



Types of Attacks and Malicious Software: Avenues of Attack

Slide 1



Principles of Computer Security, Fifth Edition


Types of Attacks and Malicious Software



Avenues of Attack

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Howdy! In this video, we review avenues of attack and how to minimize them.



Principles of Computer Security, Fifth Edition

Avenues of Attack


- A computer system is attacked for one of two general reasons:
 - It is specifically targeted by an attacker.
 - Reason based on attacker's motivation
 - E.g. political, financial
 - Not reliant on target system's hardware and software
 - Difficult and take time and effort
 - It is a target of opportunity.
 - Systems with hardware or software vulnerable to a specific exploit

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Generally, computer systems are attacked either as specific targets or as targets of opportunity.

In a targeted attack, an attacker's motivation may be anything. Common reasons are typically political or financial in nature. In a targeted attack, the choice to attack does not rely on the exact hardware and software being used in the targeted system but rather proceeds in spite of these details.

Target of opportunity attacks succeed when systems are found which have not been updated with the most current security patches and are therefore vulnerable to specific exploits.



Principles of Computer Security, Fifth Edition

Minimizing Possible Avenues of Attack

- Steps to minimize possible attacks include:
 - Ensure all patches for the operating system and applications are installed.
 - Limit the services that are running on the system.
 - Limit public disclosure of private information about your organization and its computing resources.

Copyright © 2018 by McGraw-Hill Education. All rights reserved.


By understanding the steps an attacker can take, you can limit the exposure of your system and minimize the possible avenues an attacker can exploit.

Malware often exploits known vulnerabilities for which patches exist. Therefore, a first step is to make sure that the operating system and all installed applications have up-to-date patches installed. I guess this would be a good time for a public service announcement: how up-to-date is the system you are currently using? When's the last time you updated your OS and software? If you can't remember, now would be a good time to check for updates.

Next, limit the services that run on the system to only those that are absolutely necessary. This reduces the attack surface and keeps the system simpler. You probably have unnecessary services running on your machine right now.

Another step is to limit public disclosure of private information about your organization and its computing resources. Since attackers are interested in that kind of information, don't make it easy to obtain (or make it easier to obtain inaccurate or misleading information than it is to obtain accurate information).

Slide 4



Principles of Computer Security, Fifth Edition

Attribution


- The course slides are based on slides developed by the textbook authors, W. "Art" Conklin, Ph.D. (University of Houston), and Greg White, Ph.D. (University of Texas at San Antonio).
- These slides remain the intellectual property protected by US Copyright law of the authors and the textbook company.
- There have been some changes to the slide deck.

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Thank you and take care.


Types of Attacks and Malicious Software: Malware

Slide 1



Principles of Computer Security, Fifth Edition


Types of Attacks and Malicious Software



Malware

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Howdy! In this video, we discuss several examples of malware.



Principles of Computer Security, Fifth Edition

Malicious Code

- Malicious code, or **malware**, refers to software that has been designed for some nefarious purpose.
- Patching of vulnerabilities is important, for it closes the point of entry for most malware.
- Types of malicious software include:
 - Viruses, Trojan horses, logic bombs, spyware, and worms
- Malware can be fairly complex in its construction.
 - *Multipartite, polymorphic, and metamorphic*

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Malware is a portmanteau for malicious software.

Most malware exploits vulnerabilities in software, which is why keeping systems patched is so important.


There are many different kinds of malware, like viruses, trojan horses, logic bombs, spyware, and worms. They differ in the ways they are installed and their purposes.

Malware can be fairly complex in its construction, with specific features designed to assist malware in avoiding detection. Modern malware can be multipart in construction, where several pieces work together to achieve a desired effect.

When malware has multiple different objects that it specifically attacks, it is called multipartite.

Many types of malware can include a changing encryption layer to resist pattern-matching detection. These are called polymorphic.

If the malware actually changes the code at time of infection, the malware is called metamorphic.



Principles of Computer Security, Fifth Edition

Viruses

- Best-known type of malware
 - Replicates by injecting itself into executable code.
- Boot sector Virus
- Program Virus
- Macro Virus
- Avoiding Virus Infection
- Stealthy and Polymorphic Viruses
- Armored Virus
- Virus Hoaxes

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

The best-known type of malware is the virus. A virus is a piece of malicious code that replicates by injecting itself into executable code, especially code which is likely to be executed soon or often. When the other executable code is run, the virus also executes and has the opportunity to infect other files and perform any other nefarious actions it was designed to do. The specific way that a virus infects other files, and the type of files it infects, depends on the type of virus. The first viruses created were of two types: Boot sector viruses and program viruses.

A boot sector virus infects the boot sector portion of either a floppy disk or a hard drive. (Raise your hand if you remember when computers booted from a floppy disk.). When a computer is first turned on, a small portion of the operating system is initially loaded from hardware. This small operating system then attempts to load the rest of the operating system from a specific location (sector) on either the floppy or the hard drive. A boot sector virus infects this portion of the drive.

A program virus attaches itself to executable files—typically files ending in .exe or .dll on Windows-based systems. The virus is attached in such a way that it is executed before the program executes. Most viruses have a nefarious purpose, such as deleting the hard drive data, which is triggered by a specific event, such as a date, or after a certain number of other files are infected. Like other types of viruses, program viruses are often not detected until after they execute their malicious payload. One method that has been used to detect this sort of virus before it has an opportunity to damage a system is to calculate checksums for commonly used programs or utilities. Should the checksum for an executable ever change unexpectedly, it is quite likely that it is due to a virus infection.

In the late 1990s, another type of virus appeared that now accounts for the majority of viruses. As systems and operating systems became more powerful, the boot sector virus, which once accounted for most reported infections, became less common. Systems no longer commonly booted from floppies, which were the main method for boot sector viruses to spread. Instead, the proliferation of software that included macro-programming languages (like Microsoft office) resulted in a new breed of virus—the macro virus.

The popularity and success of this type of virus led to the now-ubiquitous security best-practice advice to never open a document attached to an email if it seems at all suspicious (and to be careful even when it doesn't). You may have noticed that Microsoft office will warn you before letting you open or interact with files that came from the Internet. The reason for that is due to the possibility that the file contains a macro virus that will execute as soon as you open the file. Many organizations now routinely have their mail servers eliminate any attachments containing Visual Basic macros.

To avoid virus infection, be cautious about executing programs or opening documents sent to you, install and run (and keep running) an antivirus program and keep it updated with new virus signatures through regular updates.

That being said, two advances in virus writing have made it more difficult for antivirus software to detect viruses: the Stealth virus and the Polymorphic virus.

A stealth virus employs techniques to help evade being detected and captured by antivirus software. One way it might do this is to copy itself to other files (even non-executable files) and remove itself from the infected file and then re-infect some other executable file. Basically, it makes the antivirus play a very challenging game of whack-a-mole, or whack-a-virus, in this case.


Polymorphic viruses also attempt to evade detection, but they do so by changing the virus' own executable code. Because the virus changes, signatures for that virus may no longer be valid, and the virus may escape detection by antivirus software. This is one of the key weaknesses of signature-based virus detection.

When a new form of malware/virus is discovered, antivirus companies and security researchers will decompile the program in an attempt to reverse engineer its functionality. Much can be determined from reverse engineering, such as where the malware came from, how it works, how it communicates, how it spreads, and so forth. Armoring malware can make the process of determining this information much more difficult, if not impossible. One way of doing this is to encrypt the malware. Not only does this make it harder for security researchers to analyze the malware, it also protects the malware author's intellectual property so that other malware authors cannot use it in their own products.

Viruses have caused so much damage to systems that many Internet users become extremely cautious anytime they hear a rumor of a new virus. Some very cautious people will not connect to the Internet when they hear about a virus outbreak, just to be sure their machines don't get infected. This has given rise to virus hoaxes, in which word is spread about a new virus and the extreme danger it poses. It may warn users to not read certain files or connect to the Internet. Hoaxes can actually be even more destructive than just wasting time and bandwidth. Some hoaxes warning of a dangerous virus have included instructions to delete certain files if they're found on the user's system. Unfortunately for

those who follow the advice, the files may actually be part of the operating system, and deleting them could keep the system from booting properly. This suggests another good piece of security advice: make sure of the authenticity and accuracy of any virus report before following somebody's advice. Antivirus software vendors are a good source of factual data for this sort of threat as well.

Slide 4



Principles of Computer Security, Fifth Edition

Worms

- Malware that propagates by penetrating and copying itself through a network
 - Does not require user interaction to spread
- Protection against worms
 - Depends on the type
 - Email: don't open email attachments
 - Software and Networks: patch, minimize attack surface, defend in depth
 - Sophisticated attacks: all of the above and pray

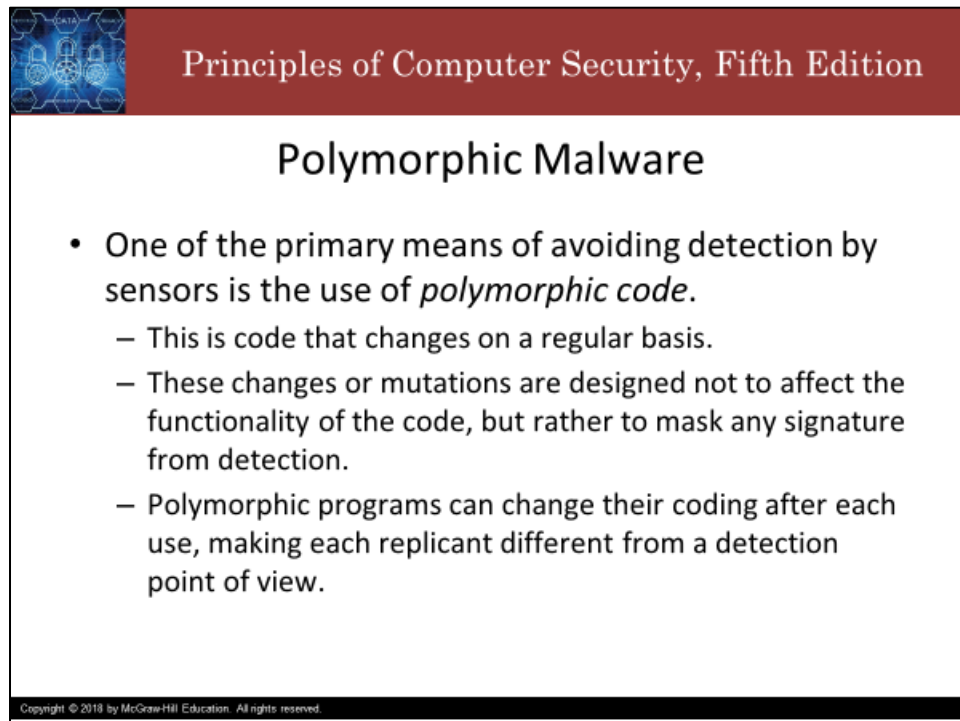
Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Worms are pieces of code that attempt to penetrate networks and computer systems. Once a penetration occurs, the worm will create a new copy of itself on the penetrated system. Reproduction of a worm does not rely on the attachment of the virus to another piece of code or to a file, which is the definition of a virus. Viruses were generally thought of as a system-based problem, and worms were

network-based. If the malicious code is sent throughout a network, it may subsequently be called a worm. The distinction between Virus and Worm is that a virus needs the user to do something in order for the virus to be executed. A worm does not require any user interaction. So, it's like a virus needs a host, but a worm can survive on its own. It was once easy to distinguish between a worm and a virus. However, recently, with the introduction of new breeds of sophisticated malicious code, the distinction has blurred.

How you protect your system against worms depends on the type of worm. Those attached and propagated through email can be avoided by following the same guidelines about not opening files and not running attachments unless you are absolutely sure of their origin and integrity. Protecting against worms involves securing systems and networks against penetration in the same way you would protect your systems against human attackers: install patches, eliminate unused and unnecessary services, enforce good password security, and use firewalls and intrusion detection systems. The most sophisticated attacks are almost impossible to avoid.

Slide 5



The slide features a dark red header with the text "Principles of Computer Security, Fifth Edition" in white. To the left of the header is a small graphic of a network with nodes and connections. The main content area is white with a black border. The title "Polymorphic Malware" is centered in a large, bold, black font. Below the title is a bulleted list with three items. The first item is a primary point, and the other two are sub-points. At the bottom of the slide, there is a small black bar with white text: "Copyright © 2018 by McGraw-Hill Education. All rights reserved."


Principles of Computer Security, Fifth Edition

Polymorphic Malware

- One of the primary means of avoiding detection by sensors is the use of *polymorphic code*.
 - This is code that changes on a regular basis.
 - These changes or mutations are designed not to affect the functionality of the code, but rather to mask any signature from detection.
 - Polymorphic programs can change their coding after each use, making each replicant different from a detection point of view.

