Lecture 1

1.2.1 Classical Cryptography

The earliest known use of cryptography is found in non-standard hieroglyphs carved into monuments from Egypt's Old Kingdom (4500+ years ago).

Scytale:

Scytale

The ancient Greeks and the Spartans in particular, are said to have used this cipher to communicate during military campaigns. Sender and recipient each had a cylinder (called a scytale) of exactly the same radius. The sender wound a narrow ribbon of parchment around his cylinder, and then wrote on it lengthwise . After the ribbon is unwound, the writing could be read only by a person who had a cylinder of exactly the same circumference.

Polybius Square:

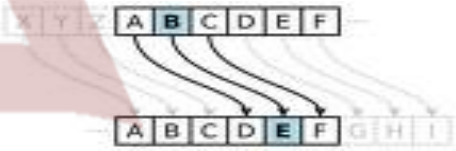
	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	I	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

Another Greek method was developed by Polybius (now called the "Polybius Square"). Each letter is represented by its coordinates in the grid. For example, "BAT" becomes "12 11 44"**.** Developed for telegraphy e.g. pairs of torches

Caesar Cipher:

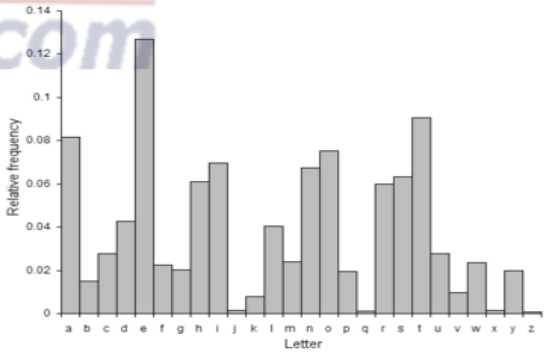
The Romans knew something of cryptography (e.g., the Caesar cipher and its variations). The method is named after Julius Caesar, who used it to communicate with his generals. The Caesar Cipher is an example of what is called a shift cipher. To encode a message, letters are replaced with a letter that is a fixed number of letters beyond the current letter.

Atbash cipher:



Later still, Hebrew scholars made use of simple mono alphabetic substitution ciphers (such as the Atbash cipher) beginning perhaps around 500 to 600 BC. The Atbash cipher is a very specific case of a substitution cipher where the letters of the alphabet are reversed. In other words, all As are replaced with Zs, all Bs are replaced with Ys, and so on.

Cryptography from Muslim History (Medieval Cryptography):



Al- Kindi, wrote a book on cryptology, the "Risalah fi Istikhraj al-Mu'amma" (Manuscript for the Deciphering Cryptographic Messages), circa 850CE. This book apparently antedates Western European cryptography works by 300 years and predates writings on probability and statistics by Pascal and Fermat by nearly 800 years. He was a pioneer in cryptanalysis and cryptology, and devised new methods of breaking ciphers, including the frequency analysis method. In his book Al-Kindi described the first cryptanalysis techniques, including some for polyalphabetic ciphers, cipher classification, Arabic phonetics and syntax, and, most importantly, gave the first descriptions on frequency analysis. He also covered methods of encipherments, cryptanalysis of certain encipherments, and statistical analysis of letters and letter combinations in Arabic.

Cryptography in the Renaissance Period:

Essentially all ciphers remained vulnerable to the cryptanalytic technique of frequency analysis until the development of the polyalphabetic cipher, and many remained so thereafter. The polyalphabetic cipher was most clearly explained by Leon Battista Alberti around the year 1467, for which he was called the "father of Western cryptology".

Modern Cryptography:

Both cryptography and cryptanalysis have become far more mathematical since World War II. Even so, it has taken the wide availability of computers, and the Internet as a communications medium, to bring effective cryptography into common use by anyone other than national governments or similarly large enterprises. The era of modern cryptography really begins with Claude Shannon, arguably the father of mathematical cryptography, with the work he did during WWII on communications security.

The First Encryption Standard:

The mid-1970s saw two major public (i.e., non-secret) advances. First was the publication of the draft Data Encryption Standard in the U.S. Federal Register on 17 March 1975. The proposed DES cipher was submitted by a research group at IBM, at the invitation of the National Bureau of Standards (now NIST), in an effort to develop secure electronic communication facilities for businesses such as banks and other large financial organizations. The aging DES was officially replaced by the Advanced Encryption Standard (AES) in 2001 when NIST announced FIPS 197. After an open competition, NIST selected Rijndael, submitted by two Belgian cryptographers, to be the AES.

Lecture 2&3

The Threat Environment: Attackers and Their Attacks

The world today is a dangerous place for corporations. The Internet has given firms access to billions of customers and other business partners. But the Internet has also given criminals access to hundreds of millions of corporations and far more individuals. Wireless transmission has brought new mobility but has also allowed attackers to enter corporations stealthily. Bypassing firewalls designed to keep intruders from coming in through the Internet

The Threat Environment:

The threat environment consists of the types of attackers and attacks that company face.

Security Goals:

Three common core goals are referred to as CIA:

- Confidentiality
- Integrity
- Availability

Confidentiality: Confidentiality means that people cannot read sensitive information, either while it is on a computer or while it is traveling across a network

Integrity: Integrity means that attackers cannot change or destroy information, either while it is on a computer or while it is traveling across a network. Or, at least, if information is changed or destroyed, then the receiver can detect the change or restore destroyed data.

Availability: Availability means that people who are authorized to use information are not prevented from doing so

Countermeasures:

- Tools used to thwart attacks
- Also called safeguards, protections, and controls

- Countermeasures can be technical, human, or a mixture of the two

Three Types Of Countermeasures:

- Preventative: keep attacks from succeeding. Most controls are preventative
- Detective: identify when a threat is attacking, and especially when it is succeeding. Fast detection can minimize damage
- Corrective: get the business process back on track after a compromise.

The TJX Data Breach: The TJX Companies, Inc. (TJX)_____ It is an online business company. A group of more than 2,500 retail stores companies operating in the United States, Canada, England, Ireland, and several other countries. On December 18, 2006, TJX detected "suspicious software" on its computer systems. They called in security experts who confirmed an intrusion and probable data loss. They notified law enforcement immediately. Company estimated that 45.7 million records with limited personal information included. Much more information was stolen on 455,000 of these customers. TJX was not in compliance with Industry-Data Security Standard (PCI-DSS) (later found to meet only 3 of 12 control objectives). TJX has suffered from damages of $256 million.

Lecture 4

Employee and Ex-employee threats:

Are "insiders" the biggest threat? They are dangerous because they have knowledge of internal systems. They often have the permissions to access systems. They often know how to avoid detection. Employees generally are trusted. Employees and ex-employees are dangerous because they have extensive knowledge of systems, have the credentials needed to access sensitive parts of the systems, often know how to avoid detection, and can benefit from the trust that usually is accorded to "our people" . IT and especially IT security professionals are the greatest employee threats. "Who will guard the guards themselves?"

Sophisticated computer knowledge not required:

In 23 financial cybercrimes committed between 1996 and 2002, 87 percent were accomplished without any "sophisticated programming" – Keeney. The US department of justice has a website **http://www.cybercrime.gov** which lists federal cybercrime prosecutions. Roughly half the defendants are IT or IT security employees or ex-employees.

Employee Sabotage:

They may destruction of hardware, software, or data and plant time bomb or logic bomb on computer. Sabotage comes from the French word for "shoe" because disgruntled workers in the early years of the Industrial Revolution supposedly threw their wooden shoes into machines to stop production. Tim Lloyd, a computer systems administrator, was fired for being threatening and disruptive. In retaliation, Lloyd planted a logic bomb on a critical server. When pre-set conditions occurred, the logic bomb destroyed the programs that ran the company's manufacturing machines. Lloyd also took home and erased the firm's backup tapes to prevent recovery. Lloyd's sabotage resulted in USD 10 million in immediate business losses, USD 2 million in reprogramming costs, and 80 layoffs. The attack led to a permanent loss of the company's competitive status in the hi-tech instruments and measurements market because the company could not re-build the proprietary software it had been using. *-Sharon Gaudin, Computerworld, 2002*

Two traffic Engineers working for the city of Los Angeles pleaded guilty to hacking the city's traffic center and disconnecting traffic signals at four LA's busiest intersections. They then locked out the controls to these intersections so that it took four days to restore control. They did this a few hours before their union's scheduled job action against the city in support of contract negotiations. For this infraction, they received

240 days of community service, and were required to have their computers at home and work monitored . *Dan Goodin, 2008*

Employee Hacking:

Hacking is intentionally accessing a computer resource without authorization or in excess of authorization . First documented use of the word "hacker" was in Steve Levy's book, Hackers in 1984. Penalties are the same whether you were trying to steal a million dollars or were merely "testing security"

Employee Financial Theft:

Misappropriation of assets: E.g. assigning them via computer to themselves

Theft of money: E.g. manipulation of an application to be paid a bonus

Two accountants at Cisco Systems illegally accessed a corporate computer to issue themselves USD 8 million worth of Cisco stocks. http://www.cybercrime.gov , 2001. In another case, Quitugua Sabathia, 31, of Vallejo, California, used her computer to embezzle more than USD 875,000 from the North Bay Health Care Group. A former accounts payable clerk at North Bay, she accessed the firm's accounting software and issued approximately 127 checks payable to herself and others. To conceal the fraud, she altered the electronic check register to make it appear that the checks had been payable to North Bay's vendors.

Employee Theft of Intellectual Property (IP):

Copyrights and patents (formally protected): Intellectual Property (IP) is the information owned by the company and protected by law.

Trade secrets: plans, product formulations, business processes, and other info that a company wishes to keep secret from competitors.

A former DuPont research scientist admitted downloading trade secrets worth USD 400 million. Only when he announced his decision to leave was his downloading behavior analyzed. The analysis found that he had downloaded 16,700 documents – 15 times more than the second-highest downloader. Most of these documents had nothing to do with his primary research area. *PC World, 2007*

Employee Extortion:

Perpetrator tries to obtain money or other goods by threatening to take actions that would be against the victim's interest. For example the employee might deploy a logic bomb on the company's computer. Stealing Intellectual Property (IP) and demanding money for not passing on the information is also extortion .

Harassment of Other Employees:

Via e-mail Displaying inappropriate material

Washington Leung left a firm and later logged into his ex-firm's servers using passwords given to him while employed there. He deleted over 900 files related to employee compensation. To frame a female co-worker, he gave her a USD 40,000 annual raise, and a USD 100,000 bonus. He created a hotmail account in the name of the female employee and sent senior managers an email containing information from the deleted files. *http://www.cybercrimes.gov*

Other Types of Abuse:

Internet Abuse

- Downloading inappropriate material, which can lead to harassment lawsuits and viruses
- Downloading pirated software, music, and video, which can lead to copyright violation penalties

- Excessive personal use of the Internet at work

Non-Internet Computer Abuse

- Unauthorized access to private personal data on internal systems by curious employees

This type of behavior was detected in the 2008 Presidential election campaign and in several celebrity hospitalizations. *Los Angeles Times, 2008*

A survey of 300 senior IT administrators in a London security conference and trade show found that one in three admitted to looking at confidential or personal information in ways unrelated to their jobs. Computerworld 2008

Carelessness:

- *Loss of computers or data media containing sensitive information*
- *Carelessness leading to the theft of such information*

A Ponemon survey in 2008 found that 630,000 laptops are lost at airports each year. Although only some of these are corporate computers, airports are not the only place where laptops are lost, and lost media (US drives) can be just as damaging. Ponemon Institute

Other "Internal" Attackers:

- *Contract workers who work for the firm for brief periods of time*
- *Workers in contracting companies*

Contract workers often get credentials that are not deleted after their engagement ends.

Claude Carpenter, a 19 year old employee of a firm managing servers for the US Internal Revenue Service (IRS) planted a logic bomb on the servers after he learned he was about to be fired. The IRS would have been the real victim had his logic bomb succeeded. He also planted the code on his supervisor's computer to frame the supervisor. http://www.cybercrime.gov (2001)

Lecture 5

Traditional External Attackers:

Traditional external attackers use the Internet to send malware into corporations, hack into corporate computer and do other damage.

Classic Malware: Viruses and Worms:

Malware: A generic name for any "evil software". The most widely known type of malware is the virus, but there are many other types including worms, Trojan horses, RATs, spam, and others. Malware is a very serious threat. In June 2006 Microsoft reported results from a survey of users who allowed their computers to be scanned for malware. The survey found 16 million pieces of malware on the 5.7 million machines examined. Viruses: Programs that attach themselves to legitimate programs on the victim's machine. Later when infected programs are transferred to other computers and run, the virus attaches itself to other programs on those machines. Spread today primarily by e-mail with infected attachments. Also by instant messaging, file transfers, file sharing programs, downloads from malicious websites, etc

Through networked applications, viruses can spread very rapidly today.

When Macintosh users searched BitTorrent sites in early 2009, they found that they were able to download the newly released CS4. They would also download a program on the downloader's computer to crack the

software. The copy of CS4 was clean, however, when the downloader ran the cracking program, he or she got a dialog box saying "Adobe CS4 Crack [intel] requires that you type your password."

-"Mac Trojan Horse Found In Pirated Adobe CS4 (http://blogs.zdnet.com)

Worms: Full programs that do not attach themselves to other programs. Like viruses, can spread by e-mail, instant messaging, and file transfers. In general worms act much like viruses and can spread via email and in other ways that viruses spread.

Direct-propagation Worms: In addition, direct-propagation worms can jump from one computer to another without human intervention on the receiving computer. Computer must have a vulnerability for direct propagation to work. Direct-propagation worms can spread extremely rapidly because they do not have to wait for users to act.

Blended Threats: Malware propagates in several ways—like worms, viruses, compromised webpages containing mobile code, etc. By propagating in multiple ways, blended threats increase their likelihood of success. MessageLabs reported in August 2006 that 1% of all email contains viruses, worms, or blended threats. During major outbreaks, one in ten email messages may contain viruses, worms, or blended threats. (http://www.messagelabs.com)

In 2004, the Aberdeen group surveyed 162 companies. They found that each firm lost an average of USD 2 million per virus or worm incident and spent an additional USD 100,000 to clean up computers after an attack. Both numbers increased with company size. Most companies reported enduring on average one incident per year, although many firms reported multiple incidents. (http://www.aberdeen.com)

Another security firm Mi2g estimated that damage from malware in 2004 alone averaged USD 290 per PC in the firms it studied. (http://www.mi2g.com)

Trojan Horses and Rootkits: (Non-mobile Malware)

They must be placed on the user's computer through one of a growing number of attack techniques. May be placed on computer by hackers or placed on computer by virus or worm as part of its payload. The victim can be enticed to download the program from a website or FTP site. Mobile code executed on a webpage can download the non-mobile malware.

- **Trojan horse:** A program that replaces an existing system file, taking its name.

  Most non-mobile malware programs are Trojan horses. Early Trojan horses were programs that pretended to be one thing, such as a game or a pirated version of a commercial program, but really were malware. Many of these classic Trojan horses still exist. Today, however, when we talk about a trojan horse, we mean a program that hides itself by deleting a system file and taking on the system file's name. Trojan horses are difficult to detect because they look like legitimate system files.

Remote Access Trojans (RATs): Remotely control the victim's PC

The attacker can remotely do pranks such as opening and closing your CD drive, or by typing text on your screen. There are many legitimate remote access programs that allow a remote user to work on a machine or do diagnostics. However, RATs are typically stealthy in order to avoid detection by the owner of the machine.

Downloader: Small Trojan horses that download larger Trojan horses after the downloader is installed.

Downloaders are usually fairly small programs which makes detection difficult. However the larger trojan horse is capable of doing much more damage.

Spyware: Programs that gather information about you and make it available to the adversary

- Cookies that store too much sensitive personal information

Although the biggest problem with spyware is the theft of information, spyware also tends to make computers run sluggishly. One new form of spyware is camera spyware which spies on the victim visually by turning on its camera and perhaps also its microphone. Websites are allowed to store small text strings called *cookies* on your PC. The next time you go to the website the website can retrieve the cookie. Cookies have many benefits like remembering your password each time you visit. Cookies can also remember what happened last in a series of screens leading to purchases. However, when cookies record too much sensitive information about you, they become spyware.

Keystroke loggers: Keystroke loggers capture all of your keystrokes. They then look through the collected keystrokes for usernames, passwords, credit card numbers, and other sensitive information. They send this information to the adversary.

Password-stealing spyware: Tells you that you have been logged out of the server you are visiting and asks you to retype your username and password. If you do the spyware sends the username and password to the attacker.

Data mining spyware: Searches through your disk drives for the same types of information sought by keystroke loggers. It also sends this information to the adversary.

Lecture 6

Mobile Code:

When you download a webpage it may contain executable code as well as text, images, sounds, and video. This is called ***mobile code*** because it executes on whatever machine downloads the webpage. In most cases mobile code is innocent and often is necessary if a user wishes to use a website's functionality.

- Executable code on a webpage
- Code is executed automatically when the webpage is downloaded
- Javascript, Microsoft Active-X controls, etc.
- Hostile code can do damage if computer has vulnerability

Social Engineering in Malware:

Social engineering attacks take advantage of flawed human judgment by convincing the victim to take actions that are counter to security policies. Social engineering is attempting to trick users into doing something that goes against security policies. For example if an employee receives an email message warning about a mass layoff being imminent, he or she may open an attachment and therefore download a virus, worm, or Trojan horse.

Social Engineering in Malware:

- Spam
- Phishing
- Spear phishing (aimed at individuals or specific groups)
- Hoaxes

Spam : The bane of all email users is spam which is defined as unsolicited commercial e-mail. In addition to being annoying, spam messages are often fraudulent or advertize dangerous products. Spam has become a common vehicle for distributing viruses, worms, trojan horses, and many other types of malware. According to MessageLabs, 73% of all e-mail messages were spam in March 2009. http://www.messagelabs.com

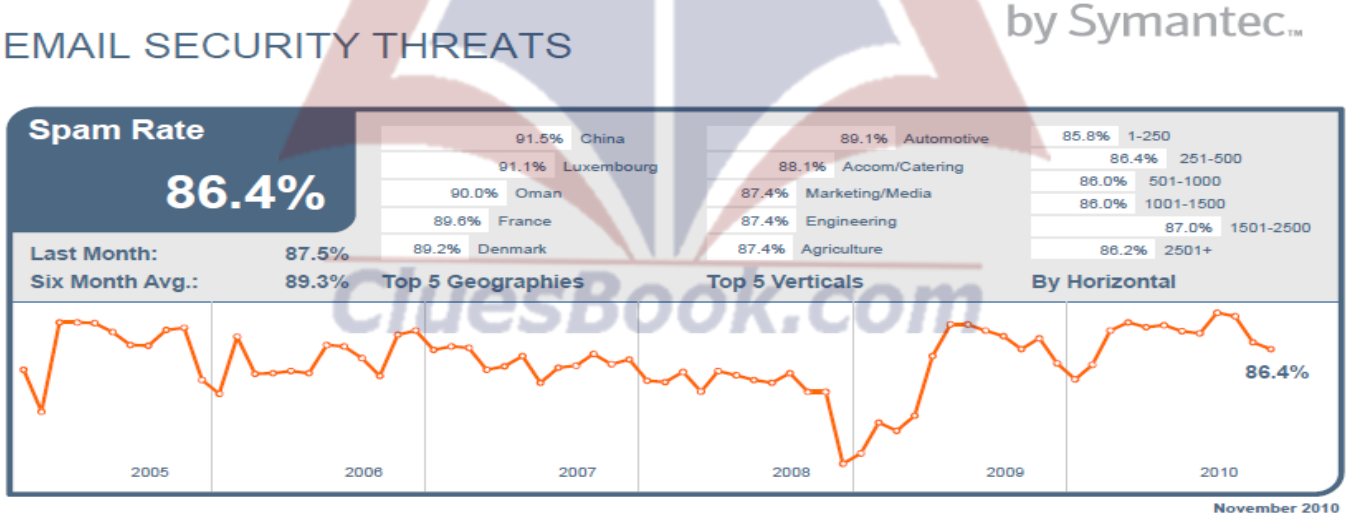
Even the load on networks caused by simply transmitting and storing spam can be significant . New forms of spam consist of image bodies instead of text bodies to avoid detection from scanning programs . Image spam messages are much larger than traditional text spam messages.

Phishing: In phishing attacks victims receive email messages that appear to come from a bank or another firm with which the victim does business. The message may even direct the victim to an authentic-looking website. The official appearance of the message and website often fool the victim into giving out sensitive information. A Gartner survey in 2007 revealed that US consumers were scammed out of USD 3.2 billion that year. In 2004 when phishing was fairly new but already well known to consumers, a study showed consumers a group of email messages and asked whether each email was a phishing attack or not. The consumers judged 28% of the phishing messages to be legitimate messages. They also believed a fair number of legitimate messages were phishing messages.

Spear phishing: (aimed at individuals or specific groups) Normally phishing attacks tend to appeal broadly to many people so they can dupe as many people as possible. In one case a number of CEOs received a message disguising itself as a court order. The message directed the CEOs to a website uscourts.com. There the CEOs could find court documents and a plug-in to view the documents. The plug-in was spyware!
(http://www.pcworld.com) 2008

Hoaxes: The sulfnbk.exe hoax told computers that a virus called AOL.exe was travelling around the Internet. The hoax said that they should delete the file sulfnbk.exe. Victims who did so were really deleting their AOL access. Other hoaxes have tried to persuade victims to delete their antivirus protection and even critical operating files needed for their computers operation.

MessageLabs Intelligence (http://www.messagelabs.com)



McAfee Spam Checklist:

1. Unsubscribe from legitimate mailings that you no longer want to receive.
2. Be selective about the Web sites where you register your email address.
3. Avoid publishing your email address on the Internet.
4. Delete all spam.
5. Avoid clicking on suspicious links in email or IM messages as these may be links to spoofed websites.
6. Open unknown email attachments. These attachments could infect your computer.

7. Reply to spam. Typically the sender's email address is forged, and replying may only result in more spam.
8. Fill out forms in messages that ask for personal or financial information or passwords
9. Buy products or services from spam messages.
10. Open spam messages.

Lecture 7

Traditional External Attackers

Hackers

Traditional Hackers: In the 1970s, malware writers were joined by external hackers who began to break into corporate computers that were connected to modems. Today nearly every firm is connected to the Internet which harbors millions of external hackers

- Motivated by thrill, validation of skills, sense of power
- Motivated to increase reputation among other hackers
- Often do damage as a byproduct
- Often engage in petty crime (ComputerWeekly.com2008)

In 2009, vandals broke into a computerized road sign in Austin, Texas, and changed its message to read: "The end is near! Caution! Zombies Ahead!" Dallasnews.com, January 2009

Raymond Torricelli, broke into NASA computers and used up processor time, money, and disk resources to divert chat users to a website for which he admitted to being paid USD 400 per week. Cybercrime.gov, 2001

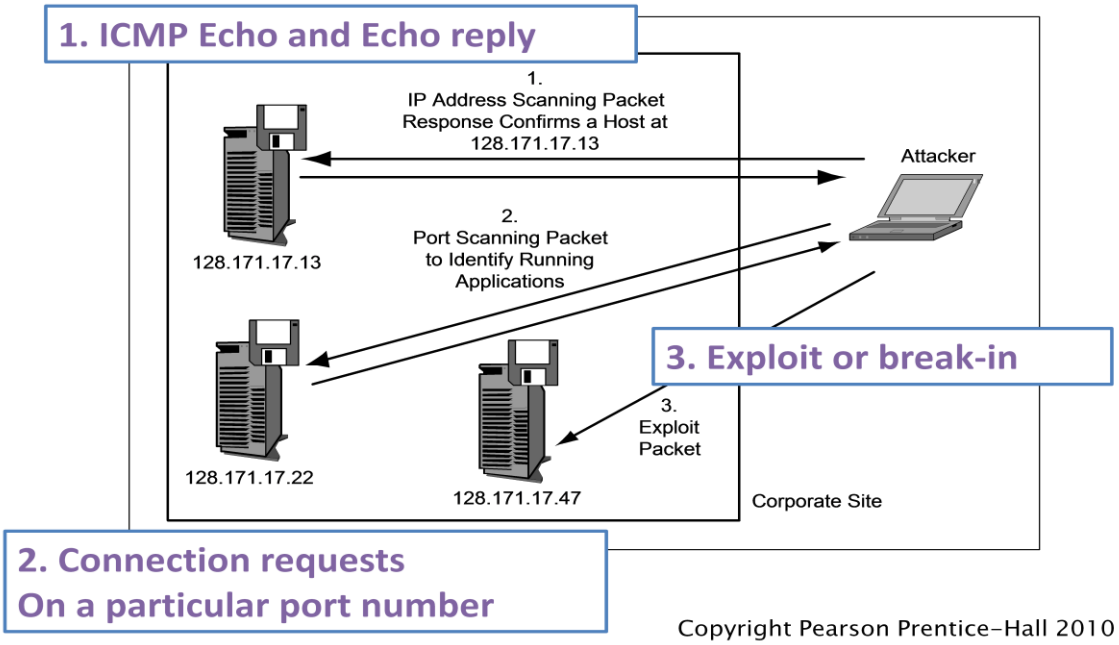
Just like a thief who wants to rob a home does reconnaissance of the neighborhood and gathers information to determine which house to break into, hackers also tend to do reconnaissance before breaking into a computer? As the figure shows, the attacker often sends probe packets into a network. These probe packets are designed to elicit replies from internal hosts and routers.

Anatomy of a Hack:

Reconnaissance probes (see figure)

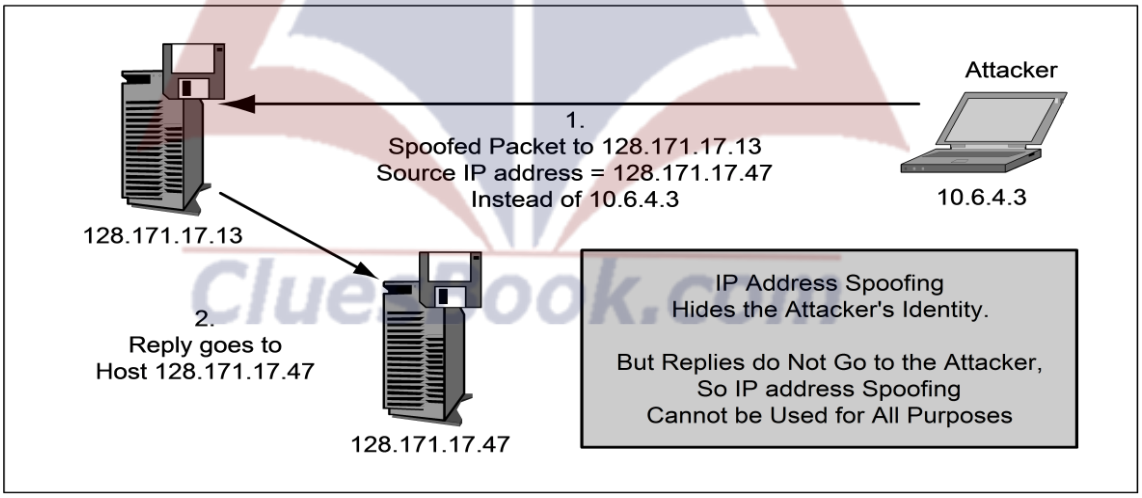
- IP address scans to identify possible victims
- Identify active hosts
- ICMP Echo and Echo reply messages
- Port scans (connection requests) to learn which services are open on each potential victim host

Port 80 is the well known port for HTTP web servers. There are many well known port numbers between 0 and 1023. Each indicates the presence of a particular type of application.



The exploit: The specific attack method that the attacker uses to break into the computer is called the attacker's exploit. The act of implementing the exploit is called exploiting the host

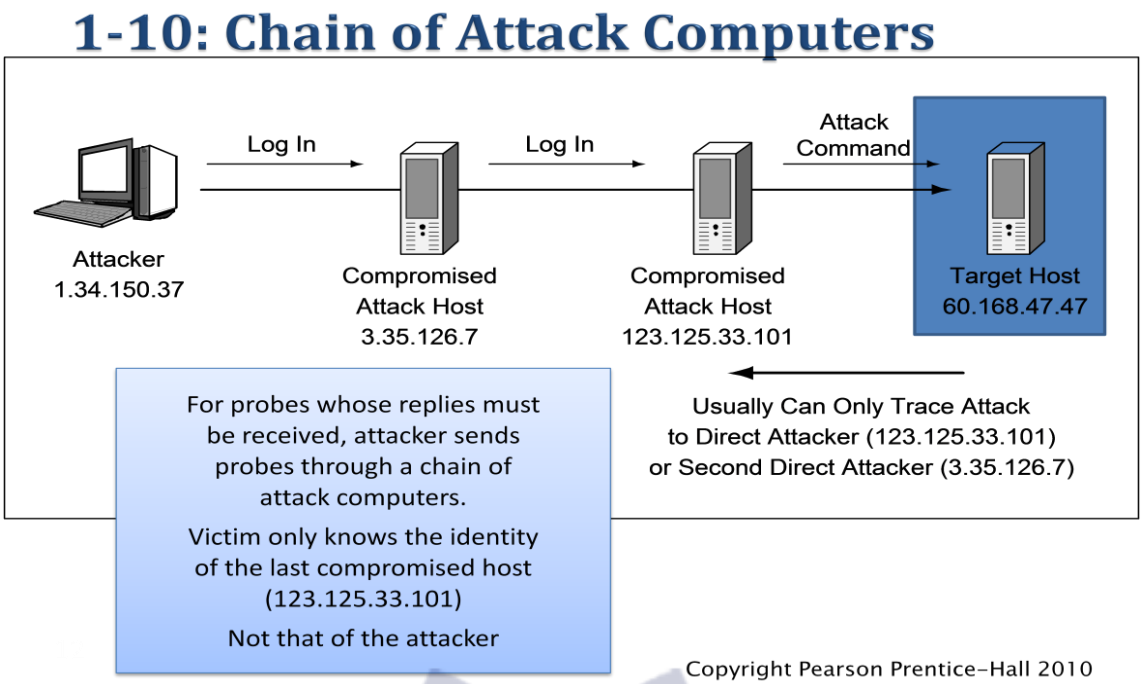
Source IP Address Spoofing



Some exploit packets cannot be spoofed so the attacker uses chain of attack

Chain of attack computers:

- The attacker attacks through a chain of victim computers
- Probe and exploit packets contain the source IP address of the last computer in the chain
- The final attack computer receives replies and passes them back to the attacker
- Often, the victim can trace the attack back to the final attack computer
- But the attack usually can only be traced back a few computers more



Social Engineering: Social engineering is often used in hacking. Social engineering (as we saw earlier) is attempting to trick users into doing something that goes against the interests of security. It is often successful because it focuses on human weaknesses instead of technological weaknesses.

In one social engineering ploy, a hacker calls a secretary claiming to be working with the secretary's boss. The hacker then asks for sensitive information such as a password or sensitive file. In one study, US Treasury Dept. inspectors posing as computer technicians called 100 Internal Revenue Service (IRS) employees and managers. The treasury agents asked each target for his or her username and asked the user to change the password to a specific password chosen by the "technician." This was a clear violation of policy but 35% fell for the ploy in 2005. *Washington Post, March 2005*

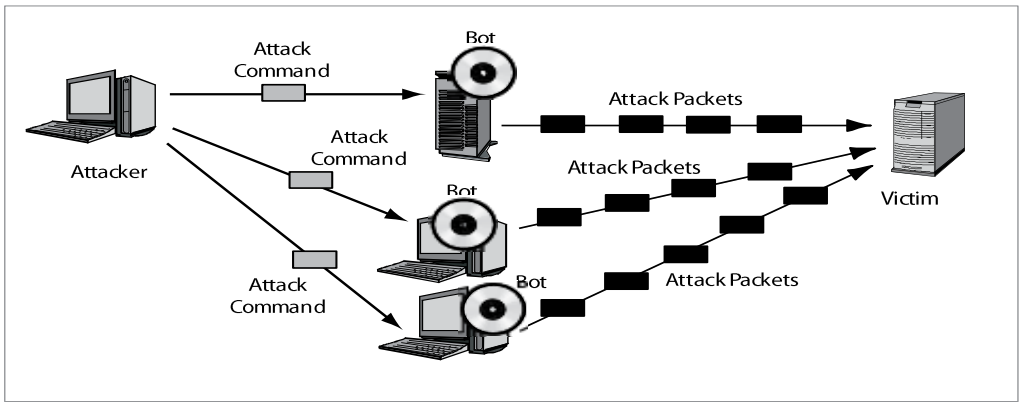
Methods of Social Engineering:

- *E-mail attack messages with attractive subjects*
- *Piggybacking*
- *Shoulder surfing*
- *Pretexting*

Piggybacking is following someone through a secure door, without entering a pass code. Looking over someone's shoulder when he or she types a password is shoulder surfing. In pretexting the attacker calls claiming to be a certain customer in order to get private information about that customer.

Bots: They are updatable attack programs (see figure). Botmaster can update the software to change the type of attack the bot can do. May sell or lease the botnet to other criminals. Botmaster can update the bot to fix bugs. Jeanson Ancheta was indicted for crimes involving a large botnet. Ancheta was charged with creating the botnet and installing adware programs, which constantly pop up advertisements, on these computers for a fee. He was also charged with renting parts of his botnet to other criminals for spam and denial of service attacks. http://www.cybercrime.gov (Nov 2005)

Denial-of-Service (DoS) Attacks: Make a server or entire network unavailable to legitimate users. Typically send a flood of attack messages to the victim. Distributed DoS (DDoS) Attacks (see figure). Bots flood the victim with attack packets. Attacker controls the bot.



In 2001, the Code Red virus attacked the US Whitehouse website. Fortunately the attacks were made against the website's IP address in stead of its host name. The White House merely changed the IP address of whitehouse.gov To attack a server, the bots might flood the server with TCP connection-opening requests (TCP SYN segments). A server reserves a certain amount of capacity each time it receives a SYN segment. By flooding a computer with SYN segments, the attacker can cause the server to run out of resources and therefore crash.

Skill Levels:

- Expert attackers are characterized by strong technical skills and dogged persistence
- Expert attackers create hacker scripts to automate some of their work
- Scripts are also available for writing viruses and other malicious software

Todays hacker scripts often have easy to use graphical user interfaces and look like commercial products. Many scripts are available on the Internet…These easy to use scripts have created a new type of hacker "the script kiddie.

- Script kiddies use these scripts to make attacks
- Script kiddies have low technical skills
- Script kiddies are dangerous because of their large numbers

In February 2000 a number of major firms were affected by devastatingly effective DDOS attacks that blocked each of their e-commerce systems for hours at a time. Victims included CNN, ebay, Yahoo, ZDNET, and others. At first the attacks were thought to be work of an elite hacker. However, the culprit was found to be a 15 year old script kiddie in Canada. Virus and other Malware writers also have written long programs for creating new malware. Viruses have become so easy to create with these tools that Svan Jaschan, an 18 year old German student, who had never written a virus before, was responsible for 70% of the virus activity in the first half of 2004. (http://www.sophos.com)

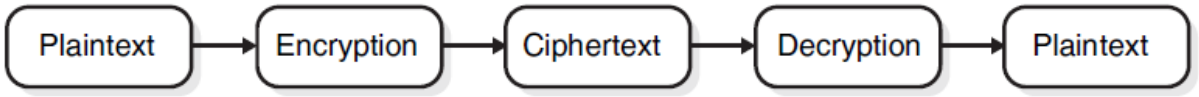
Today tools are available for creating all types of exploits . One of the most important is the MetaSploit Framework which makes it easy to take a new exploitation method and rapidly turn it into a full attack

program. Metasploit is used both by attackers to launch attacks, and by security professionals to test the vulnerability of their systems to specific exploits. (http://www.metasploit.com)

Lecture 8

An Introduction to Cryptography

When data is stored on a computer, it is usually protected by logical and physical access controls. When this same sensitive information is sent over a network, it can no longer take these controls for granted, and the information is in a much more vulnerable state. It needs to be encrypted. Encryption is a method of transforming readable data, called plaintext, into a form that appears to be random and unreadable, which is called cipher text. This enables the transmission of confidential information over insecure channels without unauthorized disclosure.



