Contents

Introduction

Security

C*omputer security is a branch of computer technology known as information security as applied to computers and computer networks. The objective of computer security includes protection of information and property from theft, corruption, or natural disaster, while allowing the information and property to remain accessible and productive to its intended users. The term computer system security means the collective processes and mechanisms by which sensitive and valuable information and services are protected from publication, tampering or collapse by unauthorized activities or untrustworthy individuals and unplanned events respectively. The strategies and methodologies of computer security often differ from most other computer technologies because of its somewhat elusive objective of preventing unwanted computer behavior instead of enabling wanted computer behavior.*

The technologies of computer security are based on logic. As security is not necessarily the primary goal of most computer applications, designing a program with security in mind often imposes restrictions on that program's behavior.

There are 4 approaches to security in computing, sometimes a combination of approaches is valid:

Trust all the software to abide by a security policy but the software is not trustworthy (this is computer insecurity).

Trust all the software to abide by a security policy and the software is validated as trustworthy (by tedious branch and path analysis for example).

Trust no software but enforce a security policy with mechanisms that are not trustworthy (again this is computer insecurity).

Trust no software but enforce a security policy with trustworthy hardware mechanisms.

Many systems have unintentionally resulted in the first possibility. Since approach two is expensive and non-deterministic, its use is very limited. Approaches one and three lead to failure. Because approach number four is often based on hardware mechanisms and avoids abstractions and a multiplicity of degrees of freedom, it is more practical. Combinations of approaches two and four are often used in a layered architecture with thin layers of two and thick layers of four.

There are various strategies and techniques used to design security systems. However, there are few, if any, effective strategies to enhance security after design. One technique enforces the principle of least privilege to great extent, where an entity has only the privileges that are needed for its function. That way even if an attacker gains access to one part of the system, fine-grained security ensures that it is just as difficult for them to access the rest.

Furthermore, by breaking the system up into smaller components, the complexity of individual components is reduced, opening up the possibility of using techniques such as automated theorem proving to prove the correctness of crucial software subsystems. This enables a closed form solution to security that works well when only a single well-characterized property can be isolated as critical, and that property is also assessable to math. Not surprisingly, it is impractical for generalized correctness, which probably cannot even be defined, much less proven. Where formal correctness proofs are not possible, rigorous use of code review and unit testing represent a best-effort approach to make modules secure.

The design should use "defense in depth", where more than one subsystem needs to be violated to compromise the integrity of the system and the information it holds. Defense in depth works when the breaching of one security measure does not provide a platform to facilitate subverting another. Also, the cascading principle acknowledges that several low hurdles does not make a high hurdle. So cascading several weak mechanisms does not provide the safety of a single stronger mechanism.

Subsystems should default to secure settings, and wherever possible should be designed 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.

In addition, security should not be an all or nothing issue. The designers and operators of systems should assume that security breaches are inevitable. Full audit trails should be kept of 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. Finally, full disclosure helps to ensure that when bugs are found the "window of vulnerability" is kept as short as possible.

Cryptography

C*ryptography is the practice and study of techniques for secure communication in the presence of third parties (called adversaries). More generally, it is about constructing and analyzing protocols that overcome the influence of adversaries and which are related to various aspects in information security such as data confidentiality, data integrity, authentication, and non-repudiation. Modern cryptography intersects the disciplines of mathematics, computer science, and electrical engineering. Applications of cryptography include ATM cards, computer passwords, and electronic commerce.*

Cryptography prior to the modern age was effectively synonymous with encryption, the conversion of information from a readable state to apparent nonsense. The originator of an encrypted message shared the decoding technique needed to recover the original information only with intended recipients, thereby precluding unwanted persons to do the same. Since World War I and the advent of the computer, the methods used to carry out cryptology have become increasingly complex and its application more widespread.

Modern cryptography is heavily based on mathematical theory and computer science practice; cryptographic algorithms are designed around computational hardness assumptions, making such algorithms hard to break in practice by any adversary. It is theoretically possible to break such a system but it is infeasible to do so by any known practical means. These schemes are therefore termed computationally secure; theoretical advances (e.g., improvements in integer factorization algorithms) and faster computing technology require these solutions to be continually adapted. There exist information-theoretically secure schemes that provably cannot be broken even with unlimited computing power—an example is the one-time pad—but these schemes are more difficult to implement than the best theoretically breakable but computationally secure mechanisms.