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

The detection of malware by antimalware programs is primarily done through the use of a signature. Files are scanned for sections of code in the executable that act as markers, unique patterns of code that enable detection. Just as the human body creates antigens that match marker proteins, antimalware programs detect malware through unique markers present in the code of the malware. Malware writers are aware of this functionality and have adapted methods to defeat it, such as the use of polymorphic code. So, if antimalware programs are like your immune system, polymorphic malware is the common cold, constantly mutating to avoid detection and capture.



Principles of Computer Security, Fifth Edition

Trojan Horses

- Appears to do one thing but hides some other functionality.
 - Standalone program. Must be copied and installed by the user.
- Prevention:
 - Never run software if you are unsure of its origin, security, and integrity
 - Antivirus

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

A **Trojan** horse is software that appears to do one thing – and it may actually do that thing -- but also includes some hidden functionality. It is named after the huge wooden horse constructed by the Greeks at the end of the Trojan war which was presented to the city of Troy as a victory trophy but which in fact held a group of Greek soldiers inside who, after the horse was pulled into the city, waited until nightfall, crept out, and opened the gates to allow the rest of the Greek army to enter the city, destroy it, and end the war.


Unlike a virus, which reproduces by attaching itself to other files or programs, a Trojan is a standalone program that must be copied and installed by the user—it must be “brought inside” the system by an authorized user.

The challenge for the attacker is enticing the user to copy and run the program. This generally means that the program must be disguised as something that the user would want to run—a special utility or game, for example.

Once it has been copied and is inside the system, the Trojan will perform its hidden purpose, with the user often still unaware of its true nature.

The single best method to prevent the introduction of a Trojan to your system is never to run software if you are unsure of its origin, security, or integrity.

An antivirus program may also be useful in detecting and preventing the installation of known Trojans.




Principles of Computer Security, Fifth Edition

Remote Access Trojan (RAT)

- Provide capability for covert surveillance and/or access to a target system.
- Employ malware to infect a system with code that can be used to facilitate the exploitation of a target.
- Commonly used by highly-skilled attackers
- Principal use: enable attacker to have a way back into a system.
 - Periodic beacon

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Remote-Access Trojans (or RATs) are toolkits designed to provide the capability of covert surveillance and/or the ability to gain unauthorized access to a target system. RATs often use keyloggers or packet sniffer applications to capture all sorts of sensitive information, but they do so with a kind of designed intelligence. Rather than just collecting information, RATs present information to an attacker in a form that facilitates the ability to gain unauthorized access to the target machine. RATs are commonly employed by the more skilled threat actors. The principal use of a RAT is to enable attackers to have a way to get back into the system later, that is, to help establish a persistent covert presence in the system. As such, one of the key functions of a RAT is to provide a periodic beacon out to bypass security checks that prohibit unrequested packets from entering the system.



Principles of Computer Security, Fifth Edition

Rootkits

- **Modify the operating system to facilitate nonstandard functionality.**
 - Can do anything the operating system does
 - Use subversion and evasion to avoid security functions of the operating system
 - Five main types:
 - Firmware – Attacks firmware on a system
 - Virtual – Attacks at the virtual machine level
 - Kernel – Attacks the kernel of the OS
 - Library – Attacks libraries used on a system
 - Application level – Attacks specific applications

Copyright © 2010 by McGraw-Hill Education. All rights reserved.

A rootkit is a form of malware that is specifically designed to modify the operation of the operating system in some fashion to facilitate nonstandard functionality.

Rootkits modify the operating system kernel and supporting functions, changing the nature of the system's operation, and they can do virtually anything that the operating system does.

Rootkits are designed to avoid, either by subversion or evasion, the security functions of the operating system.

Rootkits can change thread priorities to boost an application's performance, perform keylogging, act as a sniffer, hide other files from other applications, or create backdoors in the authentication system.

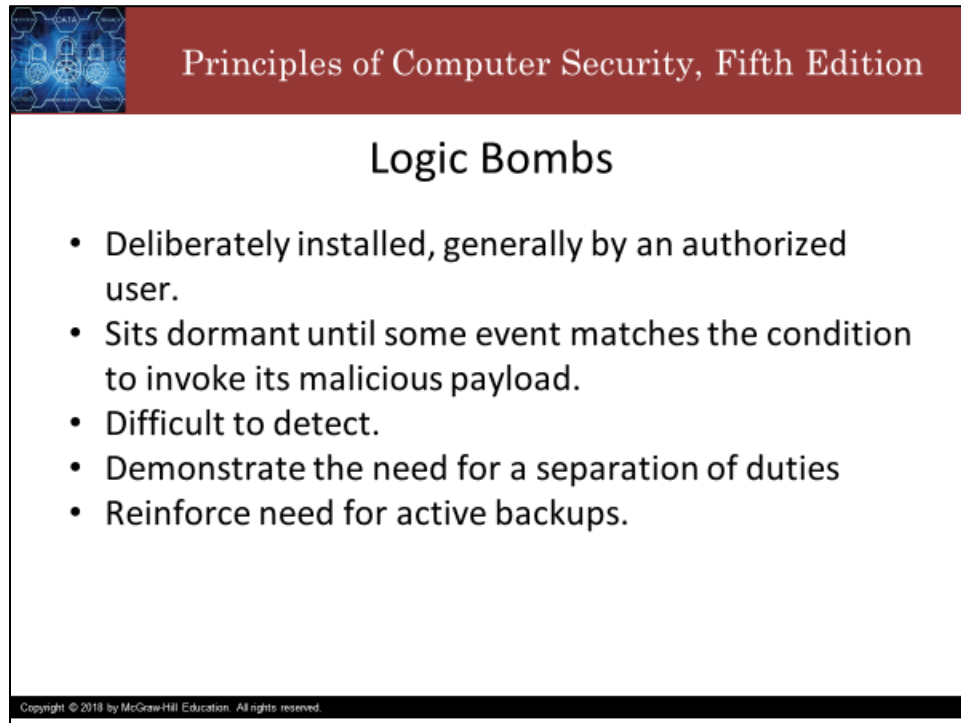
The use of rootkit functionality to hide other processes and files enables an attacker to use a portion of a computer without the user or other applications knowing what is happening. This hides exploit code from antivirus and antispyware programs, acting as a cloak of invisibility.

There are five main types of rootkits:

1. Rootkits can exist in firmware, and these have been demonstrated in both video cards and PCI expansion cards.
2. Rootkits can load before the operating system loads, acting as a virtualization layer.
3. As already mentioned, Rootkits can modify the operating system kernel and supporting functions.
4. Rootkits can exist as loadable library modules, effectively changing portions of the operating system outside the kernel.
5. And they can even exist at the application level to attack specific applications.

Once a rootkit is detected, it needs to be removed and cleaned up. Because of rootkits' invasive nature and the fact that many aspects of rootkits are not easily detectable, most system administrators don't even attempt to clean up or remove a rootkit. It is far easier to use a previously captured clean system image and reimage the system than to attempt to determine the depth and breadth of the damage and fix individual files.

Slide 9



Principles of Computer Security, Fifth Edition

Logic Bombs


- Deliberately installed, generally by an authorized user.
- Sits dormant until some event matches the condition to invoke its malicious payload.
- Difficult to detect.
- Demonstrate the need for a separation of duties
- Reinforce need for active backups.

Copyright © 2019 by McGraw-Hill Education. All rights reserved.

Logic bombs are a type of malicious software that is deliberately installed, generally by an authorized user.

They consist of some code or a program that sits dormant for a period of time, waiting until some event matches the logical condition to invoke its malicious payload.

Logic bombs are difficult to detect because they are often installed by authorized users and, in particular, by administrators who are also often responsible for security. This demonstrates the need for a separation of duties and a periodic review of all programs and services that are running on a system. It also illustrates the need to maintain an active backup program so that if your organization loses critical files to this sort of malicious code, it loses only transactions that occurred since the most recent backup and no permanent loss of data results.



Principles of Computer Security, Fifth Edition


Spyware

- Software that spies on users, recording and reporting on their activities.
 - Typically installed without user knowledge, spyware can do a wide range of activities.
 - It can record keystrokes (commonly called keylogging) when the user logs into specific web sites.
 - It can monitor how a user uses a specific piece of software (monitor attempts to cheat at games).
 - Many states have passed legislation banning the unapproved installation of software.

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

I'm sure most of us are familiar with spyware. Pretty much anyone who uses a computer (or smartphone) has some experience with spyware, but maybe many don't even realize it.

Many uses of spyware seem innocuous at first, but the unauthorized monitoring of a system can be abused very easily. In other cases, the spyware is specifically designed to steal information. Many states have passed legislation banning the unapproved installation of software, but many cases of spyware circumvent this issue through complex and confusing end-user license agreements. And many otherwise useful applications include spyware as part of their alleged basic functionality, often explained to users as something like "this application collects data to improve the user experience."



Principles of Computer Security, Fifth Edition

Adware

- Software that is supported by advertising.
 - Like Google. And Facebook.
- Comes in many different forms:
 - Legitimate adware
 - The user is aware of the advertising and agrees to the arrangement in return for free use of the software.
 - Adware in the form of malware
 - It is characterized by software that presents unwanted ads.


Copyright © 2010 by McGraw-Hill Education. All rights reserved.

Adware is software that is supported by advertisements. In practice, adware and spyware usually exist in a nefarious symbiotic relationship, where the spyware harvests data that the adware uses to target the advertisements to the user. A probably all-too-familiar example of this is using a search engine to look up something about dogs and then noticing that advertisements on Facebook, Google, Amazon, and most other websites seem to be all about dogs now.

With legitimate adware, the user is aware of the advertising and agrees to the arrangement in return for free use of the software. This type of adware often offers an alternative, ad-free version for a fee.

Adware can also refer to a form of malware, which is characterized by software that presents unwanted ads. These ads are sometimes an irritant, and at other times represent an actual security threat. Frequently these ads are in the form of pop-up browser windows, and in some cases, they cascade upon any user action.

The best defense, as always, is to practice safe surfing and be careful with what you download and install.



Principles of Computer Security, Fifth Edition

Botnets

- A collection of infected machines controlled by an attacker.
 - Hackers create armies of machines by installing malware agents on the machines, which then are called zombies.
 - Used to propagate malware, launch denial of service attacks, mine cryptocurrency, send spam emails, and more.
 - Have been observed in the wild to keep their host machines patched to protect against infection by competing botnets


Copyright © 2018 by McGraw-Hill Education. All rights reserved.

One form of malware that is seemingly benign to an average user is a botnet.

Hackers create armies of machines by installing malware agents on the machines, which then are called zombies. A collection of these machines is called a botnet.

These zombie machines are used to propagate malware, launch denial of service attacks, mine cryptocurrency, send spam emails, and more.

As botnets are a very lucrative business, botnets in the wild have been observed to keep their host machines patched to protect against infection by competing botnets.



Principles of Computer Security, Fifth Edition

Backdoors and Trapdoors

- Programs that attackers install after gaining unauthorized access to a system to ensure that they can continue to have unrestricted access to the system
- Backdoors are insecure by definition.


Copyright © 2010 by McGraw-Hill Education. All rights reserved.

Backdoors were originally (and sometimes still are) nothing more than methods used by software developers to ensure that they could get access to an application even if something were to happen in the future to prevent normal access methods.

An example would be a hard-coded password that could be used to gain access to the program in the event that administrators forgot their own system password. The obvious problem with this sort of backdoor (also sometimes referred to as a *trapdoor*) is that, since it is hard-coded, it cannot be removed. Should an attacker learn of the backdoor, all systems running that software would be vulnerable to attack.

The term backdoor is also, and more commonly, used to refer to programs that attackers install after gaining unauthorized access to a system to ensure that they can continue to have unrestricted access to the system, even if their initial access method is discovered and blocked.

Backdoors can also be installed by authorized individuals inadvertently, should they run software that contains a Trojan horse. A variation on the backdoor is the rootkit, which is established not to gain root access but rather to ensure continued root access.



Principles of Computer Security, Fifth Edition

Crypto-Malware

- Malware that encrypts files on a system and then leaves them unusable, acting as a denial of service.
- Example: Ransomware
 - Attacker offers to decrypt in exchange for paying ransom
- Completely automated
- Infection requires restoring from backups or rebuilding

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Crypto-malware is an early name given to malware that encrypts files on a system and then leaves them unusable, acting as a denial of service.

This type of malware is also behind ransomware.

Ransomware is a form of malware that performs some action and extracts a ransom from a user.


The most common form of ransomware is one that encrypts an important file or set of files, rendering a system unusable.

The attacker claims they will release the system or data after being paid, typically in a non-traceable means such as cryptocurrency.