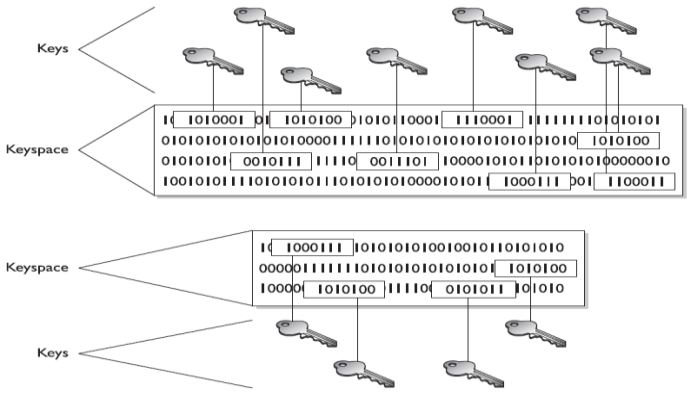
A system or product that provides encryption and decryption is referred to as a cryptosystem and can be created through hardware components or program code in an application. The cryptosystem uses an encryption algorithm. Most algorithms are complex mathematical formulas that are applied in a specific sequence to the plaintext.

A cryptosystem encompasses all of the necessary components for encryption and decryption to take place. Pretty Good Privacy (PGP) is just one example of a cryptosystem.

A cryptosystem is made up of at least the following:

1. Software 2. Protocols 3. Algorithms 4.Keys

Most encryption methods use a secret value called a ***key*** *(usually a long string of bits), which works with the algorithm to* encrypt and decrypt the text. The ***algorithm,*** the set of rules also known as the cipher, dictates how enciphering and deciphering takes place. Many of the mathematical algorithms used in computer systems today are publicly known and are not the secret part of the encryption process. If the internal mechanisms of the algorithm are not a secret, then something must be.The secret piece of using a well- known encryption algorithm is the key. In encryption, the *key (cryptovariable) is a value that comprises a large sequence of* random bits. An algorithm contains a *keyspace, which is a range of values that can be used to construct a* key. The larger the keyspace, the more available values can be used to represent different keys—and the more random the keys are, the harder it is for intruders to figure them out. When the algorithm needs to generate a new key, it uses random values from this keyspace. For example, if an algorithm allows a key length of 2 bits, the keyspace for that algorithm would be 4, which indicates the total number of different keys that would be possible. That would not be a very large keyspace, and certainly it would not take an attacker very long to find the correct key that was used. A large keyspace allows for more possible keys. Today, we are commonly using key sizes of 128, 256, 512, or even 1,024 bits and larger. So a key size of 512



bits would provide a 2^{512} possible combinations (the keyspace). The encryption algorithm should use the entire keyspace and choose the values to make up the keys as randomly as possible.

Key Length and Exhaustive Search Time

Key Length in Bits	Number of Possible Keys
1	Each extra bit
2	
4	
8	
16	
40	
56	
112	
112	
168	
256	
512	

Time Required For Decryption

Key Size (bits)	Number of Alternative Keys	Time required at 1 decryption/μs	Time required at 10^6 decryptions/μs
32	$2^{32} = 4.3 \times 10^9$	2^{31} μs = 35.8 minutes	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	2^{55} μs = 1142 years	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	2^{127} μs = 5.4×10^{24} years	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	2^{167} μs = 5.9×10^{36} years	5.9×10^{30} years
26 characters (permutation)	$26! = 4 \times 10^{26}$	2×10^{26} μs = 6.4×10^{12} years	6.4×10^6 years

Major Symmetric Key Encryption Ciphers

	RC4	DES	3DES	AES
Key Length (bits)	40 bits or more	56	112 or 168	128, 192, or 256
Key Strength	Very weak at 40 bits	Weak	Strong	Strong
Processing Requirements	Low	Moderate	High	Low
RAM Requirements	Low	Moderate	Moderate	Low
Remarks	Can use keys of variable length	Created in the 1970s	Applies DES three times with two or three different DES keys	Today's gold standard for symmetric key encryption

Kerckhoffs' Principle:

Auguste Kerckhoffs published a paper in 1883 stating that the only secrecy involved with a cryptography system should be the key. He claimed that the algorithm should be publicly known. He asserted that if security were based on too many secrets, there would be more vulnerabilities to possibly exploit. Cryptographers in the private and academic sectors agree with Kerckhoffs' principle, because making an algorithm publicly available means that many more people can view the source code, test it, and uncover any type of flaws or weaknesses. It is the attitude of "many heads are better than one." Once someone uncovers some type of flaw, the developer can fix the issue, and provide society with a much stronger algorithm. But, not everyone agrees with this philosophy. Governments around the world create their own algorithms that are not released to the public. Their stance is that if a smaller number of people know how the algorithm actually works, then a smaller number of people will know how to possibly break it. Cryptographers in the private sector do not agree with this practice and do not trust algorithms they cannot examine. It is basically the same as the open-source versus compiled software debate that is in full force today. The ***strength*** *of an encryption method comes from the algorithm, the secrecy of the* key, the length of the key, the initialization vectors, and how they all work together within the cryptosystem. When strength is discussed in encryption, it refers to how hard it is to figure out the algorithm or key, whichever is not made public. The strength of an encryption method correlates to the amount of necessary processing power, resources, and time required to break the cryptosystem or to figure out the value of the key. Breaking a cryptosystem can be accomplished by a brute force attack, which means trying every possible key value until the resulting plaintext is meaningful. Depending on the algorithm and length of the key, this can be an easy task or one that is close to impossible. The goal when designing an encryption method is to make compromising it too expensive or too time-consuming. Another name for cryptography strength is ***work factor,*** which is an estimate of the effort and resources it would take an attacker to penetrate a cryptosystem. Important elements of encryption are to use an algorithm without flaws, use a large key size, use all possible values within the keyspace, and to protect the actual key. If one element is weak, it could be the link that dooms the whole process. Even if a user employs an algorithm that has all the requirements for strong encryption, including a large keyspace and a large and random key value, if he shares his key with others, the strength of the algorithm becomes almost irrelevant.

Services of Cryptosystems

1. **Confidentiality** **2. Integrity** **3. Authentication** **4. Authorization**
5. **Nonrepudiation**

Confidentiality: Renders the information unintelligible except by authorized entities

Integrity: Data has not been altered in an unauthorized manner since it was created, transmitted, or stored

Authentication: Verifies the identity of the user or system that created information

Authorization: Upon proving identity, the individual is then provided with the key or password that will allow access to some resource

Nonrepudiation: Ensures that the sender cannot deny sending the message.

If David sends a message and then later claims he did not send it, this is an act of repudiation. When a cryptography mechanism provides nonrepudiation, the sender cannot later deny he sent the message.(He can try to deny it, but the cryptosystem proves otherwise). Suppose your boss sends you a message telling you that you will be receiving a raise that doubles your salary. The message is encrypted, so you can be sure it really came from your boss (authenticity. Someone did not alter it before it arrived at your computer (integrity). No one else was able to read it as it traveled over the network (confidentiality). Your boss cannot deny sending it later when he comes to his senses (nonrepudiation). Military and intelligence agencies are very concerned about keeping information confidential, so they would choose encryption mechanisms that provide a high degree of secrecy. Financial institutions care about confidentiality, but they also care about the integrity of the data being transmitted, so the encryption mechanism they would choose may differ from the military's encryption methods. If messages were accepted that had a misplaced decimal point or zero, the ramifications could be far reaching in the financial world. Legal agencies may care most about the authenticity of the messages they receive. If information received ever needed to be presented in a court of law, its authenticity would certainly be questioned; therefore, the encryption method used must ensure authenticity, which confirms who sent the information

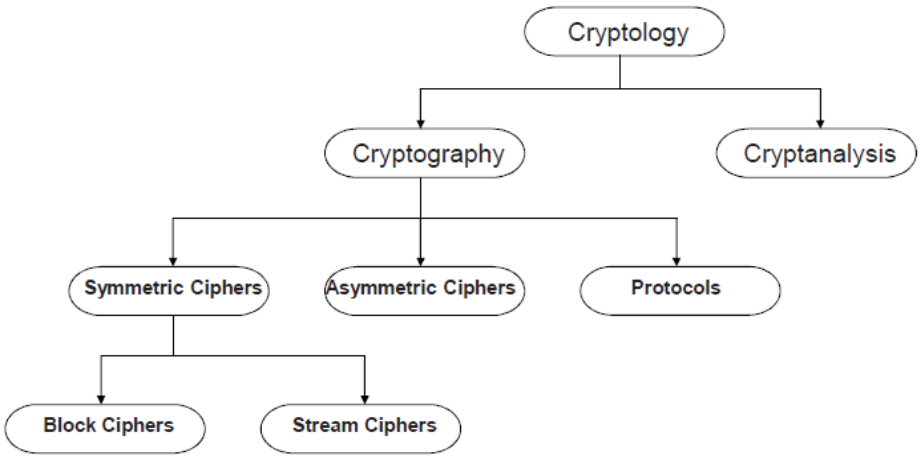
Lecture 9

Cryptanalysis: Practice of breaking cryptic systems

Entity authentication Proving the identity of the entity that sent a message

Work factor Estimated time, effort, and resources necessary to break a cryptosystem

Classification Of The Field Of Cryptology



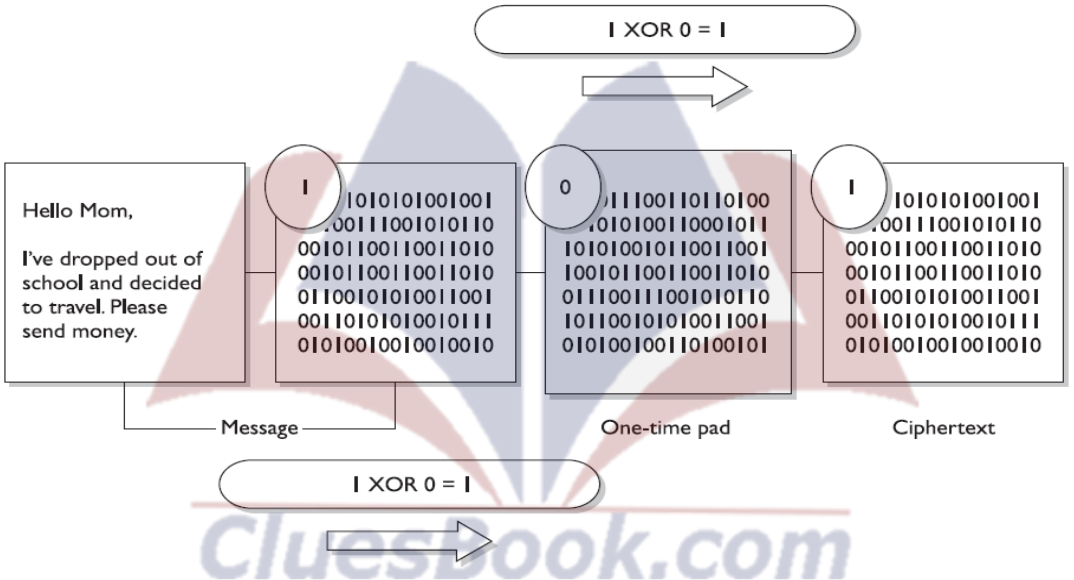
One-time Pad:

A one-time pad is a perfect encryption scheme because it is considered unbreakable if implemented properly. It was invented by Gilbert Vernam in 1917, so sometimes it is referred to as the Vernam cipher. This cipher does not use shift alphabets, as do the Caesar and Vigenere ciphers discussed earlier, but instead uses a pad made up of random values. Our plaintext message that needs to be encrypted has been converted into bits, and our one-time pad is made up of random bits. This encryption process uses a binary mathematic function called exclusive-OR, usually abbreviated as XOR.

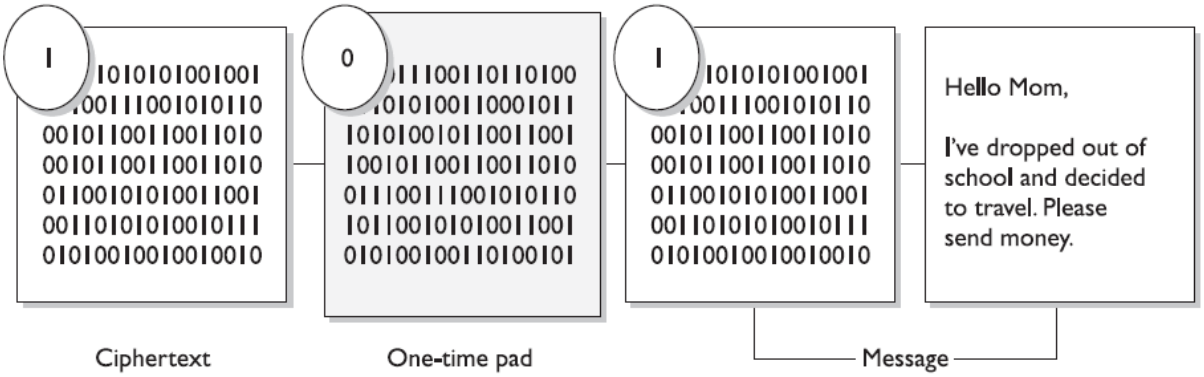
Message stream 1001010111

Keystream 0011101010

Ciphertext stream 1010111101



The first bit of the message is XORed to the first bit of the onetime pad, which results in the ciphertext value 1. The second bit of the message is XORed with the second bit of the pad, which results in the value 0



The receiver must have the same one-time pad to decrypt the message, by reversing the process. The receiver takes the first bit of the encrypted message and XORs it with the first bit of the pad. This results in the plaintext value.

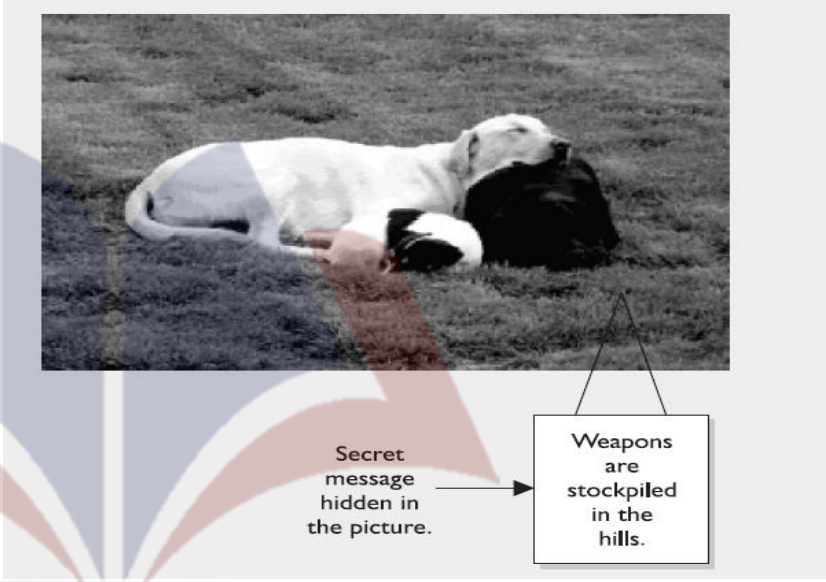
One-Time Pad Requirements:

For a one-time pad encryption scheme to be considered unbreakable, each pad in the scheme must be:

1. Made up of truly random values
2. Used only one time
3. Securely distributed to its destination
4. Secured at sender's and receiver's sites
5. At least as long as the message

A number generator is used to create a stream of random values and must be seeded by an initial value. This piece of software obtains its seeding value from some component within the computer system (time, CPU cycles, and so on). Although a computer system is complex, it is a predictable environment, so if the seeding value is predictable in any way, the resulting values created are not truly random—but pseudorandom.

Steganography:



Steganography is a method of hiding data in another media type so the very existence of the data is concealed as illustrated in the Figure. Only the sender and receiver are supposed to be able to see the message because it is secretly hidden in a graphic, wave file, document, or other type of media. The message is not encrypted, just hidden. A method of embedding the message into some type of medium is to use the ***least significant bit (LSB).*** Many types of files have some bits that can be modified and not affect the file they are in, which is where secret data can be hidden without altering the file in a visible manner. In the LSB approach, graphics with a high resolution or an audio file that has many different types of sounds (high bit rate) are the most successful for hiding information within. There is commonly no noticeable distortion, and the file is usually not increased to a size that can be detected.

Image of a cat extracted from the image of the tree

Image of a tree. Removing all but the two least significant bits of each color component produces an almost completely black image. Making that image 85 times brighter produces the image of the cat.

Digital Watermarking:

The embedded logo or trademark is called a digital watermark. Instead of having a secret message within a graphic that is supposed to be invisible to you, digital watermarks are usually visible. These are put into place to deter people from using material that is not theirs. This type of steganography is referred to as Digital Rights Management (DRM). The goal is to restrict the usage of material that is owned by a company or individual.

Types of Ciphers:

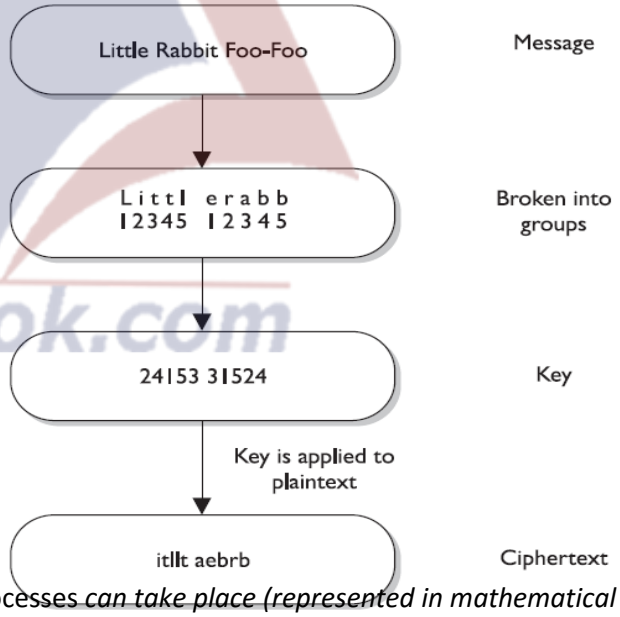
Symmetric encryption ciphers come in two basic types:

1. Substitution
2. Transposition

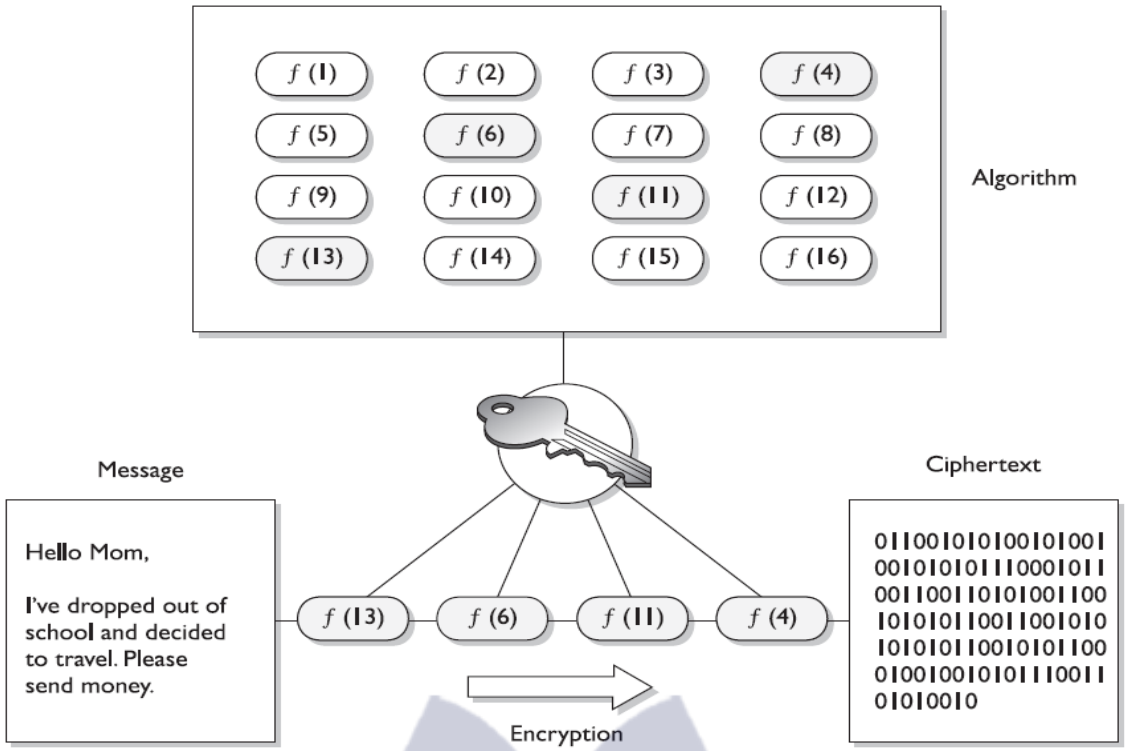
The ***substitution cipher*** replaces bits, characters, or blocks of characters with different bits, characters, or blocks. A substitution cipher uses a key to dictate how the substitution should be carried out. In the ***Caesar cipher,*** each letter is replaced with the letter three places beyond it in the alphabet. The algorithm is the alphabet, and the key is the instruction "shift up three."Substitution is used in today's symmetric algorithms, but it is extremely complex compared to this example

Transposition Ciphers:

In a transposition cipher, the values are scrambled, or put into a different order**.** The key determines the positions the values are moved to, as illustrated in the Figure. This is a simplistic example of a transposition cipher and only shows one way of performing transposition. When implemented with complex mathematical functions, transpositions can become quite sophisticated and difficult to break.



Symmetric algorithms employed today use both long sequences of complicated substitutions and transpositions on messages. The algorithm contains the possible ways that substitution and transposition processes *can take place (represented in mathematical formulas).* The key is used as the instructions for the algorithm, dictating exactly how these processes *will happen and in what order. To understand the relationship between an algorithm and a key, let's look at the Figure. Conceptually, an algorithm is made up of different boxes, each of which has a different set of mathematical formulas that dictates the substitution and transposition steps that will take place on the bits that enter the box. To encrypt our message, the bit values must go through these different boxes.*



If each of our messages goes through each of these different boxes in the same order with the same values, the evildoer will be able to easily reverse-engineer this process and uncover our plaintext message. To foil an evildoer, we use a key, which is a set of values that indicates which box should be used, in what order, and with what values. So if message A is encrypted with key 1, the key will make the message go through boxes 1, 6, 4, and then 5. When we need to encrypt message B, we will use key 2, which will make the message go through boxes 8, 3, 2, and then 9. It is the key that adds the randomness and the secrecy to the encryption process.

Lecture 10

3.6 Methods Of Encryption

For two entities to be able to communicate via encryption, they must use the same algorithm and, many times , the same key. In some encryption technologies, the receiver and the sender use the same key, and in other encryption technologies, they must use different but related keys for encryption and decryption purposes

Symmetric vs. Asymmetric Algorithms

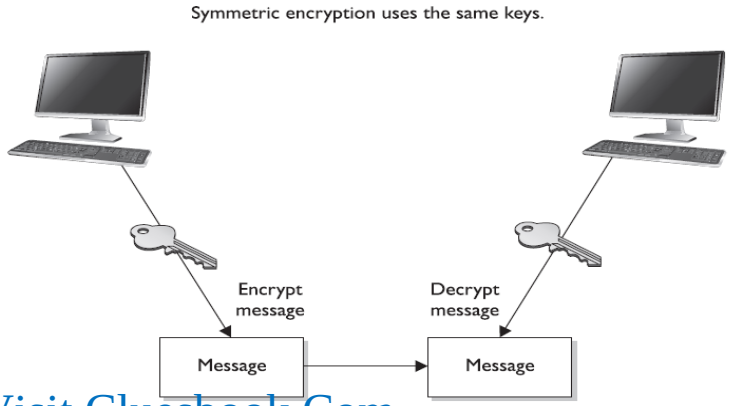
Cryptography algorithms are either ***symmetric algorithms****, which use symmetric keys* (also called secret keys), or ***asymmetric algorithms,*** which use asymmetric keys (also called public and private keys)

3.6.1 Symmetric Cryptography

In a cryptosystem that uses symmetric cryptography, the sender and receiver use two instances of the same key for encryption and decryption, as shown in the Figure

So the key has dual functionality, in that it can carry out both encryption and decryption processes.

Each pair of users who want to exchange data using symmetric key encryption must have two instances of the same key. This means that if Alice and Bob want to communicate, both need to obtain a copy of the same key. If Bob also



wants to communicate using symmetric encryption with Norm and Dave, he needs to have three separate keys , one for each friend. If ten people needed to communicate securely with each other using symmetric keys, then 45 keys would need to be kept track of. If 100 people were going to communicate, then 4,950 keys would be involved

Sharing & Updating Symmetric Keys

If Dan wants to communicate with Norm for the first time, Dan has to figure out how to get the right key to Norm securely. It is not safe to just send it in an e-mail message, because the key is not protected and can be easily intercepted and used by attackers. Thus, Dan must get the key to Norm through an ***out-of-band*** *method. Dan can save the key on a thumb drive and walk over to Norm's desk,* or have a secure courier deliver it to Norm. This is a huge hassle, and each method is insecure.

Strengths & Weaknesses Of Symmetric Encryption

Strengths

Much faster (less computationally intensive) than asymmetric systems

Hard to break if using a large key size

Weaknesses

Requires a secure mechanism to deliver keys properly

Each pair of users needs a unique key, so as the number of individuals increases, so does the number of keys, possibly making key management overwhelming

Provides confidentiality but not authenticity or nonrepudiation

Examples of Symmetric Algorithms

Data Encryption Standard (DES)

Triple-DES (3DES)

Blowfish

IDEA (International Data Encryption Algorithm)

RC4, RC5, and RC6

Advanced Encryption Standard (AES)

3.6.2 Asymmetric Cryptography

In symmetric key cryptography, a single secret key is used between entities, whereas in public key systems, each entity has different keys, or ***asymmetric keys.*** The two different asymmetric keys are mathematically related. If a message is encrypted by one key, the other key is required in order to decrypt the message**.** In a public key system, the pair of keys is made up of one public key and one private key. The *public key can be known to everyone, and the* ***private key*** *must be known and used only by the owner.* Many times, public keys are listed in directories and databases of e-mail addresses so they are available to anyone who wants to use these keys to encrypt or decrypt data when communicating with a particular person**.** The public and private keys of an asymmetric cryptosystem are mathematically related …but if someone gets another person's public key, he should not be able to figure out the corresponding private key**.** This means that if an evildoer gets a copy of Bob's public key, it does not mean he can employ some mathematical magic and find out Bob's private key. But if someone got Bob's private key, then there is big trouble—no one other than the owner should have access to a private key

Authentication

Bob can encrypt data with his private key, and the receiver can then decrypt it with Bob's public key

By decrypting the message with Bob's public key, the receiver can be sure the message really came from Bob.

A message can be decrypted with a public key only if the message was encrypted with the corresponding private key. This provides authentication, because Bob is the only one who is supposed to have his private key

If authentication is the most important security service to the sender, then he would encrypt the data with his private key. This provides assurance to the receiver that the only person who could have encrypted the data is the individual who has possession of that private key. If the sender encrypted the data with the receiver's public key, authentication is not provided because this public key is available to anyone **Confidentiality**

Bob can encrypt data with the receivers public key, and the receiver can then decrypt it with his private key . By decrypting the message with his private key, the receiver can be sure no one else can view this message. This provides confidentiality, because the receiver is the only one who is supposed to have his private key

Public Key Encryption for Confidentiality

If confidentiality is the most important security service to a sender, he would encrypt the file with the receiver's public key

This is called a ***secure message format*** because it can only be decrypted by the person who has the corresponding private key

Asymmetric algorithms are slower than symmetric algorithms because they use much more complex mathematics to carry out their functions, which requires more processing time

Although they are slower, asymmetric algorithms can provide authentication and nonrepudiation, depending on the type of algorithm being used

Strengths & Weaknesses Of Asymmetric Encryption

Strengths	**Weaknesses**
Better key distribution than symmetric systems Better scalability than symmetric systems Can provide authentication and nonrepudiation	Works much more slowly than symmetric systems Mathematically intensive tasks

Examples of Asymmetric Key Algorithms:

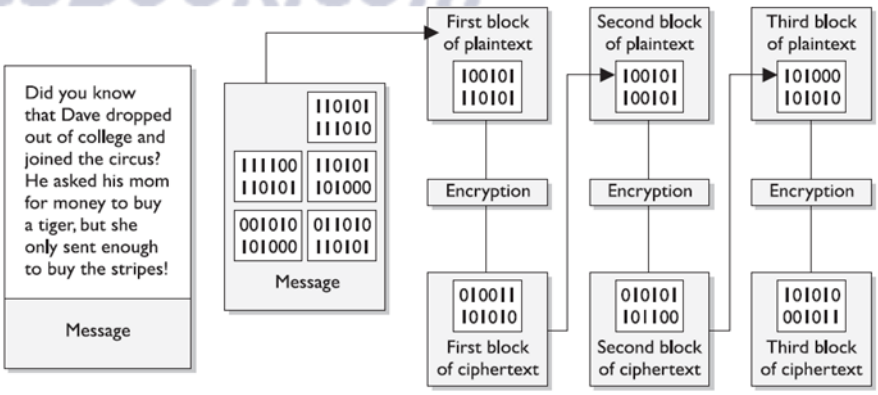
RSA (Rivest-Shamir-Adleman)
Diffie-Hellman
Digital Signature Algorithm (DSA)
Symmetric Vs. Asymmetric
Elliptic curve cryptosystem (ECC)
El Gamal
Merkle-Hellman Knapsack
Core Cryptographic Processes

Lecture 11

3.7 Symmetric Algorithms (Block and Stream

Ciphers) 3.7.1 Block Ciphers

When a ***block cipher*** is used for encryption and decryption purposes, the message is divided into blocks of bits. These blocks are then put through mathematical functions, one block at a time. Suppose you need to encrypt a message you are sending to your friend and you are using a block cipher that uses 64 bits. Your message of 640 bits is chopped up into 10 individual blocks of 64 bits. Each block is put through a succession of mathematical formulas, and what you end up with is 10 blocks of encrypted text. You send this encrypted message to your friend. He has to have the same block cipher and key, and those 10 ciphertext blocks go back through the algorithm in the reverse sequence and end up in your plaintext message.



A strong cipher contains the right level of two main attributes: **confusion and diffusion**.

Confusion is commonly carried out through substitution, while diffusion is carried out by using transposition.

For a cipher to be considered strong, it must contain both of these attributes, to ensure that reverse-engineering is basically impossible. The randomness of the key values and the complexity of the mathematical functions dictate the level of confusion and diffusion involved.

Confusion and Diffusion Example

Suppose I have 500 wooden blocks with individual letters written on them. I line them all up to spell out a paragraph (plaintext). Then I substitute 300 of them with another set of 300 blocks (confusion through substitution). Then I scramble all of these blocks up (diffusion through transposition) and leave them in a pile. For you to figure out my original message, you would have to substitute the correct blocks and then put them back in the right order.

Confusion

Confusion pertains to making the relationship between the key and resulting ciphertext as complex as possible so the key cannot be uncovered from the ciphertext
Each ciphertext value should depend upon several parts of the key, but this mapping between the key values and the ciphertext values should seem completely random to the observer.

Diffusion

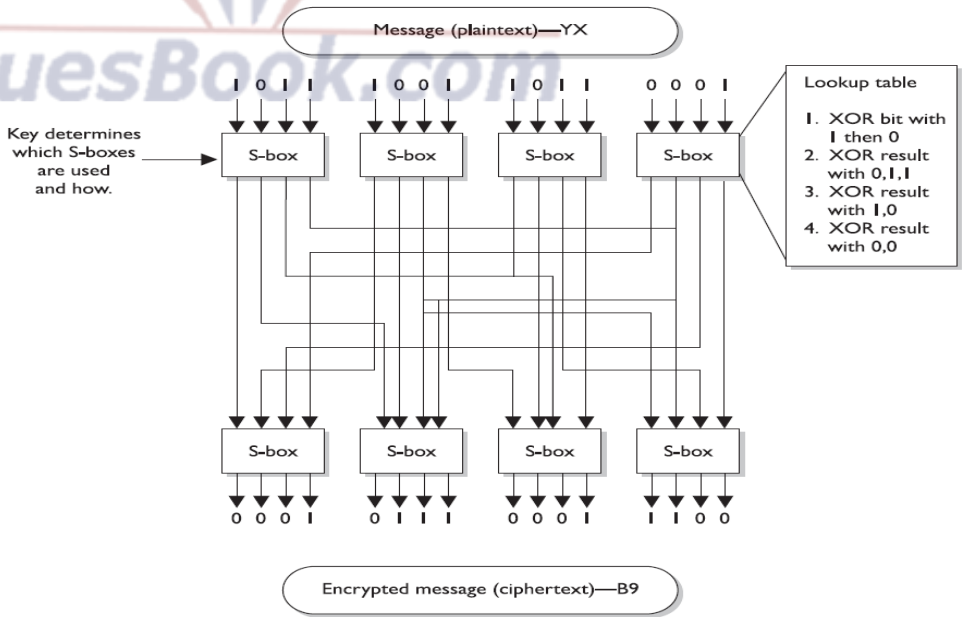
Diffusion (transposition) means that a single plaintext bit has influence over several of the ciphertext bits. Changing a plaintext value should change many ciphertext values, not just one. In fact, in a strong block cipher, if one plaintext bit is changed, it will change every ciphertext bit with the probability of 50 percent. This means that if one plaintext bit changes, then about half of the ciphertext bits will change.

Example Of A Block Cipher

- Block ciphers use diffusion and confusion in their methods. The Figure shows a conceptual example of a simplistic block cipher. It has four block inputs, and each block is made up of four bits

- The block algorithm has two layers of four-bit substitution boxes called *S-boxes.*
- *Each S-box contains a lookup table used by the algorithm as* instructions on how the bits should be encrypted

A message is divided into blocks of bits, and substitution and transposition functions are performed on those blocks

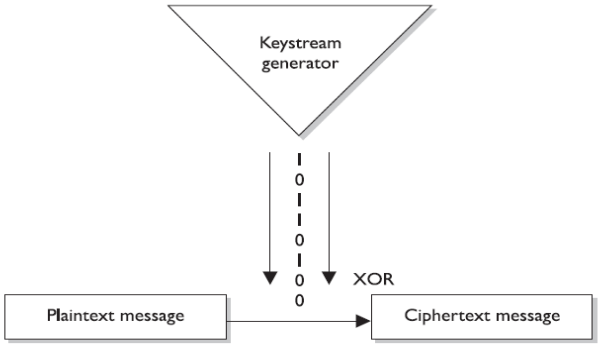


S-Boxes

The Figure shows that the key dictates what S-boxes are to be used when scrambling the original message from readable plaintext to encrypted non-readable cipher text
Each S-box contains the different substitution methods that can be performed on each block. This example is simplistic most block ciphers work with blocks of 32, 64, or 128 bits in size, and many more S-boxes are usually involved

3.7.2 Stream Ciphers

As stated earlier, a block cipher performs mathematical functions on blocks of bits. A stream cipher, on the other hand, does not divide a message into blocks. Instead, a ***stream cipher*** treats the message as a stream of bits and performs mathematical functions on each bit individually. When using a stream cipher, a plaintext bit will be transformed into a different ciphertext bit each time it is encrypted. Stream ciphers use ***keystream generators,*** which produce a stream of bits that is XORed with the plaintext bits to produce ciphertext, as shown in the Figure Stream Cipher Diagram.



