot, Nicieguyedc, LizardJr8, Trivialist, PMDrive1061, Chaserx7, Canis Lupus, Rhododendrites, Imaximax1, Vivon1, Jmaio2, Aleksd, Light show, Agilentis, Thingg, PCHS-NJROTC, Callinus, Johnuniq, MBK-iPhone, BarretB, XLinkBot, Roxy the dog, GonzoNoir, Afpre, Charco2006, Bamford, Addbot, Some jerk on the Internet, Gpershing, MrOllie, Jgkjfdls-gkjd, Fatboy500, PranksterTurtle, Glane23, Debresser, Favonian, Jaydec, 5 albert square, Tide rolls, Bultro, Jarble, HerculeBot, Matt.T, Albeiror24, Jackelfive, Ben Ben, Kurtis, Publicly Visible, Luckas-bot, Yobot, Legobot II, II MusLiM HyBRiD II, Mdolphy, Kamikaze-Bot, JackCoke, Lessandmore, IW.HG, Irccpresident, Vörös, Backslash Forwardslash, AnomieBOT, DemocraticLuntz, Kerfuffle, Jim1138, IRP, Darkblazikenex2, NickK, Materialscientist, ArthurBot, Quebec99, Justwiki, Xqbot, JimVC3, Capricorn42, RoodyAlien, Mrc1028, Srich32977, Pradameinhoff, Wikieditor1988, Tankrider, Lior1075, Shadowjams, FrescoBot, Weyesr1, Yashansi, YOKOTA Kuniteru, Blockyeyes, Ka4, Buchana4, Dejan33, Sfafinski, Bobmack89x, Pinethicket, I dream of horses, Gajic32, Professional7, MJ94, Serols, Mentmic, Full-date unlinking bot, Merlion444, FoxBot, Lotje, Callanec, Vrenator, Aoidh, Reaper Eternal, ThinkEnemies, Reach Out to the Truth, Minimac, DARTH SIDIOUS 2, Fred11111111, RjwilmsiBot, VernoWhitney, Agent Smith (The Matrix), Beccritical, Emaus-Bot, John of Reading, Immunize, Sophie, Angrytoast, Katherine, Dewritech, Minimac's Clone, RenamedUser01302013, Tommy2010, Wikipelli, Dcirovic, Ida Shaw, Pragnesh89, Josve05a, Jenks24, Michael Essmeyer, Empty Buffer, Forgottenking, Bustermythmonger, En-eMsty12, Christina Silverman, Kjg0972, Erianna, Umni2, Donner60, Carmichael, Yulli67, ChispastonBot, Trickmind, PetrB, ClueBot NG, Mechanical digger, Sagaa2010, Gareth Griffith-Jones, Iiii I I I, Catlemur, 6ii9, Hirai NJITWILL, Widr, Leearo4, Helpful Pixie Bot, Aigendon, HMSSolent, Nightenbelle, Markthing Inc., Titodutta, KLBot2, BG19bot, VasundraTaneja, Jhanov1999, Ramesh Ramaiah, FxHVC, Najma El Shelhi, Frze, AvocatoBot, SusanBREN, Metricopolus, Mark Arsten, Lochfyneman, Dainomite, Harizoth9, MrBill3, Glacialfox, Klilidiplomas, Yash101, Aisteco, CrimeWeb, Fylbecatulous, Agent 78787, Darylgolden, Riley Huntley, Aristotle, Pratyya Ghosh, Cyberbot II, Padenton, Khazar2, Abowker, Bamachick20, HelicopterLlama, Lugia2453, Frosty, Metalytics, FrostieFrost, Mason Doering, PinkAmpersand, Greengreengreenred, Dddge, LectriceDuSoir, Reziebear, Glaisher, Bullblade, EdynBliss, Ginsuloft, Quenhitran, Cindy123456, Jnguyenx3, Keatonhouse, M3osol1301, Fixture, JaconaFrere, Skr15081997, KacyConnor14, Lordangel101, Altaythegooner, AKS.9955, Cybersecurity101, Pinklights2323, S166865h, StaceyHutter, JohnC123456, Willhesucceed, Vanyaxd, Juli-edeltafima, Hellys320, Destor918, Lymaniffy, Rishab Elangovan, Guegreen, FormerPatchEditor, DarthWyyrlok, Notnot9, Erosen15, Drdebaratiwiki, KasparBot, Dmonshaugen, Airplane Maniac, DebaratiH, ScottAvalon, Zackie1990, Priyadarshivishal23, Josh9791, Rahul Tibile, Mkm8dy, Tejalpatel and Anonymous: 680

- **Vulnerability (computing)** *Source:* [https://en.wikipedia.org/wiki/Vulnerability_\(computing\)?oldid=689403515](https://en.wikipedia.org/wiki/Vulnerability_(computing)?oldid=689403515) *Contributors:* Kku, CesarB, Ronz, Joy, Eugene van der Pijll, Phil Boswell, ZimZalaBim, Waldo, Sdfisher, Jason Quinn, Wmahan, Utcursch, Beland, White-Dragon, Quarl, FrozenUmbrella, Mozerati, Discospinster, Xezbeth, Mani1, Adequate-enwiki, InShanee, Velleda, Mindmatrix, Ahouseholder, Ruud Koot, Macaddct1984, Mandarax, Tslocom, BD2412, Ketiltrot, Rjwilmsi, Jweiss11, ElKevbo, Naraht, Brownh2o, Chobot, YurikBot, Gardar Rurak, Gaius Cornelius, Irishguy, Gruffi-enwiki, Perry Middlemiss, Mugunth Kumar, Abune, SmackBot, Mmernex, AnOddName, Gilliam, PJTraill, Chris the speller, Persian Poet Gal, Manuc66-enwiki, JonHarder, Solarapex, Chris palmer, Mistress Selina Kyle, FlyHigh, Lambiam, Derek farn, Xandi, Beetstra, Ehheh, Nevuer, Dreftymac, JoeBot, Jbolden1517, Penbat, Vanished user fj0390923rok4t1km2pkd, Thjs1bot, EdJohnston, Obiwankenobi, Dman727, Eleschinski2000, S.C.F, Esanchez7587, CliffC, Fleetflame, Ash, Jesant13, Anant k, Sarveshbathija, Touisiau, Jramsey, Tanjstafl, TXiKiBoT, Softtest123, Zhenqinli, Michaeldsuarez, Haseo9999, Swwiki, LittleBenW, Sassy410, JuTiLiu, Securityphreaks, Phe-bot, Cenzic, Jojalozzo, Jruderman, Ottawahitech, Dcampbell30, Liquifried, WalterGR, DanielPharos, PotentialDanger, Sensiblekid, Fathisules, Addbot, Larry Yuma, SpBot, Tide rolls, Luckas-bot, BaldPark, Yobot, Djptechie, Sweerek, AnomieBOT, MistyHora, Blueraspberry, ArthurBot, The Evil IP address, RibotBOT, Pradameinhoff, Bentisa, Erik9, FrescoBot, Kitaura, HamburgerRadio, Pinethicket, LittleWink, Guriaz, Tool789789, Dtang2, Lotje, DARTH SIDIOUS 2, VernoWhitney, EmausBot, John of Reading, T3dkjn89q00v102Cxplqks3x7, Timtempleton, Pastore Italy, ClueBot NG, Ptrb, Shajure, Emilyisdistinct, J23450N, AvocatoBot, Exercisephys, Mrebe1983, Mdann52, Mrt3366, Mediran, Codename Lisa, Mogism, Pharrel101, Wieldthespade, Krazy alice, OccultZone, Pat power11, Monkbot, S166865h, Balancesheet, Greenmow, Gustasoz and Anonymous: 105

- **Eavesdropping** *Source:* <https://en.wikipedia.org/wiki/Eavesdropping?oldid=681073187> *Contributors:* Magnus Manske, Waveguy, Kku, Menchi, TakuyaMurata, Skysmith, Docu, Kingturtle, Kaihsu, John K, Lee M, DJ Clayworth, Kieran, Joy, Khym Chanur, Securiger, Postdlf, Wereon, Zigger, Abu el mot-enwiki, Jh51681, Talkstosocks, Longhair, Jag123, Pearle, Alansohn, Gargaj, AndreasPraefcke, Man-gojuice, Stefanomione, BD2412, FlaBot, Ground Zero, YurikBot, Mikeblas, Occono, EEMIV, Elkman, Zzuuzz, Nikkimaria, Ekeb, Rearden9, Junglecat, SmackBot, Betacommand, Rmosler2100, Chris the speller, Bluebot, Shalom Yechiel, Ortzinator, Cybergobra, Givevez,

Joystick74, Tazmaniacs, Loodog, Ckatz, 16@r, Colonel Warden, Lord E, McQuack, JForget, CBM, HalJor, Cydebot, Playtime, Clayquot, SimonDeDancer, MafiaCapo, M. B., Jr., Luigifan, NigelR, Fayenatic london, Jaysweet, Escorial82, MartinBot, Morki, Keith D, LedgendGamer, Pharaoh of the Wizards, Jeepday, Flatscan, KimiSan, Sağlamci, Melsaran, SieBot, Dreamafter, Flyer22 Reborn, Kjoto, Pinkadelica, Denisarona, Sandy of the CSARs, Martarius, ClueBot, Stevehs, DionysosProteus, Mike Klaassen, Shaliya waya, BOTarate, Vigilius, Mitch Ames, Kbdankbot, Cewvero, Addbot, RFBugExpert, Ielvis, OmgDALE, SamatBot, Herr Gruber, Luckas Blade, Luckasbot, Yobot, Citation bot, Xqbot, Mononomic, Charles D. Ward, GrouchoBot, SassoBot, Thehelpfulbot, AlexanderKaras, Citation bot 1, Bsandoval, Lotje, MrJackCole, Werieth, Youngeuropean, Cobaltcigs, ClueBot NG, Helpful Pixie Bot, Mark Arsten, Stevenbeaupre, Es-sam4002, Frosty, Anonz 8431, Flat Out, DavidLeighEllis, Dodi 8238, Abracombie, ChrisHF, Scarlettail, Suctioninfo2, Amortias, Mathjam, Bumbum01 and Anonymous: 131

- **Exploit (computer security)** *Source:* [https://en.wikipedia.org/wiki/Exploit_\(computer_security\)?oldid=687578202](https://en.wikipedia.org/wiki/Exploit_(computer_security)?oldid=687578202) *Contributors:* Axel-Boldt, Mav, Aldie, SimonP, Stevertigo, Michael Hardy, TakuyaMurata, Karada, Ronz, Nikai, Smaffy, RI, Enigmasoldier, Altenmann, Pengo, Alerante, SWAdair, Utcursch, Bluefoxicy, Discospinster, Rich Farmbrough, Pie4all88, Syp, El C, Matthe, Bobo192, La goutte de pluie, Ramsey, Walter Görlitz, Adequate-enwiki, Ringbang, Nuno Tavares, Mindmatrix, Georgia guy, Apokrif, Vargc0, Mindfuq, RainR, FlaBot, Ground Zero, Latka, Arunkoshy, Chobot, KDK, YurikBot, Hydrargyrum, Stephenb, Pseudomonas, Dpakoha, Irishguy, Ugnius, Zwobot, Yudiweb, Raistolo, Papergrl, SmackBot, Pgk, Bomac, BiT, Jerome Charles Potts, Abaddon314159, JonHarder, Sloverlord, Nakon, Tompsc, Pilotguy, Lambiam, Putnamehere3145, LebanonChild, Ehheh, Dreftymac, SkyWalker, Fabio-cots, Skittleys, Omicronpersei8, Ebrahim, Dreaded Walrus, PC Master, Zorro CX, Ghostwo, SpigotMap, Crakkpot, TXiKiBoT, Wolfrock, Jamespolco, Irsdl, Swwiki, PeterCanthropus, PabloStraub, ClueBot, Excirial, SchreiberBike, DanielPharos, Fathisules, SkyLined, GD 6041, Legobot, Luckasbot, Amirobot, Nallimbot, Galoubet, ExploitSolutions, ArthurBot, Sionus, Boyrussia, Waterloox, Weltersmith, Pradameinhoff, Erik9, Erik9bot, HamburgerRadio, Guriue, Guriaz, PleaseStand, EmausBot, WikitanvirBot, Dewritech, ZéroBot, IGeMiNix, Pastore Italy, ClueBot NG, Neynt, BG19bot, Who.was.phone, Compfreak7, T2kien, Kelly McDaniel, W. P. Uzer, Fixture, Shellcode 64, Favone, In Harry Potter We Trust, TragicEnergy, FoxStudios, Pktuzov314, Potayo, S166865h and Anonymous: 137
- **Trojan horse (computing)** *Source:* [https://en.wikipedia.org/wiki/Trojan_horse_\(computing\)?oldid=685971672](https://en.wikipedia.org/wiki/Trojan_horse_(computing)?oldid=685971672) *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, Alexandre.ip, Cmdrjameson, Chirag, DCEdwards1966, Haham hanuka, Jjon, 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, Easys12c, Optichan, Gyrae, Mekong Bluesman, Graham87, Jclemens, Enzo Aquarius, Rjwilmsi, JoshuacUK, Blacktoxic, NeonMerlin, ElKevbo, Aapo Laitinen, AySz88, Andrzej P. Wozniak, RainR, RobertG, JFisF, Bubbleboys, Ewlyahooocom, AlexjohnC3, TheDJ, DevastatorIIC, Ben-w, Gr8dude, M7bot, Ahunt, Chobot, DVdm, Roboto de Ajvol, Angus Lepper, Sceptre, Ytg111, 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, Deldot, Kelley-Cook, Jpvinal, Arsenaldc1988, Yamaguchi¶, Gilliam, Ohnoitsjamie, Spahnress, Snori, Tree Biting Conspiracy, Miquonranger03, Gareth, LaggedOnUser, Lexlex, DHN-bot~enwiki, Jeffreyarcand, Abaddon314159, Can't sleep, clown will eat me, MyNameIsVlad, Frap, Christian80, KaiserBot, Rrburke, TKD, Emre D., Nibuod, Sjaxon, 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, IvanLanin, JoeE, Blehfu, Courcelles, Linkspamremover, Astral9, Mzub, ChrisCork, Switchercat, SkyWalker, JForget, DJPhazer, CmdrObot, Wafulz, Makeemlighter, ParadoX, CWY2190, Rikva, Lishy Guy, Jesse Viviano, INVERTED, Neelix, Funnyfarmofdoom, Eqwendil, Slazenger, MC10, Red Director, SnootyClaus, Strom, Mr. XYZ, Shirulashem, UnDeRsCoRe, Rudá Almeida, Omicronpersei8, Rocket000, Thijs!bot, Epbr123, Blademaster313, N5iIn, 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, Freedom-linux, VoABot II, Nyq, Jrg7891, SineWave, GODHack~enwiki, Indon, Caill, Esanchez7587, Shuini, DidierStevens, Charitwo, Gwern, Atulsnischal, MartinBot, Axlq, Jonathan Hall, R'n'B, JohnNapier, J.delanoy, Patsanks06, Legoboy2000, Catmoongirl, Digideman, McCajor, McSly, RichJizz123, Demizh, Evils Dark, Gurchzilla, AntiSpamBot, Dividing, LeighsOptimvsMaximvs, Shoesss, Cue the Strings, Andrewcmcardle, Darryl L James, Bonadea, Martial75, Ditre, Anapologetos, ThePointblank, CardinalDan, Burlywood, Deor, VolkovBot, ABF, Jeff G., Sulcege, Rtrace, VasilevVV, Jacroe, Ryan032, Philip Truman, 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, Firefoxbession, Ramesseum, Softpile, Copana2002, SieBot, Teh nubkilr, BotMultichill, Krawi, Josh the Nerd, Caltas, Eagleal, RJaguar3, X-Fi6, Chiroz, Sephiroth storm, Johnnnyeagleisrocker, Happyssailor, Flyer22 Reborn, Caidh, Oxymoron83, Kosack, Hobartimus, Drsango, 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, Zarkthehackeralliance, Mild Bill Hiccup, Piriczki, Infogaufire, CounterVandalismBot, Dandog77, Aabrol19, Dennistang2007, Gunnar Kreitz, Somno, Aua, Excirial, Jusdafax, PixelBot, Eekster, Bde1982, Rhododendrites, Mac1202, Lunchscale, WalterGR, Doctor It, Jaizovic, Daniel-Pharos, JaneGrey, Taranet, VIKIPEDIA IS AN ANUS!, 7, Ranjithsutari, Berean Hunter, Egmontaz, Alchemist Jack, Polemos~enwiki, XLinkBot, Spitfire, NiveusLuna, Jovianeye, Feinoha, TFOWR, ErkinBatu, Mifter, Alexius08, Noctibus, Addbot, Some jerk on the Internet, Landon1980, A.qarta, Friginator, Markyman12, Ronjhones, 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, Fraggie81, Cflm001, Xxpivjtxxx, 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, Mi-pearc, S0aasd2sf, GrouchoBot, Megamonkeyextreme, RibotBOT, SassoBot, TrueGlue, Amaury, JulianDelphiki, Shadowjams, Schnitzel-MannGreek, 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, Pcusser42, GWPSP090, Ksanexx, DixonDBot, Lamarmote, Miiszmylove, MichaelRivers, Vrena-