My advice is to back up your data regularly, and then, if you get hit with ransomware, you just restore your data from the most recent backup and perform some self-reflection on your computer use habits. A cloud backup service is way cheaper than the cost of the ransom. And, also, don't forget: there's no guarantee the attacker doesn't leave the malware on your system to be used later for subsequent ransomware attacks.

Crypto-malware is typically completely automated.

Some infections can be recovered from by restoring the system from backups made before the infection (if that point is known, which it may not be). In the most extreme cases, the only repair mechanism is to completely rebuild the system. This can be time-consuming and/or impractical in some cases, making this attack mechanism equivalent to the physical destruction of assets.



Principles of Computer Security, Fifth Edition

Malware Defenses

- Use an antivirus program
- Keep your software up to date
 - Challenges
 - Keep track of the software that is on the system
 - Keep track of all vendor updates
 - Software exists to help with this


Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Malware in all forms—virus, worm, spyware, botnet, and so on—can be defended against in a couple of simple steps:

Step one: **Use an antivirus program.** Most major-vendor antivirus suites are designed to catch most widespread forms of malware. In some markets, the antivirus software is being referred to as anti-x software, indicating that it covers more than viruses. But because the threat environment changes literally daily, the signature files for the software need regular updates, which most antivirus programs offer to perform automatically.

Step two: **Keep your software up to date.** Many forms of malware achieve their objectives through exploitation of vulnerabilities in software, both in the operating system and applications. Although operating system vulnerabilities were the main source of problems, today, application-level vulnerabilities pose the greatest risk. Unfortunately, while operating system vendors are becoming more and more responsive to patching, most application vendors are not, and some (like Adobe) have very large footprints across many machines.

One of the challenges in keeping a system up to date is keeping track of the software that is on the system and keeping track of all vendor updates. There are software products that can scan your machine to enumerate all the software installed and verify the vendor status of each product. For standalone machines, such as the one in your home, this type of program is a great time-saving item. In even small enterprises, these tools are essential to manage the complexity of patches needed across the machines.



Principles of Computer Security, Fifth Edition

Application-Level Attacks

- Attacks are aimed at the applications primarily because this is where the objective of most attacks resides.
- Take advantage of several facts associated with computer applications.
 - Most applications are large and complicated programs
 - i.e. hard to secure, see also: <https://cve.mitre.org/>
 - End users are slow to apply patches.

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Attacks against a system can occur at any of several levels: the network, the operating system, the application, or the user (which would be social engineering).


Someone once asked Slick Willie Sutton, the bank robber, why he robbed banks. Perhaps they were hoping to uncover a dramatic tale of injustice and a lifelong search for revenge. Maybe a banker foreclosed on the old homestead, or maybe a banker's daughter spurned Sutton for another. Allegedly, Sutton replied simply: "I rob banks because that's where the money is."

Many attacks today include an element at the application level because that is where the money is or whatever other data or control the attacker is after. The most successful attacks are combinations of attacks at every level.

Application-level attacks take advantage of several facts associated with computer applications.

First, most applications are large programs written by groups of programmers and, by their nature, have errors in design and coding that create vulnerabilities. For a list of typical vulnerabilities, see the Common Vulnerability and Exposures list maintained by Mitre at cve.mitre.org.

Second, even when vulnerabilities are discovered and patched by software vendors, end users are slow to apply patches, as evidenced by the SQL Slammer incident in January 2003. The vulnerability exploited was a buffer overflow, and the vendor-supplied a patch six months prior to the outbreak, yet the worm still spread quickly due to the multitude of unpatched systems.



Principles of Computer Security, Fifth Edition

Attribution

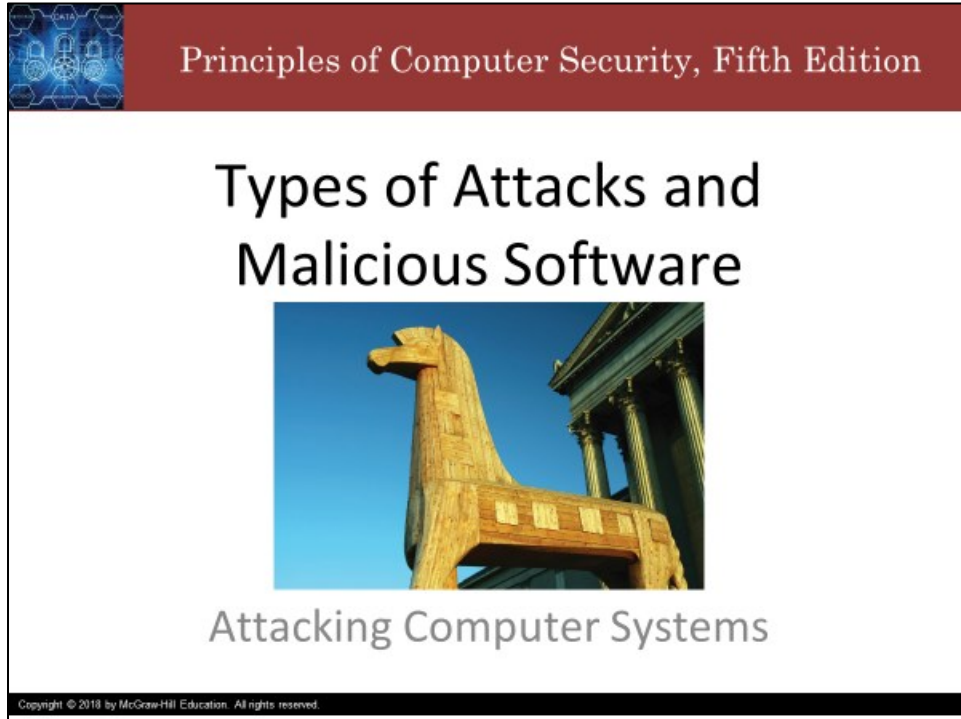
- The course slides are based on slides developed by the textbook authors, W. "Art" Conklin, Ph.D. (University of Houston), and Greg White, Ph.D. (University of Texas at San Antonio).
- These slides remain the intellectual property protected by US Copyright law of the authors and the textbook company.
- There have been some changes to the slide deck.

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Thank you and take care.


Types of Attacks and Malicious Software: Attacking Computer Systems

Slide 1



The image shows the cover of the book "Principles of Computer Security, Fifth Edition". The top left corner features a small graphic of a network with the word "DATA" above it. The title "Principles of Computer Security, Fifth Edition" is written in white serif font on a dark red background. Below this, the main title "Types of Attacks and Malicious Software" is displayed in a large, bold, black sans-serif font. Underneath the title is a photograph of a wooden Trojan horse in the foreground, with a classical building featuring columns in the background under a clear blue sky. At the bottom of the cover, the subtitle "Attacking Computer Systems" is written in a smaller, grey sans-serif font. A small copyright notice "Copyright © 2018 by McGraw-Hill Education. All rights reserved." is located at the very bottom of the cover.

Howdy! In this video, we give a small sampling of attacks against computer systems and networks. There are too many attacks to cover in a single course, much less a single video. So, please don't limit yourself to only these few attacks. I tried to pick a representative sample of some of the more common and illustrative examples, but there's still many important attacks that I am not covering here. Cybersecurity is one of those fields where one can never know everything or even enough. And that's part of what makes it fun! Let's go!



Principles of Computer Security, Fifth Edition


Attacking Computer Systems and Networks

- Attacks on computer systems and networks
 - Attacks on a specific software
 - Due to defects in design or implementation
 - Attacks on a specific protocol or service
 - Due to defects in design or implementation
 - Misuse / abuse of the protocol

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Attacks on computer systems and networks can be grouped into two broad categories:

1. Attacks on specific software (which are generally possible because of a defect in the design or implementation of the software), and
2. Attacks on a specific protocol or service (which attempt to take advantage of a specific feature of the protocol or service or to use the protocol or service in a manner for which it was not intended).



Principles of Computer Security, Fifth Edition

Denial-of-Service Attack

- Threat to Availability
 - Take system offline or make it unavailable or slow
- May be unintentional
 - Only effect matters
- Examples
 - Cable Cut
 - SYN Flood
 - Smurf
 - Hug of Death

Copyright © 2010 by McGraw-Hill Education. All rights reserved.

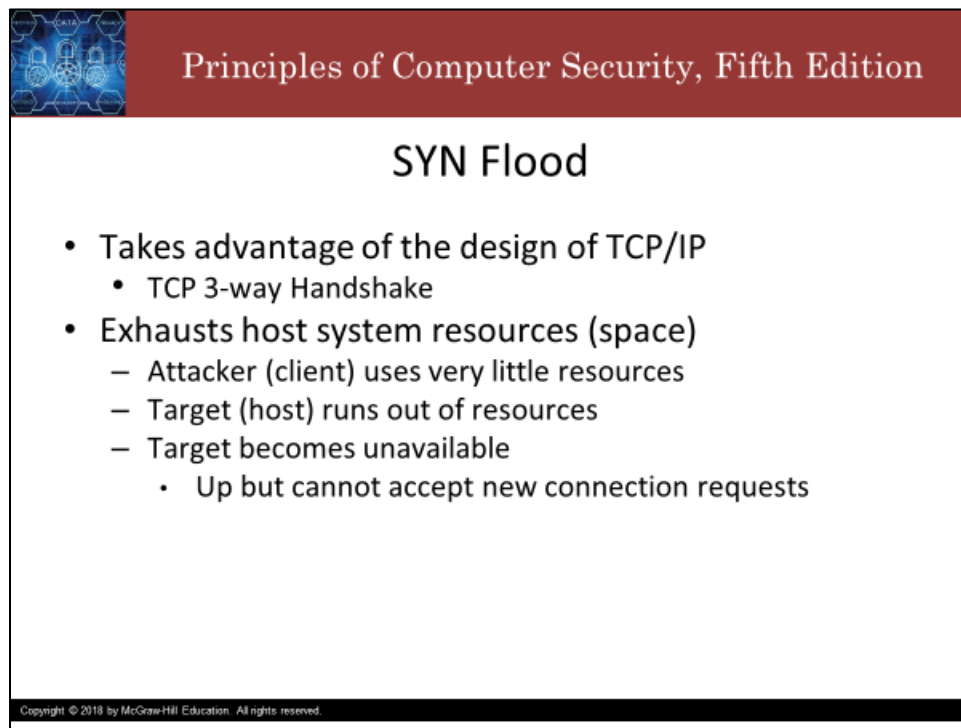
A **denial-of-service (DoS) attack** is an attack designed to prevent a system or service from functioning normally. A DoS attack can exploit a known vulnerability in a specific application or operating system, or it can attack features (or weaknesses) in specific protocols or services. In a DoS attack, the attacker attempts to deny authorized users access either to specific information or to the computer system or network itself. This can be accomplished by crashing the system—taking it offline—or by sending so many requests that the machine is overwhelmed.

The purpose of a DoS attack can be simply to prevent access to the target system, or the attack can be used in conjunction with other actions to gain unauthorized access to a computer or network. For example, a SYN flood attack can be used to prevent service to a system temporarily in order to take advantage of a trusted relationship that exists between that system and another.

Not all DoS attacks are intentional. Sometimes, mother nature literally rains on your parade. Or a utility crew accidentally cuts a fiber optic cable. Or a website gets a lot of attention in a very short amount of time, which is affectionately referred to as the hug of death. The intention of these actions is not malicious. But it doesn't matter—only the effect of the attack matters. Anything which may result in a denial of service needs to be considered as part of the security posture, and that includes weather, utility crews, and the hug of death.

I will go a bit deeper into the technical details of two intentional attacks: the SYN flood and the Smurf attack.

Slide 4



Principles of Computer Security, Fifth Edition

SYN Flood

- Takes advantage of the design of TCP/IP
 - TCP 3-way Handshake
- Exhausts host system resources (space)
 - Attacker (client) uses very little resources
 - Target (host) runs out of resources
 - Target becomes unavailable
 - Up but cannot accept new connection requests

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

SYN flooding is an example of a DoS attack that takes advantage of the way TCP/IP networks were designed to function and does a good job of illustrating the basic principles of a DoS attack.

The attacker makes the target system unavailable by exhausting the target's space resources. A SYN Flood does not necessarily crash the target, although it is possible that a bad implementation of the TCP connection handler could crash the system. A DoS attack is characterized by the observed effect on the users of the system. If the system becomes unresponsive or infuriatingly slow, the attack is succeeding because the system is not available to the authorized users. Crashing the system or physically destroying assets is but one, albeit highly effective, way to achieve this goal.

In a SYN Flood attack, the attacker uses the naïve TCP 3-way handshake protocol to make the system unavailable for new connections by reserving (but never actually using) all of the available connections on the host. To understand how this works, we need to learn about the TCP 3-way handshake protocol.