With stream ciphers, the bits generated by the keystream generator are XORed with the bits of the plaintext message.

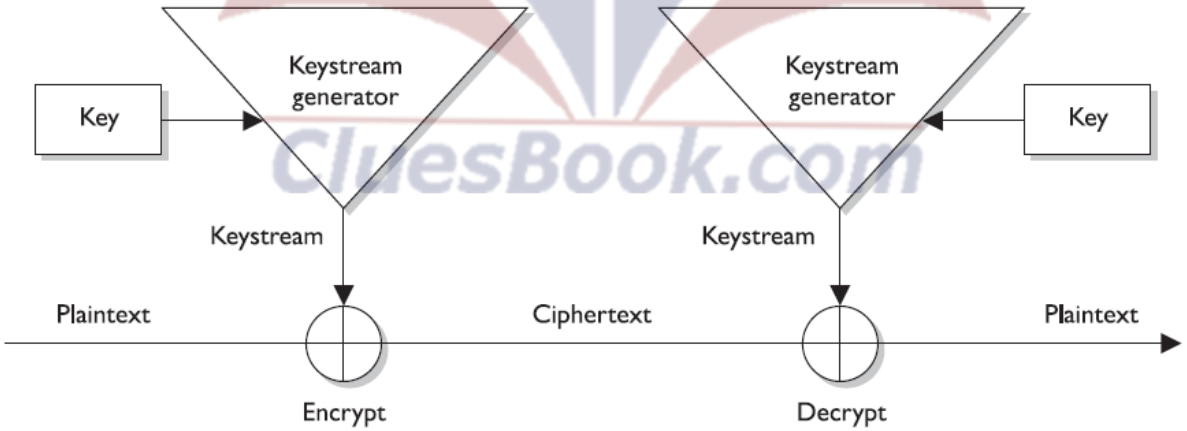
Similarity With One-time Pad

This process is very similar to the one-time pad explained earlier. The individual bits in the one-time pad are used to encrypt the individual bits of the message through the XOR function, and in a stream algorithm the individual bits created by the keystream generator are used to encrypt the bits of the message through XOR also.

Function Of Key In Stream Ciphers

If the cryptosystem were only dependent upon the symmetric stream algorithm, an attacker could get a copy of the plaintext and the resulting ciphertext, XOR them together, and find the keystream to use in decrypting other messages. So the smart people decided to stick a key into the mix.

In block ciphers, it is the key that determines what functions are applied to the plaintext and in what order. The key provides the randomness of the encryption process. As stated earlier, most encryption algorithms are public, so people know how they work. The secret ingredient is the key. In stream ciphers, the key also provides randomness, so that the stream of bits that is XORed to the plaintext is as random as possible.



Both the sending and receiving ends must have the same key to generate the same keystream for proper encryption and decryption purposes.

Initialization Vectors

Initialization vectors (IVs) are random values that are used with algorithms to ensure patterns are not created during the encryption process. They are used with keys and do not need to be encrypted when being sent to the destination. If IVs are not used, then two identical plaintext values that are encrypted with the same key will create the same ciphertext. Providing attackers with these types of patterns can make their job easier in breaking the encryption method and uncovering the key.

"IV" Example

For example, if we have the plaintext value of "See Spot run" two times within our message, we need to make sure that even though there is a pattern in the plaintext message, a pattern in the resulting ciphertext will not be created. So the IV and key are both used by the algorithm to provide more randomness to the encryption process

Characteristics Of Strong Stream Ciphers

Long periods of no repeating patterns within keystream values

Bits generated by the keystream must be random

Statistically unpredictable keystream

The bits generated from the keystream generator cannot be predicted

A keystream not linearly related to the key

If someone figures out the keystream values, that does not mean she now knows the key value

Statistically unbiased keystream (as many 0's as 1's)

There should be no dominance in the number of 0's or 1's in the keystream

Stream & Block Cipher Implementation

Stream ciphers require a lot of randomness and encrypt individual bits at a time

This requires more processing power than block ciphers require, which is why stream ciphers are better suited to be implemented at the hardware level

Because block ciphers do not require as much processing power, they can be easily implemented at the software level

Stream Ciphers Vs. One-time Pads

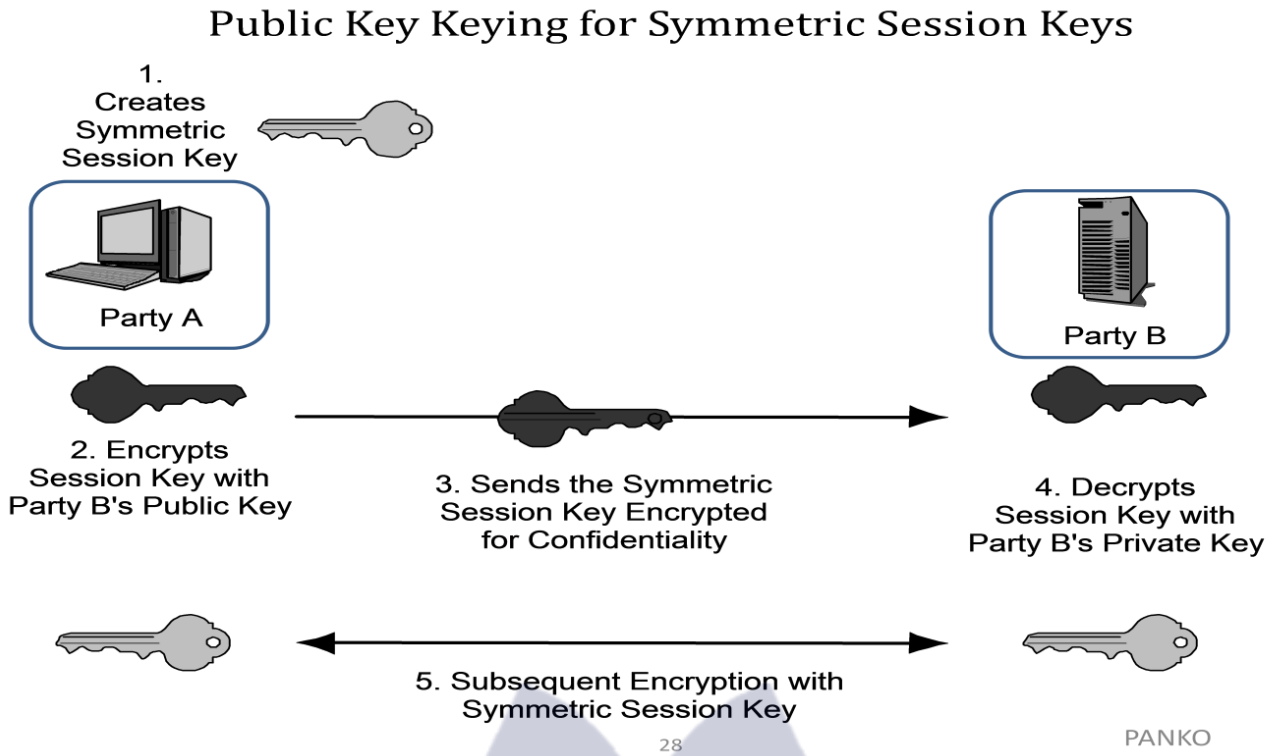
Stream ciphers were developed to provide the same type of protection one-time pads do, which is why they work in such a similar manner. In reality, stream ciphers cannot provide the level of protection one-time pads do, but because stream

ciphers are implemented through software and automated means, they are much

more practical.

3.7.3 Hybrid Encryption Systems

Up to this point, we have figured out that symmetric algorithms are fast but have some drawbacks (lack of scalability, difficult key management, and they provide only confidentiality). Asymmetric algorithms do not have these drawbacks but are very slow. We just can't seem to win. So we turn to a hybrid system that uses symmetric and asymmetric encryption methods together.



Public key cryptography uses two keys (public and private) generated by an asymmetric algorithm for protecting encryption keys and key distribution, and a secret key is generated by a symmetric algorithm and used for bulk encryption. Then there is a hybrid use of the two different algorithms: asymmetric and symmetric. Each algorithm has its pros and cons, so using them together can be the best of both worlds.

Lecture 12

3.8 Types Of Symmetric Systems

- Several types of symmetric algorithms are used today. They have different methods of providing encryption and decryption functionality
- The one thing they all have in common is that they are symmetric algorithms, meaning the sender and receiver are using two instances of the same key
- In this section, we will be walking through symmetric algorithms and their characteristics

Some examples:

• Data Encryption Standard (DES) • 3DES (Triple DES) • Blowfish	• Twofish • IDEA (International Data Encryption Algorithm) • RC4, RC5, RC6	• AES (Advanced Encryption Standard) • SAFER (Secure and Fast Encryption Routine) • Serpent

3.8.1 DES – Data Encryption Standard:

- NSA announced in 1986 that, as of January 1988, the agency would no longer endorse DES and that DES-based products would no longer fall under compliance with Federal Standard 1027.

- The NSA felt that because DES had been so popular for so long, it would surely be targeted for penetration and become useless as an official standard.

In 1998, the Electronic Frontier Foundation built a computer system for $250,000 that broke DES in three days by using a brute force attack against the keyspace. It contained 1,536 microprocessors running at 40MHz, which performed 60 million test decryptions per second per chip. Although most people do not have these types of systems to conduct such attacks, as Moore's Law holds true and microprocessors increase in processing power, this type of attack will become more feasible for the average attacker. This brought about 3 DES, which provides stronger protection, as discussed later in the chapter. DES was later replaced by the Rijndael algorithm as the *Advanced Encryption Standard (AES)* by NIST. This means that Rijndael is the new approved method of encrypting sensitive but unclassified information for the U.S. government; it has been accepted by, and is widely used in, the public arena today.

How Does DES Work ?

- DES is a symmetric block encryption algorithm. When 64-bit blocks of plaintext go in, 64-bit blocks of ciphertext come out.
    - It is also a symmetric algorithm, meaning the same key is used for encryption and decryption.
- It uses a 64-bit key: 56 bits make up the true key, and 8 bits are used for parity.
- When the DES algorithm is applied to data, it divides the message into blocks and operates on them one at a time.
- The blocks are put through 16 rounds of transposition and substitution functions.
- The order and type of transposition and substitution functions depend on the value of the key used with the algorithm.
- The result is 64-bit blocks of ciphertext.

What Does It Mean When an Algorithm Is Broken?

In most instances, an algorithm is broken if someone is able to uncover a key that was used during an encryption process. So let's say Ali encrypted a message and sent it to Bilal. Zaheer captures this encrypted message and carries out a brute force attack on it, which means he tries to decrypt the message with different keys until he uncovers the right one. Once he identifies this key, the algorithm is considered broken. So does that mean the algorithm is worthless? If an algorithm is broken through a brute force attack, this just means the attacker identified the one key that was used for one instance of encryption. But in proper implementations, we should be encrypting data with session keys, which are good only for that one session. So even if the attacker uncovers one session key, it may be useless to the attacker, in which case he now has to work to identify a new session key. So breaking an algorithm can take place through brute force attacks or by identifying weaknesses in the algorithm itself. Brute force attacks have increased in potency because of the increased processing capacity of computers today. An algorithm that uses a 40-bit key has around 1 trillion possible key values. If a 56-bit key is used, then there are approximately 72 quadrillion different key values. Relative to today's computing power, these key sizes do not provide much protection at all.

DES Modes:

Block ciphers have several modes of operation. Each mode specifies how a block cipher will operate. One mode may work better in one type of environment for specific functionality, whereas another mode may

work better in another environment with totally different requirements. It is important that vendors who employ DES (or any block cipher) understand the different modes and which one to use for which purpose.

We will look at five DES Modes:

- Electronic Code Book (ECB)
- Cipher Block Chaining (CBC)
- Cipher Feedback (CFB)
- Output Feedback (OFB)
- Counter Mode (CTR)

Electronic Code Book Mode:

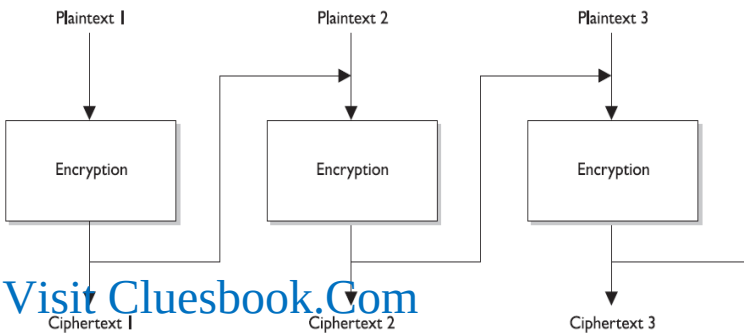
ECB mode operates like a code book. A 64-bit data block is entered into the algorithm with a key, and a block of ciphertext is produced. For a given block of plaintext and a given key, the same block of ciphertext is always produced. Not all messages end up in neat and tidy 64-bit blocks, so ECB incorporates padding to address this problem. ECB is the easiest and fastest mode to use, but as we will see, it has its dangers. A key is basically instructions for the use of a code book that dictates how a block of text will be encrypted and decrypted. The code book provides the recipe of substitutions and permutations that will be performed on the block of plaintext. The security issue that comes up with using ECB mode is that each block will be encrypted with the exact same key, and thus the exact same code book. So, two bad things can happen here:

- An attacker could uncover the key and thus have the key to decrypt all the blocks of data, or
- An attacker could gather the ciphertext and plaintext of each block and build the code book that was used, without needing the key.

The crux of the problem is that there is not enough randomness to the process of encrypting the independent blocks, so if this mode is used to encrypt a large amount of data, it could be cracked more easily than the other modes that block ciphers can work in. So the next question to ask is, why even use this mode? This mode is the fastest and easiest, so we use it to encrypt small amounts of data, such as PINs, challenge-response values in authentication processes, and encrypting keys. Because this mode works with blocks of data independently, data within a file does not have to be encrypted in a certain order. This is very helpful when using encryption in databases. A database has different pieces of data accessed in a random fashion. If it is encrypted in ECB mode, then any record or table can be added, encrypted, deleted, or decrypted independently of any other table or record . Other DES modes are dependent upon the text encrypted before them. This dependency makes it harder to encrypt and decrypt smaller amounts of text, because the previous encrypted text would need to be decrypted first. Because ECB mode does not use chaining, you should not use it to encrypt large amounts of data, because patterns would eventually show themselves.

Cipher Block Chaining Mode:

In ECB mode, a block of plaintext and a key will always give the same ciphertext. This means that if the word “balloon” were encrypted and the resulting ciphertext were “hwicssn,” each time it was encrypted using



the same key, the same ciphertext would always be given. This can show evidence of a pattern, enabling an evildoer, with some effort, to discover the pattern and get a step closer to compromising the encryption process . ***Cipher Block Chaining (CBC) does not reveal a pattern,*** because each block of text, the key, and the value based on the previous block are processed in the algorithm and applied to the next block of text, as shown in the Figure **.** This results in more random ciphertext. Ciphertext is extracted and used from the previous block of text. This provides dependence between the blocks, in a sense chaining them together.

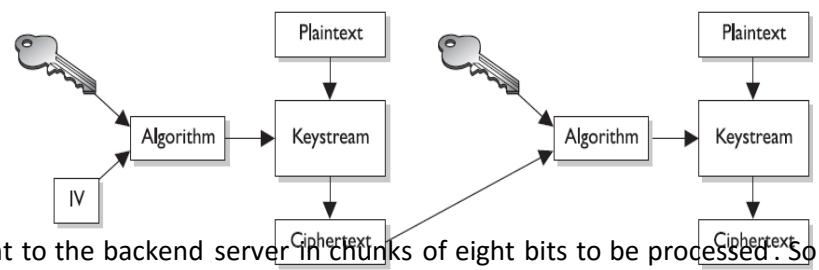
In CBC mode, the ciphertext from the previous block of data is used in encrypting the next block of data. This is where the name Cipher Block Chaining comes from, and it is this chaining effect that hides any patterns

The results of one block are XORed with the next block before it is encrypted, meaning each block is used to modify the following block. This chaining effect means that a particular ciphertext block is dependent upon all blocks before it, not just the previous block. As an analogy, let's say you have five buckets of marbles. Each bucket contains a specific color of marbles: red, blue, yellow, black, and green. The first bucket of red marbles (block of bits) you shake and tumble around (encrypt) to get them all mixed up. Then you take the second bucket of marbles, which are blue, and pour in the red marbles and go through the same exercise of shaking and tumbling them. You pour this bucket of marbles into your next bucket and shake them all up. When we encrypt our very first block using CBC, we do not have a previous block of ciphertext to "dump in" and use to add the necessary randomness to the encryption process. If we do not add a piece of randomness when encrypting this first block, then the bad guys could identify patterns, work backward, and uncover the key. So, we use an initialization vector. The 64-bit IV is XORed with the first block of plaintext, and then it goes through its encryption process. The result of that (ciphertext) is XORed with the second block of plaintext, and then the second block is encrypted. This continues for the whole message. It is the chaining that adds the necessary randomness that allows us to use CBC mode to encrypt large files. Neither the individual blocks nor the whole message will show patterns that will allow an attacker to reverse-engineer and uncover the key. If we choose a different IV each time we encrypt a message, even if it is the same message, the ciphertext will always be unique. This means that if you send the same message out to 50 people and encrypt each one using a different IV, the ciphertext for each message will be different.

Lecture 13

Cipher Feedback Mode (CFB):

If you are going to send an encrypted e-mail to your boss, your e-mail client will use a symmetric block cipher working in CBC mode. The e-mail client would not use ECB mode, because most messages are long enough to show patterns that can be used to reverse-engineer the process and uncover the encryption key. The CBC mode is great to use when you need to send large chunks of data at a time. But what if you are not sending large chunks of data at one time, but instead are sending a steady stream of data to a destination? If you are working on a terminal that communicates with a back-end terminal server, what is really going on is that each keystroke and mouse movement you make is sent to the backend server in chunks of eight bits to be processed. So even though it seems as though the computer you are working on is carrying out your commands and doing the processing you are requesting, it is not—this is happening on the server. Thus, if you need to encrypt the



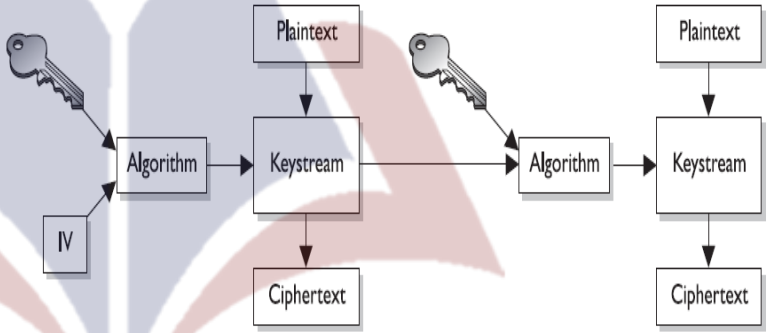
data that go from your terminal to the terminal server, you could not use CBC mode because it only encrypts blocks of data 64 bits you need to encrypt. So we use CFB mode…..

The Figure illustrates how CFB mode works, which is really a combination of a block cipher and a stream cipher.

For the first block of eight bits that needs to be encrypted, we do the same thing we did in CBC mode, which is to use an IV. Recal the IV are used by the algorithm to create a keystream, which is just a random set of bits. This set of bits is XORed to the block of block of ciphertext. So the first block (eight bits) is XORed to the set of bits created through the keystream generator. Two thing block of ciphertext. One copy goes over the wire to the destination (in our scenario, to the terminal server), and another copy is us plaintext. Adding this copy of ciphertext to the encryption process of the next block adds more randomness to the encryption where eight-bit blocks needed to be encrypted, but in reality CFB mode can be used to encrypt any size blocks, even blocks of just maps eight bits to one character, using CFB to encrypt eight- bit blocks is very common.

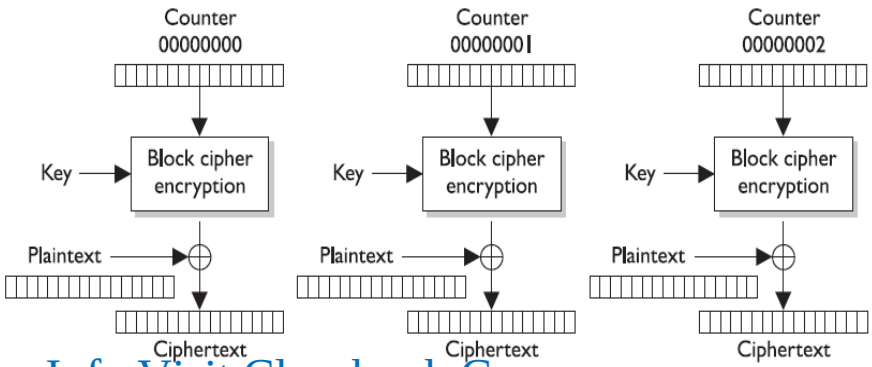
Output Feedback Mode:

Now look at the Figure . It looks terribly similar to the previous Figure (CFB), but notice that the values used to encrypt the next block of plaintext are coming directly from the keystream, not from the resulting ciphertext. This is the difference between the two modes.



If you need to encrypt something that would be very sensitive to these types of errors, such as digitized video or digitized voice signals, you should not use CFB mode. You should use OFB mode instead, which reduces the chance that these types of bit corruptions can take place. So OFB is a mode that a block cipher can work in when it needs to emulate a stream, because it encrypts small amounts of data at a time, but it has a smaller chance of creating and extending errors throughout the full encryption process. To ensure OFB and CFB are providing the most protection possible, the size of the ciphertext (in CFB) or keystream values (in OFB) needs to be the same size as the block of plaintext being encrypted. This means that if you are using CFB and are encrypting eight bits at a time, the ciphertext you bring forward from the previous encryption block needs to be eight bits. Otherwise, you are repeating values over and over, which introduces patterns. (This is the same reason why a one-time pad should be used only one time and should be as long as the message itself.)

Counter Mode (CTR): Counter Mode (CTR) is very similar to OFB mode, but instead of using a randomly unique IV value to generate the keystream values, this mode uses an IV counter that increments for each



plaintext block that needs to be encrypted. The unique counter ensures that each block is XORed with a unique keystream value. The other difference is that there is no chaining involved, which means no ciphertext is brought forward to encrypt the next block. Since there is no chaining, the encryption of the individual blocks can happen in parallel, which increases the performance. The main reason CTR would be used instead of the other modes is performance. **This mode has been around for quite some time and is used in encrypting ATM cells for virtual circuits, in IPSec, and is now integrated in the new wireless security standard, 802.11i.**

Why Use CTR ?

A developer would choose to use this mode in these situations because individual ATM cells or packets going through an IPSec tunnel or over radio frequencies may not arrive at the destination in order. Since chaining is not involved, the destination can decrypt and begin processing the packets without having to wait for the full message to arrive and *then decrypt all the data.*

3.8.2 Triple DES (3DES):

We went from DES to Triple-DES (3DES), so it might seem we skipped Double-DES. We did. Double-DES has a key length of 112 bits, but there is a specific attack against Double-DES that reduces its work factor to about the same as DES. Thus, it is no more secure than DES. So let's move on to 3DES. Many successful attacks against DES and the realization that the useful lifetime of DES was about up brought much support for 3DES. NIST knew that a new standard had to be created, which ended up being AES, but a quick fix was needed in the meantime to provide more protection for sensitive data. The result: 3DES (also known as TDEA—Triple Data Encryption Algorithm).

3DES Performance

3DES uses 48 rounds in its computation, which makes it highly resistant to differential cryptanalysis. However, because of the extra work 3DES performs, there is a heavy performance hit. It can take up to three times longer than DES to perform encryption and decryption.

3DES Modes:

Although NIST has selected the Rijndael algorithm to replace DES as the AES, NIST and others expect 3DES to be around and used for quite some time. 3DES can work in different modes, and the mode chosen dictates the number of keys used and what functions are carried out

- **DES-EEE3** Uses three different keys for encryption, and the data are encrypted, encrypted, encrypted
- **DES-EDE3** Uses three different keys for encryption, and the data are encrypted, decrypted, and encrypted
- **DES-EEE2** The same as DES-EEE3 but uses only two keys, and the first and third encryption processes use the same key
- **DES-EDE2** The same as DES-EDE3 but uses only two keys, and the first and third encryption processes use the same key

EDE ?

EDE may seem a little odd at first. How much protection could be provided by encrypting something, decrypting it, and encrypting it again? The decrypting portion here is decrypted with a different key. When data are encrypted with one symmetric key and decrypted with a different symmetric key, it is jumbled even more. So the data are not actually decrypted in the middle function, they are just run through a decryption process with a different key. Pretty tricky.

Lecture 14

3.8.3 Advanced Encryption Standard (AES)

After DES was used as an encryption standard for over 20 years and it was cracked in a relatively short time once the necessary technology was available, NIST decided a new standard, the Advanced Encryption Standard (AES), needed to be put into place. In January 1997, NIST announced its request for AES candidates and outlined the requirements in FIPS PUB 197. AES was to be a symmetric block cipher supporting key sizes of 128, 192, and 256 bits. The following five algorithms were the finalists:

AES Finalists

MARS: Developed by the IBM team that created Lucifer

• RC6: Developed by RSA Laboratories

• Serpent: Developed by Ross Anderson, Eli Biham, and Lars Knudsen

• Twofish: Developed by Counterpane Systems

• Rijndael: Developed by Joan Daemen and Vincent Rijmen

Rijndael:

Out of these contestants, Rijndael was chosen. The block sizes that Rijndael supports are 128, 192, and 256 bits. The number of rounds depends upon the size of the block and the key length:

- If both the key and block size are 128 bits, there are 10 rounds
- If both the key and block size are 192 bits, there are 12 rounds
- If both the key and block size are 256 bits, there are 14 rounds

Rijndael Advantages:

Rijndael works well when implemented in software and hardware in a wide range of products and environments. It has low memory requirements and has been constructed to easily defend against timing attacks. Rijndael was NIST's choice to replace DES. It is now the algorithm required to protect sensitive but unclassified government information.

3.8.4 International Data Encryption Algorithm (IDEA):

- *A block cipher and operates on 64-* bit blocks of data.
- The 64-bit data block is divided into 16 smaller blocks, and each has eight rounds of mathematical functions performed on it.
- The key is 128 bits long, and IDEA is faster than DES when implemented in software.

The IDEA algorithm offers different modes similar to the modes described in the DES section, but it is considered to be harder to break than DES because it has a longer key size. IDEA is used in the PGP and other

encryption software implementations. It was thought to replace DES, but it is patented, meaning that licensing fees would have to be paid to use it.

3.8.5 Blowfish:

Blowfish is a block cipher that works on 64-bit blocks of data. The key length can be anywhere from 32 bits up to 448 bits, and the data blocks go through 16 rounds of cryptographic functions. It was intended as a replacement to the aging DES. While many of the other algorithms have been proprietary and thus encumbered by patents or kept as government secrets, this wasn't the case with Blowfish. Bruce Schneier, the creator of Blowfish, has stated, "Blowfish is un-patented, and will remain so in all countries. The algorithm is hereby placed in the public domain, and can be freely used by anyone."

3.8.6 RC4:

RC4 is one of the most commonly implemented stream ciphers. It has a variable key size, is used in the SSL protocol, and was (improperly) implemented in the 802.11 WEP protocol standard. RC4 was developed in 1987 by Ron Rivest and was considered a trade secret of RSA Data Security, Inc. until someone posted the source code on a mailing list. Since the source code was released nefariously, the stolen algorithm is sometimes implemented and referred to as ArcFour or ARC4 because the title RC4 is trademarked. The algorithm is very simple, fast, and efficient, which is why it became so popular.

3.8.7 RC5:

RC5 is a block cipher that has a variety of parameters it can use for block size, key size, and the number of rounds used. It was created by Ron Rivest and analyzed by RSA Data Security, Inc. The block sizes used in this algorithm are 32, 64, or 128 bits, and the key size goes up to 2,048 bits. The number of rounds used for encryption and decryption is also variable. The number of rounds can go up to 255**.**

3.8.8 RC6:

RC6 is a block cipher that was built upon RC5, so it has all the same attributes as RC5. The algorithm was developed mainly to be submitted as AES, but Rijndael was chosen instead. There were some modifications of the RC5 algorithm to increase the overall speed, the result of which is RC6.

Revision Symmetric Key Cryptography

Examples of Symmetric Algorithms:

- Data Encryption Standard (DES)
- Triple-DES (3DES)
- Blowfish
- IDEA (International Data Encryption Algorithm)
- RC4, RC5, and RC6
- Advanced Encryption Standard (AES)

Strengths & Weaknesses Of Symmetric Encryption:

Strengths

- Much faster (less computationally intensive) than asymmetric systems
- Hard to break if using a large key size

Weaknesses

- Requires a secure mechanism to deliver keys properly
- Each pair of users needs a unique key, so as the number of individuals increases, so does the number of keys, possibly making key management overwhelming
- Provides confidentiality but not authenticity or nonrepudiation

Limitations Of Symmetric Key Cryptography

Security services

Scalability

Secure key distribution

Review Of Asymmetric Cryptography

Strengths & Weaknesses Of Asymmetric Encryption:

Strengths

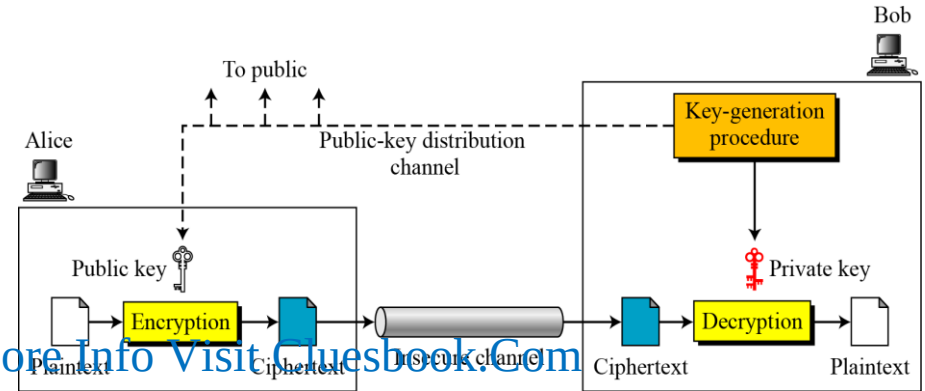
- Better key distribution than symmetric systems
- Better scalability than symmetric systems
- Can provide authentication and nonrepudiation

Weaknesses

- Works much more slowly than symmetric systems
- Mathematically intensive tasks

Examples of Asymmetric Key Algorithms:

- Diffie-Hellman
- RSA (Rivest-Shamir-Adleman)
- El Gamal
- Elliptic curve cryptosystem (ECC)
- Digital Signature Algorithm (DSA)
- Merkle-Hellman Knapsack

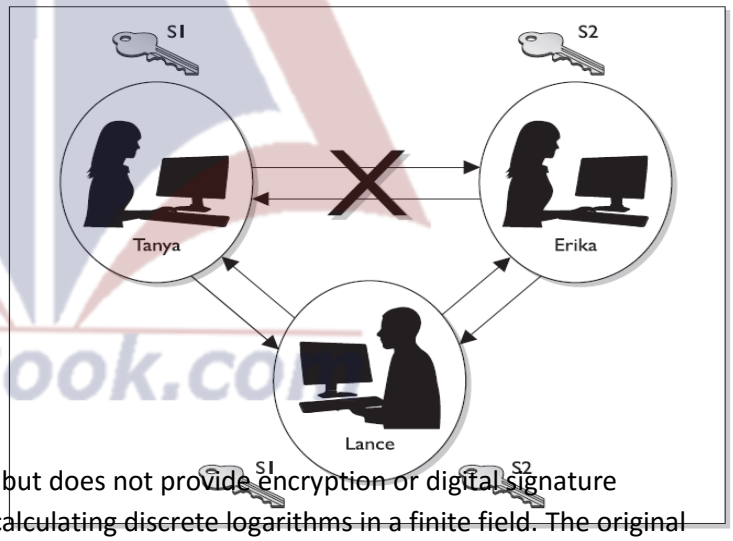


In a public key system, the pair of keys is made up of one public key and one private key. The *public key can be known to everyone, and the private key must be known and used only by the owner*. Many times, public keys are listed in directories and databases of e-mail addresses so they are available to anyone who wants to use these keys to encrypt or decrypt data when communicating with a particular person

Lecture 15

3.10.1 The Diffie-Hellman Algorithm:

The first group to address the shortfalls of symmetric key cryptography decided to attack the issue of secure distribution of the symmetric key. Whitfield Diffie and Martin Hellman worked on this problem and ended up developing the first asymmetric key agreement algorithm, called, naturally, Diffie-Hellman. Let's say that Ahmad and Bilal would like to communicate over an encrypted channel by using Diffie-Hellman. They would both generate a private and public key pair and exchange public keys. Ahmad's software would take his private key (which is just a numeric value) and Bilal's public key (another numeric value) and put them through the Diffie-Hellman algorithm. Bilal's software would take his private key and Ahmad's public key and insert them into the Diffie-Hellman algorithm on his computer. Through this process, Ahmad and Bilal derive the same shared value, which is used to create instances of symmetric keys. So, Ahmad and Bilal exchanged information that did not need to be protected (their public keys) over an untrusted network, and in turn generated the exact same symmetric key on
each system. They both can now use these symmetric keys to encrypt, transmit, and decrypt information as they communicate with each other. The preceding example describes key *agreement, which is different* from key *exchange, the functionality used by the other asymmetric algorithms* that will be discussed in this chapter. With key exchange functionality, the sender encrypts the symmetric key with the receiver's public key before transmission. The Diffie-Hellman algorithm enables two systems to exchange a symmetric key securely without requiring a previous relationship or prior arrangements. The algorithm allows for key distribution, but does not provide encryption or digital signature functionality. The algorithm is based on the difficulty of calculating discrete logarithms in a finite field. The original Diffie-Hellman algorithm is vulnerable to a man-in-the-middle attack, because no authentication occurs before public keys are exchanged. In our example, when Ahmad sends his public key to Bilal, how does Bilal really know it is Ahmad's public key? What if Rashid spoofed his identity, told Bilal he was Ahmad, and sent over his key? Bilal would accept this key, thinking it came from Ahmad. Let's walk through the steps of how this type of attack would take place, as illustrated in the Figure.



1. Ahmad sends his public key to Bilal, but Rashid grabs the key during transmission so it never makes it to Bilal.

2. Rashid spoofs Ahmad's identity and sends over his public key to Bilal. Bilal now thinks he has Ahmad's public key.

3. Bilal sends his public key to Ahmad, but Rashid grabs the key during transmission so it never makes it to Ahmad.

4. Rashid spoofs Bilal's identity and sends over his public key to Ahmad. Ahmad now thinks he has Bilal's public key.

5. Ahmad combines his private key and Rashid's public key and creates symmetric key S1.

6. Rashid combines his private key and Ahmad's public key and creates symmetric key S1.

7. Bilal combines his private key and Rashid's public key and creates symmetric key S2.

8. Rashid combines his private key and Bilal's public key and creates symmetric key S2.

9. Now Ahmad and Rashid share a symmetric key (S1) and Bilal and Rashid share a different symmetric key (S2). Ahmad and Bilal think they are sharing a key between themselves and do not realize Rashid is involved.