tor, 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, L235, Tommy2010, Wikipelli, TheGeomaster, Skaera, Ida Shaw, Dalek32, Traxs7, Eldruin, 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, Complol2234343, Robbieee2, Wiki13, MusiAnimal, AvocatoBot, Desenagator, Mark Arsten, Sbd01, Onewhohelps, 1ravensnffan, Snow Blizzard, MrBill3, Glacialfox, Kelvinruttmann, Eduardofeld, Tuteylar, Niraj.adyyyy, Th4n3r, Hsr.rautela, Adhithyan15, Cyberbot II, ChrisGualtieri, MadGuy7023, JayMyers-NJITWILL, Ghostman1947, Rezonansowy, SoledadKabocha, Djairhorn, Lugia2453, JoshLyman2012, Jc86035, Siravneetsingh, Soda drinker, Discuss-Dubious, Sourov0000, Cablewoman, Bugzeecolboy, NimaBoscarino, RootSword, Dave Braunschweig, Epicgenius, CatBallSack, Eyesnore, Gaman0091, DavidLeighEllis, Ugog Nizdast, Khabir123, Kushay titanium, Someone not using his real name, Manish2911, Oranjelo100, Dannyruthe, Sathishguru, STH235SilverLover, Joseph 0515, Marp pro, Rkpayne, Monkbot, Sidharta.mallick, Filedelinkerbot, Abcdfeghtys, Laura J. Pyle, Biblioworm, TerryAlex, Classofthewise, Earthquake58, ChamithN, HamadPervaiz, Helpguy77, TQuentin, KasparBot, James the king12, JeremiahY, TeacherWikipedia, OldMcDonald12345, Drakeblair08, Kurousagi, Majneeds2chill, Aditya3929, User0071987, Bradley!90432754, Pa19602030fu and Anonymous: 1169

- **Computer virus** *Source:* https://en.wikipedia.org/wiki/Computer_virus?oldid=690396415 *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 Obfuscous, 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, Smatty, Rob Hooft, Samuel~enwiki, GRAHAMUK, Hashar, Dblaidsell, Adam Bishop, Dcoetzee, RickK, Dysprosia, Doradus, WhisperToMe, Zoicon5, Tp-bradbury, 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, Mat-tflaschen, Cordell, Carnildo, Tobias Bergemann, 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, Freakofnurtture, Monkeyman, Imroy, Jiy, Discospinster, Solitude, Rich Farmbrough, Agnitus, Rhobite, Qwerty1234, EliasAlucard, Lemontea, Stereotek, Bender235, Rubicon, ESkog, ZeroOne, TerraFrost, CMC, JoeSmack, MisterSheik, CanisRufus, Ginnns, 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, MattGiua, Pol098, MONGO, Miss Madeline, Nfearnley, Jok2000, Wayward, Prashanthns, Dysepson, Seishirou Sakurazuka, Graham87, Marcg106, Magister Mathematicae, Cuchullain, CivilCasualty, Kbdkank71, FreplySpang, JIP, Jclemens, Reisio, Grammarbot, Zoz, Spot Color Process, Ketiltrout, Sjakkalle, Rjwilmsi, Whatcanbwndo, Xgamer4, Koavf, Isaac Rabinovitch, Tarnas, Kinu, Adjusting, Collins.mc, Commander, Astronaut, Rillian, Tangotango, Bruce1ee, Authr, Raffaele Megabyte, Captain Disdain, ErikHaugen, Gudeldar, Frenchman113, Darkasasami, Bubba73, Jdmurray, GregAsche, Sango123, DirkvdM, Antimatt, Leithp, Audunv, Munahaf, RainR, FlaBot, RobertG, Winhunter, Crazycomputers, JiFish, Nivix, RexNL, Gurch, Alexjohn3c, 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, Bullseye, 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.Nevilsteen, Zzuuzz, Imaninjapirate, Theda, Closedmouth, Spondoolicks, Dspradau, BorgQueen, Petri Krohn, GraemeL, Rlove, JoanneB, Mario23, Pursin1, Chrishmt0423, SingDeep, LeonardoRob0t, Rex Nebular, Scoutersig, Amren, Kevin, AVazquezR, Splifly, Curpsbot-unicodify, RunOrDie, RG2, Eptin, Tyomitch, GrinBot~enwiki, DVD R W, Theroachman, Rahul s55, 机器人 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, Oscarcecat, 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, Manhattan 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, Eddetic, Wtwilson3, Slowmover, Sfivaz, Sir Nicholas de Mimsy-Porpington, Minna Sora no Shita, CaptainVindaloo, Aleenf1, IronGargoyle, 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, Tawkerbot2, 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, Enevile, 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, Dddgggg, Brainbox 112, Renamed user 5197261az5af96as6aa, Dfrg.msc, Philippe, Big Bird, Natalie Erin, Oldmanbiker, Dainis, Dzubint, Danthemana531, Men-tifisto, Porqin, AntiVandalBot, Fedayee, Luna Santin, Widefox, Ibigewald leaven, Wildboy211, Oducado, Quintote, Jayron32, Qa Plar, TimVickers, Scopia, 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, 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, Alu042, Leliro19, 0612, Murray paul, 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, Paronomia, J.delanoy, MRFraga, Legitimate Editor, Carre, Rgoodermote, Bitethesilverbullet, Inspi-geon, 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, Draugo, TomasBat, NewEnglandYankee, Srpnor, Dougmarlowe, 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, NoticeBored, BuickCenturyDriver, Technopat, Hqb, Yuma en, Miranda, Crazypete101, Lord Vaedeon, JayC, Qxz, Shindo9Hikaru, Mattinsson, WraithTDK, Rfcrosssp, Anna Lincoln, Codyjames7, Lradrama, Sanjivdinakar, Xaraikex, Tricky Wiki44, JhsBot, Phillyfan1111, Mind23, LeaveSleaves, Tpk5010, Peter Dontchev, Miketsa, Air-con noble, David in DC, Csdorman, Da31989, Slipknotmetal, Madhero88, Lolroflomg, Jacky15, Aryeh Grosskopf, Lerdthenerd, Haseo9999, Tikuko, Gorank4, Falcon8765, Purgatory Fubar, LarsBK, Brianga, Mike4ty4, Bobo The Ninja, LittleBenW, AlleborgoBot, Kehrbykid, NHRHS2010, Kpa4941, Wraithdart, ChipChamp, Repy, Sepetro, Copana2002, Anindianblogger, SieBot, Coffee, J800eb, Dino911, YonaBot, Eryalus, WereSpielChequers, Dawn Bard, Eagleal, BloodDoll, Triwe, 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, Maelgwn-bot, Vice regent, MichaelIvan, Mminoche, 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, Cenarius, WalterGR, TheRedPenOfDoom, Trxman307, Hans Adler, Frozen4322, Schreiber-Bike, Arvind007, ChrisHodgesUK, BOTarate, El bot de la dieta, DanielPharos, Thingg, Aitias, Scalhotrod, Jester5x5, Versus22, PCHS-NIROTC, MelonBot, Qwfp, Johnuniq, Jpg1954, Egmontaz, Apparition11, Editor2020, SF007, Runefrost, DumZiBoT, Bones000sw, Gkaukonen, TheNameWithNoMan, XLinkBot, Stickee, Rrror, Feinoha, Poosebag, Avoided, Mitch Ames, Skarebo, WikHead, Silvonent-Bot, 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 Jaficuk, 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, Washburnmov, Taxisfolder, THEN WHO WAS PHONE?, Golftheimer, KamikazeBot, Theornamentalist, Kablakang, 花魁の花魁, Bility, Kmoultrey, AnomieBOT, AmritasyPutra, Valueyou, La Corona, Rubinbot, 1exec1, ThaddeusB, Arzanish, Jim1138, Galoubet, Dwayne, Danielt998, AdjustShift, Kingpin13, Materialscientist, Racconish, Cameron Scott, Xqbot, Spidern, Cureden, What?!?Why?Who?, Gilo1969, S0aaasdf2sf, Coretheapple, Sci-Fi Dude, Basvb, Frosted14, Shas1194, Ute in DC, Wizardist, ProtectionTaggingBot, Shirik, Mark Schierbecker, ReformatMe, Amaury, Drvikaschahal, Int2, Aurush kazemini, N419BH, Mrbillgates, Lelapindore, Erik9, Sesu Prime, Legobot III, FreeKnowledgeCreator, Komitsuki, Kitaure, Jono20201, Zero Thrust, Scott A Herbert, HamburgerRadio, Intelligentium, Pinethicket, I dream of horses, Elockid, HRoestBot, Calmer Waters, Bejinhan, RedBot, Phearnson, Serols, HaiyaTheWin, Footwarrior, Lineslarge, Reconsider the static, IIBall, Caramelldansener, 3centosoap, Xeroxli, FoxBot, Mercy11, Trappist the monk, LogAntiLog, Lotje, Callanec, Vrenator, Singlemaltscotch, Zacker150, Extra999, January, Rudy16, Chimpso, Jeffrd10, Nazizombies!, MuFc ftw, Canuckian89, Suffusion of Yellow, Richardsaugst, Tbhotch, TheMesquito, DARTH SIDIOUS 2, Onel5969, Mean as custard, The Utahraptor, RjwilmsiBot, Bento00, Fletcher707, Regancy42, DRAGON BOOSTER, Chroniccommand, Ghymes21, NerdyScience-Dude, Deagle AP, DASHBot, Deadlyops, EmausBot, Immunize, Never give in, Ajraddatz, Sevvie, Zerotonin, Dewritech, Jujitsuthird-dan, Ben10Joshua, Olof nord, RenamedUser01302013, Vanished user zq46pw21, Elee, Huctwitha, Tommy2010, Wikipelli, Soadfan112, K6ka, Kurenai96, Papelbon467, Elison2007, Rajnish357, AvicBot, ZéroBot, Rhinstone42, Checkingfax, Fae, Josve05a, 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, ChuipastonBot, Pastore Italy, Dylan Flaherty, Czeror, SlowPhoton, Neil P. Quinn, Brad117, Mikitei, DASHBotAV, Fungo4shezzo, Myfunkypbear123456789, ClueBot NG, Horoporo, Matthiaspaul, This lousy T-shirt, Baseball Watcher, Mesoderm, Rezabot, Massly, Widr, Anupmehra, Helpful Pixie Bot, Alexbee2, Lowercase signabot, BG19bot, Janendra, Kaltenmeyer, PatrickCarbone, AvocatoBot, Jobin RV, Tobias B. Besemer, Eman2129, Tony Tan, Agent190, Napsync, BattyBot, Biosthmors, Justincheng 12345-bot, Smileguy91, Cyberbot II, Codeh, Arcandam, Orioncaspar, Dexbot, Cwoobe, Codename Lisa, Makecat-bot, Lugia2453, Frosty, Arrayoutofbounds, Andyhowlett, Athomeinkobe, Cdwn, Cadillac000, Palmbeachguy, Epicgenius, Jamesmcmahon0, Ayush 691, Melonkelon, EvergreenFir, Ronburgandee, EHef, Origamite, Babitaarora, Sam Sailor, Dannyruthe, Arfaest Ealdwritere, Digiwyse, JaconaFrere, Lakun.patra, 32RB17, Monkbot, Happy Attack Dog, Thisiswikidome, Pantel, Shuaibshaikh84, Virus victim, Amortias, Shahean Cozad, NQ, The Last Arietta, Edityouranswer11, ChamithN, Crystallizedcarbon, WikiGopi, Vivek56, Charlotte Churche, Abistevenson22, Tech Aid, Pishcal, Juned kadri, IEditEncyclopedia, Jalen stephens, Wredants, Selim Abou Rahal, Deanwalt123, Hamza190949, Miniredeyes999, GeneralizationsAreBad, Ilikecakeandstuff, Bankofworld, KasparBot, Seanpatrickgray, Astolfo0, Gautamnarayan, MCFinest Anthony, Mac1817;-), CJwar, Muttaiyab Ahmad, TheGlatiator, DelugeRPG, PC-MarcLondon, Ameya Kanojia, Flashgernade99, Imdnj, Matthews david and Anonymous: 1952