Slide 5

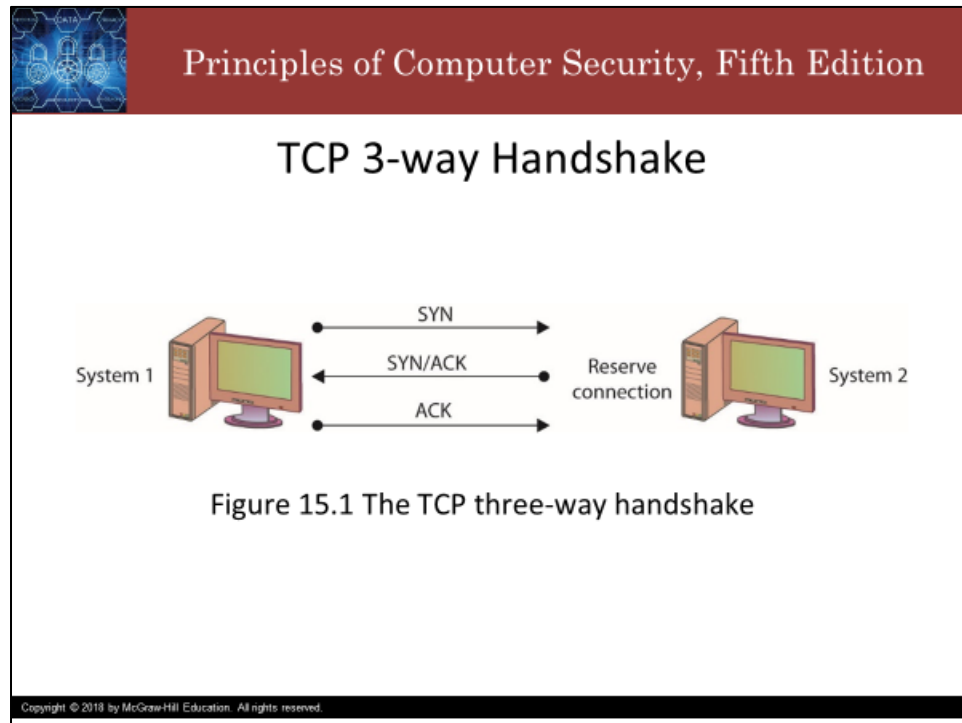


Figure 15.1 The TCP three-way handshake

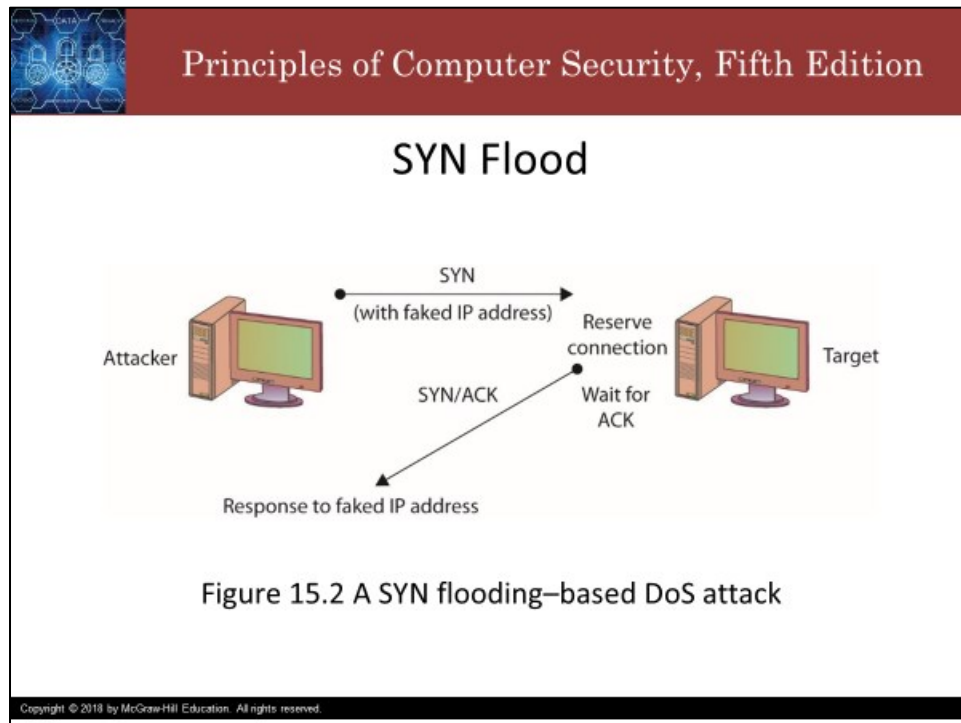
The TCP 3-way handshake is a 3-step process for establishing a new TCP connection for data transmission between two systems, a client who makes the initial connection request and a host (or server) which receives it. A TCP connection is a 2-way connection, the client has a connection to send data to the server, and the server has a connection to send data to the client. So, technically, there are 2 connections, each being a 1-way street.

The first step is for System 1 (the client) to send a SYN packet to System 2 (the server), indicating a desire to communicate with the system. SYN, like synthesis, means to create a connection. The SYN packet is the initial connection request.

In the second step, System 2 (the server) responds to System 1 (the client) by sending an ACK packet, an acknowledgment of the client's SYN packet, and a SYN packet of its own, requesting a connection to the client. These two packets, the ACK and the SYN are combined into one SYN/ACK packet.

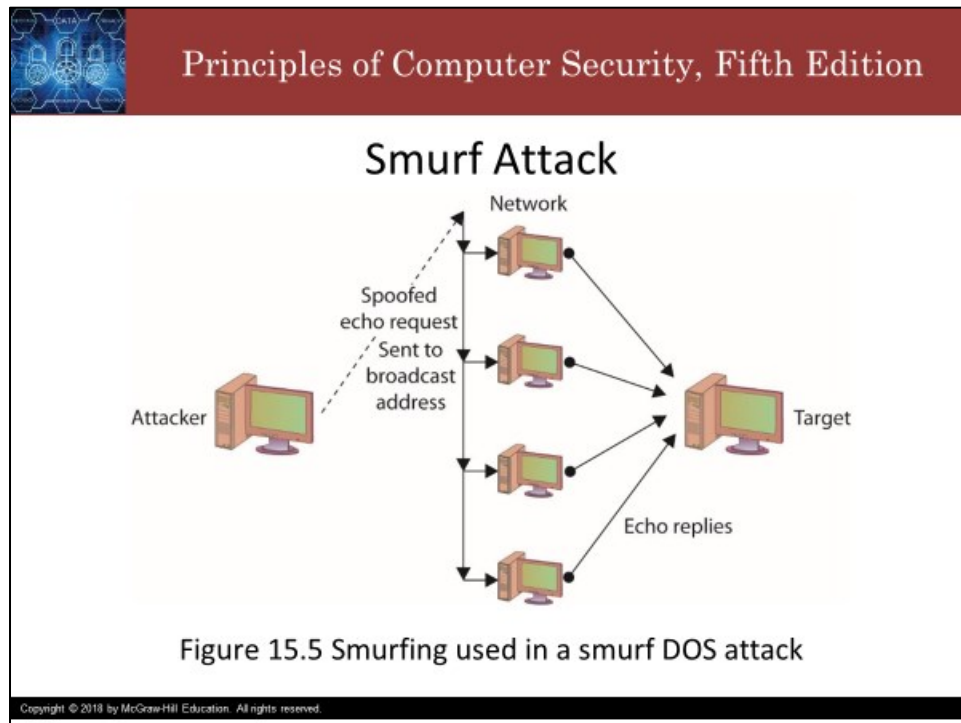
Once System 1 receives the SYN/ACK packet, it responds with an ACK packet, and the TCP connection is then established between the systems.

So, the 3-way handshake goes SYN, SYN/ACK, ACK. It is at step 2 when the server replies with SYN/ACK that the connection from the client to the server is reserved on the server. There is a table which holds all of the active (or pending) connections. When this table gets full, no additional connections can be made (in other words, the servers hands are full). This is the step that the SYN flood attacker abuses.

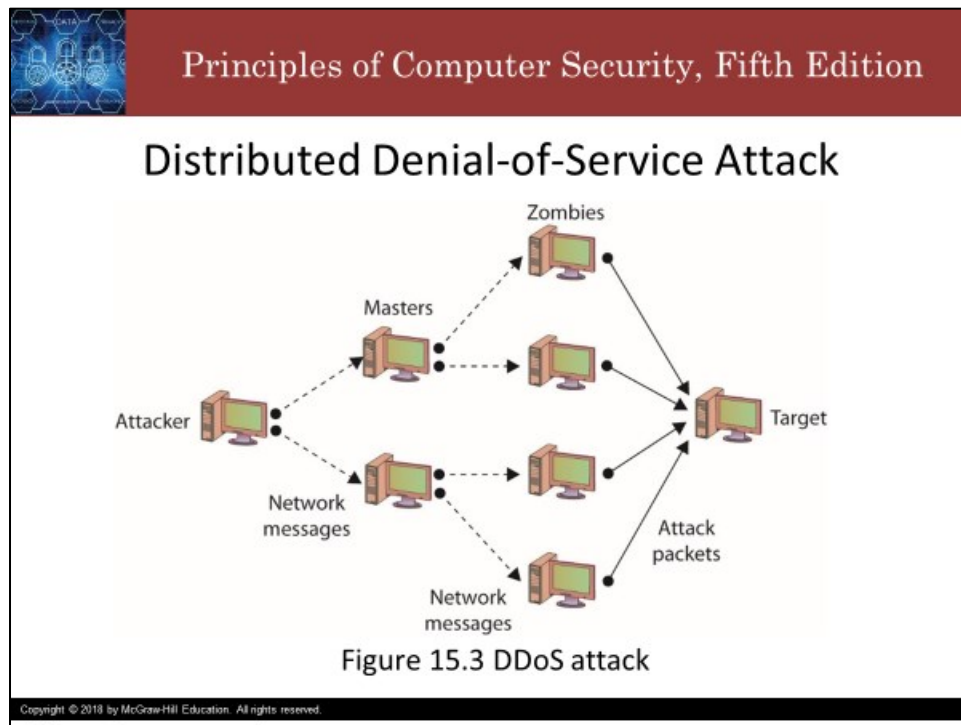


Under normal circumstances, the first system sends a SYN packet to the system with which it wants to communicate. The second system responds with a SYN/ACK if it is able to accept the request. When the initial system receives the SYN/ACK from the second system, it responds with an ACK packet, and communication can then proceed. However, in a SYN flood, the attacker never replies with the final ACK. The server is left waiting on the client, but the client will never respond (and the poor server can't tell the difference between a NO and a WAIT).


In a SYN flooding attack, the attacker sends spoofed connection requests (SYN) to the targeted system. Each of these requests will be answered (SYN/ACK) by the target system, which then waits for the third part (ACK) of the handshake. Since the requests are fake -- a nonexistent IP address is used in the requests, so the target system is responding to a system that doesn't exist or isn't listening -- the target will wait for responses that never come. The target system will drop these connections after a specific time-out period, but if the attacker sends requests faster than the time-out period eliminates them -- the attacker floods the target with SYN packets -- the system will quickly be filled with requests. The number of connections a system can support is finite, so when more requests come in than can be processed, the system will soon be reserving all its connections for fake requests. At this point, any further requests are simply dropped, and legitimate users who want to connect to the target system will not be able to do so because use of the system has been denied to them.



In a **smurf attack**, the attacker sends a spoofed packet to the broadcast address for a network, which distributes the packet to all systems on that network. The packet the attacker sends is an echo request with the source address forged so that it appears that another system (the target system) has made the echo request. The normal response of a system to an echo request is an echo reply, and it is used in the ping utility to let a user know whether a remote system is reachable and is responding. In the smurf attack, the request is broadcast by the router or switch to all systems on the network, so all will respond with an echo reply to the target system. The attacker sends a single packet that generates as many as 254 responses aimed at the target. Should the attacker send several of these spoofed requests or send them to several different networks, the target can quickly become overwhelmed.



A **Distributed Denial of Service (DDoS)** attack is a DoS attack in which the attack comes from many systems rather than just one. The smurf attack is like a mini DDoS attack in that the one attacker tricks hundreds of hosts into all spamming a single target. A DDoS attack typically involves thousands up to millions of attacking machines. A very large army of attack agents can be commandeered and controlled by an attacker. The combined systems can then overwhelm the target with traffic under the direction of the attacker. One of the common functions of botnets is to launch DDoS attacks. If the attack network is large enough, even simple web traffic can quickly overwhelm even the largest websites. The hug of death is an example of a usually unintentional DDoS attack.



Principles of Computer Security, Fifth Edition

Defense against the DoS Arts

- IP banning doesn't work when IP addresses are spoofed
- Upgrade and patch hardware and applications
- Against SYN Flooding: use SYN cookies
 - Tricky protocol that delays creating the connection until the final ACK, thus preventing resource exhaustion from a SYN flood
- Distribute load across many servers
 - Harder to exhaust many systems at once
 - If one gets targeted, shift load to the others

Copyright © 2010 by McGraw-Hill Education. All rights reserved.