10. Ahmad writes a message to Bilal, uses his symmetric key (S1) to encrypt the message, and sends it.

11. Rashid grabs the message and decrypts it with symmetric key S1, reads or modifies the message and re-encrypts it with symmetric key S2, and then sends it to Bilal.

12. Bilal takes symmetric key S2 and uses it to decrypt and read the message.

The countermeasure to this type of attack is to have authentication take place before accepting someone's public key, which usually happens through the use of digital signatures and digital certificates. Although the Diffie-Hellman algorithm is vulnerable to a man-in-the-middle attack, it does not mean this type of compromise can take place anywhere this algorithm is deployed. Most implementations include another piece of software or a protocol that compensates for this vulnerability. But some do not. As a security professional, you should understand these issues.

3.10.2 RSA:

RSA, named after its inventors Ron Rivest, Adi Shamir, and Leonard Adleman, is a public key algorithm that is the most popular when it comes to asymmetric algorithms. RSA is a worldwide de facto standard and can be used for digital signatures, key exchange, and encryption. It was developed in 1978 at MIT and provides authentication as well as key encryption. The security of this algorithm comes from the difficulty of factoring large numbers. The public and private keys are functions of a pair of large prime numbers, and the necessary activity required to decrypt a message from ciphertext to plaintext using a private key is comparable to factoring a product into two prime numbers. A prime number is a positive whole number with no proper divisors, meaning the only numbers that can divide a prime number are 1 and the number itself. One advantage of using RSA is that it can be used for encryption and digital signatures. Using its one-way function, RSA provides encryption and signature verification, and the inverse direction performs decryption and signature generation. RSA has been implemented in applications; in operating systems by Microsoft, Apple, Sun, and Novell; and at the hardware level in network interface cards, secure telephones, and smart cards. It can be used as a key exchange protocol, meaning it is used to encrypt the symmetric key to get it securely to its destination. RSA has been most commonly used with the symmetric algorithm DES, which is quickly being replaced with AES. So, when RSA is used as a key exchange protocol, a cryptosystem generates a symmetric key using either the DES or AES algorithm. Then the system encrypts the symmetric key with the receiver's public key and sends it to the receiver. The symmetric key is protected because only the individual with the corresponding private key can decrypt and extract the symmetric key.

Lecture 16

3.10.3 What Is the Difference Between Public Key Cryptography andPublic Key Infrastructure?

Public key cryptography is the use of an asymmetric algorithm. Thus, the terms asymmetric algorithm and public key cryptography are interchangeable and mean the same thing. Examples of asymmetric algorithms are RSA, elliptic curve cryptosystem (ECC), Diffie-Hellman, El Gamal, LUC, and Knapsack. These algorithms are used to create public/private key pairs, perform key exchange or agreement, and generate and verify digital signatures. Note that Diffie-Hellman can only perform key agreement and cannot generate or verify digital signatures.

Public key infrastructure (PKI) is different. It is not an algorithm, a protocol, or an application—it is an infrastructure based on public key cryptography.

3.10.4 One Way Functions:

A one-way function is a mathematical function that is easier to compute in one direction than in the opposite direction. An analogy of this is when you drop a glass on the floor. Although dropping a glass on the floor is easy, putting all the pieces back together again to reconstruct the original glass is next to impossible. This concept is similar to how a one-way function is used in cryptography, which is what the RSA algorithm, and all other asymmetric algorithms, is based upon. The easy direction of computation in the one-way function that is used in the RSA algorithm is the process of multiplying two large prime numbers. Multiplying the two numbers to get the resulting product is much easier than factoring the product and recovering the two initial large prime numbers used to calculate the obtained product, which is the difficult direction. RSA is based on the difficulty of factoring large numbers that are the product of two large prime numbers. Attacks on these types of cryptosystems do not necessarily try every possible key value, but rather try to factor the large number, which will give the attacker the private key.

When a user encrypts a message with a public key, this message is encoded with a one-way function (breaking a glass). This function supplies a *trapdoor* (knowledge of how to put the glass back together), but the only way the trapdoor can be taken advantage of is if it is known about and the correct code is applied. The private key provides this service. The private key knows about the trapdoor, knows how to derive the original prime numbers, and has the necessary programming code to take advantage of this secret trapdoor to unlock the encoded message (reassembling the broken glass). Knowing about the trapdoor and having the correct functionality to take advantage of it are what make the private key private.

When a one-way function is carried out in the easy direction, encryption and digital signature verification functionality are available. When the one-way function is carried out in the hard direction, decryption and signature generation functionality are available. This means only the public key can carry out encryption and signature verification and only the private key can carry out decryption and signature generation.

As explained earlier in this chapter, *work factor is the amount of time and resources* it would take for someone to break an encryption method. In asymmetric algorithms, the work factor relates to the difference in time and effort that carrying out a one-way function in the easy direction takes compared to carrying out a one-way function in the hard direction. In most cases, the larger the key size, the longer it would take for the bad guy to carry out the one-way function in the hard direction (decrypt a message).

Asymmetric Algorithms Key Concept:

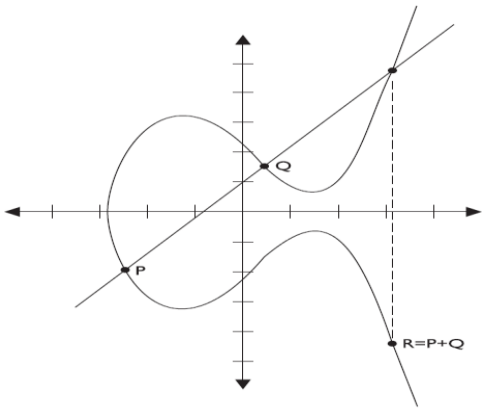
The crux of this section is that all asymmetric algorithms provide security by using mathematical equations that are easy to perform in one direction and next to impossible to perform in the other direction. The "hard" direction is based on a "hard" mathematical problem. RSA's hard mathematical problem requires factoring large numbers into their original prime numbers. Diffie-Hellman and El Gamal are based on the difficulty of calculating logarithms in a finite field.

3.10.5 El Gamal:

El Gamal is a public key algorithm that can be used for digital signatures, encryption, and key exchange. It is based not on the difficulty of factoring large numbers but on calculating discrete logarithms in a finite field. El Gamal is actually an extension of the Diffie-Hellman algorithm. Although El Gamal provides the same type of functionality as some of the other asymmetric algorithms, its main drawback is performance. When compared to other algorithms, this algorithm is usually the slowest.

3.10.6 Elliptic Curve Cryptosystems:

Elliptic curves are rich mathematical structures that have shown usefulness in many different types of applications. ECC is more efficient than RSA and any other asymmetric algorithm. An elliptic curve cryptosystem (ECC) provides much of the same functionality RSA provides: digital signatures, secure key distribution, and encryption. One differing factor is ECC's efficiency.



In this field of mathematics, points on the curves compose a structure called a group.

Elliptic Curve Cryptosystems:

The Figure is an example of an elliptic curve. In this field of mathematics, points on the curves compose a structure called a group. These points are the values used in mathematical formulas for ECC's encryption and decryption processes. The algorithm computes discrete logarithms of elliptic curves, which is different from calculating discrete logarithms in a finite field (which is what Diffie-Hellman and El Gamal use). Some devices have limited processing capacity, storage, power supply, and bandwidth, such as wireless devices and cellular telephones. With these types of devices, efficiency of resource use is very important. ECC provides encryption functionality, requiring a smaller percentage of the resources needed by RSA and other algorithms, so it is used in these types of devices. In most cases, the longer the key, the more protection that is provided, but ECC can provide the same level of protection with a key size that is shorter than what RSA requires. Because longer keys require more resources to perform mathematical tasks, the smaller keys used in ECC require fewer resources of the device.

3.10.7 Knapsack:

Over the years, different versions of knapsack algorithms have arisen. The first to be developed, Merkle-Hellman, could be us ed only for encryption, but it was later improved upon to provide digital signature capabilities. These types of algorithms are based on the "knapsack problem," a mathematical dilemma that poses the following question:

If you have several different items, each having its own weight, is it possible to add these items to a knapsack so the knapsack has a specific weight? This algorithm was discovered to be insecure and is not currently used in cryptosystems.

Zero Knowledge Proof:

If I encrypt something with my private key, you can verify my private key was used by decrypting the data with my public key. By encrypting something with my private key, I am proving to you I have my private key—but I do not give or show you my private key.

Lecture 17

3.11 Message Integrity:

Parity bits and cyclic redundancy check (CRC) functions have been used in protocols to detect modifications in streams of bits as they are passed from one computer to another, but they can usually detect only unintentional modifications. Unintentional modifications can happen if a spike occurs in the power supply, if there is interference or attenuation on a wire, or if some other type of physical condition happens that causes the corruption of bits as they travel from one destination to another. Parity bits cannot identify whether a message was captured by an intruder, altered, and then sent on to the intended destination. The intruder can just recalculate a new parity value that includes his changes, and the receiver would never know the difference. For this type of protection, hash algorithms are required to successfully detect intentional and unintentional unauthorized modifications to data. We will now dive into hash algorithms and their characteristics.

The One-Way Hash:

A one-way hash is a function that takes a variable-length string and a message and produces a fixed-length value called a hash value. For example, if Ahmad wants to send a message to Bilal and he wants to ensure the message does not get altered in an unauthorized fashion while it is being transmitted, he would calculate a hash value for the message and append it to the message itself. When Bilal receives the message, he performs the same hashing function Ahmad used and then compares his result with the hash value sent with the message. If the two values are the same, Bilal can be sure the message was not altered during transmission. If the two values are different, Bilal knows the message was altered, either intentionally or unintentionally, and he discards the message. The hashing algorithm is not a secret—it is publicly known. The secrecy of the oneway hashing function is its “one-wayness.” The function is run in only one direction, not the other direction . This is different from the one-way function used in public key cryptography, in which security is provided based on the fact that, without knowing a trapdoor, it is very hard to perform the one- way function backward on a message and come up with readable plaintext.

The hashing one-way function takes place without the use of any keys. Lets take a look at an example…

One-way Hash Example:

if Irfan writes a message, calculates a message digest, appends the digest to the message, and sends it on to Furqan, Khalid can intercept this message, alter Irfan’s message, recalculate another message digest, append it to the message, and send it on to Furqan. When Furqan receives it, he verifies the message digest, but never knows the message was actually altered by Khalid. Furqan thinks the message came straight from Irfan and it was never modified, because the two message digest values are the same. If Irfan wanted more protection than this, he would need to use message authentication code (MAC).

3.11.1 Message Authentication Codes (MACs):

A MAC function is an authentication scheme derived by applying a secret key to a message in some form. This does not mean the symmetric key is used to encrypt the message, though. You should be aware of two basic types of MACs: a hash MAC (HMAC), and CBC-MAC.

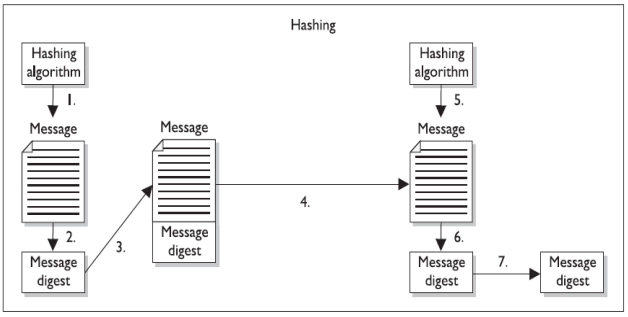
HMAC Example:

In the previous example, if Irfan were to use an HMAC function instead of just a plain hashing algorithm, a symmetric key would be concatenated with his message. The result of this process would be put through a

hashing algorithm, and the result would be a MAC value. This MAC value is then appended to his message and sent to Furqan. If Khalid were to intercept this message and modify it, he would not have the necessary symmetric key to create the MAC value that Furqan will attempt to generate.

Terminology:

The idea of a hashing function is simple. You run a message through a hashing algorithm, which in turn generates a hashing value. **A hashing value can also be called a message digest or fingerprint.**



1. The sender puts the message through a hashing function.

2. A message digest value is generated.

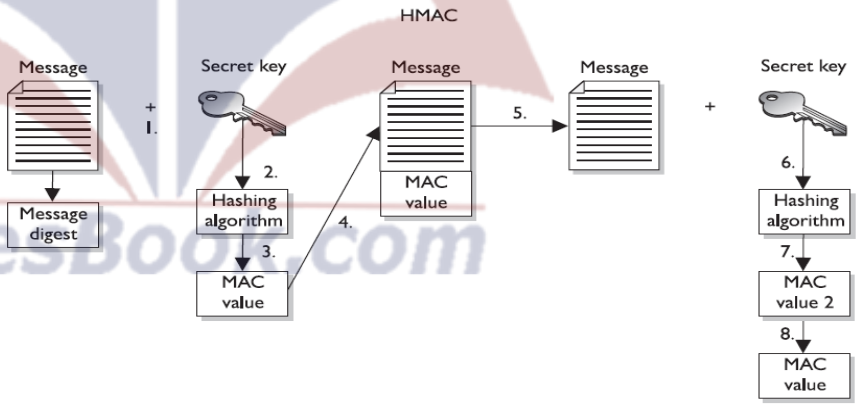
3. The message digest is appended to the message.

4. The sender sends the message to the receiver.

5. The receiver puts the message through a hashing function.

6. The receiver generates his own message digest value.

7. The receiver compares the two message digest values. If they are the same, the message has not been altered.



1. The sender concatenates a symmetric key with the message.

2. The result is put through a hashing algorithm.

3. A MAC value is generated.

4. The MAC value is appended to the message.

5. The sender sends the message to the receiver. (Just the message with the attached MAC value. The sender does not send the symmetric key with the message.)

6. The receiver concatenates a symmetric key with the message.

7. The receiver puts the results through a hashing algorithm and generates his own MAC value.

8. The receiver compares the two MAC values. If they are the same, the message has not been modified.

Concatenation:

Now, when we say that the message is concatenated with a symmetric key, we don't mean a symmetric key is used to encrypt the message. The message is not encrypted in an HMAC function, so there is no confidentiality being provided. Think about throwing a message in a bowl and then throwing a symmetric key in the same bowl . If you dump the contents of the bowl into a hashing algorithm, the result will be a MAC value.

HMAC & Symmetric Keys:

This type of technology requires the sender and receiver to have the same symmetric key. The HMAC function does not involve getting the symmetric key to the destination securely. That would have to happen through one of the other technologies we have discussed already (Diffie-Hellman and key agreement, or RSA and key exchange).

3.11.2 CBC-MAC:

If a CBC-MAC is being used, the message is encrypted with a symmetric block cipher in CBC mode, and the output of the final block of ciphertext is used as the MAC. The sender does not send the encrypted version of the message, but instead sends the plaintext version and the MAC attached to the message. The receiver receives the plaintext message and encrypts it with the same symmetric block cipher in CBC mode and calculates an independent MAC value. The receiver compares the new MAC value with the MAC value sent with the message. This method does not use a hashing algorithm as does HMAC.

CBC-MAC:

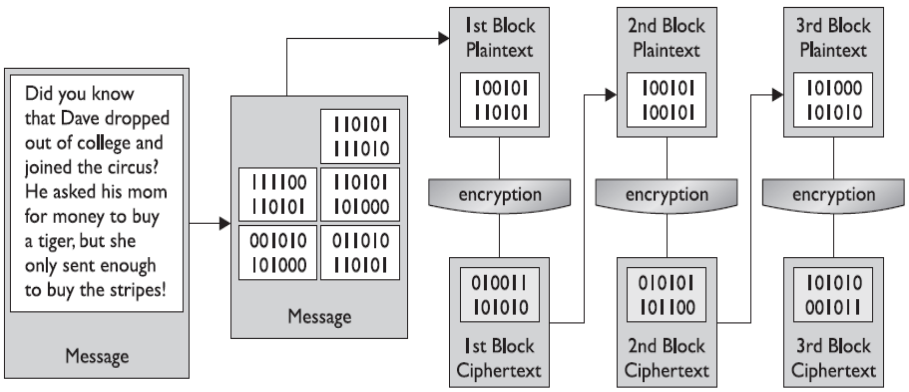
The use of the symmetric key ensures that the only person who can verify the integrity of the message is the person who has a copy of this key. No one else can verify the data's integrity, and if someone were to make a change to the data, he could not generate the MAC value (HMAC or CBC-MAC) the receiver would be looking for. Any modifications would be detected by the receiver. Now the receiver knows that the message came from the system that has the other copy of the same symmetric key, so MAC provides a form of authentication. It provides data origin authentication, sometimes referred to as system authentication. This is different from user authentication, which would require the use of a private key.

A private key is bound to an individual; a symmetric key is not. MAC authentication provides the weakest form of authentication because it is not bound to a user, just to a computer or device.

Lecture 18

3.11.4 CMAC Cipher-Based Message Authentication Code

As with most things in security, the industry found some security issues with CBCMAC and created Cipher-Based Message Authentication Code (CMAC). CMAC provides the same type of data origin authentication and integrity as CBC-MAC, but is more secure mathematically. CMAC is a variation of CBC-MAC. It is approved to work with AES and Triple DES.



CMAC is a block cipher–based message authentication code algorithm. This means that it can provide the authentication of the data origin (as in the computer it was sent from) but not the person who sent it.

Hash Functionality:

1. Sender puts a message through a hashing algorithm and generates a message digest (MD) value.

2. Sender sends message and MD value to receiver.

3. Receiver runs just the message through the same hashing algorithm and creates an independent MD value.

4. Receiver compares both MD values. If they are the same, the message was not modified.

SERVICES PROVIDED:

Integrity; not confidentiality or authentication. Can detect only unintentional modifications**.**

HMAC Functionality:

1. Sender concatenates a message and secret key and puts the result through a hashing algorithm. This creates a MAC value.

2. Sender appends the MAC value to the message and sends it to the receiver.

3. The receiver takes just the message and concatenates it with his own symmetric key. This results in an independent MAC value.

4. The receiver compares the two MAC values. If they are the same, the receiver knows the message was not modified and knows from which system it came.

SERVICES PROVIDED:

Integrity and data origin authentication; confidentiality is not provided.

CBC-MAC Functionality:

1. Sender encrypts a message with a symmetric block algorithm in CBC mode.

2. The last block is used as the MAC.

3. The plaintext message and the appended MAC are sent to the receiver.

4. The receiver encrypts the message, creates a new MAC, and compares the two values. If they are the same, the receiver knows the message was not modified and from which system it came**.**

SERVICES PROVIDED:

Data origin authentication and integrity.

CMAC Functionality:

CMAC works the same way as the CBC-MAC, but is based on more complex logic and mathematical functions**.**

Types of Hashing Algorithms:

As stated earlier, the goal of using a one-way hash function is to provide a fingerprint of the message. If two different messages produce the same hash value, it would be easier for an attacker to break that security mechanism, because patterns would be revealed. A strong one-hash function should not provide the same hash value for two or more different messages. If a hashing algorithm takes steps to ensure it does not create the same hash value for two or more messages, it is said to be *collision free.*

3.11.5 Desired Features Of Cryptographic Hash Functions:

• The hash should be computed over the entire message.

• The hash should be a one-way function so messages are not disclosed by their values.

• Given a message and its hash value, computing another message with the same hash value should be impossible.

• The function should be resistant to birthday attacks (explained in the upcoming section "Attacks Against One-Way Hash Functions").

3.12 Hashing Algorithms Used Today:

Algorithm	Description
Message Digest 2 (MD2) algorithm	One-way function. Produces a 128-bit hash value. Much slower than MD4 and MD5.
Message Digest 4 (MD4) algorithm	One-way function. Produces a 128-bit hash value.
Message Digest 5 (MD5) algorithm	One-way function. Produces a 128-bit hash value. More complex than MD4.
HAVAL	One-way function. Variable-length hash value. Modification of MD5 algorithm that provides more protection against attacks that affect MD5.
Secure Hash Algorithm (SHA)	One-way function. Produces a 160-bit hash value. Used with DSA.
SHA-1, SHA-256, SHA-384, SHA-512	Updated version of SHA. SHA-1 produces a 160-bit hash value, SHA-256 creates a 256-bit value, and so on.

3.12.1 MD2

MD2 is a one-way hash function designed by Ron Rivest that creates a 128-bit message digest value. It is not necessarily any weaker than the other algorithms in the "MD" family, but it is much slower.

3.12.2 MD4:

MD4 is a one-way hash function designed by Ron Rivest. It also produces a 128-bit message digest value. It is used for high-speed computation in software implementations and is optimized for microprocessors**.**

3.12.3 MD5:

MD5 was also created by Ron Rivest and is the newer version of MD4. It still produces a 128-bit hash, but the algorithm is more complex, which makes it harder to break. MD5 added a fourth round of operations to be performed during the hashing functions and makes several of its mathematical operations carry out more steps or more complexity to provide a higher level of security. Recent research has shown MD5 to be subject

to collision attacks, and it is therefore no longer suitable for applications like SSL certificates and digital signatures that require collision attack resistance.

3.12.4 SHA:

SHA was designed by NSA and published by NIST to be used with the Digital Signature Standard (DSS). SHA was designed to be used in digital signatures and was developed when a more secure hashing algorithm was required for U. S. government applications. SHA produces a 160-bit hash value, or message digest. This is then inputted into an asymmetric algorithm, which computes the signature for a message. SHA is similar to MD4. It has some extra mathematical functions and produces a 160-bit hash instead of a 128-bit hash, which makes it more resistant to brute force attacks, including birthday attacks. SHA was improved upon and renamed SHA-1. Recently, newer versions of this algorithm (collectively known as the SHA-2 family) have been developed and released: SHA-256, SHA-384, and SHA-512.

3.12.5 HAVAL

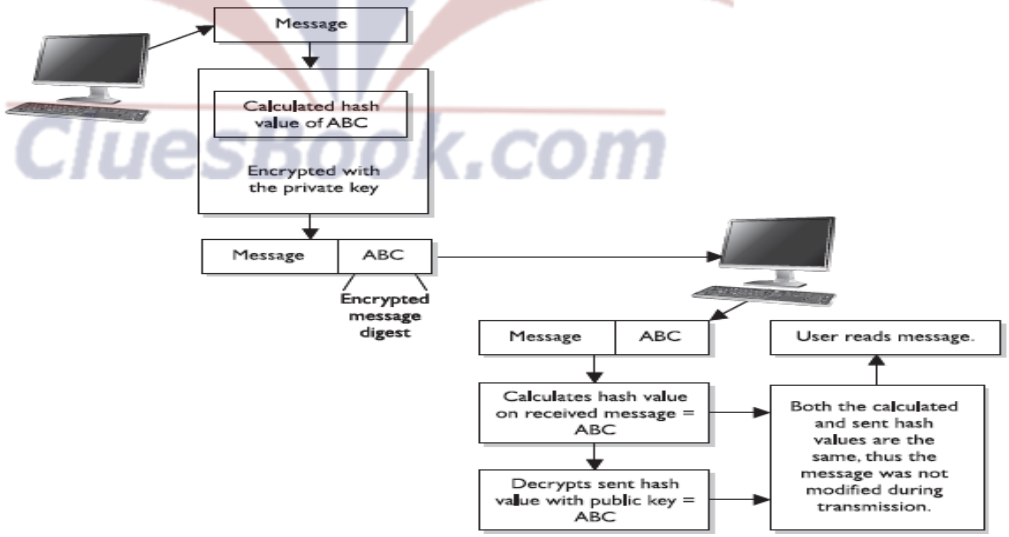
HAVAL is a variable-length one-way hash function and is a modification of MD5. It processes message blocks twice the size of those used in MD5; thus, it processes blocks of 1,024 bits. HAVAL can produce hashes from 128 to 256 bits in length.

3.12.6 Tiger:

Ross Anderson and Eli Biham developed a hashing algorithm called Tiger in 1995. It was designed to carry out hashing functionalities on 64-bit systems and to be faster than MD5 and SHA-1. The resulting hash value is 192 bits. Design wise most hash algorithms (MD5, RIPEMD, SHA-0, and SHA-1) are derivatives or have been built upon the MD4 architecture. Tiger was built upon a different type of architecture with the goal of not being vulnerable to the same type of attacks that could be successful toward the other hashing algorithms.

Lecture 19

3.13.1 Digital Signatures:



A digital signature is a hash value that has been encrypted with the sender's private key. The act of signing means encrypting the message's hash value with a private key, as shown in the Figure. From our earlier example if Ahmad wants to ensure that the message he sends to Bilal is not modified and he wants him to be sure it came only from him, he can digitally sign the message. This means that a one-way hashing function would be run on the message, and then Ahmad would encrypt that hash value with his private key.

When Bilal receives the message, he will perform the hashing function on the message, and come up with his own hash value. Then he will decrypt the sent hash value (digital signature) with Ahmad's public key. He then compares the two values, and if they are the same, he can be sure the message was not altered during transmission. He is also sure the message came from Ahmad because the value was encrypted with his private key. The hashing function ensures the integrity of the message, and the signing of the hash value provides authentication and non-repudiation. The act of signing just means the value was encrypted with a private key. We need to be clear on all the available choices within cryptography, because different steps and algorithms provide different types of security services:

Services Provided By Symmetric Key Algorithms

Algorithm Type	Encryption	Digital Signature	Hashing Function	Key Distribution
Symmetric Key Algorithms				
DES	X			
3DES	X			
Blowfish	X			
IDEA	X			
RC4	X			
SAFER	X			

Algorithm Type	Encryption	Digital Signature	Hashing Function	Key Distribution
Asymmetric Key Algorithms				
RSA	X	X		X
ECC	X	X		X
Diffie-Hellman				X
El Gamal	X	X		X
DSA		X		
LUC	X	X		X
Knapsack	X	X		X

Algorithm Type	Encryption	Digital Signature	Hashing Function	Key Distribution
Hashing Algorithms				
Ronald Rivest family of hashing functions: MD2, MD4, and MD5			X	
SHA			X	
HAVAL (variable-length hash values using a one-way function design)			X	

• A message can be encrypted, which provides confidentiality

• A message can be hashed, which provides integrity

• A message can be digitally signed, which provides authentication, nonrepudiation, and integrity

• A message can be encrypted and digitally signed, which provides confidentiality, authentication, nonrepudiation, and integrity

Some algorithms can only perform encryption, whereas others support digital signatures and encryption. When hashing is involved, a hashing algorithm is used, not an encryption algorithm. It is important to understand that not all algorithms can necessarily provide all security services. Most of these algorithms are used in some type of combination to provide all the necessary security services required of an environment. The Table shows the services provided by the algorithms.

3.13.2 Digital Signature Standard (DSS):

Because digital signatures are so important in proving who sent which messages, the U.S. government decided to establish standards pertaining to their functions and acceptable use. In 1991, NIST proposed a federal standard called the Digital Signature Standard (DSS). It was developed for federal departments and agencies, but most vendors also designed their products to meet these specifications. The federal government requires its departments to use DSA, RSA, or the elliptic curve digital signature algorithm (ECDSA) and SHA. SHA creates a 160-bit message digest output, which is then inputted into one of the three mentioned digital signature algorithms. SHA is used to ensure the integrity of the message, and the other algorithms are used to digitally sign the message. This is an example of how two different algorithms are combined to provide the right combination of security services.

RSA and DSA are the best known and most widely used digital signature algorithms. DSA was developed by the NSA. Unlike RSA, DSA can be used only for digital signatures, and DSA is slower than RSA in signature verification. RSA can be used for digital signatures, encryption, and secure distribution of symmetric keys.

3.14.1 Public Key Infrastructure (PKI):

Public key infrastructure (PKI) consists of programs, data formats, procedures, communication protocols, security policies, and public key cryptographic mechanisms working in a comprehensive manner to enable a wide range of dispersed people to communicate in a secure and predictable fashion. In other words, a PKI establishes a level of trust within an environment.

PKI is an ISO authentication framework that uses public key cryptography and the X.509 standard.

The framework was set up to enable authentication to happen across different networks and the Internet. Particular protocols and algorithms are not specified, which is why PKI is called a framework and not a specific technology. PKI provides authentication, confidentiality, non-repudiation, and integrity of the messages exchanged. PKI is a hybrid system of symmetric and asymmetric key algorithms and methods, which were discussed in earlier sections. There is a difference between public key cryptography and PKI. Public key cryptography is another name for asymmetric algorithms, while PKI is what its name states—it is an infrastructure. The infrastructure assumes that the receiver's identity can be positively ensured through certificates and that an asymmetric algorithm will automatically carry out the process of key exchange.

Lecture 20

3.14.2 Definition Of Public Key Infrastructure (PKI):

Public key infrastructure (PKI) consists of programs, data formats, procedures, communication protocols, security policies, and public key cryptographic mechanisms working in a comprehensive manner to enable a wide range of dispersed people to communicate in a secure and predictable fashion. In other words, a PKI establishes a level of trust within an environment. PKI provides authentication, confidentiality, non- repudiation , and integrity of the messages exchanged. PKI is a hybrid system of symmetric and asymmetric key algorithms and methods, which were discussed in earlier sections. The infrastructure assumes that the receiver's identity can be positively ensured through certificates and that an asymmetric algorithm will automatically carry out the process of key exchange. The infrastructure therefore contains the pieces that will identify users, create and distribute certificates, maintain and revoke certificates, distribute and maintain encryption keys, and enable all technologies to communicate and work together for the purpose of encrypted communication and authentication. Public key cryptography is one piece in PKI, but many other pieces make up this infrastructure.

Example:

An analogy can be drawn with the e-mail protocol Simple Mail Transfer Protocol (SMTP). SMTP is the technology used to get e-mail messages from here to there, but many other things must be in place before this protocol can be productive. We need e-mail clients, e-mail servers, and e-mail messages, which together build a type of infrastructure—an e-mail infrastructure. PKI is made up of many different parts: certificate authorities, registration authorities, certificates, keys, and users.

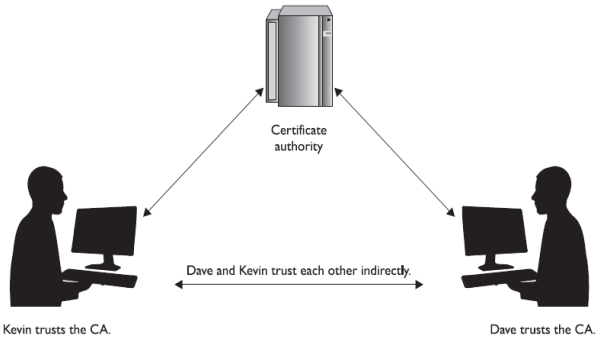
3.14.3 PKI Components:

Each person who wants to participate in a PKI requires a digital certificate, which is a credential that contains the public key for that individual along with other identifying information. The certificate is created and signed (digital signature) by a trusted third party, which is a certificate authority (CA). When the CA signs the certificate, it binds the individual's identity to the public key, and the CA takes liability for the authenticity of that individual . It is this trusted third party (the CA) that allows people who have never met to authenticate to each other and to communicate in a secure method. If Kevin has never met Dave, but would like to communicate securely with him, and they both trust the same CA, then Kevin could retrieve Dave's digital certificate and start the process.

3.14.4 Certificate Authorities (CAs):

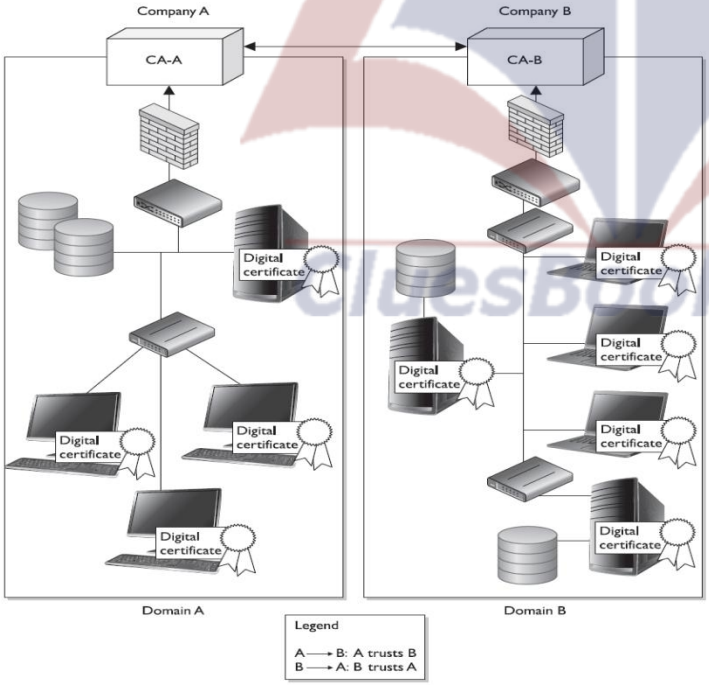
A CA is a trusted organization (or server) that maintains and issues digital certificates. When a person requests a certificate, the registration authority (RA) verifies that individual's identity and passes the certificate request off to the CA. The CA constructs the certificate, signs it, sends it to the requester, and maintains the certificate over its lifetime. When another person wants to communicate with this person, the CA will basically vouch for that person's identity. When Dave receives a digital certificate from Kevin, Dave will go through steps to validate it. Basically, by providing Dave with his digital certificate, Kevin is stating, "I know you don't know or trust me, but here is this document that was created by someone you do know and

trust. The document says I am a good guy and you should trust me." Once Dave validates the digital certificate , he extracts Kevin's public key, which is embedded within it. Now Dave knows this public key is bound to Kevin. He also knows that if Kevin uses his private key to create a digital signature and Dave can properly decrypt it using this public key, it did indeed come from Kevin. The CA can be internal to an organization. Such a setup would enable the company to control the CA server, configure how authentication takes place, maintain the certificates, and recall certificates when necessary.



Remember the man-in-the-middle attack covered earlier in the section "The Diffie-Hellman Algorithm"? This attack is possible if two users are not working in a PKI environment and do not truly know the identity of the owners of the public keys.

More and more organizations are setting up their own internal PKIs. When these independent PKIs need to interconnect to allow for secure communication to take place (either between departments or between different companies), there must be a way for the two root CAs to trust each other. The two CAs do not have a CA above them they can both trust, so they must carry out cross certification. A cross certification is the process undertaken by CAs to establish a trust relationship in which they rely upon each other's digital certificates and public keys as if they had issued them themselves. When this is set up, a CA for one company can validate digital certificates from the other company and vice versa.



3.14.5 Certificate Revocation List (CRL):

The CA is responsible for creating and handing out certificates, maintaining them, and revoking them if necessary. Revocation is handled by the CA, and the revoked certificate information is stored on a certificate revocation list (CRL). This is a list of every certificate that has been revoked. This list is maintained and updated periodically. A certificate may be revoked because the key holder's private key was compromised or because the CA discovered the certificate was issued to the wrong person.

CRL Analogy: An analogy for the use of a CRL is how a driver's license is used by a police officer. If an officer pulls over Faheem for speeding, the officer will ask to see Faheem's license. The officer will then run a check on the license to find out if Faheem is wanted for any other infractions of the law and to verify the license has not expired. The same thing happens when a person compares a certificate to a CRL. If the certificate became invalid for some reason, the CRL is the mechanism for the CA to let others know this information.

CRL Challenges: CRLs are the thorn in the side of many PKI implementations. They are challenging for a long list of reasons. It is interesting to know that, by default, web browsers do not check a CRL to ensure that a certificate is not revoked. So when you are setting up an SSL connection to do e-commerce over the Internet, you could be relying on a certificate that has actually been revoked. Not good.