- Virus hoax** Source: https://en.wikipedia.org/wiki/Virus_hoax?oldid=689898362 Contributors: Ed Poor, Smjg, Darksun, CALR, Rich Farmbrough, JoeSmack, Bobo192, Uncle G, Pol098, SchuminWeb, Ahunt, CiaPan, Aussie Evil, GunnarRene, NawlinWiki, Apokryltaros, Vlad, Midnightcoffee, Destin, SmackBot, Mmernex, Melchoir, Unforgettableid, RocketMaster, Mark alfred, Blueboar, Salamurai, IronGargoyle, JHunterJ, CmdrObot, NaBUru38, Cydebot, Nabokov, Alaibot, Kirrages, MartinBot, J.delanoy, JJGD, Dampinograaf, Jacroe, Croxley, Andrewaskew, Jerryobject, SimonTrew, RedBlade7, ClueBot, TheAmigo42, Trivialist, Rprpr, Manishearth, Xme, Daniel-Pharos, Oxz0rz, Tonyngkh, Addbot, Pejsch, Toyokuni3, This is Paul, Ipse6, Baffle gab1978, Comphelper12, Bro4, Fryderyk, MrBlueSky, KamikazeBot, Flewis, JoolzWiki, Erik9, HamburgerRadio, AntiKrisT, Peace and Passion, Ajraddatz, Kwgagel, Champion, Pastore Italy, ClueBot NG, Hz.tiang, Fernandoso, MrBill3, Codename Lisa, JRArocks, Monkbot, Crystallizedcarbon and Anonymous: 100
- Computer worm** Source: https://en.wikipedia.org/wiki/Computer_worm?oldid=691110355 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, Tobias Bergemann, David Gerard, Alerante, Fennec, Akadruid, Jtg, Noone~enwiki, Eequor, Fanf, Matt Crypto, Just Another Dan, Maximimax, 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, Stacieou, Bsadowski1, Pauli133, Bobrayner, Newnoise~enwiki, Roboshed, Woohookity, Mindmatrix, Camw, Guy M, TomTheHand, Isnow, Kralizec!, Palica, RichardWeiss, Jclemens, Rjwilmsi, Matt.whitby, Syndicate, Mcmvanbree, Nguyen Thanh Quang, RainR, Jwkpiano1, Dan Guan, JiFish, RexNL, Ewlyahocom, 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, KelleyleCook, 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, Makeemilighter, GHe, Jesse Viviano, Augrust, Oden, Slazenger, Gogo Dodo, ST47, Luckyherb, HitroMilanes, Thijs!bot, Epbr123, Luigifan, Powellflatlaw, 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, Pikolas, 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, BigDunc, RandomXYZb, MDfoo, Falcon8765, Envirobot, Burntsauce, EJF, Barkeep, SieBot, BotMultichill, Itsme2000, DarkfireInferno, Sephiroth storm, Sat84, Happyssailor, Mszegedy, Very cheap, Smaug123, Hello71, Minneapolis, Macy, OKBot, Amrishdubey2005, StaticGull, Mygerardromance, Hamilton-daniel, GioCM, Denisaron, Cellorelio, Minimosher, ClueBot, Traveler100, The Thing That Should Not Be, Lawrence Cohen, Fenwayguy, CrazyChemGuy, Eeekster, Rhododendrites, WalterGR, Dekisugi, DanielPharos, Thingg, Atilias, VIKIPEDIA IS AN ANUS!, XXXSuper-SnakeXXX, 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, Craigjones, 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, Intelligentium, 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, JDDJS, Fæ, Kalin.KOZHUHAROV, A930913, Tolly4bolly, W163, MonoAV, DennisIsMe, ChuispastonBot, Ziyad en, ClueBot NG, Henry Stanley, Borkificator, O.Koslowski, Widr, Helpful Pixie Bot, TheTrainEnthusiast, Tobias B. Besemer, Toccata quarta, Mantovanfabiomarco, Glacialfox, Derschueler, Anbu121, BattyBot, Johnthehero, Cyberbot II, ChrisGualtieri, EagerToddler39, Dexbot, Lal Thangzom, Codename Lisa, Webclient101, Djairhorn, Lugia2453, Jamesx12345, Rossumund, Muhammadbabarzaman, Smileyss, Ginsuloft, Dannyruthe, JaconaFrere, Satyajeet vit, KH-1, KasparBot, Gautamnarayan, CaseyMillerWiki, Fakersenpaipls, Manav garg, Jumblebumble12 and Anonymous: 508

- **Denial-of-service attack** *Source:* https://en.wikipedia.org/wiki/Denial-of-service_attack?oldid=691428540 *Contributors:* Magnus Manske, Derek Ross, Mav, The Anome, RobLa, Mark, Rjstott, Ed Poor, Yooden, PierreAbbat, Edward, Oystein, Michael Hardy, Mdupont, Ixfd64, Delirium, Looix~enwiki, Ahoerstemeier, Baylink, Jebba, DropDeadGorgias, Julesd, Marco Krohn, Cyan, Evercat, Ghewill, Agtx, Charles Matthews, Timwi, Pablo Mayrgundter, Kbk, Jwrosenzweig, Fuzheado, WhisperToMe, Pedant17, Maximus Rex, Furrykef, Taxman, Tempshill, Vedge, Thue, Bloodshedder, Mtcv, Olathe, Lothar Kimmeringer~enwiki, Carlossuarez46, Robbot, Fredrik, Korath, R3m0t, RedWolf, Nurg, Rfc1394, Wikibot, Victor, Lupo, Alanyst, Pengo, Tobias Bergemann, David Gerard, Ancheta Wis, Alerante, Centrx, Ryanrs, Wolfkeeper, HangingCurve, No Guru, Niteowlneils, Carlo.Ierna, Mboverload, Gracefool, Cloud200, Rchandra, Mckaysalisbury, SonicAD, Wmahan, Chowbok, Gdm, Antandrus, Mako098765, Wikimol, Rdsmith4, Gauss, Kigoe, Sam Hocevar, SamSim, Neutrality, Bluefoxicity, Stereo, Danc, Millisis, Wesha, Johan Elisson, Discospinster, Rich Farmbrough, Oliver Lineham, Mani1, Bender235, Andrejj, Violetriga, CanisRufus, Goto, Crukis, Smalljim, Mpeg4codec, Thewrite1, Giraffedata, Deryck Chan, Physicistjedi, Wrs1864, Krellis, Pearle, Alansohn, Godzig, Csabo, CyberSkull, Corporal, Andrewpmk, Echuck215, Lightdarkness, Ciaran H, Sligocki, Gblaz, Hu, Snowolf, Wtmitchell, Velella, BanyanTree, Xee, Bsdlogical, Redvers, Ceyockey, Walshga, Kenyon, Gmaxwell, Woohookity, Mindmatrix, Blaceagle, Peng~enwiki, Pol098, WadeSimMiser, Ocker3, Tckma, Scootey, Rchamberlain, Wayward, Prashanthns, Burfdl, Graham87, Deltabeignet, Clapaucius, Kakaopor, Kbdank71, Kane5187, Josh Parris, Ketiltrot, Sjö, Rjwilmsi, Friejose, Nneonneo, EIKevbo, Goncalopp, Everton137, Scartol, Bdhan, Andrzej P. Wozniak, Drnngrvy, Alliekins619, Wragge, FlaBot, VKokielov, Vegardw, Space-man85, Who, Nivix, RexNL, Revolving Bugbear, RobyWayne, Intgr, Alphachimp, Malhonen, GreyCat, Chobot, Bwoodring, DVdm, Gwernol, Kakurady, Mann Ltd~enwiki, Shaggyjacobs, YurikBot, Splintercellguy, Phantomsteve, RussBot, Sparky132, RedPen, Gardar Rurak, Hellbus, מילטון ר' גראנט, Gaius Cornelius, Shaddack, Rsrikanth05, The Cute Philosopher, NawlinWiki, Robertvan1, RattleMan, Escheffel, DavidH, Janarius, Gareth Jones, Catamorphism, Expensivehat, Irishguy, Anetode, Peter Delmonte, Rmky87, Mikeblas, Juanpd, Bucketsofg, Nethgirb, BOT-Superzero cool, DeadEyeArrow, Darkfred, Yudiweb, Black Falcon, Romal, Sierra 1, WAS 4.250, Frankgerlach~enwiki, Delirium of disorder, Wikibert~enwiki, Msuzen, Raistolo, Closedmouth, Arthur Rubin, Fang Aili, Abune, Josh3580, Matt Casey, Sean Whitton, GraemeL, JoanneB, Vicarious, RenamedUser jaskldjslak904, Huds, ViperSnake151, SoulSkorpion, GrinBot~enwiki, One, SmackBot, Fireman biff, Unschoold, McGeddon, Raven 1959, Freekee, KVDP, Eskimbot, Edgar181, David Fuchs, Buck O'Nolle, Eiler7, Yamaguchi 齋藤, Aldor, Gilliam, Brianski, Gorman, Chris the speller, Fintler, Apankrat, Jcc1, Persian Poet Gal, Thumperward, Snori, Tree Biting Conspiracy, Elecnix, Nbarth, Sunholm, Emurphy42, Zsinj, Frap, Oscar Bravo, Kaishen~enwiki, OrphanBot, JonHarder, Wiredsoul, Rrburke, Midnighthcomm, Artephius, XtAzY, Joemadeus, Shadow1, PPBlais, Paroxysm, SaintedLegion, Adcx, ElizabethFong, Vinaiwbot~enwiki, Mdavids, Rosarinjroy, Nishkid64, Kuru, Shadowlynk, Mgiganteus1, Berrick, Slakr, Roff, Dicklyon, Larrymc, Uuuuhul, Romeu, Tuspm, NeoDeGenero, Manifestation, Vashtihorvat, Kvng, Xionbox, Hu12, Gevron, White Ash, JoeBot, Wjejskenewr, Andrew Hampe, Blehfu, Courcelles, Tawkerbot2, SkyWalker, Ddcc, CmdrObot, Boborok, Alan Taylor, KyraVixen, N3X15, TheMightyOrb, GHe, Lemmio, WeggeBot, Pgr94, MeekMark, T23c, Dan Fuhy, Equendil, Cydebot, W.F.Galway, Yehaah~enwiki, Galassi, MC10, Gogo Dodo, Carl Turner, Daenney, Tawkerbot4, Christian75, Pchachten, Zzsql, Ryanmshea, Omicronpersei8, Click23, Thijs!bot, Edman274, Kubanczyk, Dschradner, Ultimus, Mbells, Loudsox, RLE64, Marek69, Woody, Bgaurav, JustAGal, Soma mishra, CTZMSC3, Escarbot, I already forgot, KrakatoaKatie, AntiVandalBot, Yonatan, Widefox, Darklord 205, Oducado, Ross cameron, QuiteUnusual, KP Botany, BenTremblay, DarkAudit, Cinnamon42, Davken1102, Oddity-, Mightywayne, Asmeurer, Ivan Velikii (2006-2008), MER-C, CosineKitty, Gmd1722, Blackmo, Livefastdieold, Hut 8.5, A.hawrylyshen, Geniac, Orpenn, Io Katai, Magioladitis, Ariel@compucall, Trigguh, Bong-warrior, VoABot II, Bradcis, Sarahj2107, Nyttend, Brusegadi, Fedvela, Bobkeyes, Web-Crawling Stickler, MetsBot, DarkMrFrank,