Cryptology-related technology has raised a number of legal issues. In the United Kingdom, additions to the Regulation of Investigatory Powers Act 2000 require a suspected criminal to hand over their encryption key if asked by law enforcement. Otherwise the user will face a criminal charge.The Electronic Frontier Foundation (EFF) is involved in a case in the Supreme Court of the United States, which may determine whether requiring suspected criminals to provide their encryption keys to law enforcement is unconstitutional. The EFF is arguing that this is a violation of the right of not being forced to incriminate oneself, as given in the fifth amendment.

Gary McKinnon

G*ary McKinnon (born 10 February 1966) is a Scottish systems administrator and hacker who was accused in 2002 of perpetrating the "biggest military computer hack of all time," although McKinnon himself – who has a diagnosis of Asperger's Syndrome – states that he was merely looking for evidence of free energy suppression and a cover-up of UFO activity and other technologies potentially useful to the public. On 16 October 2012, after a series of legal proceedings in Britain, Home Secretary Theresa May withdrew his extradition order to the United States.*

McKinnon is accused of hacking into 97 United States military and NASA computers over a 13-month period between February 2001 and March 2002, at his girlfriend's aunt's house in London, using the name 'Solo'.

The US authorities claim he deleted critical files from operating systems, which shut down the US Army's Military District of Washington network of 2,000 computers for 24 hours. McKinnon also posted a notice on the military's website: "Your security is crap". After the September 11 attacks, he deleted weapons logs at the Earle Naval Weapons Station, rendering its network of 300 computers inoperable and paralyzing munitions supply deliveries for the US Navy's Atlantic Fleet. McKinnon is also accused of copying data, account files and passwords onto his own computer. US authorities claim the cost of tracking and correcting the problems he caused was over $700,000.

While not admitting that it constituted evidence of destruction, McKinnon did admit leaving a threat on one computer:

US foreign policy is akin to Government-sponsored terrorism these days … It was not a mistake that there was a huge security stand down on September 11 last year … I am SOLO. I will continue to disrupt at the highest levels …

US authorities claim that McKinnon is trying to downplay his own actions. A senior military officer at the Pentagon told The Sunday Telegraph: "US policy is to fight these attacks as

strongly as possible. As a result of Mr McKinnon's actions, we suffered serious damage. This was not some harmless incident. He did very serious and deliberate damage to military and Nasa computers and left silly and anti-America messages. All the evidence was that someone was staging a very serious attack on US computer systems."

McKinnon has admitted in many public statements that he obtained unauthorised access to computer systems in the United States including those mentioned in the United States indictment. He claims his motivation, drawn from a statement made before the Washington Press Club on 9 May 2001 by "The Disclosure Project", was to find evidence of UFOs, antigravity technology, and the suppression of "free energy", all of which he claims to have proven through his actions.

In an interview televised on the BBC's Click programme, McKinnon claimed that he was able to get into the military's networks simply by using a Perl script that searched for blank passwords; in other words his report suggests that there were computers on these networks with the default passwords active.

In his interview with the BBC, he also claimed of "The Disclosure Project" that "they are some very credible, relied-upon people, all saying yes, there is UFO technology, there's anti-gravity, there's free energy, and it's extraterrestrial in origin and [they've] captured spacecraft and reverse engineered it." He said he investigated a NASA photographic expert's claim that at the Johnson Space Center's Building 8, images were regularly cleaned of evidence of UFO craft, and confirmed this, comparing the raw originals with the "processed" images. He claimed to have viewed a detailed image of "something not man-made" and "cigar shaped" floating above the northern hemisphere, and assuming his viewing would be undisrupted owing to the hour, he did not think of capturing the image because he was "bedazzled", and therefore did not think of securing it with the screen capture function in the software at the point when his connection was interrupted.

In 2006, a Freedom of Information Act request was filed with NASA for all documents pertaining to Gary McKinnon. NASA's documents consisted of printed news articles from the Slashdot website, but no other related documents. This is consistent with NASA employees browsing internet articles about Gary McKinnon; the records of such browsing activity are in the public domain.