3.14.6 OCSP:

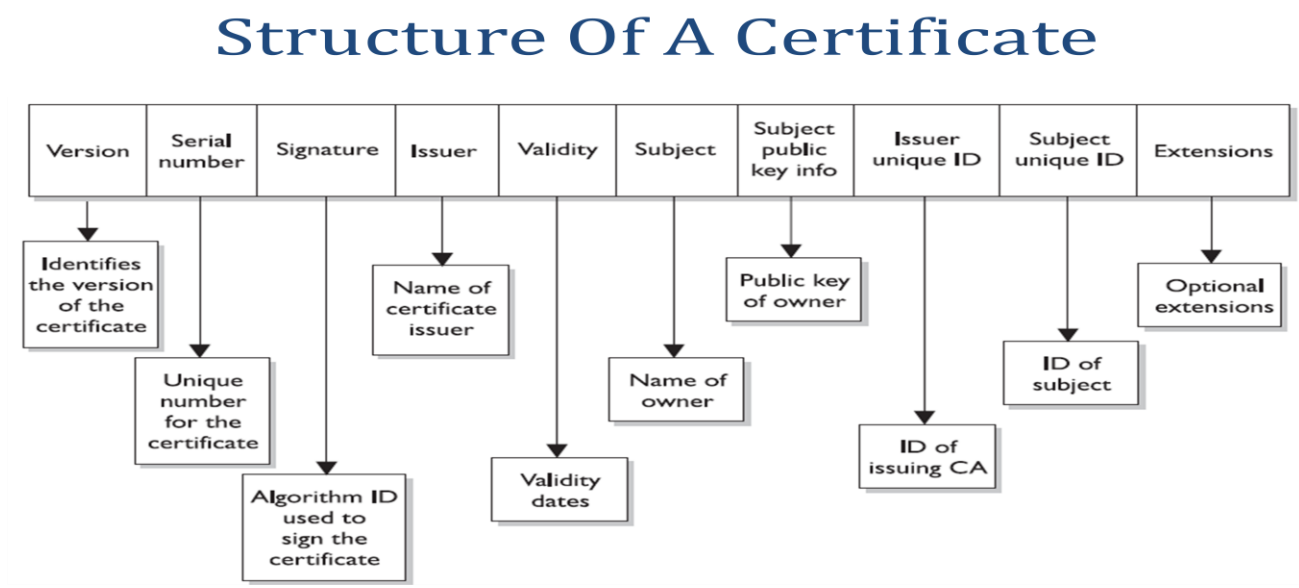
Online Certificate Status Protocol (OCSP) is being used more and more rather than the cumbersome CRL approach. When using just a CRL, the user's browser must either check a central CRL to find out if the certification has been revoked or continually push out CRL values to the clients to ensure they have an updated CRL.

OCSP Advantages: If OCSP is implemented, it does this work automatically in the background. It carries out real -time validation of a certificate and reports back to the user whether the certificate is valid, invalid, or unknown. OCSP checks the CRL that is maintained by the CA. So the CRL is still being used, but now we have a protocol developed specifically to check the CRL during a certificate validation process.

Lecture 21

3.14.7 Certificates:

One of the most important pieces of a PKI is its digital certificate. A *certificate* is the mechanism used to associate a public key with a collection of components in a manner that is sufficient to uniquely identify the claimed owner. The standard for how the CA creates the certificate is X.509, which dictates the different fields used in the certificate and the valid values that can populate those fields. We are currently at version 4 of this standard, which is often denoted as X.509v4. Many cryptographic protocols use this type of certificate, including SSL. The certificate includes the serial number, version number, identity information, algorithm information, lifetime dates, and the signature of the issuing authority, as shown in the Figure.



3.14.8 The Registration Authority:

The registration authority (RA) performs the certification registration duties. The RA establishes and confirms the identity of an individual, initiates the certification process with a CA on behalf of an end user, and performs certificate life-cycle management functions. The RA cannot issue certificates, but can act as a broker between the user and the CA. When users need new certificates, they make requests to the RA, and the RA verifies all necessary identification information before allowing a request to go to the CA.

3.14.9 PKI Example:

Now that we know some of the main pieces of a PKI and how they actually work together, let's walk through an example. First, suppose that John needs to obtain a digital certificate for himself so he can participate in a PKI. The following are the steps to do so:

1. John makes a request to the RA.

2. The RA requests certain identification information from John, such as a copy of his driver's license, his phone number, his address, and other identifying information.

3. Once the RA receives the required information from John and verifies it, the RA sends his certificate request to the CA.

4. The CA creates a certificate with John's public key and identity information embedded.

(The private/public key pair is generated either by the CA or on John's machine, which depends on the systems' configurations. If it is created at the CA, his private key needs to be sent to him by secure means. In most cases, the user generates this pair and sends in his public key during the registration process.)

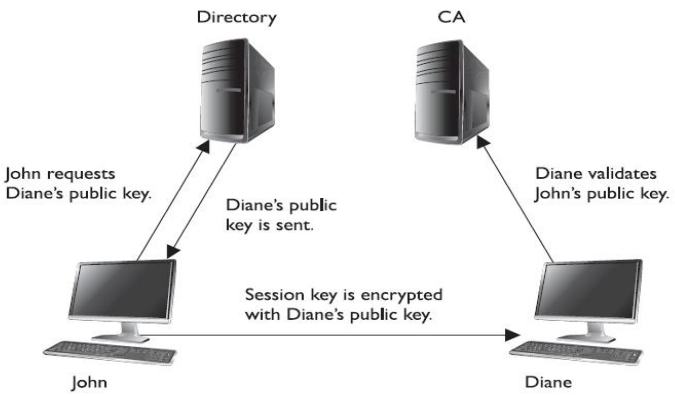
Now John is registered and can participate in a PKI. John and Diane decide they want to communicate, so they take the following steps, shown in the Figure.

5. John requests Diane's public key from a public directory.

6. The directory, sometimes called a repository, sends Diane's digital certificate.

7. John verifies the digital certificate and extracts her public key. John uses this public key to encrypt a session key that will be used to encrypt their messages. John sends the encrypted session key to Diane. John also sends his certificate, containing his public key, to Diane.

8. When Diane receives John's certificate, her browser looks to see if it trusts the CA that digitally signed this certificate. Diane's browser trusts this CA and, after she verifies the certificate, both John and Diane can communicate using encryption.



CA & User Relationships

PKI Components:

A PKI may be made up of the following entities and functions:

1. CA
2. RA
3. Certificate repository

4. Certificate revocation system
5. Key backup and recovery system
6. Automatic key update
7. Management of key histories
8. Timestamping
9. Client-side software

3.14.10 PKI Security Services:

PKI supplies the following security services:

• Confidentiality • Access control • Integrity • Authentication • Nonrepudiation

A PKI must retain a key history, which keeps track of all the old and current public keys that have been used by individual users. For example, if Kevin encrypted a symmetric key with Dave's old public key, there should be a way for Dave to still access this data. This can only happen if the CA keeps a proper history of Dave's old certificates and keys.

NOTE: Another important component that must be integrated into a PKI is a reliable time source that provides a way for secure time-stamping. This comes into play when true non-repudiation is required.

Lecture 23

3.16 Link Encryption Vs. End-to-End Encryption:

Encryption can be performed at different communication levels, each with different types of protection and implications. Two general modes of encryption implementation are link encryption and end-to-end encryption.

Link encryption encrypts all the data along a specific communication path, as in a satellite link, T3 line, or telephone circuit. Not only is the user information encrypted, but the header, trailers, addresses, and routing data that are part of the packets are also encrypted.

Link Encryption Vs. End-to-End Encryption:

The only traffic not encrypted in this technology is the data link control messaging information, which includes instructions and parameters that the different link devices use to synchronize communication methods.

Link encryption provides protection against packet sniffers and eavesdroppers.

In end-to-end encryption, the headers, addresses, routing, and trailer information are not encrypted, enabling attackers to learn more about a captured packet and where it is headed.

Link encryption, which is sometimes called *online encryption,* is usually provided by service providers and is incorporated into network protocols.

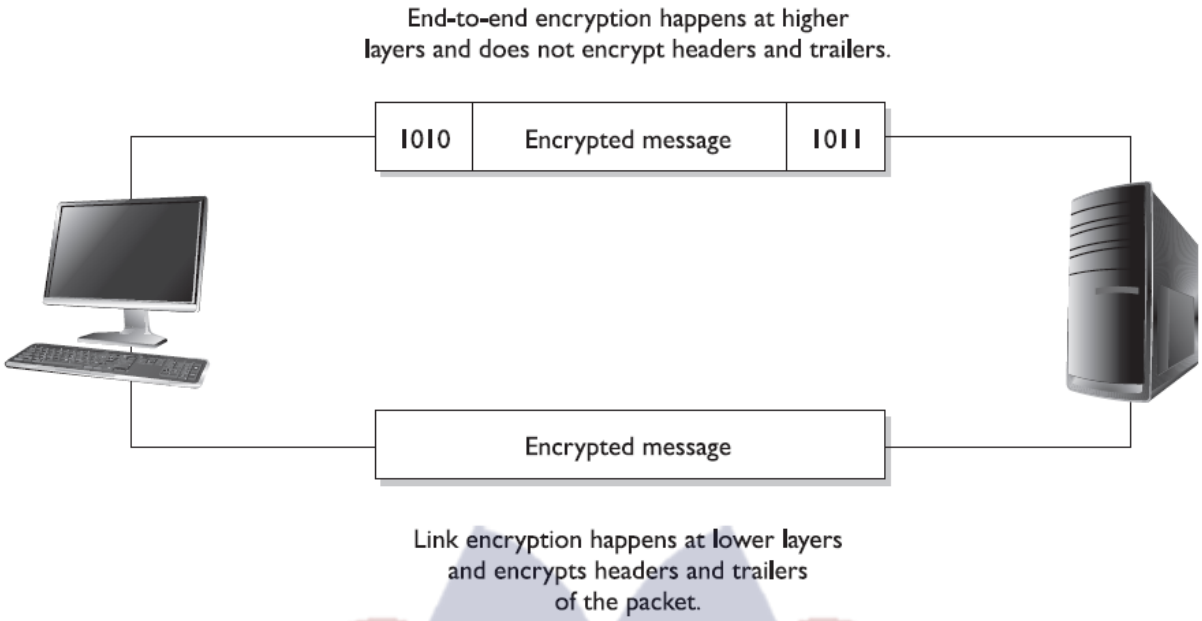
All of the information is encrypted, and the packets must be decrypted at each hop so the router, or other intermediate device, knows where to send the packet next. The router must decrypt the header portion of the packet, read the routing and address information within the header, and then re-encrypt it and send it on its way.

With end-to-end encryption, the packets do not need to be decrypted and then encrypted again at each hop, because the headers and trailers are not encrypted. The devices in between the origin and destination just read the necessary routing information and pass the packets on their way.

End-to-end encryption is usually initiated by the user of the originating computer. It provides more flexibility for the user to be able to determine whether or not certain messages will get encrypted. It is called “end-to-end encryption” because the message stays encrypted from one end of its journey to the other. Link encryption has to decrypt the packets at every device between the two ends.

Link encryption occurs at the data link and physical layers, as depicted in the Figure. Hardware encryption devices interface with the physical layer and encrypt all data that passes through them. Because no part of the data is available to an attacker, the attacker cannot learn basic information about how data flows through the environment. This is referred to as *traffic-flow security.*

Link and end-to-end encryption happen at different OSI layers.



Encryption At Different Layers:

In reality, encryption can happen at different layers of an operating system and network stack. The following are just a few examples:

- End-to-end encryption happens within the applications.

• SSL encryption takes place at the transport layer.

• PPTP encryption takes place at the data link layer.

• Link encryption takes place at the data link and physical layers.

3.16.1 End-to-end Encryption

Advantages of end-to-end encryption include the following:

• It provides more flexibility to the user in choosing what gets encrypted and how.

• Higher granularity of functionality is available because each application or user can choose specific configurations.

• Each hop computer on the network does not need to have a key to decrypt each packet.

Disadvantages of end-to-end encryption include the following:

• Headers, addresses, and routing information are not encrypted, and therefore not protected.

3.16.2 Link Encryption

Advantages of link encryption include the following:

• All data are encrypted, including headers, addresses, and routing information.

• Users do not need to do anything to initiate it. It works at a lower layer in the OSI model.

Disadvantages of link encryption include the following:

• Key distribution and management are more complex because each hop device must receive a key, and when the keys change, each must be updated.

• Packets are decrypted at each hop; thus, more points of vulnerability exist.

3.16.3 Hardware vs. Software Cryptography Systems

Encryption can be done through software or hardware, and there are trade-offs with each.

- Generally, software is less expensive and provides a slower throughput than hardware mechanisms.
- Software cryptography methods can be more easily modified and disabled compared to hardware systems, but it depends on the application and the hardware product.

If a company needs to perform high-end encryption functions at a higher speed, the company will most likely implement a h**ardwar**e solution.

3.17 Email Standards

Like other types of technologies, cryptography has industry standards and de facto standards.

Standards are necessary because they help ensure interoperability among vendor products. Standards usually mean that a certain technology has been under heavy scrutiny and properly tested and accepted by many similar technology communities.

A company still needs to decide on what type of standard to follow and what type of technology to implement.

For a cryptography implementation, the company would need to decide what must be protected by encryption, whether digital signatures are necessary, how key management should take place, what types of resources are available to implement and maintain the technology, and what the overall cost will amount to.

If a company only needs to encrypt some e-mail messages here and there, then PGP may be the best choice.

If the company wants all data encrypted as it goes throughout the network and to sister companies, then a link encryption implementation may be the best choice.

If a company wants to implement a single sign-on environment where users need to authenticate to use different services and functionality throughout the network, then implementing a PKI or Kerberos might serve it best.

To make the most informed decision, the network administrators should understand each type of technology and standard, and should research and test each competing product within the chosen technology before making the final purchase. Cryptography, including how to implement and maintain it, can be a complicated subject.

The following sections briefly describe some of the most popular e-mail standards in use**.**

3.17.1 Multipurpose Internet Mail Extension

Multipurpose Internet Mail Extension (MIME) is a technical specification indicating how multimedia data and e-mail attachments are to be transferred.

The Internet has mail standards that dictate how mail is to be formatted, encapsulated, transmitted, and opened. If a message or document contains a binary attachment, MIME dictates how that portion of the message should be handled.

When an attachment contains an audio clip, graphic, or some other type of multimedia component, the e-mail client will send the file with a header that describes the file type. For example, the header might indicate that the MIME type is Image and that the subtype is jpeg.

Although this will be in the header, many times systems also use the file's extension to identify the MIME type. So, in the preceding example, the file's name might be stuff.jpeg.

The user's system will see the extension .jpeg, or see the data in the header field, and look in its association list to see what program it needs to initialize to open this particular file. If the system has JPEG files associated with the Explorer application, then Explorer will open and present the picture to the user.

Sometimes systems either do not have an association for a specific file type or do not have the helper program necessary to review and use the contents of the file.

When a file has an unassociated icon assigned to it, it might require the user to choose the Open With command and choose an application in the list to associate this file with that program. So when the user double-clicks that file, the associated program will initialize and present the file. If the system does not

have the necessary program, the website might offer the necessary helper program, like Acrobat or an audio program that plays WAV files.

MIME is a specification that dictates how certain file types should be transmitted and handled. This specification has several types and subtypes, enables different computers to exchange data in varying formats, and provides a standardized way of presenting the data.

So if Sean views a funny picture that is in GIF format, he can be sure that when he sends it to Debbie, it will look exactly the same.

3.17.2 S/MIME

Secure MIME (S/MIME) is a standard for encrypting and digitally signing electronic mail and for providing secure data transmissions.

S/MIME extends the MIME standard by allowing for the encryption of e-mail and attachments. The encryption and hashing algorithms can be specified by the user of the mail package, instead of having it dictated to them.

S/MIME follows the Public Key Cryptography Standards (PKCS).

S/MIME provides confidentiality through encryption algorithms, integrity through hashing algorithms, authentication through the use of X.509 public key certificates, and non-repudiation through cryptographically signed message digests.

3.17.3 Privacy-Enhanced Mail

Privacy-Enhanced Mail (PEM) is an Internet standard to provide secure e-mail over the Internet and for in house communication infrastructures. The protocols within PEM provide authentication, message integrity, encryption, and key management.

This standard was developed to provide compatibility with many types of key-management processes and symmetric and public key methods of encryption. It was also designed to be compatible with PKCS.

PEM is a series of message authentication and encryption technologies developed by several governing groups. PEM can use AES for encryption and RSA for sender authentication and key management. It also provides support for non-repudiation.

The following are specific components that can be used in PEM:

Messages encrypted with AES in CBC mode

• Public key management, provided by using RSA

• X.509 standard, used for certification structure and format

PEM has not caught on to the extent the developers had planned. The main issue is that PEM provides too much structure for different environments that require more flexibility in their secure communication infrastructure.

3.17.4 Message Security Protocol

The ***Message Security Protocol (MSP)*** is the military's PEM. Developed by the NSA,

it is an X.400-compatible application-level protocol used to secure e-mail messages.

MSP can be used to sign and encrypt messages and to perform hashing functions. Like PEM, applications that incorporate MSP enable different algorithms and parameters to be used to provide greater flexibility.

3.17.5 Pretty Good Privacy

Pretty Good Privacy (PGP) was designed by Phil Zimmerman as a freeware e-mail security program and was released in 1991. It was the first widespread public key encryption program. PGP is a complete cryptosystem that uses cryptographic protection to protect e-mail and files.

It can use RSA public key encryption for key management and use IDEA symmetric cipher for bulk encryption of data, although the user has the option of picking different types of algorithms for these functions.

PGP can provide confidentiality by using the IDEA encryption algorithm, integrity by using the MD5 hashing algorithm, authentication by using the public key certificates, and non repudiation by using cryptographically signed messages. PGP uses its own type of digital certificates rather than what is used in PKI, but they both have similar purposes.

The user's private key is generated and encrypted when the application asks the user to randomly type on her keyboard for a specific amount of time. Instead of using passwords, PGP uses passphrases. The passphrase is used to encrypt the user's private key that is stored on the user hard drive.

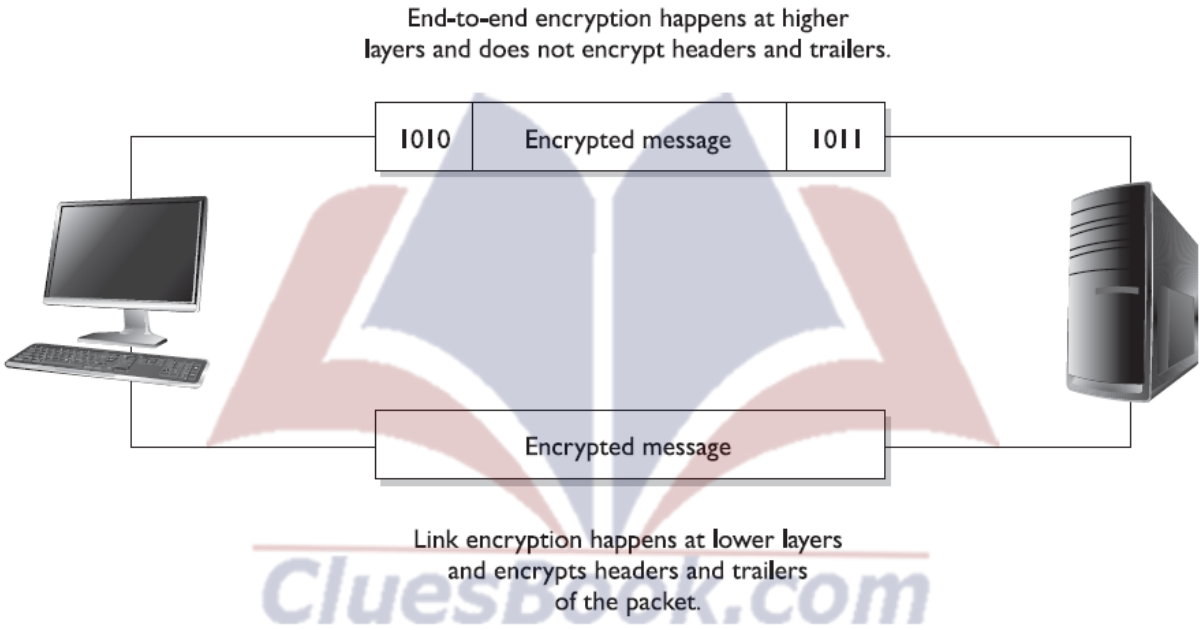
PGP does not use a hierarchy of CAs, or any type of formal trust certificates, but instead relies on a "web of trust" in its key management approach. Each user generates and distributes his or her public key, and users sign each other's public keys, which creates a community of users who trust each other. This is different from the CA approach, where no one trusts each other; they only trust the CA.

For example, if Mark and Joe want to communicate using PGP, Mark can give his public key to Joe. Joe signs Mark's key and keeps a copy for himself. Then, Joe gives a copy of his public key to Mark so they can start communicating securely. Later, Mark would like to communicate with Sally, but Sally does not know Mark and does not know if she can trust him. Mark sends Sally his public key, which has been signed by Joe. Sally has Joe's public key, because they have communicated before, and she trusts Joe. Because Joe signed Mark's public key, Sally now also trusts Mark and sends her public key and begins communicating with him.

So, basically, PGP is a system of "I don't know you, but my buddy Joe says you are an all right guy, so I will trust you on Joe's word."

Lecture 24

Link and end-to-end encryption happen at different OSI layers.



Encryption At Different Layers

- End-to-end encryption happens within the applications.
- SSL encryption takes place at the transport layer.
- PPTP encryption takes place at the data link layer.
- Link encryption takes place at the data link and physical layers.

Multipurpose Internet Mail Extension

Multipurpose Internet Mail Extension (MIME) is a technical specification indicating how multimedia data and e-mail attachments are to be transferred.

The Internet has mail standards that dictate how mail is to be formatted, encapsulated, transmitted, and opened. If a message or document contains a binary attachment, MIME dictates how that portion of the message should be handled.

When an attachment contains an audio clip, graphic, or some other type of multimedia component, the e-mail client will send the file with a header that describes the file type. For example, the header might indicate that the MIME type is Image and that the subtype is jpeg.

Although this will be in the header, many times systems also use the file's extension to identify the MIME type. So, in the preceding example, the file's name might be stuff.jpeg.

The user's system will see the extension .jpeg, or see the data in the header field, and look in its association list to see what program it needs to initialize to open this particular file. If the system has JPEG files associated with the Explorer application, then Explorer will open and present the picture to the user.

Sometimes systems either do not have an association for a specific file type or do not have the helper program necessary to review and use the contents of the file.

When a file has an unassociated icon assigned to it, it might require the user to choose the Open With command and choose an application in the list to associate this file with

that program. So when the user double-clicks that file, the associated program will initialize and present the file. If the system does not have the necessary program, the website might offer the necessary helper program, like Acrobat or an audio program that plays WAV files.

MIME is a specification that dictates how certain file types should be transmitted and handled. This specification has several types and subtypes, enables different computers to exchange data in varying formats, and provides a standardized way of presenting the data.

So if Sean views a funny picture that is in GIF format, he can be sure that when he sends it to Debbie, it will look exactly the same.

S/MIME

Secure MIME (S/MIME) is a standard for encrypting and digitally signing electronic mail and for providing secure data transmissions.

S/MIME extends the MIME standard by allowing for the encryption of e-mail and attachments. The encryption and hashing algorithms can be specified by the user of the mail package, instead of having it dictated to them.

S/MIME follows the Public Key Cryptography Standards (PKCS).

S/MIME provides confidentiality through encryption algorithms, integrity through hashing algorithms, authentication through the use of X.509 public key certificates, and non-repudiation

through cryptographically signed message digests.

Privacy-Enhanced Mail

Privacy-Enhanced Mail (PEM) is an Internet standard to provide secure e-mail over the Internet and for in house communication infrastructures. The protocols within PEM provide authentication, message integrity, encryption, and key management.

This standard was developed to provide compatibility with many types of key-management processes and symmetric and public key methods of encryption. It was also designed to be compatible with PKCS.

PEM is a series of message authentication and encryption technologies developed by several governing groups. PEM can use AES for encryption and RSA for sender authentication and key management. It also provides support for non-repudiation.

The following are specific components that can be used in PEM:

• Messages encrypted with AES in CBC mode

• Public key management, provided by using RSA

• X.509 standard, used for certification structure and format

PEM has not caught on to the extent the developers had planned. The main issue is that PEM provides too much structure for different environments that require more flexibility in their secure communication infrastructure.

Pretty Good Privacy

Pretty Good Privacy (PGP) was designed by Phil Zimmerman as a freeware e-mail security program and was released in 1991. It was the first widespread public key encryption program. PGP is a complete cryptosystem that uses cryptographic protection to protect e-mail and files.

It can use RSA public key encryption for key management and use IDEA symmetric cipher for bulk encryption of data, although the user has the option of picking different types of algorithms for these functions.

PGP can provide confidentiality by using the IDEA encryption algorithm, integrity by using the MD5 hashing algorithm, authentication by using the public key certificates, and non repudiation by using cryptographically signed messages. PGP uses its own type of digital certificates rather than what is used in PKI, but they both have similar purposes.

The user's private key is generated and encrypted when the application asks the user to randomly type on his keyboard for a specific amount of time. Instead of using passwords, PGP uses passphrases. The passphrase is used to encrypt the user's private key that is stored on the user hard drive.

PGP does not use a hierarchy of CAs, or any type of formal trust certificates, but instead relies on a "web of trust" in its key management approach. Each user generates and distributes his or her public key, and users sign each other's public keys, which creates a community of users who trust each other. This is different from the CA approach, where no one trusts each other; they only trust the CA.

For example, if Mark and Joe want to communicate using PGP, Mark can give his public key to Joe. Joe signs Mark's key and keeps a copy for himself. Then, Joe gives a copy of his public key to Mark so they can start communicating securely.

Later, Mark would like to communicate with Sally, but Sally does not know Mark and does not know if she can trust him. Mark sends Sally his public key, which has been signed by Joe. Sally has Joe's public key, because they have communicated before, and she trusts Joe. Because Joe signed Mark's public key, Sally now also trusts Mark and sends her public key and begins communicating with him.

So, basically, PGP is a system of "I don't know you, but my buddy Joe says you are an all right guy, so I will trust you on Joe's word."

3.17.6 Pretty Good Privacy

Each user keeps in a file, referred to as a ***key ring, a*** collection of ***public keys*** he has received from other users. Each key in that ring has a parameter that indicates the level of trust assigned to that user and the validity of that particular key.

If Steve has known Liz for many years and trusts her, he might have a higher level of trust indicated on her stored public key than on Tom's, whom he does not trust much at all.

There is also a field indicating who can sign other keys within Steve's realm of trust. If Steve receives a key from someone he doesn't know, like Kevin, and the key is signed by Liz, he can look at the field that pertains to whom he trusts to sign other people's keys. him.

If the field indicates that Steve trusts Liz enough to sign another person's key, Steve will accept Kevin's key and communicate with him because Liz is vouching for him. However, if Steve receives a key from Kevin and it is signed by untrustworthy Tom, Steve might choose to not trust Kevin and not communicate with him.

Lecture 25

S/MIME

Secure MIME (S/MIME) is a standard for encrypting and digitally signing electronic mail and for providing secure data transmissions.

S/MIME extends the MIME standard by allowing for the encryption of e-mail and attachments. The encryption and hashing algorithms can be specified by the user of the mail package, instead of having it dictated to them.

S/MIME provides confidentiality through encryption algorithms, integrity through hashing algorithms, authentication through the use of X.509 public key certificates, and non-repudiation through cryptographically signed message digests.

Pretty Good Privacy

Pretty Good Privacy (PGP) was designed by Phil Zimmerman as a freeware e-mail security program and was released in 1991. It was the first widespread public key encryption program. PGP is a complete cryptosystem that uses cryptographic protection to protect e-mail and files.

It can use RSA public key encryption for key management and use IDEA symmetric cipher for bulk encryption of data, although the user has the option of picking different types of algorithms for these functions.

PGP can provide confidentiality by using the IDEA encryption algorithm, integrity by using the MD5 hashing algorithm, authentication by using the public key certificates, and non repudiation by using cryptographically signed messages. PGP uses its own type of digital certificates rather than what is used in PKI, but they both have similar purposes.

3.18.1 Quantum Cryptography

Because of the need to always build a better algorithm, some very smart people have mixed quantum physics and cryptography, which has resulted in a system (if built correctly) that is unbreakable and where any eavesdroppers can be detected.

In traditional cryptography, we try to make it very hard for an eavesdropper to break an algorithm

and uncover a key, but we cannot detect that an eavesdropper is on the line. In quantum

cryptography, however, not only is the encryption very strong, but an eavesdropper *can be detected.*

Quantum cryptography can be carried out using various methods. So, we will walk through one version to review how all this works.

Let’s say Tom and Kathy are spies and need to send their data back and forth with the assurance it won’t be captured. To do so, they need to establish a symmetric encryption key on both ends, one for Tom and one for Kathy.

In quantum cryptography, photon polarization is commonly used to represent bits (1 or 0). Polarization is the orientation of electromagnetic waves, which is what photons are. Photons are the particles that make up light.

The electromagnetic waves have an orientation of horizontal or vertical, or left hand or right hand.

Think of a photon as a jellybean. As a jellybean flies through the air, it can be vertical (standing up straight), horizontal (lying on its back), left handed (tilted to the left), or right handed (tilted to the right). (This is just to conceptually get your head around the idea of polarization.)

Now both Kathy and Tom each have their own photon gun, which they will use to send photons (information) back and forth to each other. They also have a mapping between the polarization of a photon and a binary value.

The polarizations can be represented as vertical (|), horizontal (–), left (\), or right (/), and since we only have two values in binary, there must be some overlap.

3.18.2 Quantum Cryptography Example

In this example, a photon with a vertical (|) polarization maps to the binary value of 0. A left polarization (\) maps to 1, a right polarization (/) maps to 0, and a horizontal polarization (–) maps to 1. This mapping (or encoding) is the binary values that make up an encryption key.

Tom must have the same mapping to interpret what Kathy sends to him. Tom will use this as his map so when he receives a photon with the polarization of (\), he will write down a 1. When he receives a photon with the polarization of (|), he will write down a 0. He will do this for the whole key, and use these values as the key to decrypt a message Kathy sends him.

Note: If it helps, think about it this way. Tom receives a jellybean that is horizontal and he writes down a 1. The next jellybean he receives is tilted to the right, so he writes down 0. The next jellybean is vertical, so he writes down 1. He does this for all of the jellybeans Kathy sends his way. Now he has a string of binary values that is the encryption key his system will use to decrypt the messages Kathy sends to him.

So they both have to agree upon a key, which is the mapping between the polarization states of the photons and how those states are represented in a binary value. This happens at the beginning of a communication session over a dedicated fiber line.

Once the symmetric key is established, it can be used by Kathy and Tom to encrypt and decrypt messages that travel over a more public communication path, like the Internet. The randomness of the polarization and the complexity of creating a symmetric key in this manner help ensure that an eavesdropper will not uncover the encryption key.

Since this type of cryptography is based on quantum physics and not strictly mathematics, the sender and receiver can be confident that no eavesdropper is listening to the communication path used to establish their key and that a man-in-the-middle attack is not being carried out.

This is because, at the quantum level, even “looking” at an atom or a subatomic particle changes its attributes. This means that if there is an eavesdropper carrying out a passive attack, such as sniffing, the

receiver would know because just this simple act changes the characteristics (polarization) of the photons.

This means that as the jellybeans are sent from Kathy to Tom, if Sean tries to view the jellybeans, the ones that were traveling in a horizontal manner could be tilting left, and ones that were traveling in a vertical manner could now be traveling horizontally.

Some people in the industry think quantum cryptography is used between the U.S. White House and the Pentagon and between some military bases and defense contractor locations. This type of information is classified Top Secret by the U.S. government, and unless you know the secret handshake and have the right decoder ring, you will not be privy to this type of information.

3.19 Internet Security

The Web is not the Internet. The Web runs on top of the Internet, in a sense. The Web is the collection of HTTP servers that hold and process web sites we see. The Internet is the collection of physical devices and communication protocols used to traverse these web sites and interact with them.

The web sites look the way they do because their creators used a language that dictates the look, feel, and functionality of the page. Web browsers enable users to read web pages by enabling them to request and accept web pages via HTTP, and the user's browser converts the language (HTML, DHTML, and XML) into a format that can be viewed on the monitor. The browser is the user's window to the World Wide Web.

Browsers can understand a variety of protocols and have the capability to process many types of commands, but they do not understand them all. For those protocols or commands the user's browser does not know how to process, the user can download and install a viewer or plug-in, a modular component of code that integrates itself into the system or browser.

This is a quick and easy way to expand the functionality of the browser. However, this can cause serious security compromises, because the payload of the module can easily carry viruses and malicious software that users don't discover until it's too late.

Internet Protocols

Why do we connect to the Internet? At first, this seems a basic question, but as we dive deeper into the query, complexity creeps in. We connect to download MP3s, check e-mail, order security books, look at web sites, communicate with friends, and perform various other tasks.

But what are we really doing? We are using services provided by a computer's protocols and software. The services may be file transfers provided by FTP, remote connectivity provided by Telnet, Internet connectivity provided by HTTP, secure

connections provided by SSL, and much, much more. Without these protocols, there would be no way to even connect to the Internet.

3.19.1 HTTP

TCP/IP is the protocol suite of the Internet, and HTTP is the protocol of the Web. HTTP sits on top of TCP/IP. When a user clicks a link on a web page with his mouse, his browser uses HTTP to send a request to the web server hosting that web site. The web server finds the corresponding file to that link and sends it to the user via HTTP.

So where is TCP/IP in all of this? The TCP protocol controls the handshaking and maintains the connection between the user and the server, and the IP protocol makes sure the file is routed properly throughout the Internet to get from the web server to the user.

HTTP is a stateless protocol, which means the client and web server make and break a connection for each operation. When a user requests to view a web page, that web server finds the requested web page, presents it to the user, and then terminates the connection.

3.19.2 HTTP Secure

HTTP Secure (HTTPS) is HTTP running over SSL. (HTTP works at the application layer and SSL works at the transport layer.) Secure Sockets Layer (SSL) uses public key encryption and provides data encryption, server authentication, message integrity, and optional client authentication.

When a client accesses a web site, that web site may have both secured and public portions. The secured portion would require the user to be authenticated in some fashion. When the client goes from a public page on the web site to a secured page, the web server will start the necessary tasks to invoke SSL and protect this type of communication.

The server sends a message back to the client, indicating a secure session should be established, and the client in response sends its security parameters. The server compares those security parameters to its own until it finds a match.

This is the handshaking phase. The server authenticates to the client by sending it a digital certificate, and if the client decides to trust the server, the process continues. The server can require the client to send over a digital certificate for mutual authentication, but that is rare.

The client generates a session key and encrypts it with the server's public key. This encrypted key is sent to the web server, and they both use this symmetric key to encrypt the data they send back and forth. This is how the secure channel is established.

SSL keeps the communication path open until one of the parties requests to end the session. The session is usually ended when the client sends the server a FIN packet, which is an indication to close out the channel.

SSL requires an SSL-enabled server and browser. SSL provides security for the connection but does not offer security for the data once received. This means the data are encrypted while being transmitted, but not after the data are received by a computer.

So if a user sends bank account information to a financial institution via a connection protected by SSL, that communication path is protected, but the user must trust the financial institution that receives this information, because at this point, SSL's job is done.