A3nm, DerHexer, Shuini, Gwern, Stephenchou0722, Mmoneypenny, CliffC, Jeannealcid, Rhlitonjua, Bemsor, Krimanilo, Joelee.org, RockMFR, J.delanoy, Skiidoo, Herbythyme, StewE17, Uncle Dick, Jesant13, Eliz81, Duzzyman, Cadence-, Andy5421, Ncmvocalist, Clerks, Crakkpot, Zedmelon, Silas S. Brown, Naniwako, Goarany, Mysterysociety, Zhouf12, Bushcarrot, Danr2k6, Liveste, JonNiola, Mufka, Cmichael, KylieTastic, DorganBot, Useight, Steel1943, Signalhead, Lights, Jhalkompwdr, CWii, Butwhatdoiknow, Lexein, Fences and windows, J-Baptiste, Philip Trueman, Vipinhari, Rei-bot, Anonymous Dissident, Liko81, Warrush, Anna Lincoln, Lradrama, TheJae, Broadbot, Canaima, LeaveSleaves, UnitedStatesian, Da31989, Haseo9999, Dark Tea, Spartytyme, Pfreicher, Monty845, Dsarma, Shree theultimate, DeanC81, Red, Hmwith, Lsi john, Drake144, SieBot, Madman, K. Annoyomous, Klaslop, ToePeu.bot, Dawn Bard, Sephiroth storm, Yintan, Alex9788, Softwaredude, Flyer22 Reborn, Xavier6984~enwiki, Bananastalktome, Drennick, Gopher23, Tmauer, Joey Eads, Harry the Dirty Dog, La Parka Your Car, Chri\$topher, Rjc34, Raoravikiran, Kortaggio, Denisaroma, Atif.t2, Martarius, Beeblebrox, ClueBot, LAX, NicDumZ, The Thing That Should Not Be, B1atv, Starkiller88, Antisoft, Terryr rodery, Lawrence Cohen, Keraunoscopia, Jotag14, R000t, Pwitham, SuperHamster, Melgomac, Boing! said Zebedee, PentiumMMX, Trivialist, William Ortiz, Cwagmire, Swearwolf666, Kitsunesgami, Excirial, Filovirus, Wilsone9, Zac439, Rhododendrites, Usbdriver, Mikay, Muro Bot, DanielPharos, Jemcc450, MelonBot, SoxBot III, Egmontaz, Darkicebot, Bearsona, XLinkBot, Arnoldegibson, Rror, Xawieri-enwiki, WikHead, Findepi, SilvonenBot, Alexius08, Dnrvfanta, Brento1499, IsmaelLuceno, Addbot, Chakkaloresh, Viperdeck PyTh0n, Ramu50, Mortense, WeatherFug, AlexandrDmitri, Stphnm, Astt100, Msick33, Tothwolf, Totakeke423, Ronjhones, Scientus, CanadianLinuxUser, Atif0321, O.neill.kid, MrOllie, Mcnaryxc, Ld100, Favonian, ChenzwBot, Getmoreatp, Chicken Quiet, Jasper Deng, Tide rolls, Bartledan, Peter AUDEM, Titi 3bood, Luckas-bot, Yobot, Ptbotgourou, Fraggie81, Donfbreed, Ojay123, Becky Sayles, THEN WHO WAS PHONE?, GateKeeper, MrBlueSky, Echtner, Eric-Wester, Magog the Ogre, Synchronism, AnomieBOT, BlackCatN, Thexelt, Kristen Eriksen, Deathwiki, Killiondude, Jim1138, IRP, Gascreed, AdjustShift, Ulric1313, Flewis, Materialscientist, Fffrrrr, Nosperantos, Citation bot, Aneah, Armormaid, ArthurBot, LilHelpa, Xqbot, Addihockey10, Jeffrey Mall, Melkori, Tad Lincoln, Shulini, Silversam, Mechanic1c, Life Now, GrouchoBot, Frosted14, Monaarora84, Omnipaedita, MugabeeX, Sophus Bie, TakaGA, Smallman12q, Shadowjams, SchnitzelMannGreek, Teknopup, Samwb123, Shyish, GliderMaven, FrescoBot, Sandgem Addict, Balth3465, Galorr, JMS Old Al, Adnan Suomi, Zhangyongmei, Jamesrules90, Fiddler on the green, Drogenov, HamburgerRadio, I dream of horses, Haneef503, Modamoda, Anders K Berggren, Yaltaplace, Fat&Happy, RedBot, BrianHoef, Anonauthor, Ille flottante, Lineslarge, Blstormo, Random1c, Cnwilliams, Jack-HenryDyson, FoxBot, Mwikieditor, Jrmurad, DixonDBot, Xiphidae, Demonkill101, Lotje, Lionaneesh, Gazzat5, N-david-1, Dinamik-bot, Toobulkeh, JOptionPane, Allen4names, Adarw, Xitrax, Merlinsocta, Fastilysock, Gomangoman, Mean as custard, RjwilmsiBot, Ripchop Bot, Bossanoven, Enrico.cambiaso, VernoWhitney, Mchcopl, J36miles, EmausBot, Mjdjm, Dewritech, Racerx11, GoingBatty, RA0808, RenamedUser01302013, NotAnonymous0, Qrsdogg, L235, Solarra, Wikielli, K6ka, Euclidthalis, OmnidPluS, QuentinUK, John Cline, Daonguyen95, Bongoramsey, Josve05a, DannyFratella, Finnish Metal, Msamae83, Access Denied, EneMsty12, Wikfr, Whsazdfhnbfxgnh, FinalRapture, Beatles2233, Ivhtbr, Erianna, Maccoat, EricWesBrown, Demiurge1000, TyA, Anne-Caroline~enwiki, Schnoatbrax, Donner60, Ryancarpenter, Jak32797, Pastore Italy, Zabanio, Teapeat, Kellyk99, Paddingtonbaer, Petrb, ClueBot NG, TT-97976, Gareth Griffith-Jones, Manubot, Keegen123, MelbourneStar, Satellizer, Permalinks, Joefromrandb, Cogware, Stultiwikia, Lawandtech, Lasoraf, Sunndil, Jormund, 123Hedgehog456, O.Koslowski, Vlhsp, Popo41~enwiki, Widr, Soucedric, Helpful Pixie Bot, Strike Eagle, BG19bot, Murler, Rpk74 lb, Tragic8, Vagobot, Fahmedch, Streakydj, Juro2351, Mleoking, MadHaTTer666, Mfordtl, Lloydus98, Adsf1234, MusikAnimal, Stogers, Kevo1cat, Mark Arsten, Op47, Rm1271, Eduart.steiner, Writ Keeper, Jarrodaddy, Junganghansik, Gtaguy235, Minhal Mehdi, Worldnewsinformant, Skunk44, Abgelcartel, AllenZh, RichardMills65, Mdann52, Cathairawr, Tonyxc600, Cyberbot II, Vinsanity123, Run4health, Chengshoutian, Eb1511, Dwnd4, Superkc, Team Blitz, Icicilizer, Kikue26, K7L, SoledadKabocha, 331dot, TonyJunak, Broadcasterxp, Lugia2453, Spicyitalianmeatball, UNOWenNYC, Ascom99, Jmoss57, Leemon2010, Palmbeachguy, Joshua-Hall155065, Sid Shadelslayer, Epicgenius, Ashikali1607, Renoldsmartin, ExtremeRobot, HMGamerr, FrigidNinja, Mbmxpress, James-Moose, Tentinator, Sngs87, Marchino61, Webhostingtips, Johnhax, Ogh4x, Nodove, DavidLeighEllis, Dwgould, NiuWang, Reacher1989, Ginsulof, ArmitageAmy, Henrychan123, Jianhui67, UY Scuti, Quitesavy, YellowLawnChair, Akashksunny13, Meteor sandwich yum, PJone, Dodi 8238, Pvpmasters, XxWoLfxX115, JNKL, Lordangel101, Kazkade, Monkbot, Noahp15, Lucyloo10, Vieque, Jakupian, Web20DOS, Muhammadabubakar92, Cph12345, 365Adventure, Frogteam, Orthogonal1, BlackCat1978, ApolloLV, Jjsantanna, Thetech-girl, Sam-the-droid, Tommate789, ChamithN, Koen2014 7, Matiia, V1n1 pares, HoustonMade, Flated, Probinrcux, ThatOneGuyGaming, Lizard Quadrant, JohnZLegand, Delcooper11, Chiranjeev242, Restart32, Jokingrotten, Eslam Yosef, Infinite Guru, Haroly, Hazim116, Kurousagi, Pandamaury, Positronon, Misfoundings and Anonymous: 1256

- **Malware Source:** <https://en.wikipedia.org/wiki/Malware?oldid=691060608> **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, Khyrn Chanur, Finlay McWalter, Rossumcapek, Huangdi, Riddley, Donarreiskoffer, Pigsontehwing, Fredrik, Vesprisiano, 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, Icarins, TonyW, Clemwang, Trafton, D6, Monkeyman, Discospinster, Rich Farmbrough, Guanabot, Vague Rant, Vsmith, Sperling, Night Gyr, Sc147, JoeSmack, Elwikpedista~enwiki, Sietse Snel, Art LaPella, EurekaLott, One-dimensional Tangent, Xgravity23, Bobo192, Longhair, Billymac00, Smalljim, Cwlfsheep, 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, Ketiltrot, Rjwilmsi, Collins.mc, Vary, BruceLee, Frenchman113, PrivaSeeCrusade, Connorhd, Yamamoto Ichiro, Andrzej P. Wozniak, RainR, FlaBot, Fraggle, Chobot, Bornhj, Bdelisle, Cshay, Gwernol, RogerK, Siddhant, YurikBot, Wavelength, RattusMaximus, Aussie Evil, Phantomsteve, Ikester, RussBot, DMahalko, TheDoober, Coyote376, Ptomes, Wint, Thane, NawlinWiki, Hm2k, Krystyn Dominik, Trovatore, Cleared as filed, Coderzombie, Kingpomba, Ugnius, Amcfreely, Voidxor, Tony1, Alex43223, FlyingPenguins, Chriscoole, Bota47, Groink, Yudiweb, User27091, Tigalch, Flipjargandy, Romal, American2, Nikkimaria, Theda, Closedmouth, Abune, PrivaSeeCrusader, Dspradau, GraemeL, Cffrost, RealityCheck, RenamedUser jaskldjslak904, 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 9139j3, Gobonobo, Robofish, Xaldafax, Nagle, Fernando S. Aldado~enwiki, Vodicidtenore, Ian Dalziel, 16@r, Andypandy.UK, Slakr, NcSchu, Ehheh, Uuuuhuł, Vernalex, DL2000, Andreworkney, Fan-1967, Tomwood0, Iridescent, Dreftymac, NativeForeigner, Mere Mortal, UncleDougjie, 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, 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, 28421u2232nfencenc, 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, Comprman12, 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, Oshawah, 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, Nnkh00, Nosferatus2007, Evaluist, Miniapolis, Lightmouse, Helikophis, Correogsk, Stieg, Samker, Jacob.jose, BfMGH, Dabomb87, Denisarona, Ratemonth, Martarius, ClueBot, Muhammadsb1, NickCT, Vitor Cassol, The Thing That Should Not Be, VsBot, Lawrence Cohen, Wysprgr2005, Frmorrison, Jwihbey, Sam Barsoom, Ottava Rima, Paulcmnt, Excirial, Jusdafax, Dcampbell30, Rhododendrites, Ejsilver26, WalterGR, Maniago, Jaizovic, Dekisugi, Xme, DanielPharos, Versus22, Callinus, Johnuniq, Rossen4, DumZiBoT, Darkicebot, XLinkBot, BodhisattvaBot, Dal33T, Avoided, Sogle, Mifter, Noctibus, CalumH93, Kei Jo, Addbot, 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, ଆଶୀର୍ବାଦ ମନ୍ତ୍ରନାର୍କ୍, Publicly Visible, Luckas-bot, Yobot, Philmikeyz, WikiDan61, Tohd8BoaithuGh1, Ptbotgourou, Fraggle81, Evans1982, Gjohnson9894, Dmarquard, AnomieBOT, DemocraticLuntz, Rubinbot, Roman candles, IRP, Galoubet, RandomAct, Materialsscientist, CoMePrAdZ, Crimsonmargarine, Frankenpuppy, ArthurBot, Quebec99, Cameron Scott, Xqbot, TheAMmollusc, Mgaskins1207, Capricorn42, 12056, Avastik, Christopher Forster, Masonaxcte, S0aaasd2sf, Almabot, GrouchoBot, Monaarora84, Shirik, RobotBOT, PM800, Dougofborg, Luminique, 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, Thhotch, Lord of the Pit, DARTH SIDIOUS 2, Whisky drinker, Mean as custard, Moshe1962, RjwilmsiBot, Colindiffer, Panda Madrid, Thunder238, Salvio giuliano, Enauspeak, EmausBot, John of Reading, WikitanvirBot, The Mysterious El Willstro, Winner 42, K6ka, Porque123, Eliason2007, SocialAlex, Connoe, AvicBot, Bollyjeff, Dffgd, Azuris, H3llBot, EneMsty12, Simorjay, Paramecium13, Tyuioop, Tolly4bolly, Cit helper, Coasterlover1994, Techpraveen, Shrigley, Capnjim123, Ankitcktd, Ego White Tray, Mv Cristi, Pastore Italy, Corb555, Mark Martinec, Helpsome, ClueBot NG, Cwmhiraeth, Rich Smith, PizzaMuncherMan, Lzeltsr, PwnFlakes, Matthiaspaul, MelbourneStar, Catlemur, Satellizer, Piast93, Steve dixon, MarsTheGrayAdept, Mesoderm, Widr, Brandoraganza, 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, Pauly72, BYagour, Verzer, Antonio.chuh, ChrisGualtner, Tech77, Hassim1983, Hnetsec, MadGuy7023, JYBot, Ghostman1947, Tofeeopot, EagerToddler39, Dexbot, FTLK, Joshy5.crane, Dothack111, Skullmak, Webbanana1, Rajalakshmi S Kothandaraman, Salwanmohsen, CoLocate, Lugia2453, Himanshu Jha 07, Frosty, UNOwnNYC, SirkusSystems, Sourov0000, Corn cheese, Palmbeachguy, Gtangil, 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, Ajitkumar.pu, JaconaFrere, Vandraag, SnoozeKing, Newlywoos, Gatndr Burkllg, Alan24308, Worzzy, Lucius.the, Andrei Marzan, Joeleo123, MLP Eclipse, Stevebrown164, Qwertyp2000, Andrdema, A4freees, Ianamada, AnastasiaGS, Maxwell Verbeek, Oiyarbepsy, WikiGopi, Eurodyne, Shovelhead54, Spamero, Fimatic, Hvaara, Friendshipbracelet, Rubbish computer, Gabiked, ToonLucas22, Nysrtup, Davenil, Akstotz, M8b8f8, Supdiop, KasparBot, Tmar877, Marzan Chowdhury, JJMC89, Jasmineluv, ROMA1maro, Albaianakuov, Shreyassachdeva, Hearmewe, Salarmaf, Matthews david, Kosherpenguinexperience, Phils234, Srikeha selva and Anonymous: 890