The FOIA documents have been uploaded to the internet for review, and can be downloaded.

On 12 December 2007, BBC Radio 4 broadcast a 45-minute radio play about the case, The McKinnon Extradition by John Fletcher. It was re-broadcast on 2 September 2008. It was directed by Pete Atkin and produced by David Morley.

Blue-Jacking

B*luejacking is the sending of unsolicited messages over Bluetooth to Bluetooth-enabled devices such as mobile phones, PDAs or laptop computers, sending a vCard which typically contains a message in the name field (i.e., for bluedating or bluechat) to another bluetooth enabled device via the OBEX protocol.*

Bluetooth has a very limited range, usually around 10 metres (32.8 ft) on mobile phones, but laptops can reach up to 100 metres (328 ft) with powerful (Class 1) transmitters.

Origins

Bluejacking was reportedly first carried out by a Malaysian IT consultant who used his phone to advertise Sony Ericsson. He also invented the name, which purports to be an amalgam of Bluetooth and ajack, his username on Esato, a Sony Ericsson fan online forum. Jacking is, however, an extremely common shortening of hijack, the act of taking over something.

Usage

Bluejacking is usually harmless, but because bluejacked people generally don't know what has happened, they may think that their phone is malfunctioning. Usually, a bluejacker will only send a text message, but with modern phones it's possible to send images or sounds as well. Bluejacking has been used in guerrilla marketing campaigns to promote advergames.

With the increase in the availability of Bluetooth enabled devices, it is often reported that devices have become vulnerable to virus attacks and even complete take over of devices through a trojan horse program although most of these reports are easily debunked.

Bluejacking is also confused with Bluesnarfing which is the way in which mobile phones are illegally hacked via Bluetooth.

Bluejacking tools and software

Many tools have been developed for bluejacking. Most of the development happened in the 2000 to 2004, where multiple new bluetooth vulnerabilities were discovered. Most of these tools are developed by individual developers and have very specific functions. While there are many tools to assist someone in bluejacking, only a few hidden tools are available for the more sinister bluesnarfing or bluebugging. These are usually internal trade secrets which the experts guard earnestly.

One example is bluesniff, which seeks out hidden bluetooth devices. One of the most commonly used bluetooth software is bloover, which is in version 2 now. It allows users to seek then send unsolicited messages to unwary bluetooth devices.

Given the fact that most Bluetooth devices present a confirmation dialog when a remote device tries to connect, it is possible to achieve another form of Bluejacking by setting the unsolicited message as the friendly name of the Bluejacking device. For example[5] a Bluetooth device can be renamed as "You're being watched!" and then when connecting to another Bluetooth device it will provide this name and so the user will see it.

Virtual private network

A virtual private network (VPN) is a technology for using the Internet or another intermediate network to connect computers to isolated remote computer networks that would otherwise be inaccessible. A VPN provides varying levels of security so that traffic sent through the VPN connection stays isolated from other computers on the intermediate network, either through the use of a dedicated connection from one "end" of the VPN to the other, or through encryption. VPNs can connect individual users to a remote network or connect multiple networks together.

For example, users may use a VPN to connect to their work computer terminal from home and access their email, files, images, etc.

Through VPNs, users are able to access resources on remote networks, such as files, printers, databases, or internal websites. VPN remote users get the impression of being directly connected to the central network via a point-to-point link.

Types of VPN

VPNs can be either remote-access (connecting an individual computer to a network) or site-to-site (connecting two networks together). In a corporate setting, remote-access VPNs allow employees to access their company's intranet from home or while traveling outside the office, and site-to-site VPNs allow employees in geographically separated offices to share one cohesive virtual network. A VPN can also be used to interconnect two similar networks over a dissimilar middle network; for example, two IPv6 networks over an IPv4 network.[3]

VPN systems can be classified by:

- ***the protocols used to tunnel the traffic***
- ***the tunnel's termination point, i.e., customer edge or network-provider edge***
- ***whether they offer site-to-site or remote-access connectivity***
- ***the levels of security provided***
- ***the OSI layer they present to the connecting network, such as Layer 2 circuits or Layer 3 network connectivity***

Keyloggers

K*eystroke logging (more often called keylogging or "keyloggers") is the action of tracking (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. It also has very legitimate uses in studies of human-computer interaction. There are numerous keylogging methods, ranging from hardware and software-based approaches to electromagnetic and acoustic analysis.*

Application

Software-based keyloggers

A logfile from a software-based keylogger.