Defending against DoS attacks is hard.

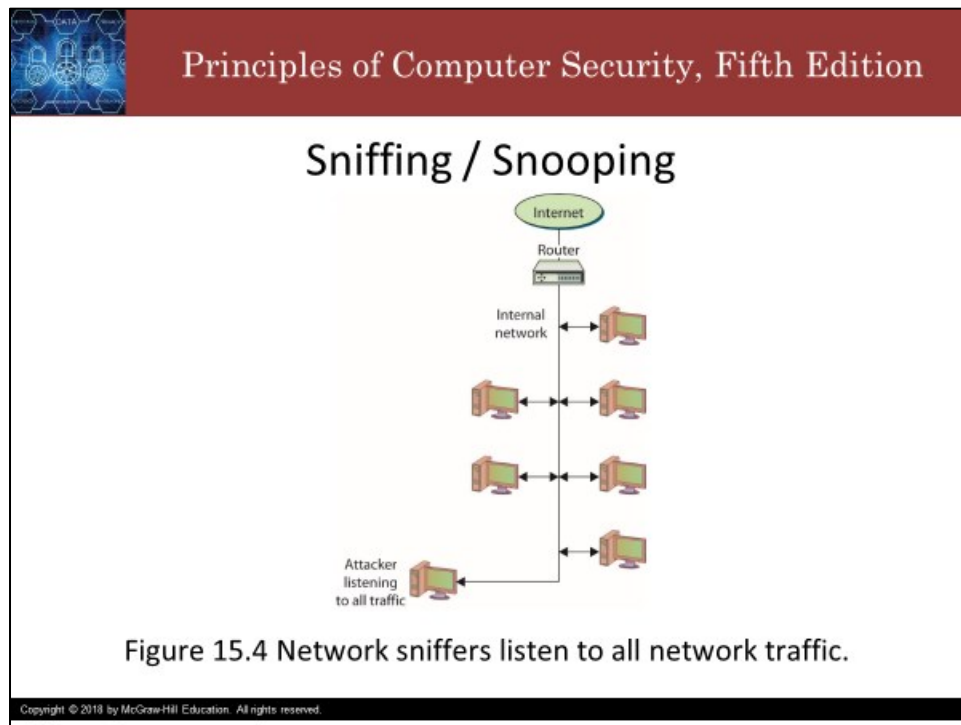
To prevent a DDoS attack, you must either be able to intercept or block the attack messages or keep the DDoS network from being established in the first place.

Blocking IP traffic from specific addresses won't work when the source addresses are spoofed. So, you must rely on the infrastructure-level defenses and the security of your Internet neighbors. That means they depend on you, too.

So, one thing you should do is make sure you have applied the latest patches and upgrades to your systems and the applications running on them. This will make it harder for your systems to be used for an attack.

There is actually a countermeasure to prevent SYN flooding, called SYN cookies. I think most systems nowadays enable SYN cookies by default. The idea is sneaky: the server doesn't actually reserve a connection when it receives a SYN. Instead, it sends a crafted SYN/ACK packet such that the ACK reply contains all the information required to set up the connection. Since SYNs have virtually no effect on resource consumption, the attacker's SYN flood power is effectively nerfed.

Another good tactic is to distribute your workload across several systems. On the one hand, every additional server increases the capacity of the system to handle connections, making it harder to exhaust those resources. But also, if one server gets hit directly with a DoS attack, the legitimate traffic can be re-routed to other servers.



The group of protocols that makes up the TCP/IP suite was designed to work in a friendly environment in which everybody who connected to the network used the protocols as they were designed.

Normally, the network device that connects a computer to a network is designed to ignore all traffic that is not destined for that computer.

Network sniffers ignore this friendly agreement and observe all traffic on the network, whether destined for that computer or others.

Sniffing is when someone examines all the network traffic that passes their network interface, whether addressed for them or not.


A network sniffer is a software or hardware device that is used to observe traffic as it passes through a network on shared broadcast media.

Sniffing can be used to view all traffic, or it can target a specific protocol, service, or even string of characters (looking for logins, for example).

Some network sniffers are designed not just to observe all traffic but to modify traffic as well.

Network sniffers have legitimate uses, such as by network administrators to monitor network performance.

But they can also be used by attackers to gather information to be used in an attack.



Principles of Computer Security, Fifth Edition

spoofing

- Making data look like it has come from a different source.
 - Possible in TCP/IP because of the friendly assumptions behind the protocols.
 - Examples
 - IP spoofing
 - Email spoofing
 - MAC spoofing
- Difficulty depends on several factors
 - Encryption, attacker's location relative to target

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Spoofing is nothing more than making data look like it has come from a different source. This is possible in TCP/IP because of the friendly assumptions behind the protocols. When the protocols were developed, it was assumed that individuals who had access to the network layer would be privileged users who could be trusted.

When a packet is sent from one system to another, it includes not only the destination IP address and port but the source IP address as well. You are supposed to fill in the source with your own address, but nothing stops you from filling in another system's address. This is called IP spoofing and is just one of the several forms of spoofing.

In email spoofing, a message is sent with a From address that differs from that of the sending system. This can be easily accomplished in several different ways using several programs and is a common feature of listservs where the reply-to address can be set so that replies do not go to the whole list.

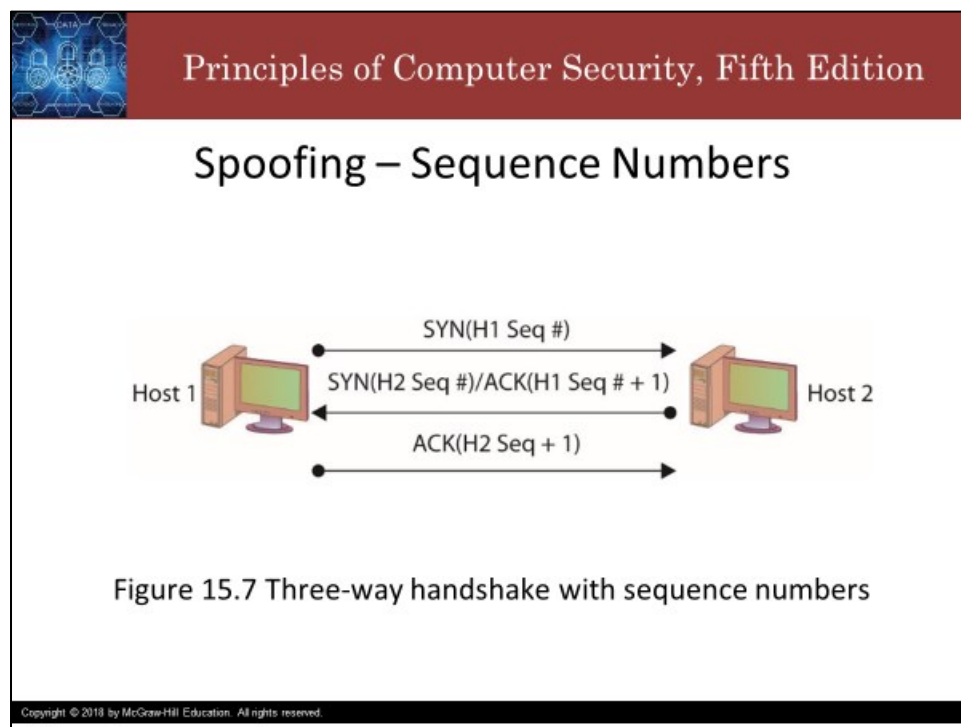
Recipients can use several methods to determine whether an email message was sent by the source it claims to have been sent from, but most users do not question their email and will accept it as authentic where it appears to have originated.

The IP address is a software-defined address and can change over time. The MAC address is supposed to be a hardware-defined address that is fixed at manufacture and never changes, supposedly uniquely identifying a particular device. MAC spoofing is the act of changing a MAC address to bypass security checks based on the MAC address. This is possible because a query for a device's MAC address is handled by software, which has the ability to effectively lie.

Spoofing can also take advantage of a trusted relationship between two systems. If two systems are configured to accept the authentication accomplished by each other, an individual logged onto one system might not be forced to go through an authentication process again to access the other system. An attacker can take advantage of this arrangement by sending a packet to one system that appears to have come from a trusted system. Since the trusted relationship is in place, the targeted system may perform the requested task without authentication. This is a violation of the principle of complete mediation, and therefore authentication based on IP address (or any other spoofable data) is considered insecure.

How difficult it is to convincingly pull off a spoof depends heavily on several factors, including whether the traffic is encrypted and where the attacker is located relative to the target. Spoofing attacks from inside a network, for example, are much easier to perform than attacks from outside of the network because the inside attacker can observe the traffic to and from the target and can do a better job of formulating the necessary packets.

Slide 12



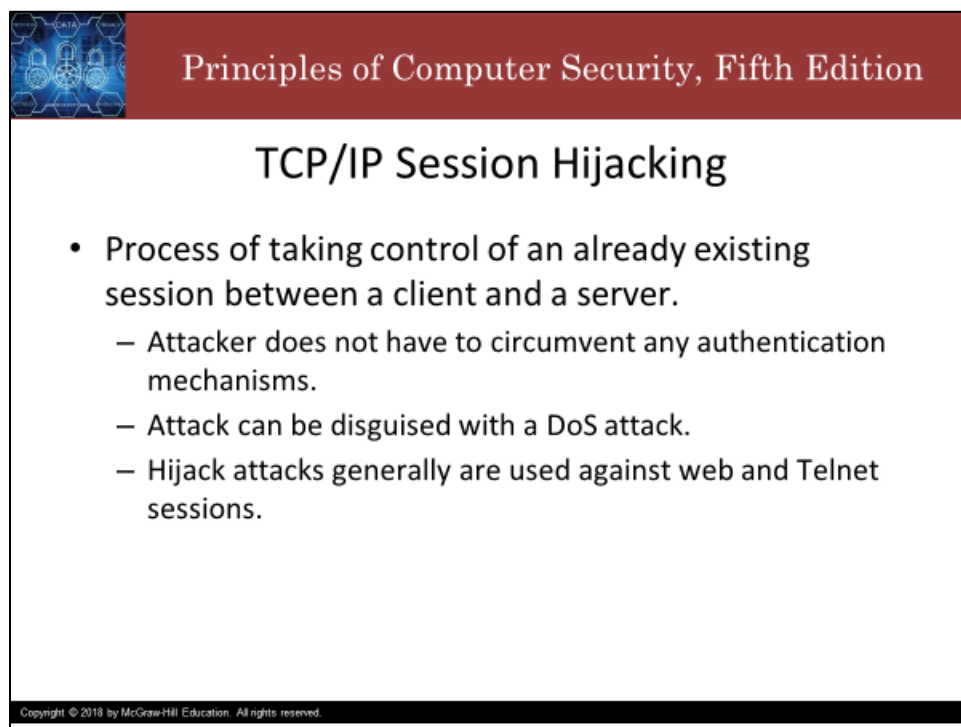
Formulating the packets is more complicated for external attackers because a sequence number is associated with TCP packets. A **sequence number** is a 32-bit number established by the host that is incremented for each packet sent. Packets are not guaranteed to be received in order, and the sequence number is used to reorder packets as they are received and to refer to packets that may have been lost in transmission.

The difference in the difficulty of attempting a spoofing attack from inside a network and from outside involves determining the sequence number.

If the attacker is inside of the network and can observe the traffic with which the target host responds, the attacker can easily see the sequence number the system creates and can respond with the correct sequence number.

If the attacker is external to the network and the sequence number the target system generates is not observed, it is next to impossible for the attacker to provide the final ACK with the correct sequence number. So the attacker has to guess what the sequence number might be.

Slide 13



Principles of Computer Security, Fifth Edition

TCP/IP Session Hijacking

- Process of taking control of an already existing session between a client and a server.
 - Attacker does not have to circumvent any authentication mechanisms.
 - Attack can be disguised with a DoS attack.
 - Hijack attacks generally are used against web and Telnet sessions.