- **Payload (computing)** Source: [https://en.wikipedia.org/wiki/Payload_\(computing\)?oldid=676340150](https://en.wikipedia.org/wiki/Payload_(computing)?oldid=676340150) Contributors: Kku, Robbot, Discospinster, H2g2bob, Pol098, GregorB, BertK, SmackBot, Hateless, Kvng, Neelix, Alaibot, Legotech, Pstanton, MikeLynch, The Transhumanist, Tinucherian, Bogey97, Derekrugerson, Synthebot, X-Fi6, Denisarona, Wdwd, WikHead, Addbot, Pradameinhoff, Erik9, Jonesy95, Oliverlyn, Jenks24, ClueBot NG, Scarlettai, Mo5254, Wambuirwn and Anonymous: 30
- **Rootkit** Source: <https://en.wikipedia.org/wiki/Rootkit?oldid=687709116> Contributors: Zundark, Fubar Obfuscator, William Avery, SimonP, Stevertigo, Frecklefoot, JohnOwens, Nixdorf, Pnm, Liftarn, Zanimus, 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, Tobias Bergemann, Unfree, David Gerard, Alison, JimD, Ezhiki, Cloud200, AlistairMcMillan, Saucapan, Taka, Alobidig, Deewiant, Creidieki, Pascav, Adashiel, Squash, Brianhe, ElTyrant, Rich Farmbrough, Agnitus, 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, Easys12c, Midnightblaze, SDC, Umofoemia, Xiong Chiamiov, RichardWeiss, Graham87, Rjwilmsi, TitaniumDreads, Syndicate, Arisa, Randolph, RainR, Flarn2006, FlaBot, RobertG, Stoph, JiFish, Harmil, Mark Luszniak, Arunkoshy, Mordin, Intgr, Mimithebrain, Dbpigeon, Martin Hinks, Poorsod, FrankTobia, Elf guy, Uriah923, YurikBot, Wavelength, Hairy Dude, Diesonne, AVM, Chrijsutinparr, 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, Rootorick, AnimeJanai, Solarusdude, Jacqui M, That Guy, From That Show!, SmackBot, Mmernex, Estoy Aquí, Reedy, Mate.tamasko, Unyooyega, 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, MichaelBillington, DMacks, J.Christopher.Wells, AndyBQ, A5b, Mitchumch, N-dy, Clicketyclack, FrostyBytes, Tasc, Tthtlc, Peyre, Simon Solts, Xionbox, LAlawMedMBA, IvanLanin, CapitalR, Prpower, Phoenixrod, Courcelles, Tawkerbot2, Davidb-spalding, FatalError, Zarex, Cyrus XIII, Megaboz, Jokes Free4Me, Jesse Viviano, Chrismo111, Racooper, Myasuda, Equentil, A876, GrahamGRA, Tryl, Irate, Fetterinity, Mewsterus, Etaon, Ambulnick, Marek69, Tocharianne, AntiVandalBot, Widefox, Obiwankenobi, Czj, Sjledet, Lfstevens, Bscottbrown, AndreasWittenstein, TivicBot, Hiddenstealth, NapoliRoma, MER-C, Minitrue, QuantumEngineer, Karsini, BCube, Repku, Raanoo, Drugonot, Chevinki, Nyttend, Cl36666, Denorios, Stromdal, Alekjds, Hamiltonstone, Cpl Syx, Xan-

droZ, 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, Ceranthon, Ggpur, MrChupon, SieBot, Technobreath, Sephiroth storm, Edans.sandes, Windowsvistafan, Aly89, General Synopsis, Fyvre, Clearshield, Capitalismojo, Bogwhistle, BfMGH, Guest141, Martarius, ClueBot, The Thing That Should Not Be, TheRasIsBack, Mild Bill Hiccup, Foss guy, 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, Kongr43open, Sergey AMTL, Elsendero, TutterMouse, Cst17, MrOllie, OIEnglish, Fiftyquid, Luckas-bot, Yobot, Fraggie81, GateKeeper, Golftheimer, Alipie42, AnomieBOT, NoKindOfName, Blueraspberry, Materialscientist, Nutsterrt, Citation bot, ArthurBot, LilHelpa, Avastik, S0aasdf2sf, Notwej, GrouchoBot, Kernel.package, Thearcher4, 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 aoiowaiuyr894isidik43, TjBot, Alph Bot, EmausBot, John of Reading, WikitanvirBot, Timtempleton, Hercules31, Diewritech, Janiko, P3+J3^u!, ZéroBot, Herman Shurger, Basheersubei, Mike735150, IceCreamForEveryone, Bender17, Chicklette1, Difflame, Macwhiz, Nero2006, 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, Crypttodd, CaSJer, A Certain Lack of Grandeur, Ginsuloft, Oranjelo100, Dust482, Monkbot, Vieque, BethNaught, Ahollypak, Shinydiscoball, Jithendran Subburaj, TQuentin, Azlan 6473, KasparBot, Sharanyanaveen and Anonymous: 580

- **Keystroke logging** *Source:* https://en.wikipedia.org/wiki/Keystroke_logging?oldid=691272525 *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, Sriskanth05, Wimt, Mipadi, Bob Stromberg, Vivaldi, Tony1, Occno, 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, SheeEt-tin, Frap, Skidude9950, Ww2censor, Flask215, Khoikhoi, Engwar, Nakon, Gamgee, Kalathalan, Clicketyclack, Torritori, Ckatz, Tu-anmd, Redboot, Ehheh, Njb, Mets501, H, Mike Doughney, Pauric, Sander Säde, OnLine, Jeremy Banks, JForget, Dycedarg, Jesse Viviano, CorpX, Alexdw, Odie5533, Tawkerbot4, Bposert, 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, Alecks, Hut 8.5, Isthisthingon, Techie guru, .anacondabot, Magioladitis, Jaysweet, Ukuser, JNW, Cheezyd, Confiteordeo, Fedia, Wikivda, Wikire, MartinBot, STBot, CliffC, Jonathan.lampe@standardnetworks.com, Anaxial, Nono64, Spider, Tresnius, 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, Nmvw, 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, Drwho-for, 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, Reallmoldycheese, Automaite, Ezen, S0aasdf2sf, Aceclub, Ruy Pugliesi, GrouchoBot, IslandLumberJack, Mark Schierbecker, Krypton3, 78.26, Aenus, Mountielee, Prari, FrescoBot, WPANI, Clubmaster3, DigitalMonster, PeramWiki, Nathancac, Waller540, HamburgerRadio, Italic, Redrose64, Rajtuhin, MKFI, AgentG, Reconsider the static, Ao5357, Lotje, Vrenator, 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, Guess4094, Carlitaeliza, MeanMotherJr, BattyBot, Abgelcartel, Jfd34, Eduardofeld, Lloydliske, EagerToddler39, Codename Lisa, Webclient101, Klabor74, Zhiwei sun, Jaericsmith, Sourov0000, Corn cheese, Way2veers, Yuvalg9, MountRainier, JadeGuardian, Kennethaw88, Lvanwaes, Mover07, Jianhui67, Dannyruthe, NewWorldOdor, Janeandrew01, Michael Dave, CNMall41, Jamesmakeon, Bobsd12, Wasill37, Crystallizedcarbon, Scyrusk, Devwebtel, JoanaRivers, ScottDNelson, Jhfhey, Awmarks, Muhammad mobeen83, BD2412bot, Wikijagon and Anonymous: 558
- **Access control** *Source:* https://en.wikipedia.org/wiki/Access_control?oldid=690834502 *Contributors:* Timo Honkasalo, The Anome, Edward, Patrick, BAxelrod, Wikiborg, Vaceituno, Texture, Auric, Tobias Bergemann, Alan Liefting, Andycjp, Omassey, Bderidder, CHoltje, El C, Sietse Snel, BrokenSegue, SPUl, Espoo, Hu, Wtmitchell, Dave.Dunford, Neetij, H2g2bob, Ringbang, Woohookitty, Mindmatrix, Camw, Daira Hopwood, Jeff3000, PeregrineAY, BD2412, Rjwilmsi, Amire80, Syced, Eubot, Old Moonraker, Nihiltres, Gurch, Intgr, Chobot, DVdm, Bgwhite, Borgx, RussBot, Stephenb, Gaius Cornelius, NawlinWiki, Grafen, Welsh, PrologFan, Auminski, Galar71, Ka-Ping Yee, Back ache, Guinness man, DEng, NeilN, Carlosguitar, Exit2DOS2000, SmackBot, McGeddon, Zzm7000, Xaosflux, Yamaguchi, Delfeye, Silly rabbit, DHN-bot~enwiki, Frap, RoyalBlueStuey, Andreijj, Trbdavies, Luís Felipe Braga, Chris0334, ArielGlenn, Kuru, Yan Kuligin, Twredfish, Slakr, E-Kartoffel, Thatcher, Kvng, Hu12, Hetar, MikeHobday, Linkspamremover, CmdrObot, Cydebot, Gogo Dodo, Jedonnelley, Soifranc, Dawnseeker2000, QuiteUnusual, MER-C, NE2, Roleplayer, Knokej, Magioladitis, McGov1258, George A. M., Web-Crawling Stickler, Americanhero, Billbl, FisherQueen, Bostonvaulter, CliffC, CommonsDelinker, Fmjohanson, Memoroid~enwiki, Hersfold, Philip Trueman, TXiKiBoT, SecurityEditor, Ziounclesi, Mazarin07, Andy Dingley, Scouttle, EverGreg, Nickbernon, Wikiscottcha, SieBot, Therepguy, Happysailor, Brankow, Lightmouse, Sfan00 IMG, ClueBot, Josang, Binksternet, Apacheguru, DragonBot, LadyAngel89, Eldub1999, The Founders Intent, Dekisugi, Aitias, Talsetrocks, Wsimonsen, Pichpich, Tonypdmr, Mitch Ames, MystBot, RyanCross, Addbot, Actatek, Leszek Jaćczuk, Fluffernutter, Rickfray, Subverted, Stantry, نویسنده, Gail, Jarble, Ben Ben, Luckas-bot, Yobot, Bunnyhop11, KamikazeBot, Timothyhouse1, AnomieBOT, Advancesafes55, Piano non troppo, Willowrock, Materialscientist, StewartNetAddict, Securitywiki, Sionk, Jgeorge60, Andriusval, SassoBot, Kernel.package, Jray123, Ruuddekeijzer, Indyaedave, Nageh, Mark Renier, DrilBot, Winterst, Swamy.narasimha, Testplt75, Jandalhandler, Lotje, Vrenator, Clarkcj12, Zeeshankhuhro, Lingliu07, Mean as custard, EmausBot, John of Reading, WikitanvirBot, Timtempleton, Immunize, Feptel, Iancbend, Gagandeeps117, Abhinavcambridge,

Josve05a, Krd, Iwatchwebmaster, Staszek Lem, RISCO Group, Ssbabudilip, Dineshkumar Ponnusamy, ChispastionBot, ClueBot NG, Sesha Sayee K V, KunjanKshetri, Chester Markel, Secguru1, Animusnovo, Mesoderm, Stewartjohnson229900, Widr, Ellerose, BG19bot, Cyberbot II, The Illusive Man, Cfeltus, Codename Lisa, Vinjadav, TheSadnessOfBeing, David.brossard, Dewoller, Fan Zhang-IHC, Luesand, Mattgavenda, TotaliTech, Mumbui, Weikrx, Mgavenda, Rsschomburg, Virgo hariom, Clark Steph, Cmogntgomery11, Cybersecurity101, Filedelinkerbot, KaiserKIS, Ggotero, Hphaikuku, Thetechgirl, Politikuserna, Zubairul, Kaspabot, Fabtron and Anonymous: 256