These are software programs designed to work on the target computer's operating system. From a technical perspective there are five 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: 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. 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; the operating system then notifies the keylogger each time a key is pressed and 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.[1] These types of keyloggers are the easiest to write, but where constant polling of each key is required, they can cause a noticeable increase in CPU usage, and can also miss the occasional key. A more recent example simply polls the BIOS for pre-boot authentication PINs that have not been cleared from memory.[2]

Form grabbing based: Form grabbing-based keyloggers log web form submissions by recording the web browsing onsubmit event functions. This records form data before it is passed over the Internet and bypasses HTTPS encryption.

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.[citation needed]

Packet analyzers: This involves capturing network traffic associated with HTTP POST events to retrieve unencrypted passwords.

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.

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).[3]

The recording of every program/folder/window opened including a screenshot of each and every website visited, also including a screenshot of each.

The recording of search engines queries, instant messenger conversations, FTP downloads and other Internet-based activities (including the bandwidth used).

Café-Latte Attack

The **Caffe Latte** attack was invented by me, the author of this book and was demonstrated in Toorcon 9, San Diego, USA. The Caffe Latte attack is a WEP attack which allows a hacker to retrieve the WEP key of the authorized network, using just the client. The attack does not require the client to be anywhere close to the authorized WEP network. It can crack the WEP key using just the isolated client.

Description

The Cafe Latte attack allows you to obtain a WEP key from a client system. Briefly, this is done by capturing an ARP packet from the client, manipulating it and then send it back to the client. The client in turn generates packets which can be captured by airodump-ng. Subsequently, aircrack-ng can be used to determine the WEP key.

These links provide a detailed explanation of the attack plus some ways to protect yourself from it:

Cafe Latte attack

The Caffe Latte Attack: How It Works—and How to Block It

Where did the attack name come from? The concept is that a WEP key could be obtained from an innocent client at a coffee bar in the time it takes to drink your cafe latte.

Usage

aireplay-ng -6 -h 00:09:5B:EC:EE:F2 -b 00:13:10:30:24:9C -D rausb0

Where:

-6 means Cafe-Latte attack

-h 00:09:5B:EC:EE:F2 is our card MAC address

-b 00:13:10:30:24:9C is the Access Point MAC (any valid MAC should work)

-D disables AP detection.

rausb0 is the wireless interface name

The Caffe Latte attack is another way to defeat WEP. It is not necessary for the attacker to be in the area of the network using this exploit. By using a process that targets the Windows wireless stack, it is possible to obtain the WEP key from a remote client. By sending a flood of encrypted ARP requests, the assailant takes advantage of the shared key authentication and the message modification flaws in 802.11 WEP. The attacker uses the ARP responses to obtain the WEP key in less than 6 minutes.

References

- Ross J. Anderson: *Security Engineering: A Guide to Building Dependable Distributed Systems*
- Cryptography & Network Security by William Stallings
- http://en.wikipedia.org/wiki/Gary_McKinnon
- The Autistic Hacker: Gary McKinnon hacked thousands of government computers by David Kushner
- http://en.wikipedia.org/wiki/Bluejacking
- Microsoft Technet. "Virtual Private Networking: An Overview"
- "VPN - Virtual Private Network and OpenVPN"
- http://en.wikipedia.org/wiki/Keystroke_logging
- Cormac Herley and Dinei Florencio (2006-02-06). "How To Login From an Internet Cafe Without Worrying About Keyloggers"
- "The Security Digest Archives". Retrieved 2009-11-22.
- http://www.keylogger.org/articles/silent-shadow/creating-a-keylogger-in-vb-3.html
- http://www.dmoz.org/Computers/Data_Communications/Wireless/Security/
- http://www.linksysbycisco.com/EU/en/learningcenter/HowtoSecureYourNetwork
- http://www.airtightnetworks.com/fileadmin/ppt/Toorcon.ppt (for caffe latte)
- Hacking Wireless Networks for Dummies