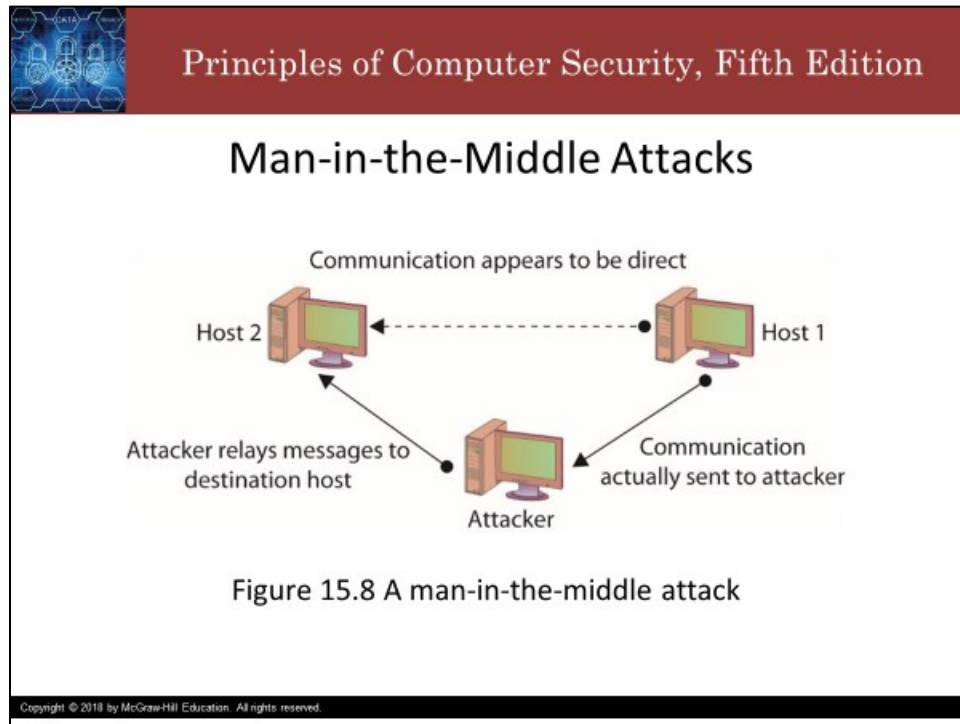
Copyright © 2010 by McGraw-Hill Education. All rights reserved.

TCP/IP hijacking and **session hijacking** are terms used to refer to the process of taking control of an already existing session between a client and a server.

The advantage to an attacker of hijacking over attempting to penetrate a computer system or network is that the attacker doesn't have to circumvent any authentication mechanisms since the targeted user has already authenticated and established the session.

To prevent the user from noticing anything unusual, the attacker can decide to attack the user's system and perform a DoS attack on it, taking it down so that the user, and the system, will not notice the extra traffic that is taking place.

Hijack attacks generally are used against web and Telnet sessions. Sequence numbers as they apply to spoofing also apply to session hijacking since the hijacker will need to provide the correct sequence number to continue the appropriated sessions.



A **man-in-the-middle attack** generally occurs when attackers are able to place themselves in the middle of two other hosts that are communicating.

A man-in-the-middle attack can be accomplished by compromising a router to alter the path of the traffic.

The attacker can then observe all traffic before relaying it and can actually modify or block traffic.

To the target host, it appears that communication is occurring normally since all expected replies are received.

There are numerous methods of instantiating a man-in-the-middle attack; one of the common methods is via session hijacking. Session hijacking can occur when information such as a cookie is stolen, allowing the attacker to impersonate the legitimate session. This can happen as a result of a cross-site scripting attack, which tricks a user into executing code resulting in cookie theft. The amount of information that can be obtained in a man-in-the-middle attack will obviously be limited if the communication is encrypted. Even in this case, however, sensitive information can still be obtained since knowing what communication is being conducted and between which individuals may, in fact, provide information that is valuable in certain circumstances.

A man-in-the-middle attack is ideally accomplished by ensuring that all communication going to or from the target host is routed through the attacker's host (which can be accomplished if the attacker can compromise the router for the target host). The attacker can then observe all traffic before relaying it

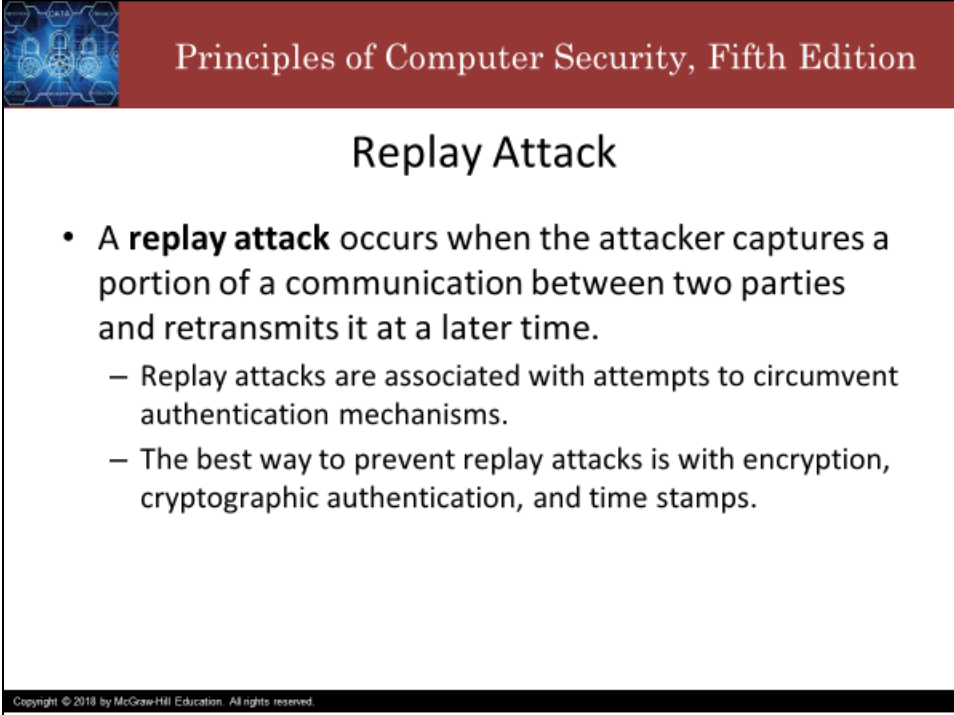
and can actually modify or block traffic. To the target host, it appears that communication is occurring normally since all expected replies are received.

The term "man-in-the-middle attack" also refers to a specific type of attack to undermine cryptographic security.

A man-in-the-middle attacker intercepts a request for the target's public key and replies with the attacker's public key. This lets the attacker decrypt, read, and modify the data before re-encrypting it to forward to the destination (who then replies to the attacker, who can read and modify data before sending it on to the victim, who is none the wiser).

Well-designed cryptographic products use techniques such as mutual authentication to avoid this problem.

Slide 15



Principles of Computer Security, Fifth Edition

Replay Attack


- A **replay attack** occurs when the attacker captures a portion of a communication between two parties and retransmits it at a later time.
 - Replay attacks are associated with attempts to circumvent authentication mechanisms.
 - The best way to prevent replay attacks is with encryption, cryptographic authentication, and time stamps.

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

A **replay attack** occurs when the attacker captures a portion of a communication between two parties and retransmits it at a later time.

Replay attacks are generally associated with attempts to circumvent authentication mechanisms, such as the capturing and reuse of a certificate or ticket.

The best way to prevent replay attacks is with encryption, cryptographic authentication, and time stamps. If a portion of the certificate or ticket includes a timestamp or an expiration date, and this portion is also encrypted as part of the ticket or certificate, replaying it at a later time will prove useless since it will be rejected as having expired.



Principles of Computer Security, Fifth Edition

Transitive Access


- *Transitive access* is a means of attacking a system by violating the trust relationship between machines.
- A simple example is when servers are well protected and clients are not, and the servers trust the clients.
 - In this case, attacking a client can provide transitive access to the servers.

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Transitive access is a means of attacking a system by violating the trust relationship between machines.

A simple example is when servers are well protected, and clients are not, and the servers trust the clients.

In this case, attacking a client can provide transitive access to the servers.



Principles of Computer Security, Fifth Edition

Spam and Spim

- Not generally considered a social engineering or security issue
 - however, can still be a security concern.
- Spam is bulk unsolicited e-mail.
 - Can be legitimate
 - Can also be malicious
- Spim is IM/chat Spam

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Though not generally considered a social engineering issue, nor a security issue for that matter, spam can, however, be a security concern.

Spam, as just about everybody knows, is bulk unsolicited email.

It can be legitimate in the sense that it has been sent by a company advertising a product or service, but it can also be malicious and could include an attachment that contains malicious software designed to harm your system or a link to a malicious website that may attempt to obtain personal information from you.

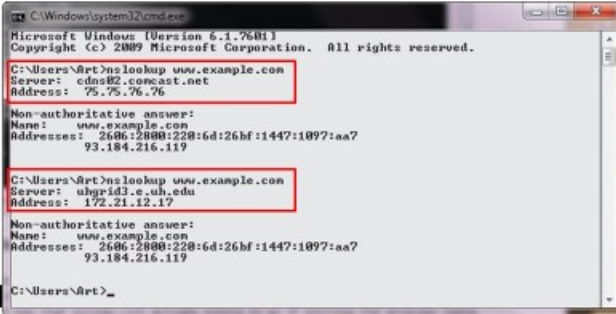
Though not as well known, a variation on spam is *spim*, which is basically spam delivered via an instant messaging or chat application.

The purpose of hostile spim is the same as that of spam—the delivery of malicious content or links.

Principles of Computer Security, Fifth Edition

Cache Poisoning

- DNS poisoning
 - Can occur at any level
- ARP poisoning
 - Corrupt the ARP table to misroute packets



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Art>nslookup www.example.com
Server: cdns82.comcast.net
Address: 75.75.76.76

Non-authoritative answer:
Name: www.example.com
Addresses: 2606:2800:228:6d:26bf:1447:1097:aa7
          93.184.216.119

C:\Users\Art>nslookup www.example.com
Server: ubgrid3.e.oh.edu
Address: 172.21.12.17

Non-authoritative answer:
Name: www.example.com
Addresses: 2606:2800:228:6d:26bf:1447:1097:aa7
          93.184.216.119