- **Application security** *Source:* https://en.wikipedia.org/wiki/Application_security?oldid=684079155 *Contributors:* SimonP, Charles Matthews, Psychonaut, DavidCary, Cloud200, Hillel, AliveFreeHappy, Discospinster, Rhobite, Enric Naval, JYolkowski, Bobrayner, OwenX, Mindmatrix, Halovivek, Vegaswikian, Pseudomonas, Welsh, Tjarrett, Slicing, NielsenGW, Rwww, Algae, Tyler Oderkirk, SmackBot, Ohnoitsjamie, Frap, JonHarder, IronGargoyle, Kvng, Iridescent, Sander Säde, Tedmarynicz, OnPatrol, Blackjackmagic, Njan, Aarnold200, Dawnseeker2000, Obiwankenobi, Dman727, Robina Fox, Toutoune25, JLEM, Grabon~enwiki, IronAlloy, JEMLA, DatabACE, Maurice Carbonaro, Maxgleeson, Alanfeld, Philip Trueman, Felmon, Pryderi100, NEUrOO, M4gnun0n, Friendlydata, Dosco, Swtechwr, Wiscoplay, Dcunited, Raysecurity, Paulmguyen, Dthomsen8, Mitch Ames, Ha runner, Bookbrad, MrOllie, Eheitzman, Jnarvey, Yobot, Fraggle81, Nickbell79, AnomieBOT, Fhuonder, Stationcall, Mwd, FrescoBot, Nageh, Geofhill, Amey.anekar, Hnguyen322, Trapist the monk, Vrenator, Mr.moyal, Super n1c, We hope, ClueBot NG, Aashbel~enwiki, Widr, RachidBM, BG19bot, MatthewJPJohnson, Swameticul, Xena77, Mdann52, Tohimanshu, Triomio, Isoron27000, Roberto Bagnara, Truehorizon, Securechecker1, Jpickel, MuscleheadNeV, Chrisdmiller5, Greenmow, Milan2377, Abigaildw, Prugile and Anonymous: 74
- **Antivirus software** *Source:* https://en.wikipedia.org/wiki/Antivirus_software?oldid=691428176 *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, Furykef, 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, Solipsit, Wmahan, Utcursh, SoWhy, Beland, Piotrus, Cynical, Gscshoyru, TonyW, Hobart, Eisnel, Discospinster, Rich Farmbrough, 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, Mattern, Cuvitxo, Kbdenk71, Yurik, Ryan Norton, Rjwilmsi, DirkvdM, RainR, FlaBot, JiFish, RexNL, Gurch, DavideAndrea, ChongDae, Born2cycle, Melancholie, Ahunt, PeterL, Gwernol, YurikBot, Wavelength, Borgx, Grizzly37, Wfried, Arado, TheDoober, Piet Delport, SpuriousQ, Akhristov, Clunaia, NawlinWiki, Hm2k, Badagnani, Arichnad, Vlad, Bota47, Bokonon~enwiki, BazookaJoe, GraemeL, Peter, Fourohfour, Hirebrand, Jaysbro, Eptin, 机器人 robot, Dunxd, Cumbiagermen, Firewall-guy, SmackBot, Although, JurgenHadley, J7, Dxco, Relaxing, Easygoeasycome, Gilliam, JorgePeixoto, Lakshmin, Gary09202000, Chris the speller, Egladkhih, Morte, EncMstr, Jerome Charles Potts, Bigs sb, DHN-bot~enwiki, Uniwares, Darth Panda, Frap, JonHarder, Korinkami, 03vaseyj, SundarBot, Cybercobra, Valenciano, Mwtows, Ihatetoregister, Oo7jeep, Gobonobo, Capmo, NongBot~enwiki, 16@r, Erotml, Beetstra, Doczilla, Qu4rk, Caiaffa, Hu12, DabMachine, SimonD, Phantomnecro, UncleDoggie, CapitalR, Kirill Chiryasov, Courcelles, Tawkerbot2, FleetCommand, CmdrObot, BENNYSOFT, Jesse Viviano, NaBUru38, Chrisahn, Cydebot, Gogo Dodo, Xhopingtearsxx, AcceleratorX, Tawkerbot4, Khattab01~enwiki, Ohadlikmsan, The Mad Bomber, SpK, Neustradamus, Mikewax, TAG.Odessa, Dimo414, Thijis'bot, Jdivakarla, Leedeth, LemonMan, Saibo, Dalahäst, TurboForce, Dawnseeker2000, Menifisto, AntiVandalBot, Sjconrad-mchedrawe, Gökhan, Serpent's Choice, JANdbot, Meinsla, MER-C, Tushard mwti, 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, Je-sant13, Turbulencepb, Neon white, Ripdog2121, Tokyogirl79, 5they, Patrickjk, AntiSpamBot, Dougmarlowe, DadaNeem, Pandawelch, White 720, Jamesontai, Idioma-bot, Javeed Safai, Melovfemale, VolkovBot, AlnoktaBOT, Philip Trueman, DoorsAjar, TXiKiBoT, Emedlin1, Mujdat61, Vipinhari, Technopat, Anonymous Dissident, Qxz, Corvus cornix, LeaveSleaves, Natg 19, Tmalcomv, Haseo9999, C45207, Ngantengyen, LittleBenW, Fredtheflyfrog, Lonwolve, Wrldwzrd89, Sahilm, Derekslater, Newspartnergroup, Swaq, Sephiroth storm, Yintan, Miremare, Mothmolevna, Jerryobject, Flyer22 Reborn, PolarBot, Nosferatus2007, Askild, Topicle, OKBot, Plati, Samker, PrimeYoshi, Escape Orbit, Arnos78, Martarius, Tanvir Ahmed, Leahtwo saints, ClueBot, Kl4m, The Thing That Should Not Be, IceUnshattered, Trotline, Spuernase, Mild Bill Hiccup, Ka vijay, LizardJr8, ChandlerMapBot, Georgest23, Rockfang, DragonBot, Excirial, Socrates2008, Pavix, Tyler, Pladook, Jotterbot, JamieS93, ChrisHodgesUK, DanielPharos, Versus22, Johnnuniq, SoxBot III, SF007, Sensiblekid, XLinkBot, Rror, Mavenkatesh, Svarya, HexaChord, Addbot, Xp54321, Wizho, Mortense, Nuno Brito, Softfreak, Sergey AMTL, Vatrena ptica, CanadianLinuxUser, Fluffernutter, Ankitguptajaipur, Kueensryche, 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, Blueraspberry, Materialscientist, Police267, Kalamkaar, Eumolpo, Cameron Scott, Misi91, Avun, XZeroBot, Sputink, Rwmoeckoe, S0aasd2sf, 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, Trappis the monk, Lamarmote, Miiszmyleove, LogAntiLog, Lotje, Wikipandaeng, Vrenator, TBloemink, 777sms, Neshemah, Dianna, Hornlitz, Exeter, Teenboi001, Mean as custard, RjwilmsiBot, Ripchip Bot, Panda Madrid, Enauspeaker, DASHBot, EmausBot, John of Reading, WikitanvirBot, Immunize, Philtweir, Heracles31, Dinhuydza, Ibbn, Ryanxo, Tommy2010, Cmlloyd1969, Emenid, Elison2007, Fa, Mats131, ElationAviation, Makecat, Skyinfo, Yabba67, Rickraptor707, Diflame, ChispastionBot, GermanJoe, Pastore Italy, EdoBot, Kandr8, Petrb, ClueBot NG, Lzelts, TheKaneDestroyer, Jack Greenmaven, Satellizer, LK20, Dfarrell07, Multiwikiswat, Piyush1992, JuventiniFan, Malijinx, Widr, Hsinghsarao, Joseph843, Helpful Pixie Bot, Dwe0008, HMSSolent, Krenair, Jezaz87, Janendra, Arthurnyc, AvocatoBot, Thekillerpenguin, Teksquiste, Irfanshararuddin, TheMw2Genius, Kremin, Newmen1020304050, BattbyBot, Justincheng12345-bot, JC.Torpey, Divonnais, Farqad, IddiKlu, Nisha1987, Rohaneknathshinde459, Garamond Lethe, JYBot, Dark Silver Crow, Codename Lisa, Cryptodd, Pcguru66, K1ngXSp3c1al, Lugia2453, Kumarworld2, Sourov0000, Seo100, M.R.V model, Gau-tamcool12, Faizan, I am One of Many, Ryan889, Matt.Sharp98, Jakec, Eddymck1, Ashajose0002, Assumption, Ginsuloft, Quenhitran, Dannyruthe, MetalFusion81, Robevans123, Monkbot, CodyHofstetter, TerryAlex, Xpasindu123, Thetechgirl, ChamithN, WikiGopi, Jacbizer, Puffle7275, Deanwalt123, Rom broke, Drop knowhow, Seanpatrickgray, The Professor123, Chenthil Vel Murugan 1986, Srilekha selva, XxXMinecraftProsnipzXxX and Anonymous: 661
- **Secure coding** *Source:* https://en.wikipedia.org/wiki/Secure_coding?oldid=618244589 *Contributors:* AliveFreeHappy, H2g2bob, Bobrayner, Malcolm, SmackBot, Frap, Alaibot, QuiteUnusual, Manione, Fabrictramp, Susan118, Addbot, Elviscrc, Luckas-bot, Shantzg001, BK314159, Helpful Pixie Bot, Richu jose, Monkbot and Anonymous: 2
- **Secure by design** *Source:* https://en.wikipedia.org/wiki/Secure_by_design?oldid=658065876 *Contributors:* Aezram, LMB, Joy, Gracefool,

Vadmium, AndrewKeenanRichardson, Sam Hocevar, Andreas Kaufmann, Sperling, Project2501a, Ciaran H, Mailer diablo, Pol098, Josh Parris, Vofly, FlaBot, Quuxplusone, Tardis, Gardar Rurak, Ptomes, Bovineone, Mcicogni, Drable, Alex Ruddick, SmackBot, Mmernex, Mauls, Commander Keane bot, Chris the speller, Wonderstruck, Hu12, CmdrObot, Towopedia, Manionc, Douger, Gwern, Nono64, Warut, Alecwh, Zoef1234, Goodone121, XLinkBot, Addbot, Luckas-bot, AnomieBOT, Teriyakistix, LucienBOT, Sae1962, John of Reading, WikitanvirBot, TheSophera, Wes.turner and Anonymous: 22

- **Security-focused operating system** *Source:* https://en.wikipedia.org/wiki/Security-focused_operating_system?oldid=689215951 *Contributors:* The Anome, Graue, PatriceNeff, David Latapie, Dysprosia, Maximus Rex, Taxman, Joy, Finlay McWalter, Chealer, Hadal, Foot, Ricky~enwiki, Kusunose, Sam Hocevar, Bluefoxicy, DMG413, Projex, Alexkon, MeltBanana, Sperling, Bender235, Spearhead, Miles Monroe~enwiki, NicM, Bobrayner, Pol098, Ruud Koot, Qwertys, Dar-Ape, Nihiltres, Jrtayloriv, BMF81, Bgwhite, Ptomes, Etbe, Di-electric, Mosquitops, Ilmaisin, Janizary, Allens, SmackBot, Gizmoguy, Isaac Dupree, Frap, JonHarder, Vdanen, Harryboyles, La Cigale, Nowhere man, Editor at Large, TurboForce, Douger, NapoliRoma, Transcendence, Wkussmaul, Doc aberdeen, Dlewis7444, Penma, Ben-bucksch, Allong, Canaima, Toddst1, Jojalozzo, Mild Bill Hiccup, Xenon54, Ravaet, SF007, Useerup, Paulmnguyen, JamieGoebel, Addbot, Hybrazil, C933103, Yobot, AnomieBOT, ChristopheS, Ojakubcik, FrescoBot, NuclearWizard, Mdjango, David Hedlund, JumpDiscont, Wrix24x7, John of Reading, WikitanvirBot, GoingBatty, Peaceray, Wafelijzer, Solbu, BG19bot, WikiTryHardDieHard, Akuckartz, Gotimer, NewGuy1001, MrFretty, Comp.arch, Oranjelo100, Ostree, Zakraye, Eclipsespark, Miraclexix, MrArsGravis, R3sJAP155M, Hail knowledge and Anonymous: 93
- **Authentication** *Source:* <https://en.wikipedia.org/wiki/Authentication?oldid=691126667> *Contributors:* SJK, Mswake, Imran, Edward, Gbroiles, Pnm, MartinHarper, (, Pde, EntmootsOfTrolls, Smack, Edmilne, M0mms, Ww, Nickshanks, Altenmann, Nurg, Lowellian, Rholton, Tobias Bergemann, Mark.murphy, Tom-, Rchandra, Matt Crypto, Pgaa002, Beland, Heavy chariot, Ablewisuk, Oneiros, Almit39, Rich Farmbrough, Tpkorra, ArnoldReinhold, Pavel Voznenik, Goochelaar, Szquirrel, Edward Z. Yang, Billymac00, John Vandenberg, SpeedyGonsales, Cherlin, HasharBot~enwiki, ClementSeveillac, Johntinker, DreamGuy, H2g2bob, HenryLi, Ceyockey, Roland2~enwiki, Rodii, Woohookitty, Mindmatrix, Daira Hopwood, Kokoriko, Shonzilla, Bernburgerin, Jorunn, Rjwilmsi, Pdelong, Husky, Syced, Fish and karate, Emarsee, Margosbot~enwiki, LeCire~enwiki, YurikBot, RobotE, RussBot, Backburner001, DanMS, Theshadow27, Bachrach44, Cryptosmith, Leotohill, Bota47, Metadigital, Zzuuzz, GraemeL, SmackBot, Imz, EJVargas, KocjoBot~enwiki, Arny, Yamaguchi, Ohnoitsjamie, Lakshmin, Chris the speller, Bluebot, DHN-bot~enwiki, Tsca.bot, Skidude9950, JonHarder, Arnneisp, Radagast83, Tjcoppet, Luís Felipe Braga, Spiel496, Nabeth, Kvng, Hu12, DouglasCalvert, AbsolutDan, WeggeBot, Thepofo, Ednar, Cydebot, Xemoi, UncleBubba, Bdpg, Jose.canedo, Dancter, AngoraFish, Lindenhills, Nunewsco, Thijs!bot, Ebpr123, Sqying, EdJohnston, Dawnseeker2000, AntiVandalBot, Fedayee, Rshoham, Blmatthews, EJNorman, AndreasWittenstein, JAnDbot, Durrantm, Ingvar2000, Jheiv, Technolgyvoices, Darth Andy, Rami R, AlephGamma, Gabriel Kielland, David Eppstein, User A1, Ludvikus, JaGa, STBot, Bnovack, Bertix, Rlsheehan, Syphertext, Nigelshaw, Homer Landskirty, Idioma-bot, VolkovBot, TXiKiBoT, NoticeBored, Technopat, Rei-bot, Aymath2, Steven J. Anderson, Jackfork, Mayahlynn, Double Dickel, SieBot, Euryalus, WereSpielChequers, Pkgx, Mdsam2~enwiki, Yerpo, Skippydo, Pinkadelica, Stlmh, Denisarona, Sfan00 IMG, ClueBot, Wikiold1, Trivialist, Neverquick, Apelbaum, Jomsborg, Captpossum, The Founders Intent, Primasz, BOTarate, Bludpojke, Versus22, Boozinf, Apparition11, XLinkBot, Dmg46664, Avoided, Dsimic, Addbot, Fyrael, Ronhjones, EconoPhysicist, Wireless friend, Yobot, Pink!Teen, MarioS, THEN WHO WAS PHONE?, PavanThatha, AnomieBOT, DemocraticLuntz, Galoubet, Materialscientist, ArthurBot, DSisypheBot, Mywikiid99, GrouchoBot, Omnipaedista, RibotBOT, Dtodorov, Shadowjams, Talion86, Jmd wiki99, Id babe, Nageh, D'ohBot, Teux, Tabbelio, MastiBot, Littledogboy, Hnguyen322, Anavaman, Lotje, Ashdo, Mean as custard, Kiko4564, AnBuKu, EmausBot, Goldenbrook, WikitanvirBot, Timtempleton, Zollerriia, Active Banana, Clymbon, Tommy2010, ValC, K6ka, Davy7891, Fakeshop, Donner60, Newstv11, ClueBot NG, Frietjes, 0x55534C, Curb Chain, Mahfudien, Microtrace, Mike2learn, OnlinepresenceHID, MusikAnimal, Wikisafety, Daniel.N.Rollins, RockOrSomething, Lugia2453, Maruleth, Pi-pat.poonP1, Kmaletsky, DavidLeighEllis, Ugog Nizdast, UKAmerican, Soaring souls, Sticky molasses, Monkbot, Mounikamm, Marianne-Miller, Ana286, Wrtr63, CAPTAIN RAJU, CLCStudent and Anonymous: 226
- **Multi-factor authentication** *Source:* https://en.wikipedia.org/wiki/Multi-factor_authentication?oldid=689453906 *Contributors:* Edward, Nealmcb, Pnm, William M. Conolley, JimTheFrog, Pabouk, Cloud200, Alexf, Robert Brockway, AlexKilpatrick, Alexanderino, Mindmatrix, BD2412, Intgr, Wavelength, Pontillo, ViperSnake151, SmackBot, McGeddon, Rojomoke, Ohnoitsjamie, Thumperward, MovGP0, Jmnbatista, Kvng, ETSkinner, Nick Number, Dougher, Froid, Firealwaysworks, David Eppstein, DGG, Bwtranch, Krishnachandranvn, Atropos235, Edvvc, Softtest123, JL-Bot, Martarius, Niceguyedc, Dsimic, Addbot, Mortense, Otisjimmy1, MrOllie, Worch, Wireless friend, Yobot, MarioS, Sweerek, AnomieBOT, Blueraspberry, Materialscientist, LilHelpa, Biker Biker, Chris Caven, ItsZippy, Anongork, Going-Batty, Bpburns88, HupHollandHup, Neil P. Quinn, Rocketrod1960, Snotbot, MerllwBot, Lowercase sigmabot, BG19bot, Turtle595, Mark Arsten, BattyBot, Suncatcher 13, Zhao Feng Li, Mdann52, Dicti0nary0, Hbiering, Radiochemist, Quukhgjels, Codename Lisa, Gacelp, Basilphilipsz, Dave Braunschweig, Lsmll, BB609, MesaBoy77, Comrade pem, NickyLarson29, Kwaimind, Dudewhereismybike, Rosstaylor3b, Sticky molasses, Benforrest, Pfalcon2, Dylan-Evans-not-the-other-one, 19JKG96, Nyashinski, Vieque, Srijanashastri, TheCoffeeAddict, Sinchdev, Baylott, Joemates, Walter.Jessica and Anonymous: 67
- **Two-factor authentication** *Source:* https://en.wikipedia.org/wiki/Two-factor_authentication?oldid=690899708 *Contributors:* Willsmith, Kabads, Julesd, Pavon, Vsmith, GregorB, Fish and karate, Barefootliam, Kanet77, IE, Lambiam, Kvng, Shirt58, Swpb, Zoombody, Vahid shirvani, Jculverhouse, Rhododendrites, AlanPater, AnomieBOT, Xqbot, I dream of horses, Checkingfax, Bigbigreader, BG19bot, Vmen1, EpicWolverine, SFK2, BB609, Brentsalter, Johnnysnel, TechHead7, CraigyDavi, Azealia911, Nt200, Wrtr63, The2faman, Nmenovic and Anonymous: 20
- **Authorization (computer access control)** *Source:* [https://en.wikipedia.org/wiki/Authorization_\(computer_access_control\)?oldid=681286666](https://en.wikipedia.org/wiki/Authorization_(computer_access_control)?oldid=681286666) *Contributors:* Michael Hardy, Pnm, Mulad, PBS, Lowellian, Wikibot, Mark.murphy, Mark T, Tagishsimon, ArnoldReinhold, Lectonar, Cjcollier, Dave.Dunford, Ringbang, HenryLi, Rodii, Mindmatrix, BD2412, BMF81, RobotE, RussBot, Ru.spider, Gaius Cornelius, Theshadow27, Rupert Clayton, Bota47, Deville, GraemeL, SmackBot, Betacommand, JonHarder, MichaelBillington, Tjcoppet, Luís Felipe Braga, 16@r, TwistOfCain, Rgr0f, The Transhumanist, Emerauda, Metrax, NewEnglandYankee, Althepal, VolkovBot, Philip True-man, TXiKiBoT, MrMelonhead, Mazarin07, Jamelan, Jaap3, Cnilep, SieBot, Gerakibot, Sean.hoyland, Josang, Niceguyedc, Socrates2008, SoxBot III, Ali farjami~enwiki, Dsimic, Addbot, Lightbot, Luckas-bot, Yobot, MacTire02, Materialscientist, Joining a hencing, Jordav, JimVC3, GrouchoBot, Nageh, Edajgi76, Danbalam, Hnguyen322, EmausBot, WikitanvirBot, Timtempleton, Akjar13, Sam Tomato, Winner 42, Shengzhongxie, BattyBot, ChrisGualtieri, Kupiakos, Crystallizedcarbon, ARIENGO ARIE and Anonymous: 56
- **Data-centric security** *Source:* https://en.wikipedia.org/wiki/Data-centric_security?oldid=691350375 *Contributors:* Idumont and Swister-Twister
- **Firewall (computing)** *Source:* [https://en.wikipedia.org/wiki/Firewall_\(computing\)?oldid=691244690](https://en.wikipedia.org/wiki/Firewall_(computing)?oldid=691244690) *Contributors:* Paul~enwiki, Nealmcb, Michael Hardy, Pnm, Egil, Ahoerstemeier, Copsewood, Haakon, Jebba, Sugarfish, RI, Dcoetzee, Jay, DJ Clayworth, Taxman, Bevo, Topbanana, Joy, Khym Chanur, Robbot, ZimZalaBim, Danutz, Auric, Jondel, Hadal, Diberri, Tobias Bergemann, Pabouk, Giftlite,