The user can verify that a connection is secure by looking at the URL to see that it includes https://. The user can also check for a padlock or key icon, depending on the browser type, which is shown at the bottom corner of the browser window.

In the protocol stack, SSL lies beneath the application layer and above the network layer. This ensures SSL is not limited to specific application protocols and can still use the communication transport standards of the Internet. Different books and technical resources place SSL at different layers of the OSI model, which may seem confusing at first.

SSL is actually made up of two protocols: one works at the lower end of the session layer, and the other works at the top of the transport layer. This is why one resource will state that SSL works at the session layer and another resource puts it in the transport layer. We will use the latter definition; the SSL protocol works at the transport layer.

Although SSL is almost always used with HTTP, it can also be used with other types of protocols. So if you see a common protocol that is followed by an *s, that protocol is* using SSL to encrypt its data.

SSL is currently at version 3.0. Since SSL was developed by Netscape, it is not an open-community protocol. This means the technology community cannot easily extend SSL to interoperate and expand in its functionality. If a protocol is proprietary in nature, as SSL is, the technology community cannot directly change its specifications and functionality.

If the protocol is an open-community protocol, then its specifications can be modified by individuals within the community to expand what it can do and what technologies it can work with. So the open-community version of SSL is Transport Layer Security (TLS). The differences between SSL 3.0 and TLS is slight, but TLS is more extensible and is backward compatible with SSL.

Lecture 26

Secure HTTP

Though their names are very similar, there is a difference between Secure HTTP (S-HTTP) and HTTP Secure (HTTPS). S-HTTP is a technology that protects each message sent between two computers, while HTTPS protects the communication channel between two computers, messages and all.

HTTPS uses SSL/TLS and HTTP to provide a protected circuit between a client and server. So, S-HTTP is used if an individual message needs to be encrypted, but if all information that passes between two computers must be encrypted, then HTTPS is used, which is SSL over HTTP.

3.19.3 Secure Electronic Transaction

Secure Electronic Transaction (SET) is a security technology proposed by Visa and MasterCard to allow for more secure credit card transaction possibilities than what is currently available. SET has been waiting in the wings for full implementation and acceptance as a standard for quite some time.

Although SET provides an effective way of transmitting credit card information, businesses and users do not see it as efficient because it requires more parties to coordinate their efforts, more software installation and configuration for each entity involved, and more effort and cost than the widely used SSL method.

SET is a cryptographic protocol and infrastructure developed to send encrypted credit card numbers over the Internet. The following entities would be involved with a SET transaction, which would require each of them to upgrade their software, and possibly their hardware:

• **Issuer** (cardholder's bank) The financial institution that provides a credit card to the individual.

• **Cardholder** The individual authorized to use a credit card.

• **Merchant** The entity providing goods.

• **Acquirer** (merchant's bank) The financial institution that processes payment cards.

• **Payment gateway** This processes the merchant payment. It may be an acquirer.

To use SET, a user must enter his credit card number into his electronic wallet software. This information is stored on the user's hard drive or on a smart card. The software then creates a public key and a private key that are used specifically for encrypting financial information before it is sent.

Let's say Ahmad wants to use his electronic credit card to buy his brother a gift from a web site. When he finds the perfect gift and decides to purchase it, he sends his encrypted credit card information to the merchant's web server. The merchant does not decrypt the credit card information, but instead digitally signs it and sends it on to its processing bank.

At the bank, the payment server decrypts the information, verifies that Ahmad has the necessary funds, and transfers the funds from Ahmad's account to the merchant's account. Then the payment server sends a message to the merchant telling it to finish the transaction, and a receipt is sent to Ahmad and the merchant.

At each step, an entity verifies a digital signature of the sender and digitally signs the information before it is sent to the next entity involved in the process. This would require all entities to have digital certificates and to participate in a PKI.

This is basically a very secure way of doing business over the Internet, but today everyone seems to be happy enough with the security SSL provides. They do not feel motivated enough to move to a different and more encompassing technology.

The lack of motivation comes from all of the changes that would need to take place to our current processes and the amount of money these changes would require.

3.19.4 Cookies

Cookies are text files that a browser maintains on a user's hard drive. Cookies have different uses, and some are used for demographic and advertising information. As a user travels from site to site on the Internet, the sites could be writing data to the cookies stored on the user's system.

The sites can keep track of the user's browsing and spending habits and the user's specific customization for certain sites. For example, if Bilal goes to mainly gardening sites on the Internet, those sites will most likely record this information and the types of items in which he shows most interest.

Then, when Bilal returns to one of the same or similar sites, it will retrieve his cookies, find he has shown interest in gardening books in the past, and present him with its line of gardening books.

This increases the likelihood of Bilal purchasing a book of his liking. This is a way of zeroing in on the right marketing tactics for the right person. The servers at the web site determine how cookies are actually used.

When a user adds items to his shopping cart on a site, such data are usually added to a cookie. Then, when the user is ready to check out and pay for his items, all the data in this specific cookie are extracted and the totals are added.

As stated before, HTTP is a stateless protocol, meaning a web server has no memory of any prior connections. This is one reason to use cookies. They retain the memory between HTTP connections by saving prior connection data to the client's computer.

For example, if you carry out your banking activities online, your bank's web server keeps track of your activities through the use of cookies. When you first go to its site and are looking at public information, such as branch locations, hours of operation, and CD rates, no confidential information is being transferred back and forth.

Once you make a request to access your bank account, the web server sets up an SSL connection and requires you to send credentials.

Once you send your credentials and are authenticated, the server generates a cookie with your authentication and account information in it. The server sends it to your browser, which either saves it to your hard drive or keeps it in memory.

So, suppose you look at your checking account, do some work there, and then request to view your savings account information. The web server sends a request to see if you have been properly authenticated for this activity by checking your cookie.

Most online banking software also periodically requests your cookie, to ensure no man-in-the-middle attacks are going on and that someone else has not hijacked the session. It is also important to ensure that secure connections time out.

This is why cookies have timestamps within them. If you have ever worked on a site that has an SSL connection set up for you and it required you to reauthenticate, the reason is that your session has been idle for a while and, instead of leaving a secure connection open, the web server software closed it out.

A majority of the data within a cookie is meaningless to any entities other than the servers at specific sites, but some cookies can contain usernames and passwords for different accounts on the Internet.

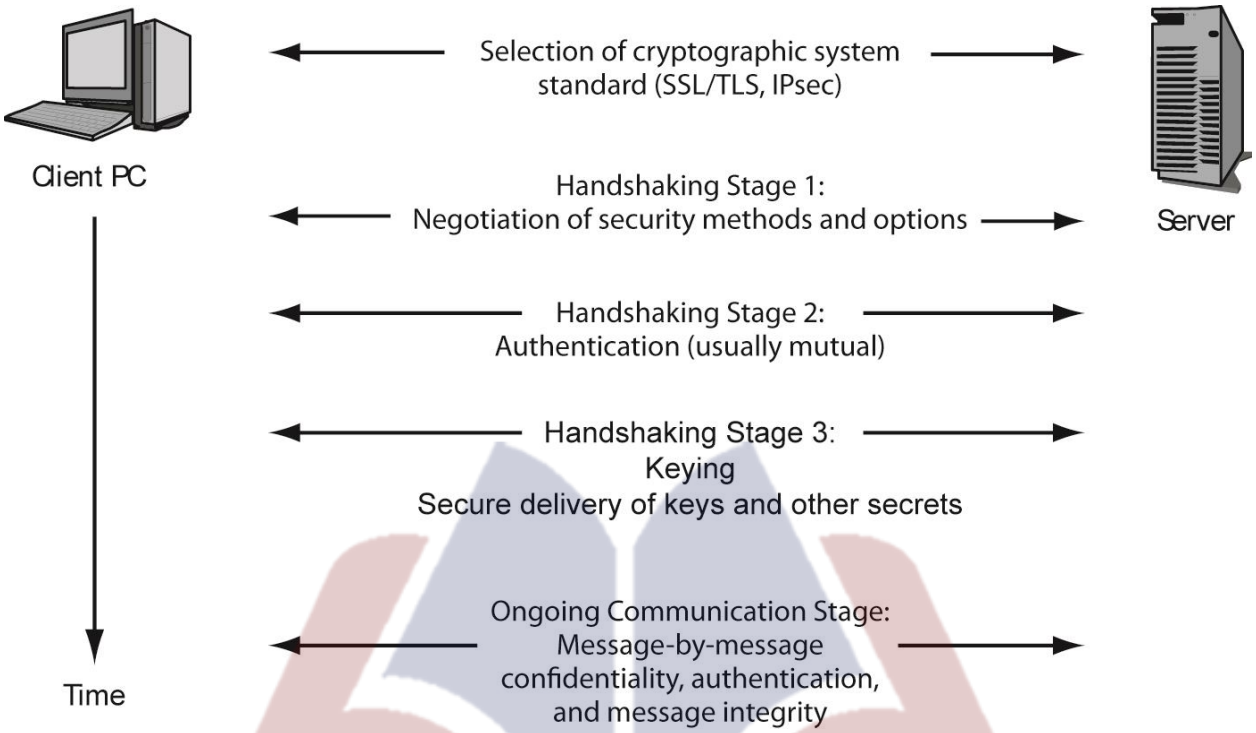
The cookies that contain sensitive information should be encrypted by the server at the site that distributes them, but this does not always happen, and a nosey attacker could find this data on the user's hard drive and attempt to use it for mischievous activity.

Some people who live on the paranoid side of life do not allow cookies to be downloaded to their systems (controlled through browser security controls). Although this provides a high level of protection against different types of cookie abuse, it also reduces their functionality on the Internet.

Some sites require cookies because there is specific data within the cookies that the site must utilize correctly in order to provide the user with the services he requested.

Lecture 27

3.19.5 Cryptographic System



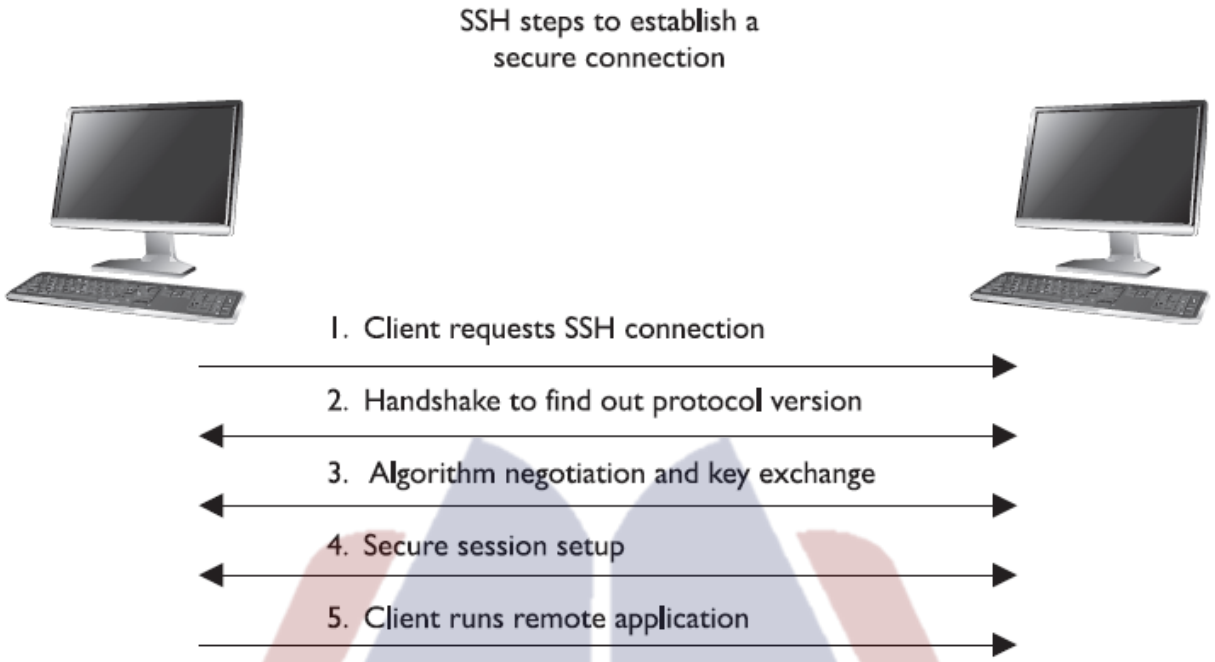
3.19.6 Secure Shell

Secure Shell (SSH) functions as a type of tunneling mechanism that provides terminal-like access to remote computers. SSH is a program and a protocol that can be used to log into another computer over a network.

For example, the program can let Paul, who is on computer A, access computer B's files, run applications on computer B, and retrieve files from computer B without ever physically touching that computer. SSH provides authentication and secure transmission over vulnerable channels like the Internet.

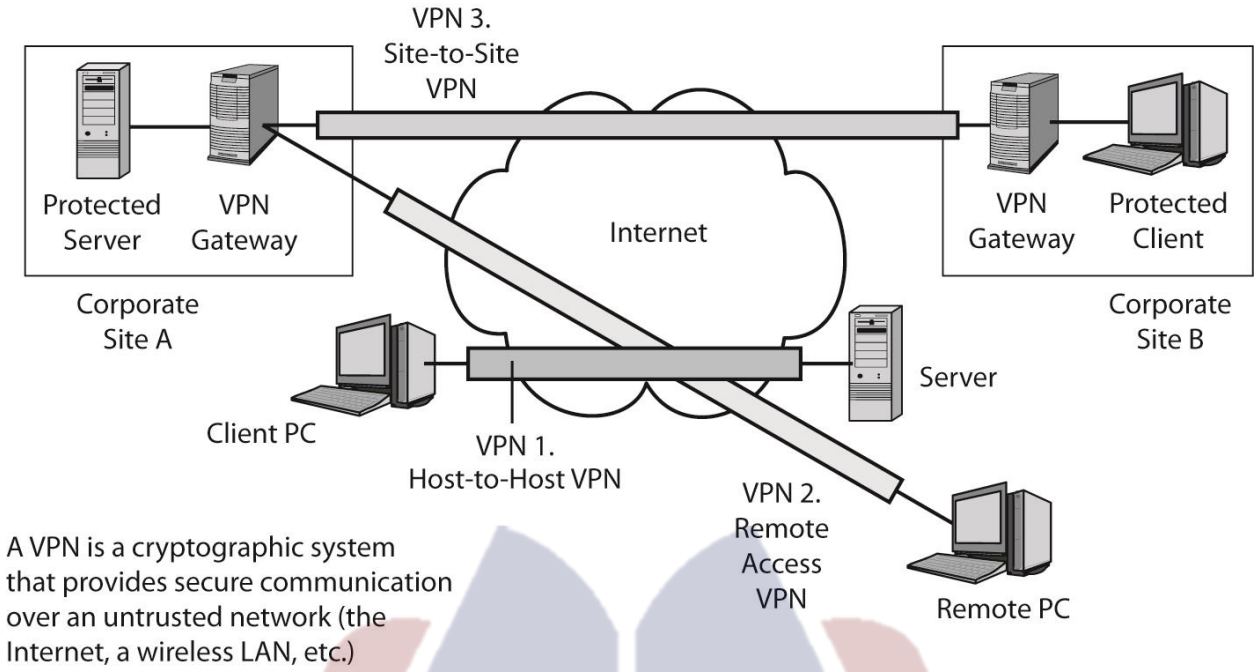
SSH should be used instead of Telnet, FTP, rlogin, rexec, or rsh, which provide the same type of functionality SSH offers but in a much less secure manner. SSH is a program and a set of protocols that work together to provide a secure tunnel between two computers.

The two computers go through a handshaking process and exchange (via Diffie-Hellman) a session key that will be used during the session to encrypt and protect the data sent. The steps of an SSH connection are outlined in the Figure.



Once the handshake takes place and a secure channel is established, the two computers have a pathway to exchange data with the assurance that the information will be encrypted and its integrity will be protected.

3.19.7 Virtual Private Networks (VPNs)



3.19.8 IPSEC (Internet Protocol Security)

The ***Internet Protocol Security (IPSec)*** protocol suite provides a method of setting up a secure channel for protected data exchange between two devices. The devices that share this secure channel can be two servers, two routers, a workstation and a server, or two gateways between different networks.

IPSec is a widely accepted standard for providing

network layer protection. It can be more flexible and less expensive than end-to-end and link encryption methods.

IPSec has strong encryption and authentication methods, and although it can be used to enable tunneled communication between two computers, it is usually employed to establish virtual private networks (VPNs) among networks across the Internet.

IPSec is not a strict protocol that dictates the type of algorithm, keys, and authentication method to use. Rather, it is an open, modular framework that provides a lot of flexibility for companies when they choose to use this type of technology.

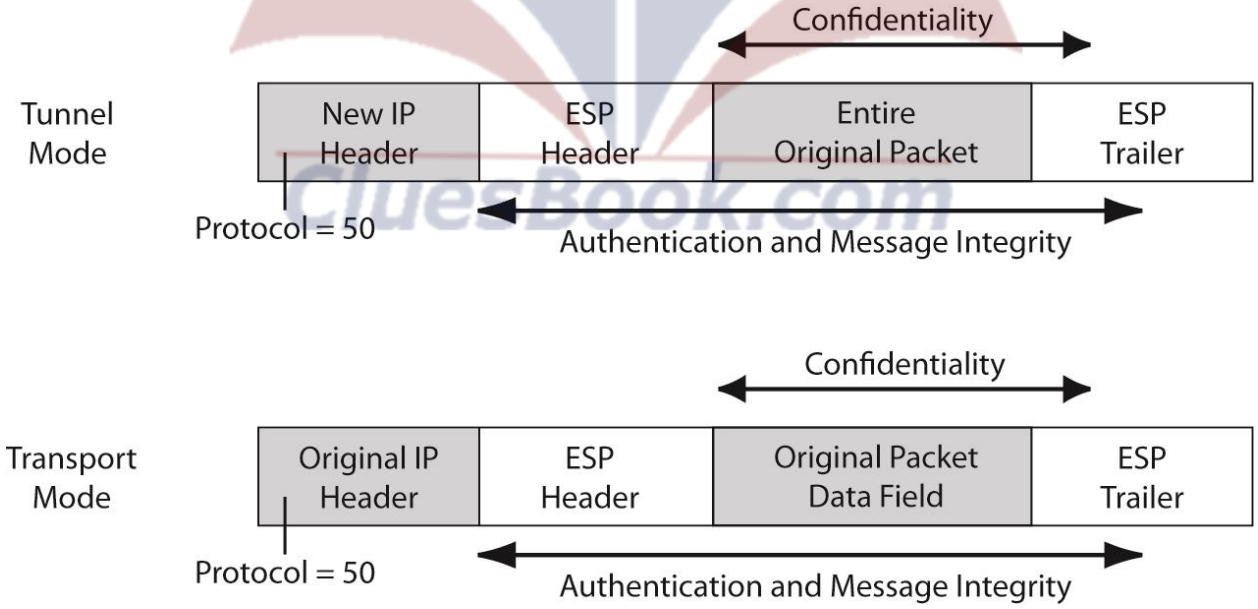
IPSec uses two basic security protocols: ***Authentication Header (AH)*** *and* ***Encapsulating Security Payload (ESP)***.

AH is the authenticating protocol, and ***ESP*** is an authenticating and encrypting protocol that uses cryptographic mechanisms to provide source authentication, confidentiality, and message integrity.

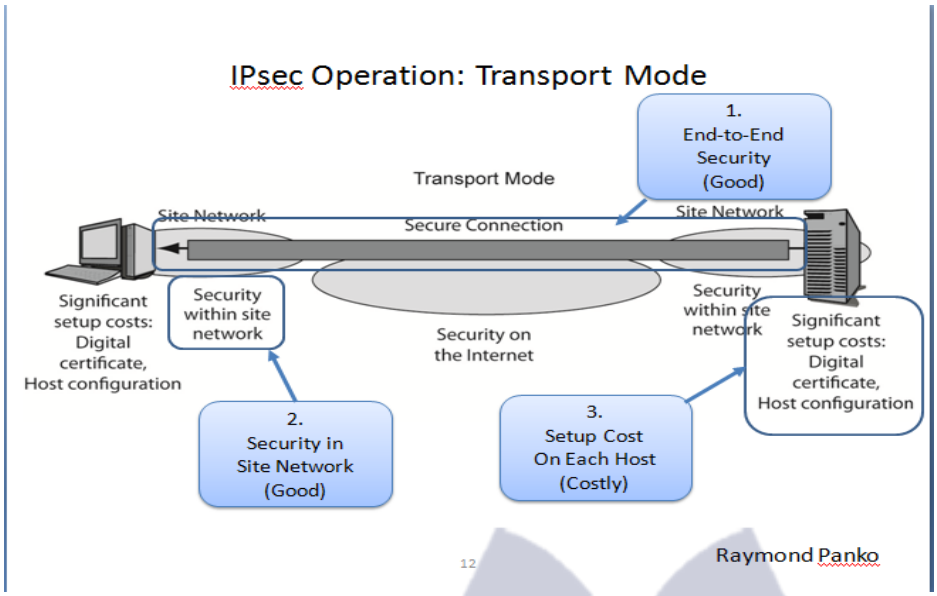
IPSec can work in one of two modes: **transport mode**, in which the payload of the message is protected, and **tunnel mode**, in which the payload and the routing and header information are protected.

ESP in transport mode encrypts the actual message information so it cannot be sniffed and uncovered by an unauthorized entity. Tunnel mode provides a higher level of protection by also protecting the header and trailer data an attacker may find useful.

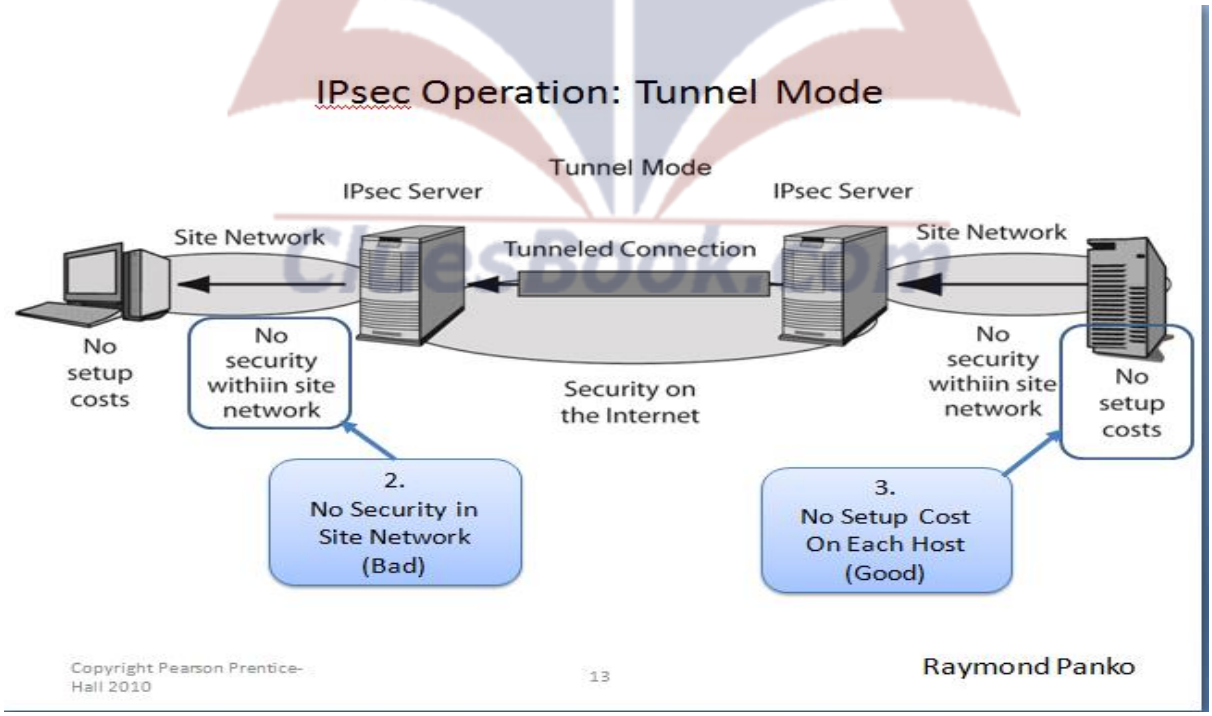
IPsec Encapsulating Security Payload (ESP) Header and Trailer in Transport and Tunnel Modes



IPsec Operation: Transport Mode



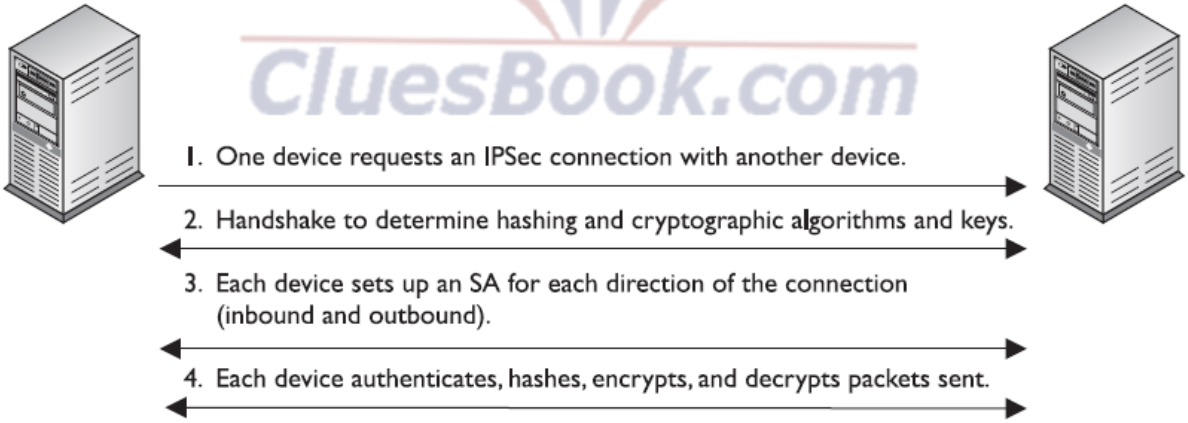
IPsec Operation: Tunnel Mode



Comparing IPsec Transport and Tunnel Modes

Characteristic	Transport Mode	Tunnel Mode
Uses an IPsec VPN Gateway?	No	Yes
Cryptographic Protection	All the way from the source host to the destination host, including the Internet and the two site networks.	Only over the Internet between the IPsec gateways. Not within the two site networks.
Setup Costs	High. Setup requires the creation of a digital certificate for each client and significant configuration work.	Low. Only the IPsec gateways must implement IPsec, so only they need digital certificates and need to be configured.

The Figure shows the high-level view of the steps of setting up an IPSec connection.

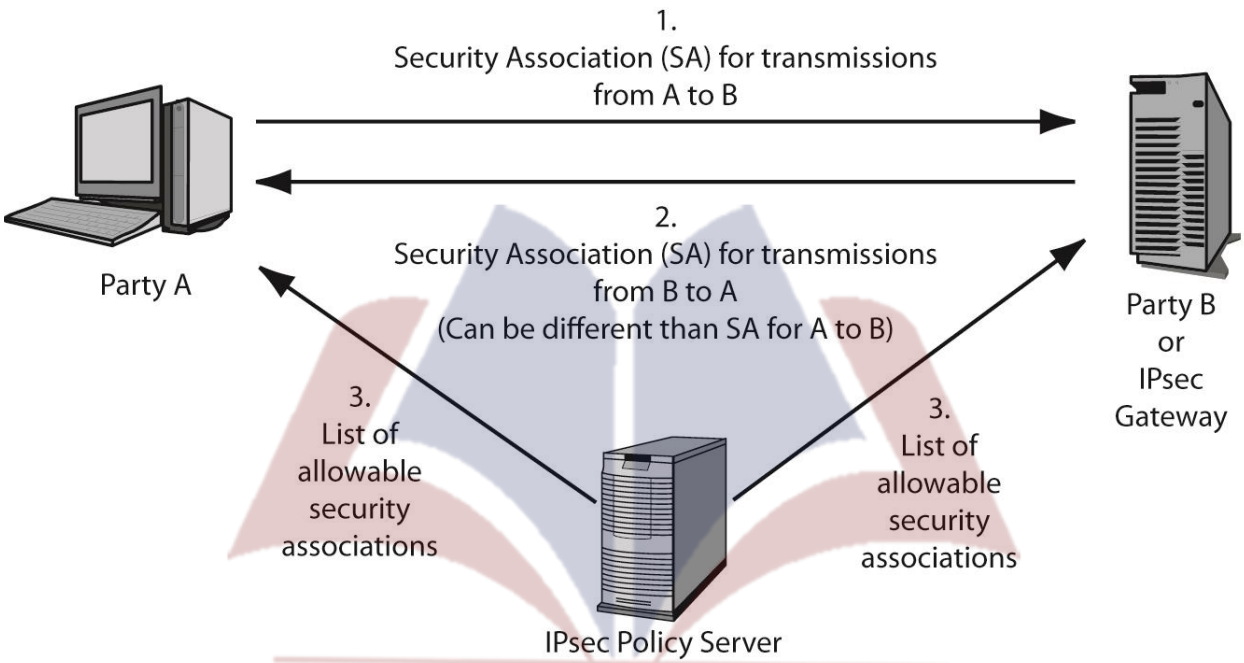


Each device will have at least one ***security association (SA) for each secure connection*** it uses. The SA, which is critical to the IPSec architecture, is a record of the configurations the device needs to support an IPSec connection.

When two devices complete their handshaking process, which means they have agreed upon a long list of parameters they will use to communicate, these data must be recorded and stored somewhere, which is in the SA. The SA can contain the authentication and encryption keys, the agreed-upon algorithms, the key lifetime, and the source IP address.

IPsec Security Associations

An IPsec security association (SA) is an agreement about what security methods and options the two hosts or two IPsec gateways will use during their communication.

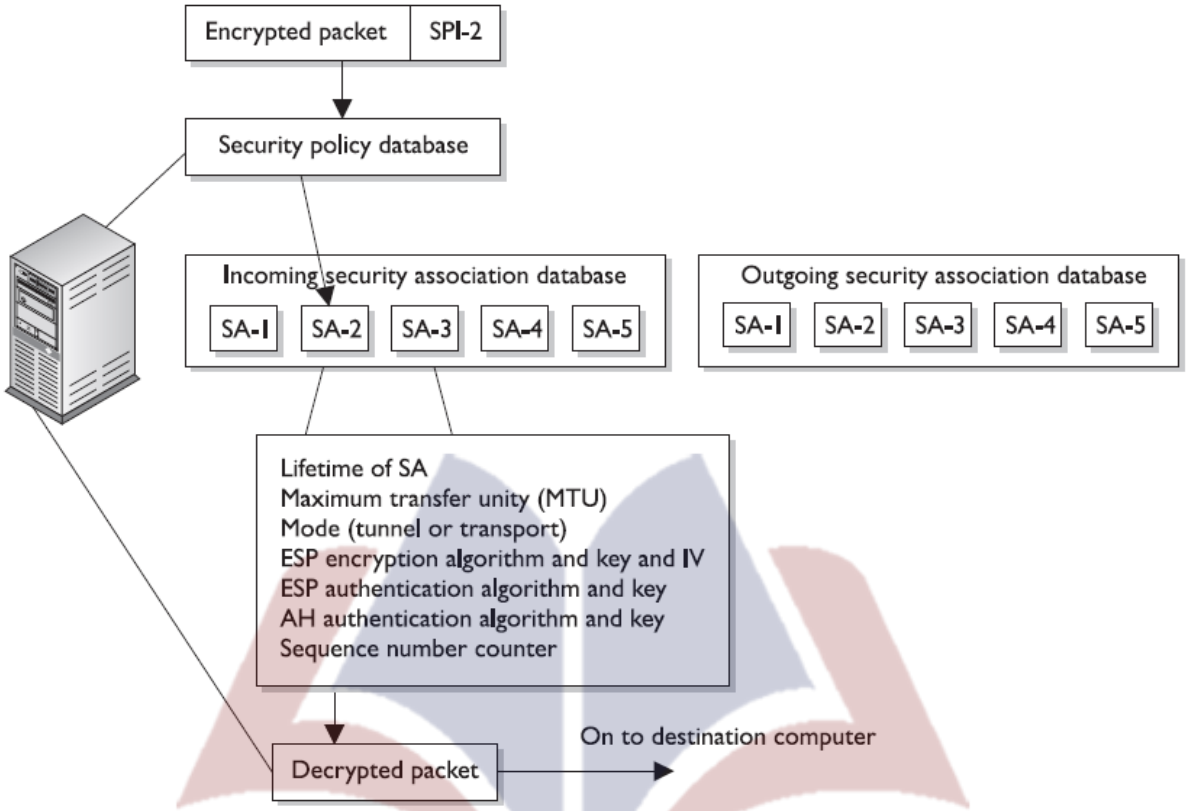


When a device receives a packet via the IPSec protocol, it is the SA that tells the device what to do with the packet. So if device B receives a packet from device C via IPSec, device B will look to the corresponding SA to tell it how to decrypt the packet, how to properly authenticate the source of the packet, which key to use, and how to reply to the message if necessary.

SAs are directional, so a device will have one SA for outbound traffic and a different SA for inbound traffic for each individual communication channel. If a device is connecting to three devices, it will have at least six SAs, one for each inbound and outbound connection per remote device.

So how can a device keep all of these SAs organized and ensure that the right SA is invoked for the right connection? With the mighty security parameter index (SPI), that's how.

Each device has an SPI that keeps track of the different SAs and tells the device which one is appropriate to invoke for the different packets it receives. The SPI value is in the header of an IPSec packet, and the device reads this value to tell it which SA to consult, as depicted in the Figure.



IPSec can authenticate the sending devices of the packet by using MAC (covered in the earlier section, "The One-Way Hash"). The ESP protocol can provide authentication, integrity, and confidentiality if the devices are configured for this type of functionality. So if a company just needs to make sure it knows the source of the sender and must be assured of the integrity of the packets, it would choose to use AH.

If the company would like to use these services and also have confidentiality, it would use the ESP protocol because it provides encryption functionality. In most cases, the reason ESP is employed is because the company must set up a secure VPN connection.

It may seem odd to have two different protocols that provide overlapping functionality. AH provides authentication and integrity, and ESP can provide those two functions

and confidentiality. Why even bother with AH then? In most cases, the reason has to do with whether the environment is using network address translation (NAT).

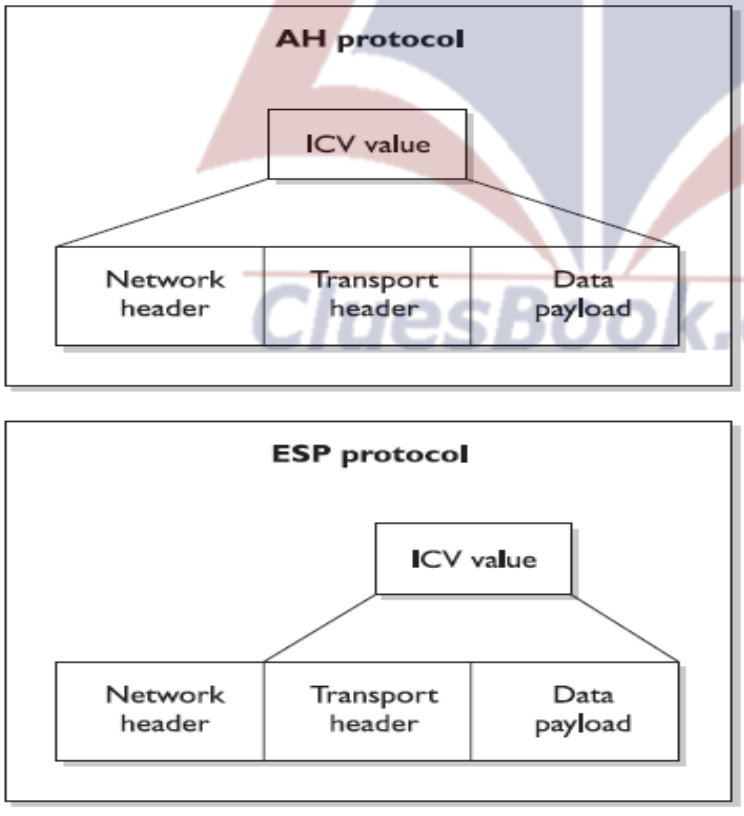
IPSec will generate an integrity check value (ICV), which is really the same thing as a MAC value, over a portion of the packet. Remember that the sender and receiver generate their own values. In IPSec, it is called an ICV value. The receiver compares his ICV value with the one sent by the sender.