C:\Users\Art>
```

Copyright © 2010 by McGraw-Hill Education. All rights reserved.

Many network activities rely on various addressing schemes to function properly. When you type a URL into the address bar of your browser, your browser consults a DNS – domain name system – to find out the numeric IP address associated with the URL. When a packet is being switched through the network, a series of address caches are used to improve efficiency by preventing repeated redundant lookups. But, caches can be poisoned, sending incorrect information to the end user's application, redirecting traffic, and changing system behaviors. Two common examples of this attack are DNS poisoning and ARP poisoning.

A DNS poisoning attack can occur when network connections are changed, resulting in different DNS lookups.

The DNS is hierarchy of DNS caches, from the backbone Internet all the way down through your ISP, router, and your machine itself. DNS poisoning can occur at any level.

DNS poisoning is a variant of a larger attack class referred to as DNS spoofing, in which an attacker changes a DNS record through any of a multitude of means.

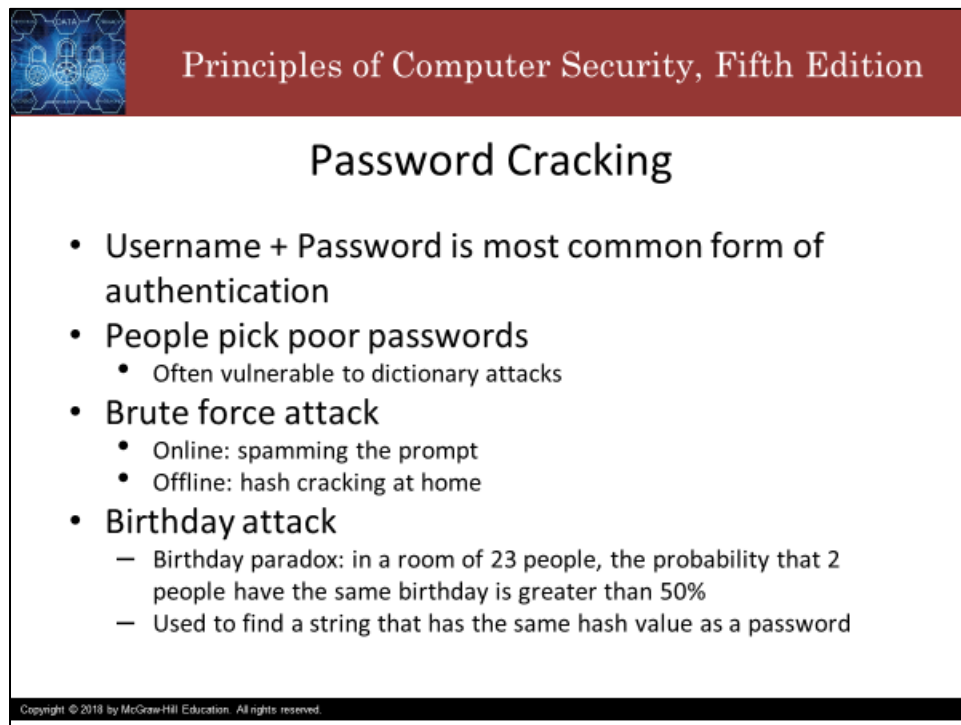
The figure shows a series of DNS queries executed on a Windows machine. In the first request, the DNS server was with an ISP, while on the second request, the DNS server was from a VPN connection. Between the two requests, the network connections were changed, resulting in different DNS lookups. This is a form of DNS poisoning attack.

Moving packets between machines requires knowing the address of the machine to which the packet should be sent. The IP protocol uses IP addresses, but the hardware uses MAC address. The Address Resolution Protocol (ARP) is like DNS for MAC addresses, allowing hosts to find out the MAC address of the machine which has a particular IP address and vice versa.

ARP poisoning involves an attacker sending messages, corrupting the ARP table, and causing packets to be misrouted.

This can allow a mechanism whereby an attacker can inject themselves into the middle of a conversation between two machines, which is to say, a man-in-the-middle attack.

Slide 19



Principles of Computer Security, Fifth Edition

Password Cracking

- Username + Password is most common form of authentication
- People pick poor passwords
 - Often vulnerable to dictionary attacks
- Brute force attack
 - Online: spamming the prompt
 - Offline: hash cracking at home
- Birthday attack
 - Birthday paradox: in a room of 23 people, the probability that 2 people have the same birthday is greater than 50%
 - Used to find a string that has the same hash value as a password

Copyright © 2010 by McGraw-Hill Education. All rights reserved.

The most common form of authentication is the user ID and password combination.

While it is not inherently a poor mechanism for authentication, the combination can be attacked in several ways.

All too often, these attacks yield favorable results for the attacker, not as a result of a weakness in the scheme but usually due to the user not following good password procedures.

People are notorious for picking poor passwords.

Users need to select a password that they can remember, so they create simple passwords.

The attacker just needs to obtain a valid user ID and some information about the user before guessing can begin.

A password-cracking program can use a list of dictionary words to try to guess the password.

Rules can also be defined so that the cracking program will substitute special characters for other characters or combine words.

In a brute-force attack, a password-cracking program attempts all possible character combinations.

A brute-force attack on a password can take place at two levels:

The attacker can use a password-cracking program to attempt to guess the password directly at a login prompt. The attack can be made more difficult if the account locks after a few failed login attempts.


The attacker can first steal a password file, use a password-cracking program to compile a list of possible passwords based on the list of password hashes contained in the password file (offline), and then use that narrower list to attempt to guess the password at the login prompt. The second attack can be thwarted if the password file is securely maintained so that others cannot obtain a copy of it.

The **birthday attack** is a special type of brute-force attack that uses the birthday paradox.

The *birthday paradox* states that in a group of at least 23 people, the chance that two individuals will have the same birthday is greater than 50 percent.

Applied to passwords, the goal is not actually to find 2 users with the same password but to find some string that has the same hash value as some user's password. By the nature of hash functions, this does not mean that the attacker definitely learns the actual password, but rather the attack finds a string which the authentication system will mistake for the user's valid password because its hash value matches that of the user's actual password (which, for security reasons, the system doesn't know).

Slide 20



Principles of Computer Security, Fifth Edition

Software Exploitation

- Takes advantage of errors in software
 - design or implementation
 - If it ain't broke, it doesn't have enough features
- **Preventable Risk** can be reduced
 - SDLC, code review, tools: threat modeling, bug tracking, fuzzing, code analysis
- **Zero-day** – attacker is first to know
- **Most common software errors**
 - Buffer overflow – don't write out of bounds
 - Integer overflow – if it doesn't fit, don't force it
 - <https://www.sans.org/top25-software-errors/>

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Software exploitation is an attack that takes advantage of errors in software.

The exploitable errors can be the result of poor design, poor testing, or poor coding practices.

Software sometimes behaves in quirky ways, sometimes on purpose and sometimes not. It's not a bug; it's a feature. Some people like their software to be very feature-rich. The problem with this is that features increase the attack surface.

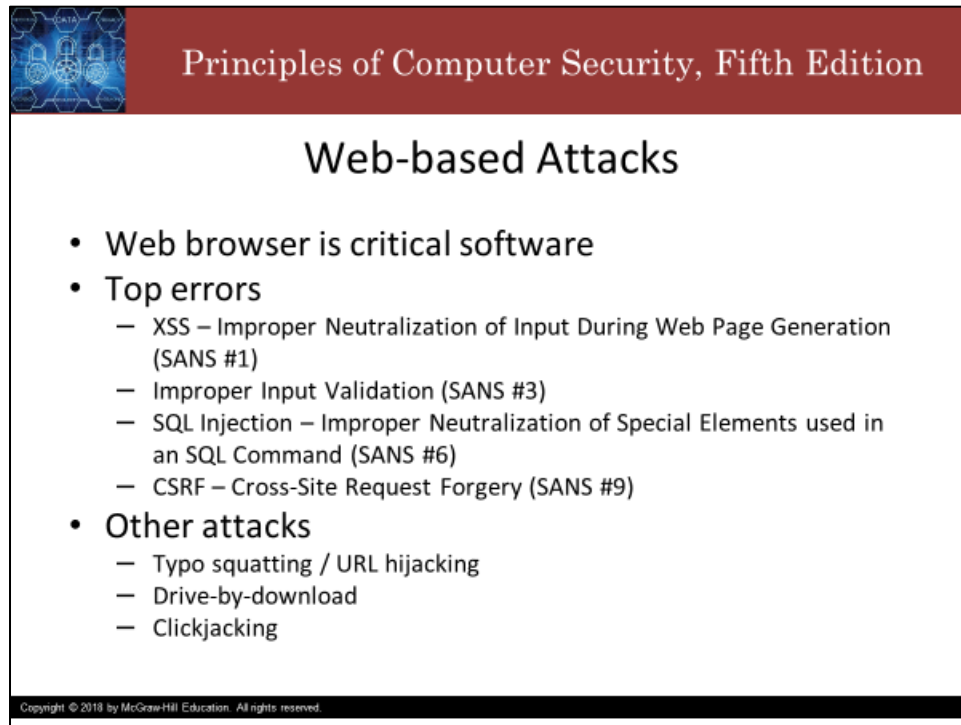
Software exploitation is not entirely preventable., but it is possible to reduce the risk. Remember: security is risk management. Through the use of a secure software development lifecycle, code review, and tools like bug tracking, fuzzing, and code analysis, the risk of a vulnerability in software can be reduced to acceptable levels.

A zero-day attack is one in which the attacker is the only one who knows of the existence of the vulnerability (or the vendor knows but hasn't told anyone and hasn't patched it, which is worse). There is no way to defend against zero-day attacks except through the consistent and thorough application of secure software development practices, which reduce the risk of such an attack by making it harder to find a zero-day, harder to exploit, and less damaging if exploited.

One of the most common errors is the buffer overflow which occurs when a program is provided more data for input than it was designed to handle and it handles the overflow badly, usually crashing the system but sometimes allowing the attacker to gain control of the system.

Another common error is the integer overflow, which is like a buffer overflow, but for a single number. Integer overflow happens when the program tries to store a number which is too big for the data type, and the value is corrupted (the high-order bits are chopped off). When using signed integers, really big numbers become negative, which can be a problem. Integer overflows are sometimes involved in buffer overflows to punish a developer for improperly guarding against buffer overflow.

CWE and SANS keep a list of the top 25 software errors. Buffer overflow (aka out-of-bounds write) is number 2.



Principles of Computer Security, Fifth Edition

Web-based Attacks

- Web browser is critical software
- Top errors
 - XSS – Improper Neutralization of Input During Web Page Generation (SANS #1)
 - Improper Input Validation (SANS #3)
 - SQL Injection – Improper Neutralization of Special Elements used in an SQL Command (SANS #6)
 - CSRF – Cross-Site Request Forgery (SANS #9)
- Other attacks
 - Typo squatting / URL hijacking
 - Drive-by-download
 - Clickjacking

Copyright © 2010 by McGraw-Hill Education. All rights reserved.

The web browser has become the major application for users to engage resources across the Web.

The popularity and the utility of this interface has made it a prime target for attackers to gain access and control over a system.

A wide variety of attacks can occur in and through the browser, typically resulting from a failure to properly validate input before use.

Unvalidated and unsanitized input can result in injection attacks, header manipulation, and more.

One of the most dangerous attacks is the cross-site scripting attack (or XSS).

Cross-site scripting (XSS) vulnerabilities occur when:

- Untrusted data enters a web application, typically from a web request.
- The web application dynamically generates a web page that contains this untrusted data.
- During page generation, the application does not prevent the data from containing content that is executable by a web browser, such as JavaScript, HTML tags, HTML attributes, mouse events, Flash, ActiveX, etc.
- A victim visits the generated web page through a web browser, which contains malicious script that was injected using the untrusted data. Since the script comes from a web page that was sent by the web server, the victim's web browser executes the malicious script in the context of the web server's domain. This effectively violates the intention of the web browser's same-origin policy, which states that scripts in one domain should not be able to access resources or run code in a different domain.

(source: <https://cwe.mitre.org/data/definitions/79.html>)

Input validation is a frequently-used technique for checking potentially dangerous inputs in order to ensure that the inputs are safe for processing within the code or when communicating with other components. When software does not validate input properly, an attacker is able to craft the input in a form that is not expected by the rest of the application. This will lead to parts of the system receiving unintended input, which may result in altered control flow, arbitrary control of a resource, or arbitrary code execution.

(source: <https://cwe.mitre.org/data/definitions/20.html>)

Without sufficient removal or quoting of SQL syntax in user-controllable inputs, the generated SQL query can cause those inputs to be interpreted as SQL instead of ordinary user data. This can be used to alter query logic to bypass security checks or to insert additional statements that modify the back-end database, possibly including execution of system commands.

SQL injection has become a common issue with database-driven websites. The flaw is easily detected and easily exploited, and as such, any site or software package with even a minimal user base is likely to be subject to an attempted attack of this kind. This flaw depends on the fact that SQL makes no real distinction between the control and data planes.

(source: <https://cwe.mitre.org/data/definitions/89.html>)


Cross-site request forgery is possible when a web server is designed to receive a request from a client without any mechanism for verifying that it was intentionally sent. An attacker may be able to trick a client into making an unintentional request to the web server, which will be treated as an authentic request. This can be done via a URL, image load, or any of several other mechanisms and can result in exposure of data or unintended code execution.

(source: <https://cwe.mitre.org/data/definitions/352.html>)

Typosquatting is an attack form that involves capitalizing upon common typo errors. This attack pattern is also referred to as *URL hijacking*, *fake URL*, or *brandjacking* if the objective is to deceive based on branding. It is a common tactic used in phishing and pharming attacks to make the phishy URL look more authentic.

A **drive-by download attack** automatically downloads malware within browser, whether a user clicks or not, simply because the user visits a site. Often, the user is tricked or forced into visiting the site by malicious code on another site.

Clickjacking uses malicious code in web pages to force the user to click on things that they did not intend to click on. This can be used to boost ad revenue or to direct victims to drive-by-download attacks.



Principles of Computer Security, Fifth Edition

Attribution

- The course slides are based on slides developed by the textbook authors, W. "Art" Conklin, Ph.D. (University of Houston), and Greg White, Ph.D. (University of Texas at San Antonio).
- These slides remain the intellectual property protected by US Copyright law of the authors and the textbook company.
- There have been some changes to the slide deck.

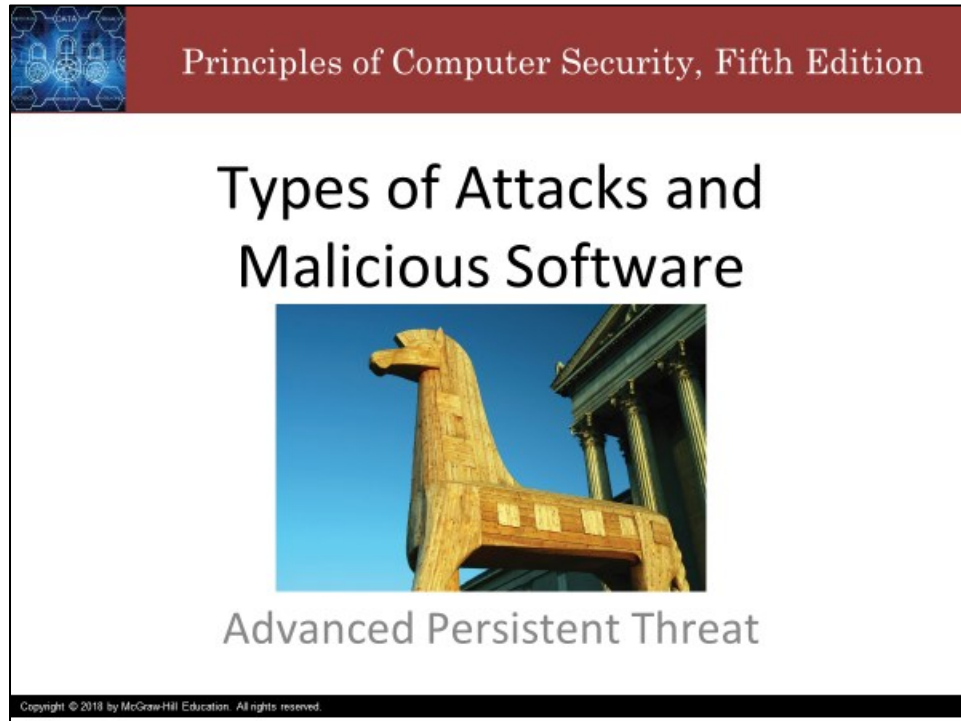
Copyright © 2018 by McGraw-Hill Education. All rights reserved.

There are many, many, many more attacks that you could learn about. The creativity of attackers to exploit vulnerabilities in computer systems and networks, as well as the creativity of the designers and builders who put them there, is boundless. If you want to enjoy some schadenfreude, click through to some of the CVE links in the observed examples sections on the CWE/SANS Top 25 list. It's like a hall of shame in there! But, seriously, you can actually learn a lot through failure.

Thanks for watching, and take care!


Types of Attacks and Malicious Software: Advanced Persistent Threat

Slide 1



Principles of Computer Security, Fifth Edition


Types of Attacks and Malicious Software



Advanced Persistent Threat

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Howdy! In this video, we review advanced persistent threats.



Principles of Computer Security, Fifth Edition

Advanced Persistent Threat

- Focuses on stealth and continuous presence on a system.
- Very advanced method
 - Requires a team, involves high-value targets, multiple backdoors
- Goal: pwn the system
- Typically nation-state actors (sometimes corporate)
- Signs of an APT attack
 - Off-hours activity, trojans, mysterious files, hacker tools, strange data flows