Yama, Everyking, Rchandra, AlistairMcMillan, Eequor, Matthäus Wander, Wiki Wikardo, DemonThing, Wmahan, Stevietheman, ConradPino, Antandrus, Ricky~enwiki, Mitaphane, Biot, Deewiant, Joyous!, HaxOrw4ng, 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, Sfrier, Dmccreary, Gurch, DevastatorIIC, Intgr, Alphachimp, OpenToppedBus, Ahunt, Marcuswittig, DVdm, FeldBum, Bgwhite, Theymos, YurikBot, Wavelength, Borgx, TexasAndroid, Quentin X, Sceptre, Alan216, MMuzammils, RussBot, Mattgibson, Lincolnite, Piet Delport, Stephenb, Manop, Rsrikanth05, Wimt, Capi, NawlinWiki, ENeville, Trevor1, Rebel, Mortein, Cryptosmith, Jpbowen, Voidxor, Bkil, Zwobot, Bucketsofg, Black Falcon, Mcicogni, CraigB, Nlu, Wknight94, Rxwxrwx, Dse, JonnyJinx, Closedmouth, E Wing, Pb30, ILRainyday, Chriswaterguy, Talyian, Cffrost, Anclation~enwiki, Maxamegalon2000, Bswilson, A13ean, SmackBot, Unschool, Rbmennut, 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 jr, Megaboz, JohnCD, Topsplnslams, Kgentryjr, Random name, Lazulilasher, WeggeBot, Josemi, Nnp, Equendil, Phatom87, Cydebot, T Houdijk, Mashby, UncleBubba, Gogo Dodo, Tbird1965, Hamzanaqvi, Guitardemon666, Irate, Omicronperse8, Thijs!bot, Danhm, Epbr123, Barticus88, Kubanczyk, Dschrader, Pajz, Randilyn, Simeon H, Marek69, SGHH, Chrisdab, CharlotteWebb, Wai Wai, 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, 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, Jiges, 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, VasiliievVV, Venom8599, Philip Trueman, Apy886, Jackrockstar, Cedric dblb, Ulrichlang, OlavN, Anna Lincoln, Corvus cornix, David.bar, Sanfrannman59, Justin20, LeaveSleaves, Seb az86556, Lolsalad, 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, Goody-husband, 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, Nicceguyedc, Blanchardb, Harland1, ChandlerMapBot, Bencejoful, Jusdafax, Tim874536, Dcampbell30, Estrirabot, Shiro jdn, Aurora2698, Peter.C, Mxbuck, Creed1928, ChrisHodgesUK, BOTarate, La Pianista, 9Nak, Aitias, Apparition11, Vanished user uih38riiw4hjlsd, Sensiblekid, DumZiBoT, BarretB, Wordwizz, Gnowor, Booster4324, Gonzonoir, Rror, NelleBly, 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, Kran0, June, Bluebusy, WikiDreamer Bot, Shawnj99, Luckas-bot, Yobot, Terronis, Fraggie81, Amirobot, Fightingirishfan, AnomieBOT, JDavis680, Jlavepoze, Tcosta, Killiondude, Jim1138, Gascreed, Piano non troppo, Elieb001, Gc9580, Fahadsadah, Kyleflaherty, Flewis, Materialscientist, Citation bot, Aneah, Neurolysis, Obersachsebot, Xqbot, TheAMollusc, 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, MrBen-Cai, December21st2012Freak, Cougar w, Weyling, Danshelb, TobeBot, WilliamSun, FunkyBike1, Vrenator, Clarkcj12, Stephanman882, 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, K6ka, Aejr120, Shujipzv3, Athn, Ebambot, Kandarp.pande.kandy, Sg313d, Cit helper, IntelligentComputer, Rafiwiki, OisinisiO, NTox, Cubby-house, Zabanio, DASHBotAV, Sepersann, 28bot, Socialservice, ClueBot NG, AAriel42, Lord Roem, Vakanuvis789, 123Hedgehog456, Vlhsrp, Wldr, Johnny C. Morse, Debby5.0, HMSSolent, Titodutta, Kanwar47, Wbm1058, Wiki13, Silvrous, Dentalplanlisa, Zune0112, Paulwray97, Nperrakis, Klilidiplomus, Sk8erPrince, Cimorus, Fastcatz, CGuerrero-NJITWILL, Cvarta, PhilipFoulkes, Dexbot, Sendar, SimonWiseman, Codename Lisa, Avinash7075, Pete Mahen, CaSJer, Jamesx12345, Rob.bosc, VikiED, Palmbeachguy, Epicgenius, Camayoc, Melonkelon, Anupasinha.20, Praemonitus, SamoaBot, EvergreenFir, Indiesingh, Ginsulof, ScotXW, Harshad1310, Nyashinski, Monkbot, The Original Filfi, Darshansham, WikiGopi, Jeremy.8910, Kenkutengu, AMLIMSON, Miraclexix, Jerodlycett, KasparBot, ProprioMe OW, Risc64, Natasha Miranda, Radhwann, 0600K078 and Anonymous: 980

- **Intrusion detection system** Source: https://en.wikipedia.org/wiki/Intrusion_detection_system?oldid=687853987 Contributors: Mav, Harperf, Michael Hardy, Willsmith, (, Ellywa, Ronz, Julesd, Smatty, Schnellocke, Hashar, Joy, Astronautics~enwiki, Tbutzon, Wereon, Tobias Bergemann, 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, Joonga, Borgx, Michael@thelander.com, Wolfmankurd, Boonebytes, Msoos, Falcon9x5, Abune, Josh3580, LeonardoRob0t, NeilN, SimmondP, SmackBot, Mmernex, Charlesrh, Unforgettableid, Jprg1966, Mhecht, Frap, JonHarder, Midnightcomm, UU, Grover cleveland, Radagast83, Wegerbil, Kolmigabrouil, Drphilharmonic, DMacks, Kuru, Its-geneb, Tomhubbard, MarkSutton, Slakr, Yms, Visnum, EdC~enwiki, FleetCommand, Sir Vicious, Difference engine, Cydebot, Mikebrand, Studerby, Cs califonia, Malleus Fatuorum, Jojan, AndreasWittenstein, JAnDbot, PhilKnight, Caffeinepuppy, BigChicken, Ronbarak, Pete Wall, CliffC, Tamer ih~enwiki, CraigMonroe, Haiauphixu, NewEnglandYankee, Sgorton, Derekrogerson, Wilson.canadian, VolkovBot, Tburket, Butseriouslyfolks, TXiKiBot, Qxz, Ferengi, Clangin, Nitin.skd, Softtest123, Madhero88, Finity, Hotmixclass, Dremeda, Phishes, Shahmirj, Jerryobject, SouthLake, Jasonhatwki, ClueBot, Lokipro, PolarYukon, Foss guy, Anubis1055, Gunnar Kreitz, Crazy-ChemGuy, Dcampbell30, Footballfan190, SoxBot III, Jovianeye, Tackat, Addbot, Some jerk on the Internet, Ronhjones, J7387438, MrOlillie, Chrismcnab, Oakleeman, Bond0088, Yobot, Omarmalal, AnomieBOT, Jim1138, Hiihammuk, Materialscientist, Aneah, Xqbot, Capricorn42, Kpcatch6, Jesse5656, Kernel.package, Locobot, Shadowjams, Aghsajiy, Weyesr1, Access-bb, RedBot, Banej, Lotje, Offnfopt, Autumnnalmonk, 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, Wbm1058, BG19bot, NewsAndEventsGuy, PhnomPencil, RobVeggett, Archos331, Baszerr, Cristianocosta, Mogism, Cerabot~enwiki, Bgibbs2, Namnatulco, Spicyitalianmeatball, Frosty, SFK2, Jose Manuel Caballero, KBhokray, Rajamca66, Acc12345acc, Rsschomburg, Drkhataniar, Monkbot, Sofia Koutsouveli, S166865h, Malayks, 115ash, Crystallizedcarbon, Jchango301, Ahmed embedded and Anonymous: 300