If the values match, the receiver can be assured the packet has not been modified during transmission. If the values are different, the packet has been altered and the receiver discards the packet.

The AH protocol calculates this ICV over the data payload, transport, and network headers. If the packet then goes through a NAT device, the NAT device changes the IP address of the packet. That is its job.

This means a portion of the data (network header) that was included to calculate the ICV value has now changed, and the receiver will generate an ICV value that is different from the one sent with the packet, which means the packet will be discarded automatically.

The ESP protocol follows similar steps, except it does not include the network header portion when calculating its ICV value. When the NAT device changes the IP address, it will not affect the receiver's ICV value because it does not include the network header when calculating the ICV. The differences are shown in the Figure.



AH and ESP use different portions of the packet to calculate the ICVs.

Because IPSec is a framework, it does not dictate which hashing and encryption algorithms are to be used or how keys are to be exchanged between devices. Key management

can be handled manually or automated by a key management protocol. The de facto standard for IPSec is to use Internet Key Exchange (IKE), which is a combination of the ISAKMP and OAKLEY protocols.

The Internet Security Association and Key Management

Protocol (ISAKMP) is a key exchange architecture that is independent of the type of keying mechanisms used. Basically, ISAKMP provides the framework of what can be negotiated to set up an IPSec connection (algorithms, protocols, modes, keys). The **OAKLEY** protocol is the one that carries out the negotiation process.

You can think of ISAKMP as providing the playing field (the infrastructure) and OAKLEY as the guy running up and down the playing field (carrying out the steps of the negotiation).

IPSec is very complex with all of its components and possible configurations. This complexity is what provides for a great degree of flexibility, because a company has many different configuration choices to achieve just the right level of protection.

IP Security (IPsec) versus SSL/TLS

	SSL/TLS	IPsec
Cryptographic security standard	Yes	Yes
Cryptographic security protections	Good	Gold Standard
Supports central management	No	Yes
Complexity and expense	Lower	Higher
Layer of operation	Transport	Internet
Transparently protects all higher-layer traffic	No	Yes
Works with IPv4 and IPv6	NA	Yes
Modes of operation	NA	Transport, Tunnel

Raymond Panko

3.20 Attacks: Passive & Active

Eavesdropping and sniffing data as it passes over a network are considered **passive attacks** because the attacker is not affecting the protocol, algorithm, key, message, or any parts of the encryption system. Passive attacks are hard to detect, so in most cases methods are put in place to try to prevent them rather than to detect and stop them.

Altering messages, modifying system files, and masquerading as another individual are acts that are considered ***active attacks because the attacker is actually doing something*** instead of sitting back and gathering data. Passive attacks are usually used to gain information prior to carrying out an active attack. The following sections address some active attacks that relate to cryptography.

Cipher-Only Attacks

In this type of attack, the attacker has the cipher text of several messages. Each of the messages has been encrypted using the same encryption algorithm. The attacker's goal is to discover the key used in the encryption process. Once the attacker figures out the key, he can decrypt all other messages encrypted with the same key.

A cipher text-only attack is the most common type of active attack because it is very easy to get cipher text by sniffing someone's traffic, but it is the hardest attack to actually be successful at because the attacker has so little information about the encryption process.

Known-Plaintext Attacks

In known-plaintext attacks, the attacker has the plaintext and corresponding ciphertext of one or more messages. Again, the goal is to discover the key used to encrypt the messages so other messages can be deciphered and read.

Messages usually start with the same type of beginning and close with the same type of ending. An attacker might know that each message a general sends out to his commanders always starts with certain greetings and ends with specific salutations and the general's name and contact information.

In this instance, the attacker has some of the plaintext (the data that are the same on each message) and can capture an encrypted message, and therefore capture the cipher text.

Once a few pieces of the puzzle are discovered, the rest is accomplished by reverse-engineering, frequency analysis, and brute force attempts. Known-plaintext attacks were used by the United States against the Germans and the Japanese during World War II.

Chosen-Plaintext Attacks

In chosen-plaintext attacks, the attacker has the plaintext and cipher text, but can choose the plaintext that gets encrypted to see the corresponding cipher text. This gives him more power and possibly a deeper understanding of the way the encryption process works so he can gather more information about the key being used.

Once the key is discovered, other messages encrypted with that key can be decrypted.

How would this be carried out? I can e-mail a message to you that I think you not only will believe, but that you will also panic about, encrypt, and send to someone else. Suppose I send you an e-mail that states, "The meaning of life is 42."

You may think you have received an important piece of information that should be concealed from others, everyone except your friend Bob, of course. So you encrypt my message and send it to Bob. Meanwhile I am sniffing your traffic and now have a copy of the plaintext of the message, because I wrote it, and a copy of the cipher text.

In chosen-cipher text attacks, the attacker can choose the cipher text to be decrypted and has access to the resulting decrypted plaintext. Again, the goal is to figure out the key.

This is a harder attack to carry out compared to the previously mentioned attacks, and the attacker may need to have control of the system that contains the cryptosystem.

Adaptive Attacks

NOTE: All of these attacks have a derivative form, the names of which are the same except for putting the word "adaptive" in front of them: such as adaptive chosen-plaintext and adaptive chosen-ciphertext.

What this means is that the attacker can carry out one of these attacks and, depending upon what he gleaned from that first attack, modify his next attack. This is the process of reverse-engineering or cryptanalysis attacks: using what you learned to improve your next attack.

Differential Cryptanalysis

This type of attack also has the goal of uncovering the key that was used for encryption purposes. This attack looks at cipher text pairs generated by encryption of plaintext pairs with specific differences and analyzes the effect and result of those differences.

One such attack was invented in 1990 as an attack against DES, and it turned out to be an effective and successful attack against DES and other block algorithms.

The attacker takes two messages of plaintext and follows the changes that take place to the blocks as they go through the different S-boxes. (Each message is being encrypted with the same key.) The differences identified in the resulting cipher text values are used to map probability values to different possible key values.

The attacker continues this process with several more sets of messages and reviews the common key probability values. One key will continue to show itself as the most probable key used in the encryption processes. Since the attacker chooses the different plaintext messages for this attack, it is considered to be a type of chosen-plaintext attack.

Linear Cryptanalysis

Linear cryptanalysis is another type of attack that carries out functions to identify the highest probability of a specific key employed during the encryption process using a block algorithm.

The attacker carries out a known-plaintext attack on several different messages encrypted with the same key. The more messages the attacker can use and put through this type of attack, the higher the confidence level in the probability of a specific key value.

The attacker evaluates the input and output values for each S-box. He evaluates the probability of input values ending up in a specific combination.

Identifying specific output combinations allows him to assign probability values to different keys until one shows a continual pattern of having the highest probability.

Side-Channel Attacks

All of the attacks we have covered thus far have been based mainly on the mathematics of cryptography. Using plaintext and cipher text involves high-powered mathematical tools that are needed to uncover the key used in the encryption process.

But what if we took a different approach? Let's say we see something that looks like a duck, walks like a duck, sounds like a duck, swims in water, and eats bugs and small fish. We could confidently conclude that this is a duck. Similarly, in cryptography, we can review facts and infer the value of an encryption key.

For example, we could detect how much power consumption is used for encryption and decryption (the fluctuation of electronic voltage). We could also intercept the radiation emissions released and then calculate how long the processes take.

Looking around the cryptosystem, or its attributes and characteristics, is different from looking into the cryptosystem and trying to defeat it through mathematical computations.

If I want to figure out what you do for a living, but I don't want you to know I am doing this type of reconnaissance work, I won't ask you directly. Instead, I will find out when you go to work and come home, the types of clothing you wear, the items you carry, whom you talk to . . . or I can just follow you to work.

These are examples of side channels. So, in cryptography, gathering "outside" information with the goal of uncovering the encryption key is just another way of attacking a cryptosystem.

An attacker could measure power consumption, radiation emissions, and the time it takes for certain types of data processing. With this information, he can work backward by reverse-engineering the process to uncover an encryption key or sensitive data.

A power attack reviews the amount of heat released. This type of attack has been successful in uncovering confidential information from smart cards. In 1995, RSA private keys were uncovered by measuring the relative time cryptographic operations took.

Replay Attacks

A big concern in distributed environments is the **replay attack,** in which an attacker captures some type of data and resubmits it with the hopes of fooling the receiving device into thinking it is legitimate information.

Many times, the data captured and resubmitted are authentication information, and the attacker is trying to authenticate himself as someone else to gain unauthorized access.

Timestamps and sequence numbers are two countermeasures to replay attacks. Packets can contain sequence numbers, so each machine will expect a specific number on each receiving packet. If a packet has a sequence number that has been previously used, this is an indication of a replay attack.

Packets can also be time stamped. A threshold can be set on each computer to only accept packets within a certain timeframe. If a packet is received that is past this threshold, it can help identify a replay attack.

Algebraic Attacks

Algebraic attacks analyze the vulnerabilities in the mathematics used within the algorithm and exploit the intrinsic algebraic structure. For instance, attacks on the "textbook" version of the RSA cryptosystem exploit properties of the algorithm such as the fact that the encryption of a raw "0" message is "0".

Analytical Attacks

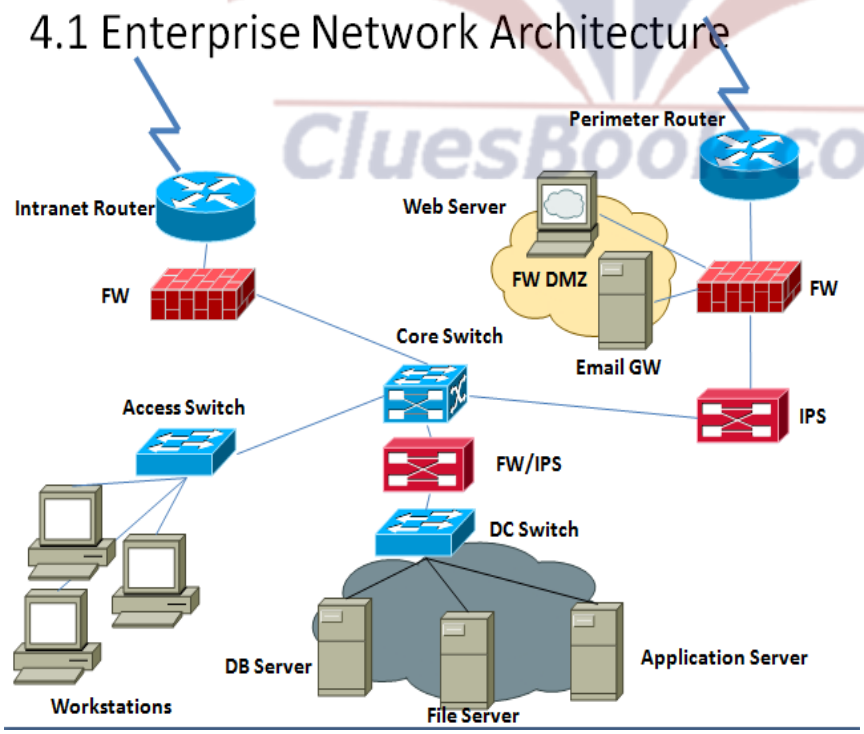
Analytic attacks identify algorithm structural weaknesses or flaws, as opposed to brute force attacks, which simply exhaust all possibilities without respect to the specific properties of the algorithm. Examples = Double DES attack and RSA factoring attack.

Statistical Attacks

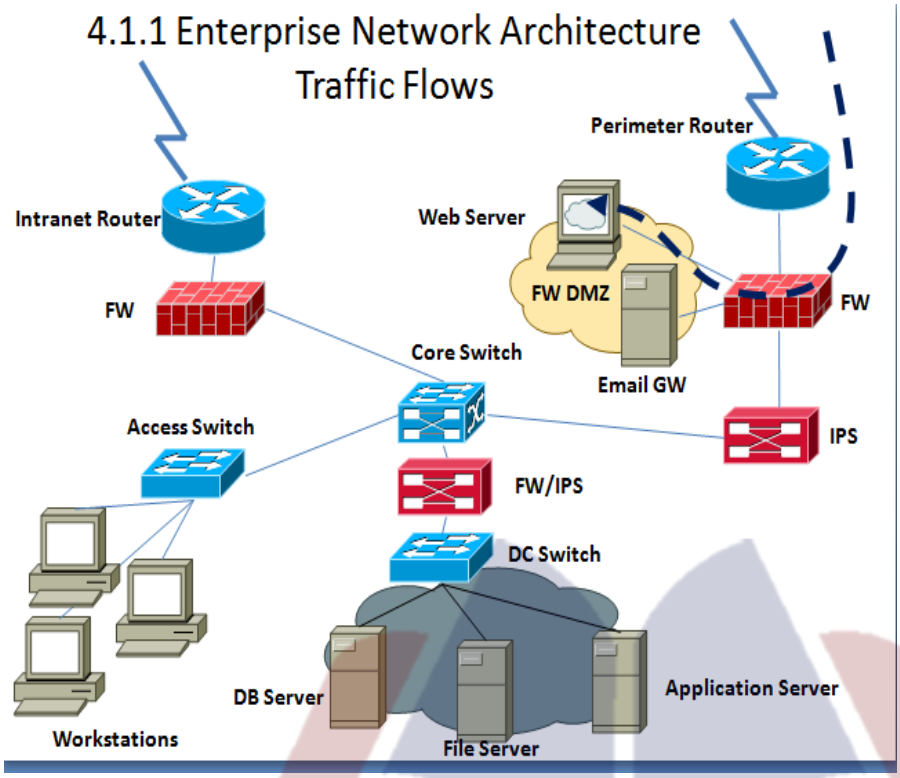
Statistical attacks identify statistical weaknesses in algorithm design for exploitation — for example, if statistical patterns are identified, as in the number of 0's compared to the number of 1's.

Lecture 29

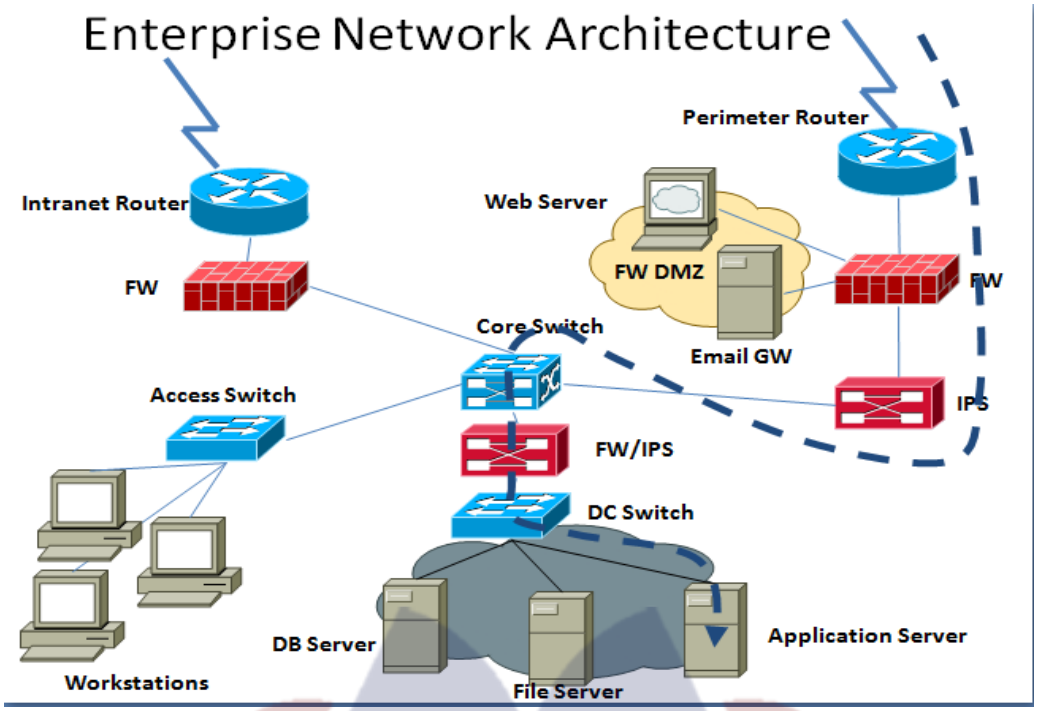
Enterprise Network Architecture



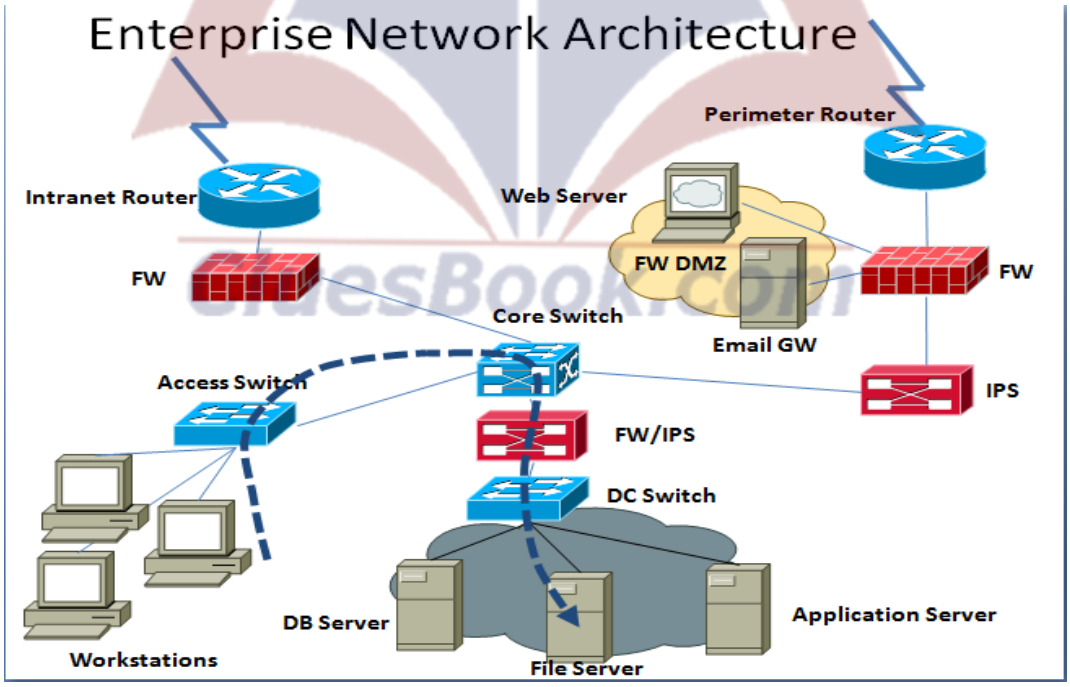
4.1.1 Enterprise Network Architecture

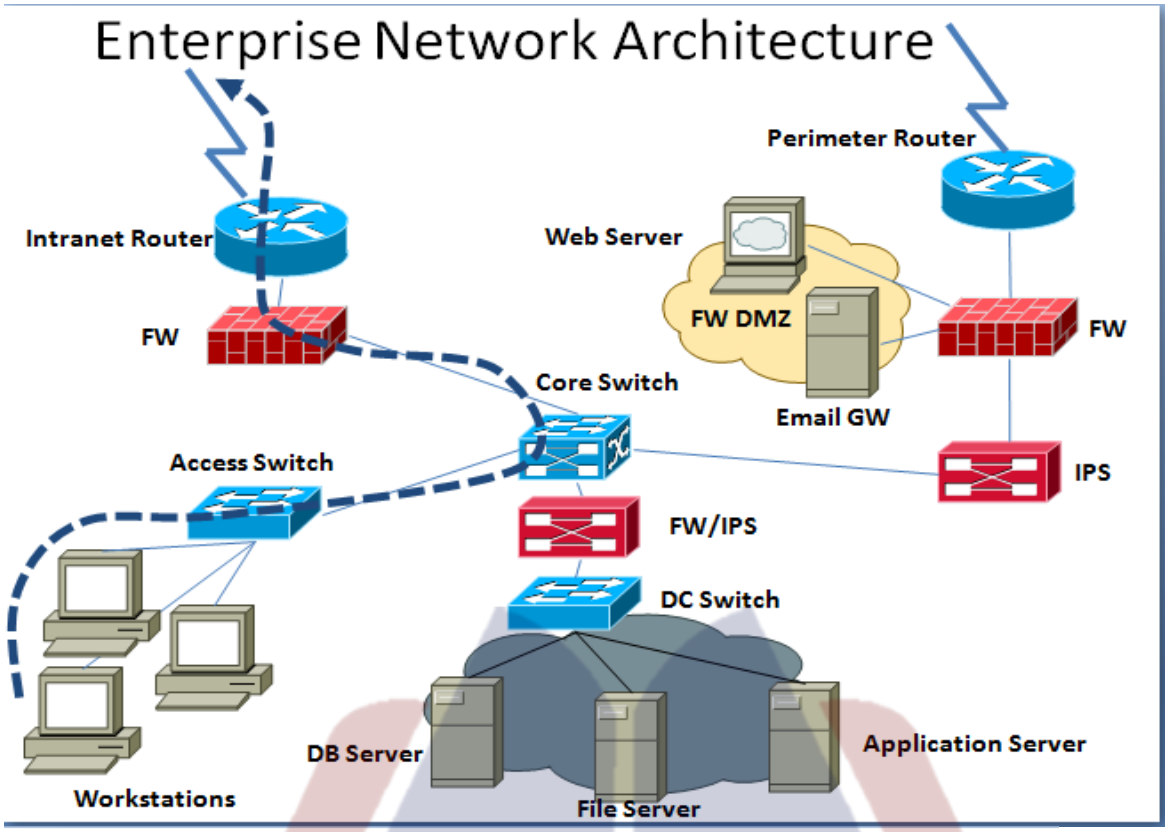


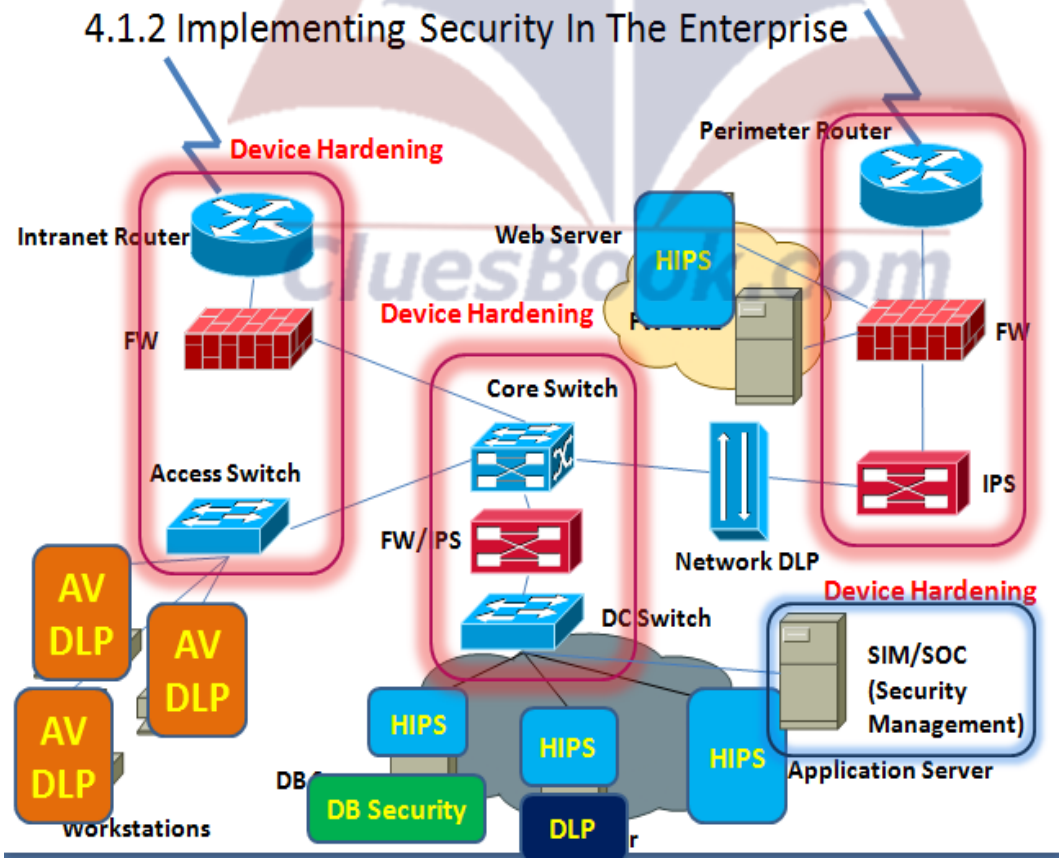
Enterprise Network Architecture

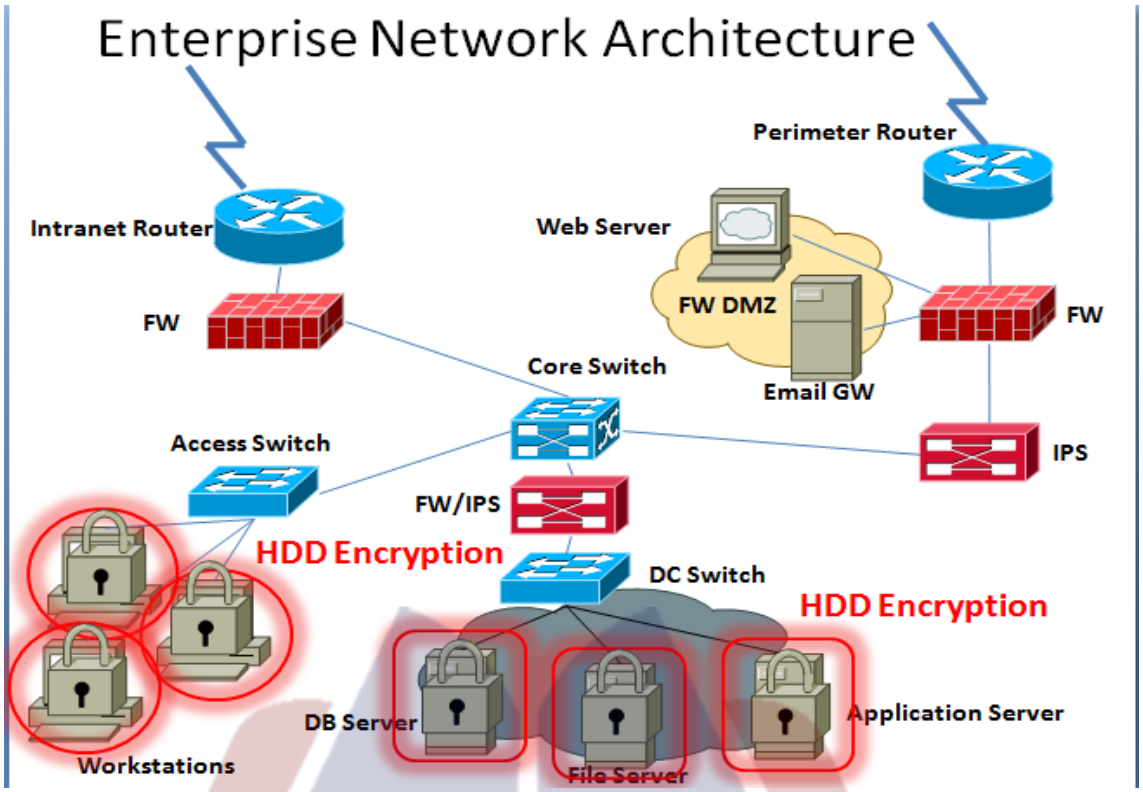


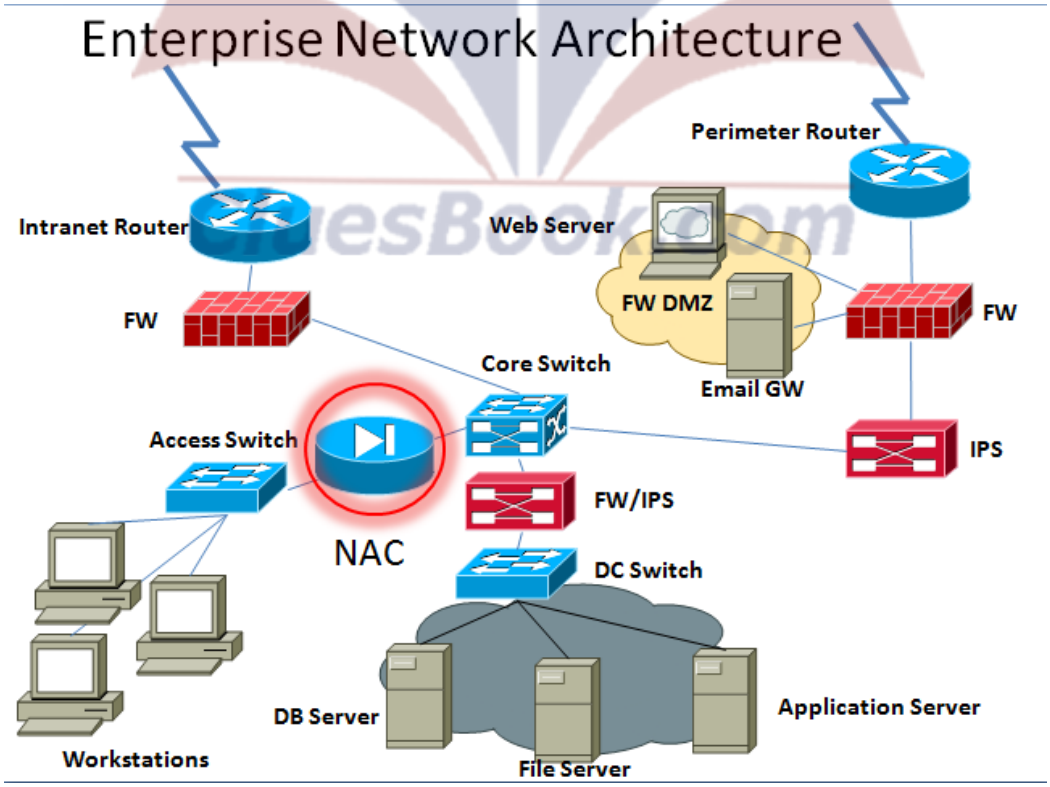
Enterprise Network Architecture

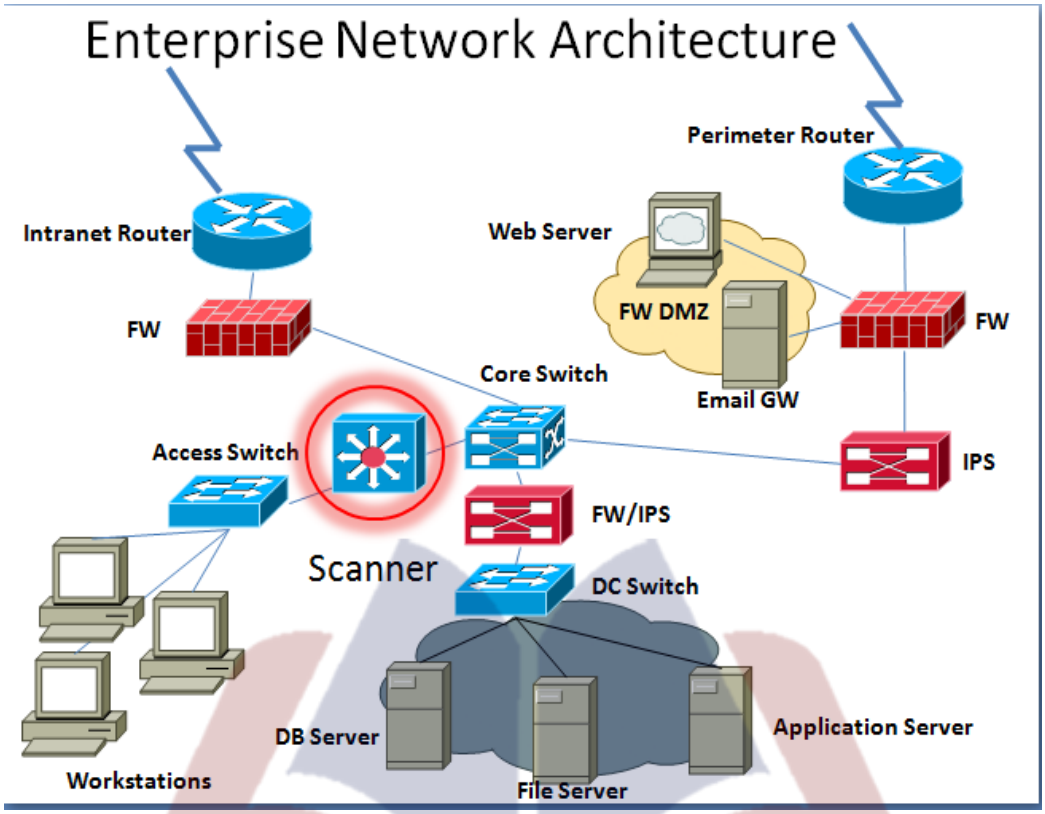


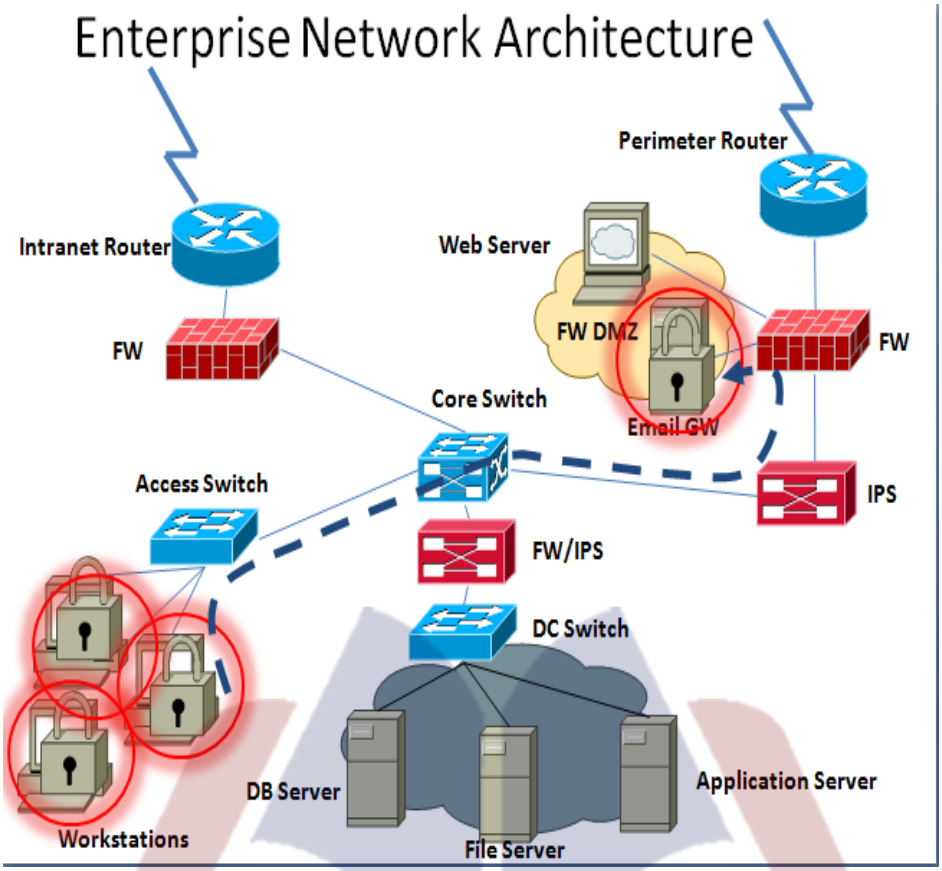

Enterprise Network Architecture
Perimeter Router
Intranet Router
Web Server
FW DMZ
FW
FW
Core Switch
Email GW
Access Switch
IPS
FW/IPS
DC Switch
DB Server
Application Server
Workstations
File Server