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

The advanced persistent threat (APT) is a method of attack that primarily focuses on stealth and continuous presence on a system.

APT is a very advanced method, requiring a team to maintain access and typically involves high-value targets.


APT typically involves specially crafted attack vectors coupled with phishing or spear-phishing for the initial entry.

Techniques are then employed to develop backdoors and multiple account access routes.

The skill level of the attackers is usually extremely high, as the goal of the attack is to completely own the system and maintain that ownership for a long time without being detected. In this sense, to own means to be able to see and do anything in the system.

Due to the resources required, APT actors are typically at the nation-state level.

Signs of an APT attack include off-hours activity, finding trojans on the system, large files of unknown origin, phishing emails, finding hacker tools like password crackers and network sniffers, and strange data flows. A very stealthy APT will hide all of their activity, but sometimes we get lucky and catch them when they make a mistake. Sometimes, that's the only way we catch them.



Principles of Computer Security, Fifth Edition

Attribution

- The course slides are based on slides developed by the textbook authors, W. "Art" Conklin, Ph.D. (University of Houston), and Greg White, Ph.D. (University of Texas at San Antonio).
- These slides remain the intellectual property protected by US Copyright law of the authors and the textbook company.
- There have been some changes to the slide deck.

<https://www.fireeye.com/current-threats/apt-groups.html>


Copyright © 2018 by McGraw-Hill Education. All rights reserved.

I put a link on the slide to Fireeye's who's who of nation-state APT actors. If you skim through the attack vectors, you'll notice a lot of them use spear phishing.

That's all for now. Thank you for watching. Take care.


Types of Attacks and Malicious Software: Tools

Slide 1



Principles of Computer Security, Fifth Edition


Types of Attacks and Malicious Software



Tools





Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Howdy! In this video, we introduce some of the most famous tools that security professionals and hackers use.



Principles of Computer Security, Fifth Edition

Tools


- Metasploit 
- Social Engineer Toolkit 
- Burp Suite 
- Kali 

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

There are a variety of toolsets used by security professionals that could also be used for malicious purposes.

These toolsets are used by penetration testers when testing the security posture of a system.

The same tools in the hands of an adversary can be used for malicious purposes.



Principles of Computer Security, Fifth Edition

Metasploit metasploit®

- Framework that provides information about security vulnerabilities and aids in penetration testing
- On the bleeding edge of exploits
- Open source, widely distributed, powerful, one of the most popular tools.
- <https://www.metasploit.com/>

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Metasploit is a framework that provides information about security vulnerabilities and aids in penetration testing.

When new vulnerabilities are discovered in systems, Metasploit exploit modules are quickly created in the community.

Metasploit is open source, widely distributed, powerful, and one of the most popular tools used by penetration testers and attackers alike.



Principles of Computer Security, Fifth Edition

Social-Engineer Toolkit



- The Social-Engineering Toolkit (SET) is a set of tools that can be used to target attacks toward the people using systems.
- It supports many attack vectors
 - Spear-phishing, website attacks, infectious media, mass mailing, SMS spoofing, WAP attacks, QR Codes, Powershell, and more.
- <https://www.social-engineer.org/framework/se-tools/computer-based/social-engineer-toolkit-set/>


Copyright © 2018 by McGraw-Hill Education. All rights reserved.

The Social-Engineer Toolkit (SET) is designed to perform advanced attacks against the human element. SET is a standard tool in a penetration testers arsenal. The attacks built into the toolkit are designed to be targeted and focused attacks against a person or organization used during a penetration test.

(source: <https://www.social-engineer.org/framework/se-tools/computer-based/social-engineer-toolkit-set/>)

The organization behind SET is Social Engineer (their motto is “Security through Education,” which I love), and they have a TON of information on the art and science of social engineering.

Slide 5



Principles of Computer Security, Fifth Edition

Burp Suite Burp Suite

- Web application security testing
- Began as port scanner with benefits
 - View web traffic, limited control
- Today is a full service tool
 - Mapping, vulnerability scanning, automation
- Free, Pro, and Enterprise version
 - Pro is \$400 / user, Enterprise is \$6000 - \$24000 / year
- <https://portswigger.net/burp>

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Burp Suite is a graphical tool for testing Web application security.

It began as a port scanner tool with limited additional functionality for controlling and analyzing web traffic.


Today, burp supports the entire testing process, from initial mapping and analysis of an application's attack surface, through finding and exploiting security vulnerabilities.

Burp Suite is a commercial tool, but the free version is reasonably priced, well-liked, and utilized in the pen-testing marketplace.



Principles of Computer Security, Fifth Edition

Kali Linux



- Kali is a Linux distribution that is preloaded with many security tools.
- It includes a whole host of preconfigured, preloaded tools, including Metasploit, Social-Engineer Toolkit, Burp Suite, and others.
- Undisputed industry standard Open-source penetration testing platform.
- <https://www.kali.org/>
- <https://www.offensive-security.com/kali-linux-vm-vmware-virtualbox-image-download/>

Copyright © 2018 by McGraw-Hill Education. All rights reserved.


As legend tells it, years ago, there was a penetration test in an isolated environment where the assessment team was not able to bring in any computers or have network access in or out of the target environment. In order to do the work, the first penetration testing distribution was born. It was a bootable Live CD configured with various tools needed to do the work, and after the assessment was completed, the Live CD was shared online and became very popular.

Kali Linux has a direct lineage from this [original distribution](#), running on through [BackTrack Linux](#), and now is Kali Linux.

(source: <https://www.kali.org/features/>)

[Kali Linux](#) is an [open-source](#), [Debian-based Linux](#) distribution aimed at advanced Penetration Testing and Security Auditing. It contains [several hundred tools](#) targeted towards various information security tasks, such as Penetration Testing, Security Research, Computer Forensics, and Reverse Engineering. It runs on multiple platforms and is freely available to information security professionals and hobbyists.

(source: <https://www.kali.org/docs/introduction/what-is-kali-linux/>)



Principles of Computer Security, Fifth Edition

Attribution

- The course slides are based on slides developed by the textbook authors, W. "Art" Conklin, Ph.D. (University of Houston), and Greg White, Ph.D. (University of Texas at San Antonio).
- These slides remain the intellectual property protected by US Copyright law of the authors and the textbook company.
- There have been some changes to the slide deck.


Copyright © 2018 by McGraw-Hill Education. All rights reserved.

There are many resources freely available online that demonstrate how to use the tools. There are also books and courses (very expensive courses, I might add) that do the same. Even if you don't intend to use them yourself, it is worthwhile to know that they exist and to familiarize yourself with the kinds of things they can do.

Thank you and take care.


Types of Attacks and Malicious Software: Auditing

Slide 1



Principles of Computer Security, Fifth Edition


Types of Attacks and Malicious Software



Auditing

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Howdy! In this video, we discuss auditing: what it is, why it is important, and what should be audited.



Principles of Computer Security, Fifth Edition

Auditing

- Process of assessing the security state of an organization compared against an established standard
- Differs from vulnerability and security assessments
- Should be conducted on a regular basis
- Assessment should include
 - security perimeter
 - policies, procedures, and guidelines governing security
 - employee training
- Security logs are a powerful tool
 - Recall: Auditability, Compromise Recording

Copyright © 2010 by McGraw-Hill Education. All rights reserved.

Auditing in the computer security world is a process of assessing the security state of an organization compared against an established standard.

The important elements here are the standards. Organizations from different communities may have widely different standards, and any **audit** will need to consider the appropriate elements for the specific community. Audits differ from security or vulnerability assessments in that assessments measure the security posture of the organization but may do so without any mandated standards against which to compare them. In a security assessment, general security "best practices" can be used, but they may lack the regulatory teeth that standards often provide. Penetration tests can also be encountered—these tests are conducted against an organization to determine whether any holes in the organization's security can be found. The goal of the penetration test is to penetrate the security rather than measure it against some standard. Penetration tests are often viewed as *white-hat hacking* in that the methods used often mirror those that attackers (often called *black hats*) might use, but the white hat pen testers are doing their work for the benefit of the organization while the black hats are doing it for their own benefit.


You should conduct some form of security audit or assessment on a regular basis. Your organization might spend a lot of money on security, so it is important to measure the effectiveness of the efforts to establish and maintain the desired (or required) security level. In certain communities, audits may be required on a periodic basis with very specific standards against which the security of the organization must be measured. Even if your organization is not part of such a community, periodic assessments are worthwhile.

At a minimum, security assessments should include the security perimeter with all of its components, including both host and network security, the organization's policies, procedures, and guidelines

governing security, and employee training (since people are very often the weakest link and are targets for social engineering and password cracking attacks).

Security logs are a powerful tool for detecting and understanding security incidents. But, preparation is needed to determine what to log and when and how to review the logs. Log too little or review too infrequently, and you miss too many things. Log too much or review too often, and you can't keep up.

Slide 3



Principles of Computer Security, Fifth Edition

Performing Routine Audits


- Examples of items that should be audited on a regular basis include:
 - User access – who, what, when, etc.
 - User rights – ensure least privilege
 - Storage – who, what, how much
 - Retention – what, how, how long, disposal
 - Firewall rules – clean up

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

As part of any good security program, administrators must perform periodic audits to ensure things "are as they should be" with regard to users, systems, policies, and procedures. Installing and configuring security mechanisms is important, but they must be reviewed on a regularly scheduled basis to ensure

they are effective, up to date, and serving their intended function. Here are some examples, but by no means a complete list, of items that should be audited on a regular basis:

- **User access** – Administrators should review which users are accessing the systems, when they are doing so, what resources they are using, and so on. Administrators should look closely for users accessing resources improperly or accessing legitimate resources at unusual times.
- **User rights** – When a user changes jobs or responsibilities, she will likely need to be assigned different access permissions; she may gain access to new resources and lose access to others. To ensure that users have access only to the resources and capabilities they need for their current positions, all user rights should be audited periodically.
- **Storage** – Many organizations have policies governing what can be stored on "company" resources and how much space can be used by a given user or group. Periodic audits help to ensure that no undesirable or illegal materials exist on organizational resources.
- **Retention** – In some organizations, how long a particular document or record is stored can be as important as what is being stored. A records retention policy helps to define what is stored, how it is stored, how long it is stored, and how it is disposed of when the time comes. Periodic audits help to ensure that records or documents are removed when they are no longer needed.
- **Firewall rules** – Periodic audits of firewall rules are important to ensure the firewall is filtering traffic as desired and to help ensure that "temporary" rules do not end up as permanent additions to the ruleset.



Principles of Computer Security, Fifth Edition

Attribution

- The course slides are based on slides developed by the textbook authors, W. "Art" Conklin, Ph.D. (University of Houston), and Greg White, Ph.D. (University of Texas at San Antonio).
- These slides remain the intellectual property protected by US Copyright law of the authors and the textbook company.
- There have been some changes to the slide deck.

Copyright © 2018 by McGraw-Hill Education. All rights reserved.

Thank you and take care.