- **Intrusion prevention system** *Source:* https://en.wikipedia.org/wiki/Intrusion_prevention_system?oldid=652729170 *Contributors:* The Anome, Yaronf, Jiang, Pieterclaassen, Topbanana, Robbot, Astronautics~enwiki, Wmahan, Hellisp, Bluefoxicity, CanisRufus, Espoo, Dran-gon, Cjcollier, Ringbang, Rtcpenguin, Jclemens, Rjwilmsi, ElKevbo, CarolGray, Timpala, Melancholie, Chobot, YurikBot, Tony1, Zwobot, BOT-Superzero cool, DeadEyeArrow, Werdna, CraigB, AmpleFire, Fsiler, Michael Pietroforte, KnightRider~enwiki, SmackBot, Jurgen-Hadley, Ohnoitsjamie, Jeffnathan, Oli Filth, Onceier, Frap, JonHarder, A5b, DSV~enwiki, Kuzaar, M3rc3r0, Kuru, Arman4, Noah Salzman, IReceivedDeathThreats, Dylanmathews~enwiki, Marysunshine, JLCA, Artemgy, Unixguy, CmdrObot, Raysonho, Gmctaggart, Equendil, Tbird1965, TheJC, Thijis!bot, Loudsox, Drgigglesmagoo, QuiteUnusual, BenTremblay, Rjathar, Davidcl, AndreasWittenstein, Captainlegacy, Raanoo, Ariel@compucall, Starfighter~enwiki, Aka042, VegKilla, R'n'B, Shyaam007, SJP, Toon05, STBotD, Ng8838, Jramsey, Greg gti, Mikeblyth, Clangin, Eriberto mota, Weirdarms, ManjuShrestha, SieBot, Tresiden, IllusiveMonkey, Martarius, Clue-Bot, Mild Bill Hiccup, Lokipro, PolarYukon, Neilhenry, Kirklanda, Andybmondeo, Kellerpm, Addbot, Xp54321, Itsjustaj, Fluffernutter, Mike A Quinn, Pruzicka, Amirobot, Jcraft50, Piano non troppo, Coret3z, Kyleflaherty, Materialscientist, Aneah, Xqbot, Kpcatch6, Mag-ixcian, Impakti, Solphusion~enwiki, SCARECROW, Saliari2, Jamieicitra, Breadtk, Mniesenhmb, Lescarolnyc, Kalki101, Madridclm, Banej, Yalon, Acatyes, Mjdtjm, Samance, Miles Montgomery, Mahdi.s.m, Sepersann, ClueBot NG, MerllwBot, Helpful Pixie Bot, Murat-TURUNCOGLU07, Ikt123, BattyBot, Kingston Dominik, Cerabot~enwiki, Faizan, EvergreenFir, Awais01khan, Nyashinski, Daveginsburg and Anonymous: 189
- **Mobile secure gateway** *Source:* https://en.wikipedia.org/wiki/Mobile_secure_gateway?oldid=689451198 *Contributors:* Kvng, Yobot, Champion, BG19bot, BattyBot, Alesteska, P.enderle23 and Anonymous: 3

5.2 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-T10-ArchitectureDiagram.png** *Source:* <https://upload.wikimedia.org/wikipedia/commons/8/86/2010-T10-ArchitectureDiagram.png> *License:* CC BY-SA 3.0 *Contributors:* <http://www.owasp.org/index.php/File:2010-T10-ArchitectureDiagram.png> *Original artist:* Neil Smithline
- **File:Access_control_door_wiring.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/1/10/Access_control_door_wiring.png *License:* Public domain *Contributors:* Own work *Original artist:* Andriusval
- **File:Access_control_door_wiring_io_module.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/f/f2/Access_control_door_wiring_io_module.png *License:* Public domain *Contributors:* Own work *Original artist:* Andriusval
- **File:Access_control_topologies_IP_controller.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/5/5b/Access_control_topologies_IP_controller.png *License:* Public domain *Contributors:* Own work *Original artist:* Andriusval
- **File:Access_control_topologies_IP_master.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/9/95/Access_control_topologies_IP_master.png *License:* Public domain *Contributors:* Own work *Original artist:* Andriusval
- **File:Access_control_topologies_IP_reader.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/e/ef/Access_control_topologies_IP_reader.png *License:* Public domain *Contributors:* Midpoint Systems, UAB *Original artist:* Andriusval
- **File:Access_control_topologies_main_controller_a.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/8/8f/Access_control_topologies_main_controller_a.png *License:* Public domain *Contributors:* Own work *Original artist:* Andriusval
- **File:Access_control_topologies_main_controller_b.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/a/a5/Access_control_topologies_main_controller_b.png *License:* Public domain *Contributors:* Own work *Original artist:* Andriusval
- **File:Access_control_topologies_serial_controllers.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/e/e0/Access_control_topologies_serial_controllers.png *License:* Public domain *Contributors:* Own work *Original artist:* Andriusval
- **File:Access_control_topologies_terminal_servers.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/a/ac/Access_control_topologies_terminal_servers.png *License:* Public domain *Contributors:* Own work *Original artist:* Andriusval
- **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:Ambox_rewrite.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/1/1c/Ambox_rewrite.svg *License:* Public domain *Contributors:* self-made in Inkscape *Original artist:* penubag
- **File:Audio_drill.jpg** *Source:* https://upload.wikimedia.org/wikipedia/commons/a/a8/Audio_drill.jpg *License:* Public domain *Contributors:* ? *Original artist:* ?
- **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:CIAJMK1209.png** *Source:* <https://upload.wikimedia.org/wikipedia/commons/9/9a/CIAJMK1209.png> *License:* CC BY-SA 3.0 *Contributors:* Made and uploaded by John Manuel - JMK. *Original artist:* John M. Kennedy T.
- **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: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 Mauron

- **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; Transfer was stated to be made by User:Mardus. *Original artist:* Original uploader was SF007 at en.wikipedia
- **File:Commons-logo.svg** *Source:* <https://upload.wikimedia.org/wikipedia/en/4/4a/Commons-logo.svg> *License:* ? *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:Crypto_key.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/6/65/Crypto_key.svg *License:* CC-BY-SA-3.0 *Contributors:* Own work based on image:Key-crypto-sideways.png by MisterMatt originally from English Wikipedia *Original artist:* MesserWoland
- **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:Defense_In_Depth_-_Onion_Model.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/4/4c/Defense_In_Depth_-_Onion_Model.svg *License:* CC BY-SA 3.0 *Contributors:* Own work *Original artist:* K. Bolino
- **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:Emoji_u1f4bb.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/d/d7/Emoji_u1f4bb.svg *License:* Apache License 2.0 *Contributors:* <https://code.google.com/p/noto/> *Original artist:* Google
- **File:Encryption_-_decryption.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/b/bf/Encryption_-_decryption.svg *License:* CC-BY-SA-3.0 *Contributors:* based on png version originally uploaded to the English-language Wikipedia by mike40033, and moved to the Commons by MichaelDiederich. *Original artist:* odder
- **File:EvilTwinWireless.jpg** *Source:* <https://upload.wikimedia.org/wikipedia/commons/6/66/EvilTwinWireless.jpg> *License:* CC BY-SA 3.0 *Contributors:* Own work *Original artist:* Thomas.Baguette
- **File:Fiber_optic_tap.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/f/fc/Fiber_optic_tap.png *License:* GFDL *Contributors:* Own work *Original artist:* Roens
- **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:Fob-at-proximity-reader_532_130xauto.jpg** *Source:* https://upload.wikimedia.org/wikipedia/commons/c/cf/Fob-at-proximity-reader_532_130xauto.jpg *License:* CC BY-SA 3.0 *Contributors:* Own work Template:Self photograph Template:[Http://www.accesscontrol.ie/products/category/pro/](http://www.accesscontrol.ie/products/category/pro/) *Original artist:* Mgavenda
- **File:GatewayTracingHologramLabel.jpg** *Source:* <https://upload.wikimedia.org/wikipedia/commons/d/da/GatewayTracingHologramLabel.jpg> *License:* CC BY-SA 3.0 *Contributors:* self-made <http://www.tmg.com.vn> *Original artist:* Newone <http://www.tmg.com.vn>
- **File:Graphiquemalware.jpeg** *Source:* <https://upload.wikimedia.org/wikipedia/commons/0/07/Graphiquemalware.jpeg> *License:* CC BY-SA 3.0 *Contributors:* Own work *Original artist:* Thomas.Baguette
- **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:Henri_Adolphe_Laissement_Kardin%C3%A4le_im_Vorzimmer_1895.jpg** *Source:* https://upload.wikimedia.org/wikipedia/commons/7/7b/Henri_Adolphe_Laissement_Kardin%C3%A4le_im_Vorzimmer_1895.jpg *License:* Public domain *Contributors:* Hampel Kunstaustionen *Original artist:* Henri Adolphe Laissement (1854-1921)
- **File:ID-card-spain-(01).png** *Source:* <https://upload.wikimedia.org/wikipedia/commons/2/23/ID-card-spain-%2801%29.png> *License:* CC SA 1.0 *Contributors:* ? *Original artist:* ?
- **File:Intelligent_access_control_door_wiring.PNG** *Source:* https://upload.wikimedia.org/wikipedia/commons/5/53/Intelligent_access_control_door_wiring.PNG *License:* Public domain *Contributors:* Own work *Original artist:* Andriusval
- **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:Locksmiths-11211.jpg** *Source:* <https://upload.wikimedia.org/wikipedia/commons/7/7a/Locksmiths-11211.jpg> *License:* CC0 *Contributors:* <http://nyclocksmithbrooklyn.com/security-systems/fast-locksmith-anytime-locks/> *Original artist:* Locksmithwilli
- **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_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:Merge-arrow.svg** *Source:* <https://upload.wikimedia.org/wikipedia/commons/a/aa/Merge-arrow.svg> *License:* Public domain *Contributors:* ? *Original artist:* ?
- **File:Merge-arrows.svg** *Source:* <https://upload.wikimedia.org/wikipedia/commons/5/52/Merge-arrows.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:Monitor_padlock.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/7/73/Monitor_padlock.svg *License:* CC BY-SA 3.0 *Contributors:* Transferred from en.wikipedia; transferred to Commons by User:Logan using CommonsHelper. *Original artist:* Lunarbunny (talk). Original uploader was Lunarbunny at en.wikipedia
- **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: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:* Jengelh
- **File:NewYorkCitySubwayEntranceInterior.jpg** *Source:* <https://upload.wikimedia.org/wikipedia/commons/5/55/NewYorkCitySubwayEntranceInterior.jpg> *License:* CC BY 3.0 *Contributors:* Own work *Original artist:* Canadaolympic989
- **File:PalmPilot5000.jpg** *Source:* <https://upload.wikimedia.org/wikipedia/commons/e/e7/PalmPilot5000.jpg> *License:* CC-BY-SA-3.0 *Contributors:* en.wikipedia.org: 00:35, 3. Jul 2004 . . Mfatic . . 220x296 (12719 Byte) (Palm Pilot 5000)

This template will categorize into Category:Wikipedia license migration candidates. *I dug this out of a drawer and took the photo myself.)* *Original artist:* Mfatic

- **File:PersonalStorageDevices.agr.jpg** *Source:* <https://upload.wikimedia.org/wikipedia/commons/8/87/PersonalStorageDevices.agr.jpg> *License:* CC-BY-SA-3.0 *Contributors:* I took this photograph of artifacts in my possession *Original artist:* --agr 15:53, 1 Apr 2005 (UTC)
- **File:Physical_security_access_control_with_a_fingerprint_scanner.jpg** *Source:* https://upload.wikimedia.org/wikipedia/commons/8/8b/Physical_security_access_control_with_a_fingerprint_scanner.jpg *License:* CC BY 3.0 *Contributors:* Own work *Original artist:* Lgat74
- **File:Privacy_International_2007_privacy_ranking_map.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/2/2c/Privacy_International_2007_privacy_ranking_map.png *License:* CC BY-SA 3.0 *Contributors:* own work, based on Image:Privacy_International_2006_privacy_ranking_map.png *Original artist:* Wüstling
- **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; transferred to Commons by User:IngerAlHaosului using CommonsHelper. *Original artist:* Original uploader was CyberSkull at en.wikipedia. Later version(s) were uploaded by Eliashc at en.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:Scale_of_justice_2.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/0/0e/Scale_of_justice_2.svg *License:* Public domain *Contributors:* Own work *Original artist:* DTR
- **File:SecureID_token_new.JPG** *Source:* https://upload.wikimedia.org/wikipedia/commons/8/8f/SecureID_token_new.JPG *License:* Public domain *Contributors:* Own work *Original artist:* Ocrho
- **File:Space_Saver_Drop_Arm_Turnstiles.jpg** *Source:* https://upload.wikimedia.org/wikipedia/commons/4/4e/Space_Saver_Drop_Arm_Turnstiles.jpg *License:* CC BY-SA 4.0 *Contributors:* Own work *Original artist:* Ed Jacobsen
- **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:System-installer.svg** *Source:* <https://upload.wikimedia.org/wikipedia/commons/d/db/System-installer.svg> *License:* Public domain *Contributors:* The Tango! Desktop Project *Original artist:* The people from the Tango! project
- **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:* Transferred from en.wikipedia; transferred to Commons by User:Faigl.ladislav using CommonsHelper.
Original artist: tkgd2007. Original uploader was Tkgd2007 at en.wikipedia
- **File:U.S._Navy_Cyber_Defense_Operations_Command_monitor.jpg** *Source:* https://upload.wikimedia.org/wikipedia/commons/d/d6/U.S._Navy_Cyber_Defense_Operations_Command_monitor.jpg *License:* Public domain *Contributors:* <http://www.navy.mil/management/photodb/photos/081203-N-2147L-390.jpg> *Original artist:* Mass Communications Specialist 1st Class Corey Lewis , U.S. Navy
- **File:US_Navy_050308-N-2385R-029_Master-at-Arms_Seaman_Carly_Farmer_checks_an_identification_card_(ID)_before_allowing_a_driver_to_enter_the_gate_at_U.S._Fleet_Activities_Sasebo,_Japan.jpg** *Source:* https://upload.wikimedia.org/wikipedia/commons/4/42/US_Navy_050308-N-2385R-029_Master-at-Arms_Seaman_Carly_Farmer_checks_an_identification_card_%28ID%29_before_allowing_a_driver_to_enter_the_gate_at_U.S._Fleet_Activities_Sasebo%2C_Japan.jpg *License:* Public domain *Contributors:*
This Image was released by the United States Navy with the ID 050308-N-2385R-029 (next). This tag does not indicate the copyright status of the attached work. A normal copyright tag is still required. See Commons:Licensing for more information.
Original artist: U.S. Navy photo by Photographer's Mate 3rd Class Yesenia Rosas
- **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_cropped.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/1/1c/Wiki_letter_w_cropped.svg *License:* CC-BY-SA-3.0 *Contributors:*
- **Wiki_letter_w.svg** *Original artist:* Wiki_letter_w.svg: Jarkko Piironen
- **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-en.svg** *Source:* <https://upload.wikimedia.org/wikipedia/commons/f/f8/Wiktionary-logo-en.svg> *License:* Public domain *Contributors:* Vector version of [Image:Wiktionary-logo-en.png](#). *Original artist:* Vectorized by Fvasconcellos (talk · contribs), based on original logo tossed together by Brion Vibber

5.3 Content license

- Creative Commons Attribution-Share Alike 3.0