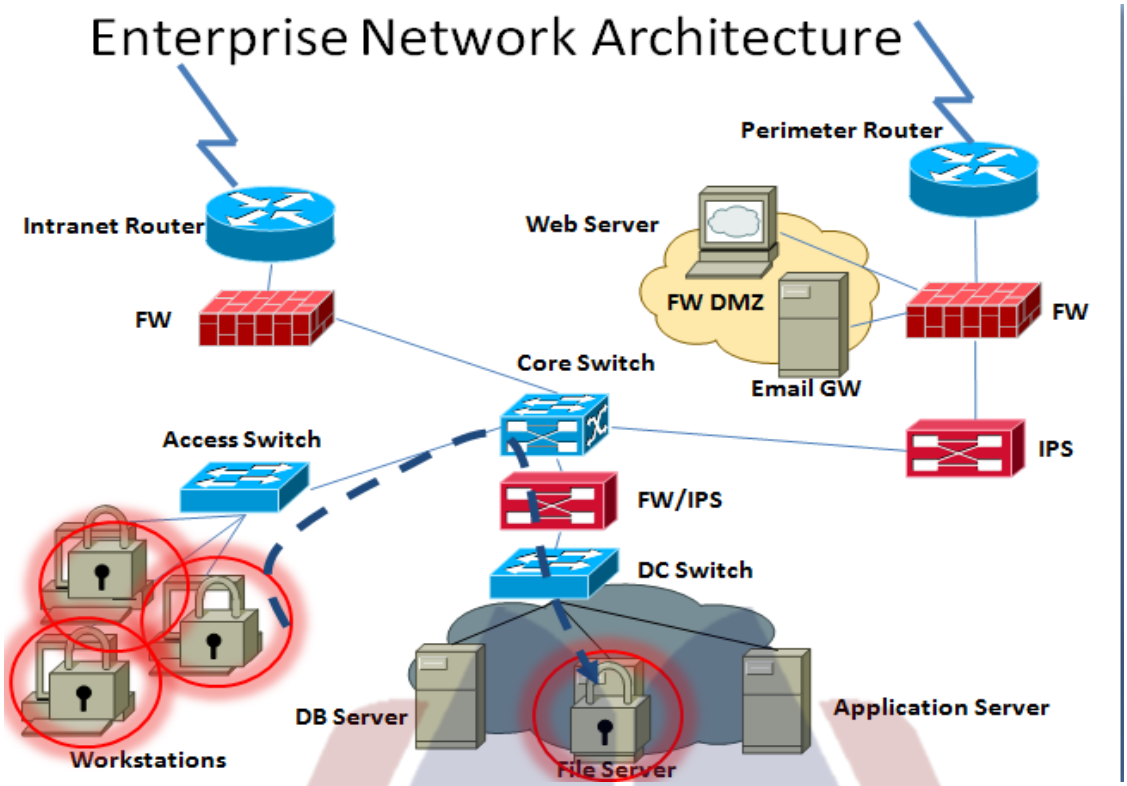

4.1.2 Implementing Security In The Enterprise
Device Hardening
Perimeter Router
Intranet Router
Web Server
HIPS
Device Hardening
FW
FW
Core Switch
Access Switch
IPS
FW/ PS
Network DLP
Device Hardening
AV DLP
AV DLP
AV DLP
DC Switch
SIM/SOC (Security Management)
HIPS
HIPS
HIPS
Application Server
DB Security
DLP
Workstations

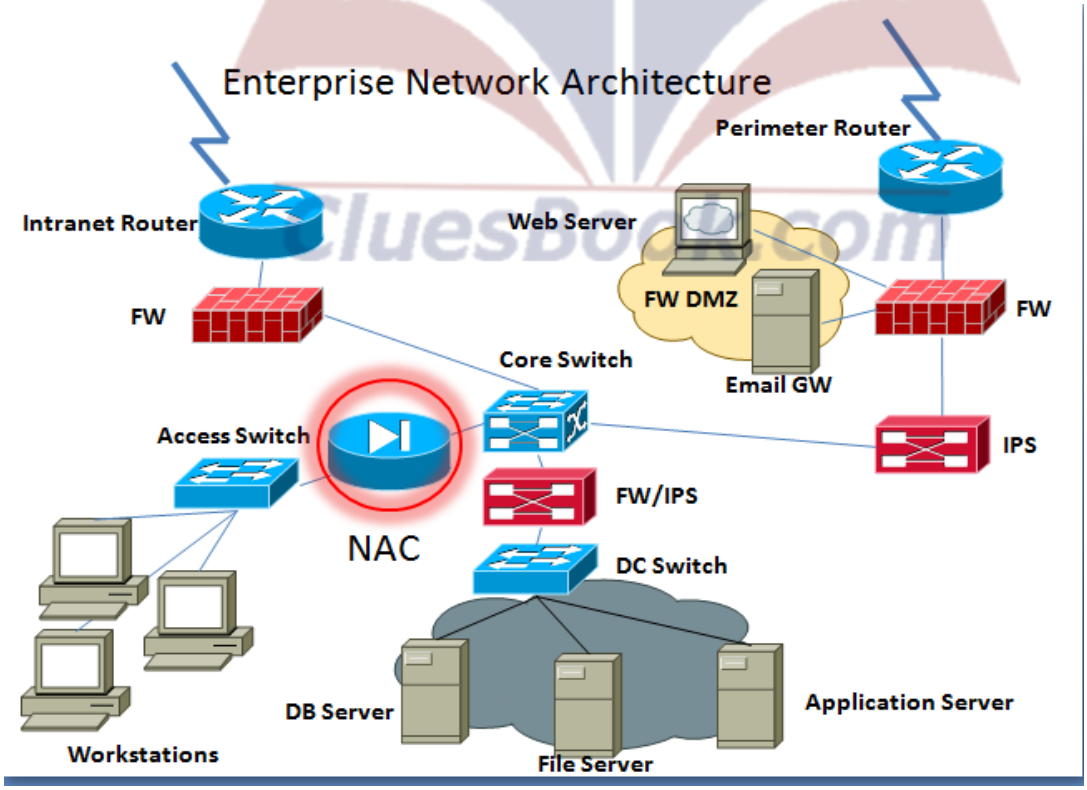

Enterprise Network Architecture
Perimeter Router
Intranet Router
Web Server
FW DMZ
FW
FW
Core Switch
Email GW
Access Switch
IPS
FW/IPS
HDD Encryption
DC Switch
HDD Encryption
DB Server
Application Server
Workstations
File Server

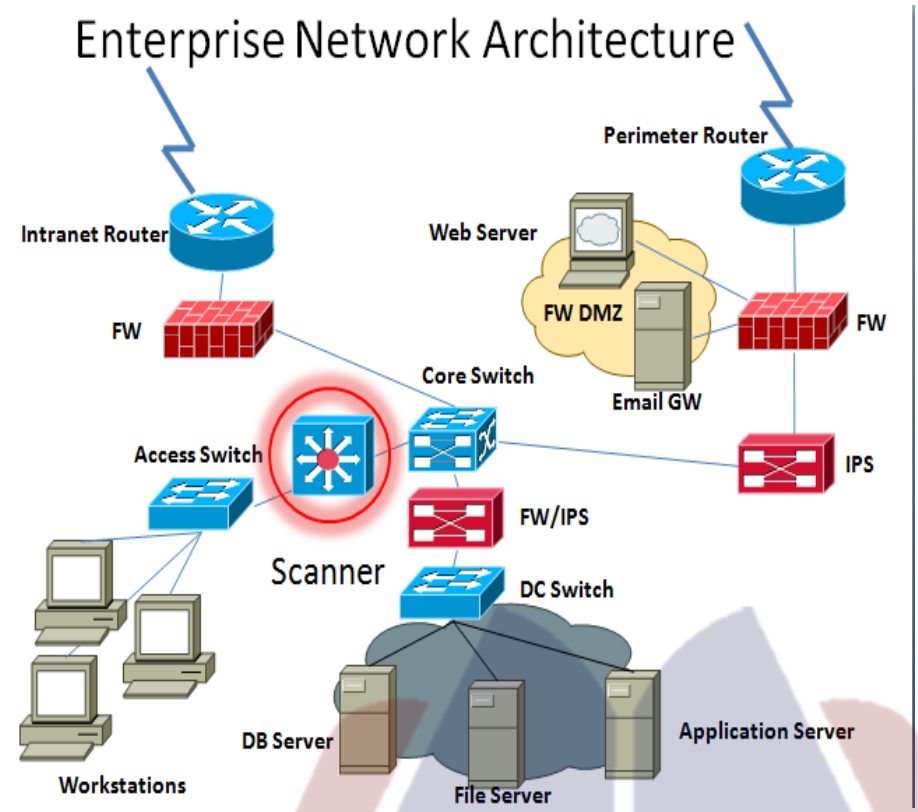

Enterprise Network Architecture
Perimeter Router
Intranet Router
Web Server
FW DMZ
FW
FW
Core Switch
Email GW
Access Switch
IPS
FW/IPS
NAC
DC Switch
DB Server
Application Server
Workstations
File Server

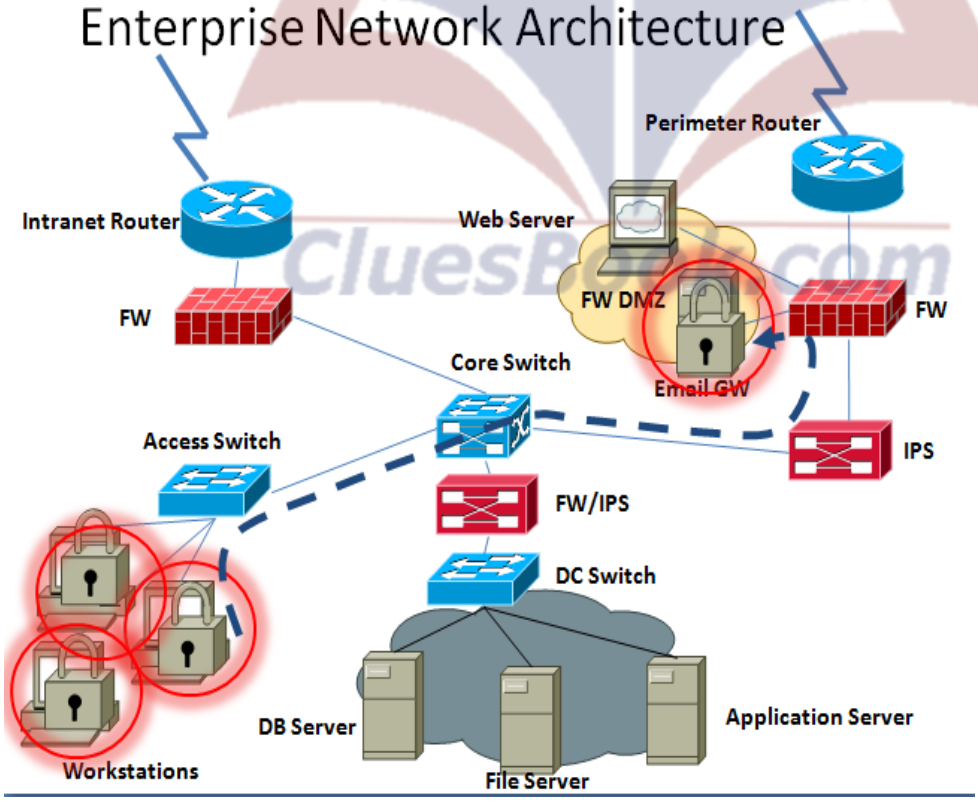


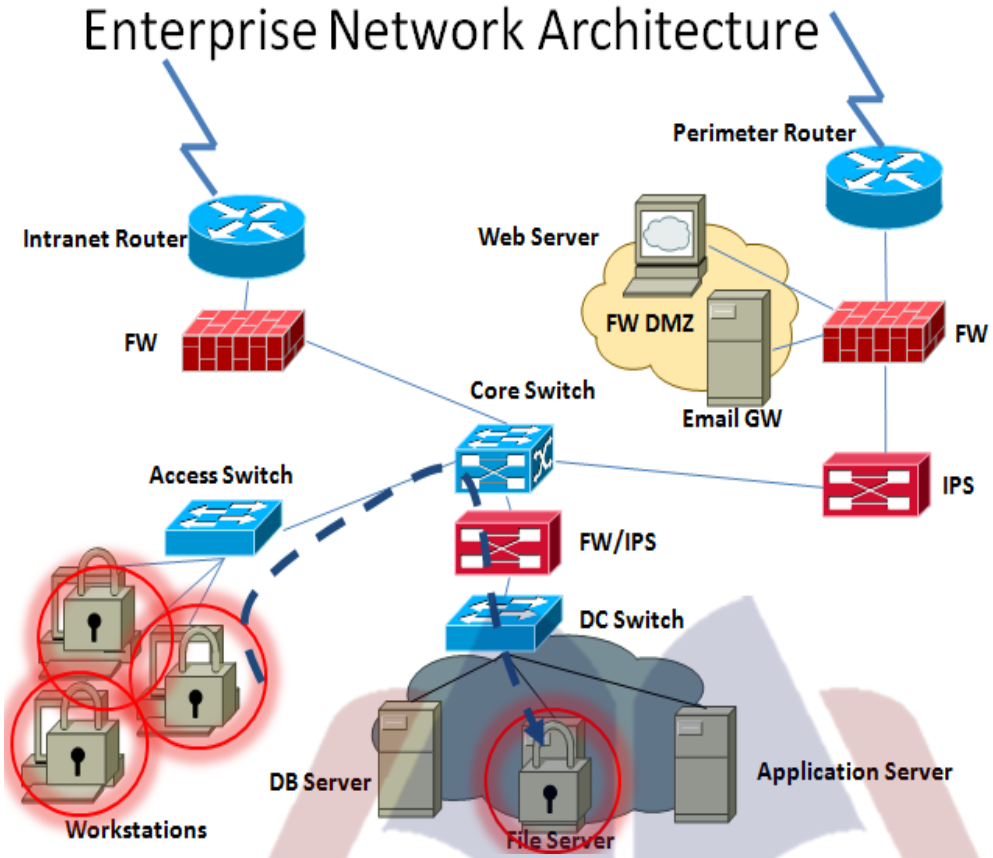



Enterprise Network Architecture
Perimeter Router
Intranet Router
Web Server
FW DMZ
Email GW
FW
FW
Core Switch
Access Switch
IPS
FW/IPS
DC Switch
DB Server
Application Server
Workstations
File Server


Enterprise Network Architecture
Perimeter Router
Intranet Router
Web Server
FW DMZ
Email GW
FW
FW
Core Switch
Access Switch
IPS
FW/IPS
NAC
DC Switch
DB Server
Application Server
Workstations
File Server


Enterprise Network Architecture
Perimeter Router
Intranet Router
Web Server
FW DMZ
FW
FW
Email GW
Core Switch
Access Switch
IPS
FW/IPS
Scanner
DC Switch
DB Server
Application Server
Workstations
File Server


Enterprise Network Architecture
Perimeter Router
Intranet Router
Web Server
FW DMZ
FW
FW
Email GW
Core Switch
Access Switch
IPS
FW/IPS
DC Switch
DB Server
Application Server
Workstations
File Server



4.2.1 Perimeter Router

- Also called edge router
- Provides WAN interfaces and protocol support
- First line of defense
- Access-lists can be applied here
- Router hardening is performed

4.2.2 Firewall (FW)

- It is a device or set of devices that is configured to permit or deny network transmissions based upon a set of rules and other criteria. Firewalls are thus a specialized type of router focusing on specific types of network security functions
- All messages entering or leaving the intranet pass through the firewall, which inspects each message and blocks those that do not meet the specified security criteria.
- VPN termination point

- Rules-based filters
- Zones, DMZ
- NAT

4.2.3 Intrusion Prevention System (IPS)

- Transparent device that inspects traffic (deep packet inspection)
- The main functions of intrusion prevention systems are to identify malicious activity, log information about said activity, attempt to block/stop activity, and report activity
- More specifically, IPS can take such actions as sending an alarm, dropping the malicious packets, resetting the connection and/or blocking the traffic from the offending IP address
- Signature-based Detection: This method of detection utilizes signatures, which are attack patterns that are preconfigured and predetermined. A signature-based intrusion prevention system monitors the network traffic for matches to these signatures.

4.2.4 Antivirus (AV)

- **Anti-virus software** is used to prevent, detect, and remove computer viruses, worms and trojan horses. It may also prevent and remove adware, spyware, and other forms of malware
- Signature-based detection involves searching for known patterns of data within executable code.
- However, it is possible for a computer to be infected with new malware for which no signature is yet known.
- To counter such so-called zero-day threats, heuristics can be used. One type of heuristic approach, generic signatures, can identify new viruses or variants of existing viruses by looking for known malicious code, or slight variations of such code, in files.
- Some antivirus software can also predict what a file will do by running it in a sandbox and analyzing what it does to see if it performs any malicious actions.

4.2.5 Host-based Intrusion Prevention (HIPS)

- The main functions of 'intrusion prevention systems' are to identify malicious activity, log information about said activity, attempt to block/stop activity, and report activity
- A host-based IPS monitors all or parts of the dynamic behavior and the state of a computer system.

- Much as a NIPS will dynamically inspect network packets, a HIPS might detect which program accesses what resources and discover that, for example, a word-processor has suddenly and inexplicably started modifying the system password database.
- Similarly a HIPS might look at the state of a system, its stored information, whether in RAM, in the file system, log files or elsewhere; and check that the contents of these appear as expected
- One can think of a HIPS as an agent that monitors whether anything or anyone, whether internal or external, has circumvented the system's security policy.
- Provides security features to protect servers
- Secure areas on HDD (integrity)
- Audit trail
- Permitted applications
- IPS functionality
- Is separate from AV, although the lines become blurred

4.2.6 Network Admission Control (NAC)

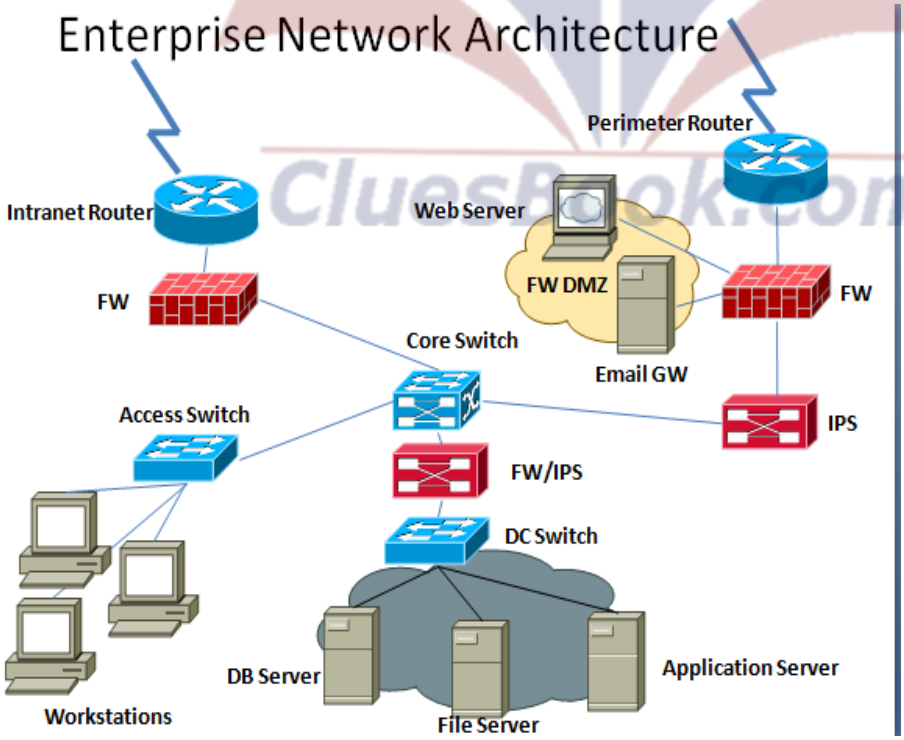
- NAC restricts access to the network based on identity or security posture
- When a network device is configured for NAC, it can force user or machine authentication prior to granting access to the network.
- In addition, guest access can be granted to a quarantine area for remediation of any problems that may have caused authentication failure
- Network admission control systems allow noncompliant devices to be denied access, placed in a quarantined area, or given restricted access to computing resources, thus keeping insecure nodes from infecting the network

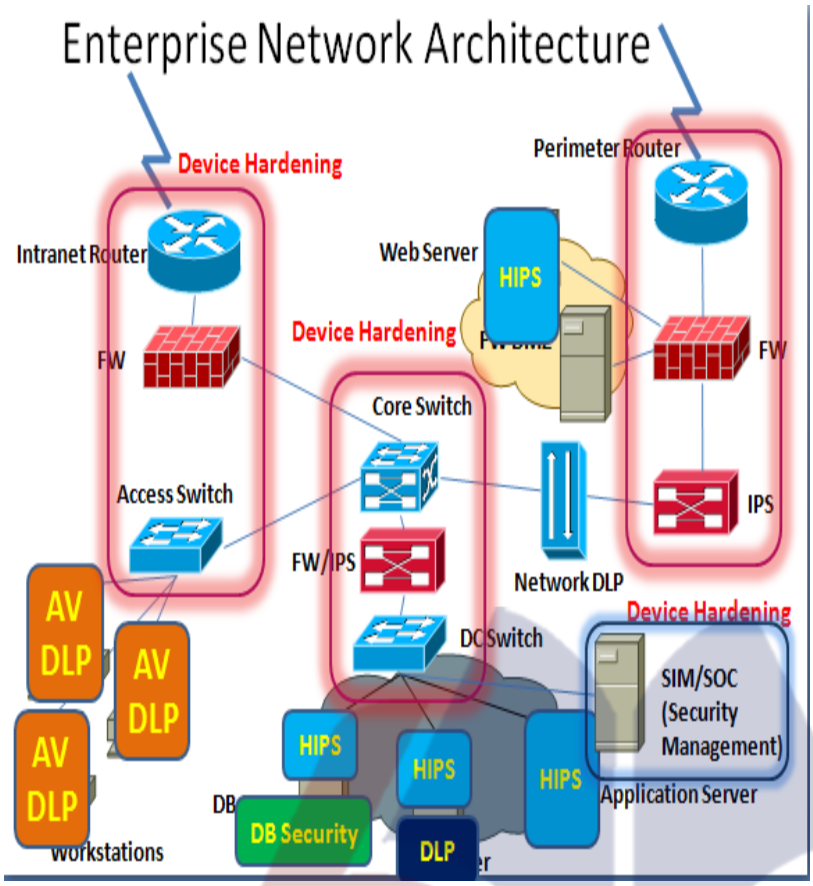
4.2.7 Data Loss Prevention (DLP)

- **Data Loss Prevention** (**DLP**) refers to systems that identify, monitor, and protect data
  - in use (e.g., endpoint actions),
  - data in motion (e.g., network actions),
  - and data at rest (e.g., data storage)

- DLP uses deep content inspection, contextual security analysis of transaction (attributes of originator, data object, medium, timing, recipient/destination, etc.), and with a centralized management framework
- The systems are designed to detect and prevent the unauthorized use and transmission of confidential information
- Can be agent based or network based
- Documents in an organization are classified
- Users in an organization are given rights for the documents (tags)
- Agent-based DLP:
  - Can restrict printing, print-screen, copying onto USB, attaching to email, attaching to web mail, transferring via instant messaging
- Network-based DLP:
  - Detects which documents are going out of the network and blocks if this is unauthorized

Lecture 31





4.2.8 Vulnerability Assessment Scanner

- A **vulnerability scanner** is a computer program designed to assess computers, computer systems, networks or applications for weaknesses.
- While functionality varies between different types of vulnerability scanners, they share a common, core purpose of enumerating the vulnerabilities present in one or more targets.
- Vulnerability scanners are a core technology component of vulnerability management.
- A vulnerability scanner can be used to conduct network reconnaissance, which is typically carried out by a remote attacker attempting to gain information or access to a network on which it is not authorized or allowed.
- Network reconnaissance is increasingly used to exploit network standards and automated communication methods.
- The aim is to determine what types of computers are present, along with additional information about those computers—such as the type and version of the operating system.
- This information can be analyzed for known or recently discovered vulnerabilities that can be exploited to gain access to secure networks and computers.

4.2.9 Web & Application Security

- **Web application security** is a branch of information security that deals specifically with security of websites and web applications
- **Application security** encompasses measures taken throughout the application's life-cycle to prevent exceptions in the security policy of an application or the underlying system (vulnerabilities) through flaws in the design, development, deployment, upgrade, or maintenance of the application
- The majority of web application attacks occur through cross-site scripting (XSS) and SQL injection attacks which typically result from flawed coding, and failure to sanitize input to and output from the web application

4.2.10 Database Security

- **Database security** is the system, processes, and procedures that protect a database from unintended activity
- Unintended activity can be categorized as authenticated misuse, malicious attacks or inadvertent mistakes made by authorized individuals or processes
- Databases provide many layers and types of information security:
  - Access control
  - Auditing
  - Authentication
  - Encryption
  - Integrity controls

4.2.11 Security Operations Center (SOC)

- A **security operations center (SOC)** is an Information Security function within the company or of separate organization that delivers IT security services
- It attempts to detect unauthorized access in any form to prevent and manage security related incidents using processes and procedures
- The mission is risk management through centralized analysis using the combined resources consisting of personnel, dedicated hardware and specialized software
- Typically, these systems operate constantly. These resources offer continuous events monitoring and risk analysis to detect intrusion to guarantee protection against it

- The SOC consists of monitoring and analyzing all types of systems, devices, or applications events such as users activities, firewall activity, Intrusion Detection System (IDS/IPS) activity, antivirus activity, individual vulnerabilities, etc.
- SOC Services
    - Proactive analysis & system management
    - Security device management
    - Reporting
    - Security alert
    - DDoS mitigation
    - Security assessment
    - Technical assistance

4.2.12 Secure Remote Access & VPNs

- Employees may want to access the corporate network from a remote location
    - Email
    - Fileserver
    - Other corporate network resources
- VPN connections terminate into a firewall or secure remote access server solution
- Virtual Private Networks (VPNs) may be established to connect to remote sites (site-to-site VPNs) within the intranet

4.2.13 Penetration Testing

- A **penetration test**, occasionally **pentest**, is a method of evaluating the security of a computer system or network by simulating an attack from a malicious source, known as a *Black Hat Hacker*, or *Cracker*.
- The process involves an active analysis of the system for any potential vulnerabilities that could result from poor or improper system configuration, both known and unknown hardware or software flaws, or operational weaknesses in process or technical countermeasures.
- This analysis is carried out from the position of a potential attacker and can involve active exploitation of security vulnerabilities.

- Any security issues that are found will be presented to the system owner, together with an assessment of their impact, and often with a proposal for mitigation or a technical solution.
- The intent of a penetration test is to determine the feasibility of an attack and the amount of business impact of a successful exploit, if discovered.
- It is a component of a full security audit

4.2.14 Network Forensics

- **Network forensics** is a sub-branch of digital forensics relating to the monitoring and analysis of computer network traffic for the purposes of information gathering, legal evidence or intrusion detection
- Unlike other areas of digital forensics, network investigations deal with volatile and dynamic information. Network traffic is transmitted and then lost, so network forensics is often a pro-active investigation.

Network forensics generally has two uses.

- The first, relating to security, involves monitoring a network for anomalous traffic and identifying intrusions
  - An attacker might be able to erase all log files on a compromised host; network-based evidence might therefore be the only evidence available for forensic analysis
- The second form of Network forensics relates to law enforcement
  - In this case analysis of captured network traffic can include tasks such as reassembling transferred files, searching for keywords and parsing human communication such as emails or chat sessions

lecture 32

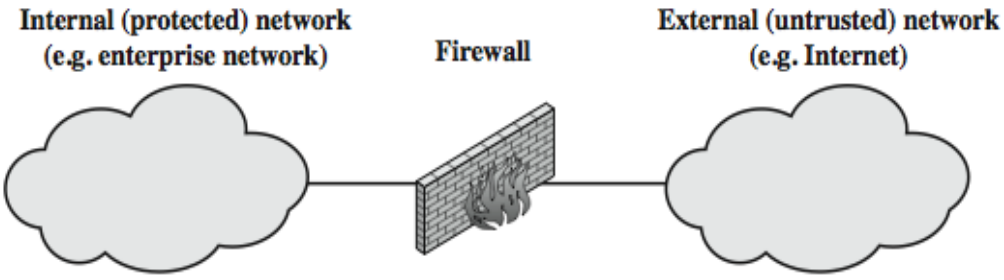
Firewalls

4.3.1 Firewalls

- Typically **Firewalls are used** to provide **perimeter defence** as part of a comprehensive security strategy
- Firewalls can be an effective means of protecting a local system or network of systems from network-based security threats while at the same time affording access to the outside world via wide area networks and the Internet
- However they need to be part of a wider security strategy including host security

What Is A Firewall ?

- Provide a **choke point** of control and monitoring
- Interconnect networks with differing trust
- Impose restrictions on network services
  - only authorized traffic is allowed
- Auditing and controlling access
  - can implement alarms for abnormal behavior
- Provide NAT & usage monitoring (e.g. Audit logs)
- Implement VPNs using IPSec
- Must be immune to penetration since it will be a target of attack



Firewall Limitations

- Cannot protect from attacks bypassing it

  - Utility modems, trusted organisations, trusted services (eg SSL/SSH)

- Cannot protect against internal threats
  - eg disgruntled or colluding employees
- Cannot protect against access via WLAN
  - if improperly secured against external use
- Cannot protect against malware imported via laptop, PDA, storage infected outside the corporate network, and then attached and used internally

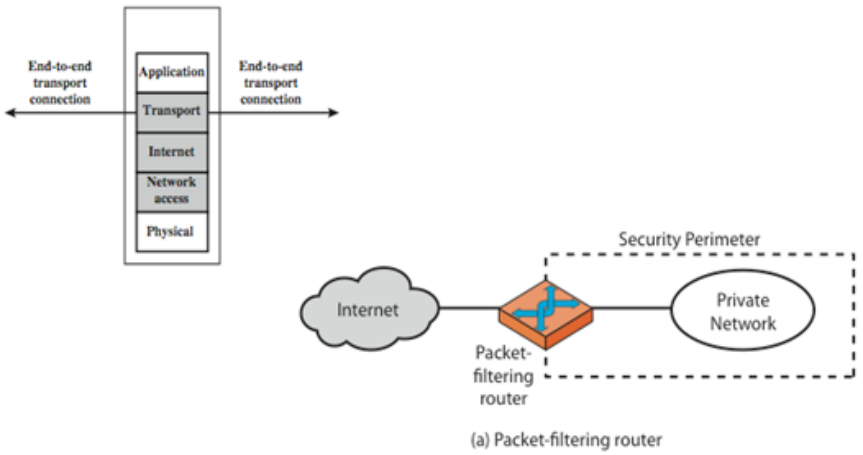
4.3.2 Types Of Firewalls

1. Packet Filters

- A packet-filtering router applies a set of rules to each incoming and outgoing IP packet to forward or discard the packet.
- Filtering rules are based on information contained in a network packet such as src & dest IP addresses, ports, transport protocol & interface Some advantages are simplicity, transparency & speed.

- Foundation of any firewall system
- possible default policies:
- that not expressly permitted is prohibited
- that not expressly prohibited is permitted

Firewalls – Packet Filters



(a) Packet-filtering router

The Figure illustrates the packet filter firewall role as utilising information from the transport, network & data link layers to make decisions on allowable traffic flows, and its placement in the border region between the external less-trusted Internet and the internal more trusted private network.

Attacks On Packet FWs

Some of the attacks that can be made on packet-filtering routers & countermeasures are:

- IP address spoofing: where intruder transmits packets from the outside with internal host source IP addr,
  - need to filter & discard such packets
- Source routing attacks: where source specifies the route that a packet should take to bypass security measures,
  - should discard all source routed packets
- Tiny fragment attacks: intruder uses the IP fragmentation option to create extremely small fragments and force the TCP header information into separate fragments to circumvent filtering rules needing full header info,
  - can enforce minimum fragment size to include full header

2. Stateful Packet Filters

- Traditional packet filters do not examine higher layer context

  - ie matching return packets with outgoing flow

- Stateful packet filters address this need
- They examine each IP packet in context
  - keep track of client-server sessions
  - check each packet validly belongs to one
- Hence are better able to detect bogus packets out of context
- May even inspect limited application data

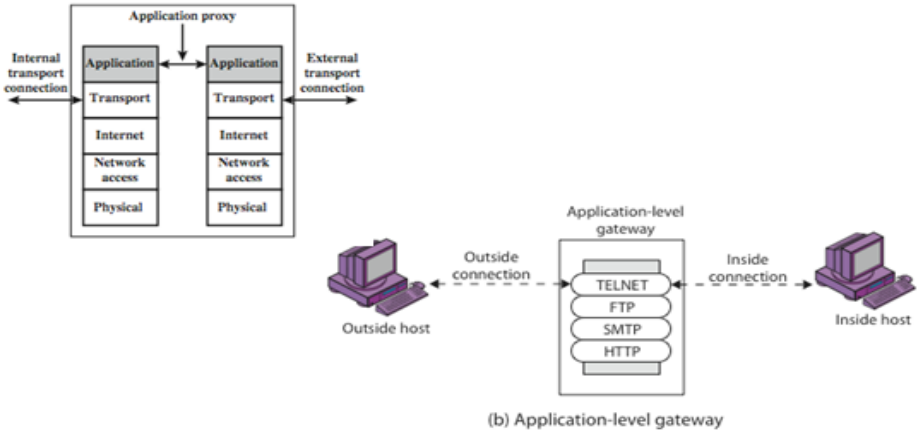
Firewalls – Stateful Packet Filters

- A stateful inspection packet filter tightens up the rules for TCP traffic by creating a directory of outbound TCP connections, and will allow incoming traffic to high-numbered ports only for those packets that fit the profile of one of the entries in this directory.
- Hence they are better able to detect bogus packets sent out of context.
- A stateful packet inspection firewall reviews the same packet information as a packet filtering firewall, but also records information about TCP connections.

3. Application Level Gateway (or Proxy)

- An application-level gateway (or proxy server), acts as a relay of application-level traffic
- A user contacts the gateway to access some service, provides details of the service, remote host & authentication details. The gateway contacts the application on the remote host and relays all data between the two endpoints
- If the gateway does not implement the proxy code for a specific application, then it is not supported and cannot be used
- Note that some services naturally support proxying, whilst others are more problematic. Application-level gateways tend to be more secure than packet filters, & can log and audit traffic at application level

3. Application Level Gateway (or Proxy)



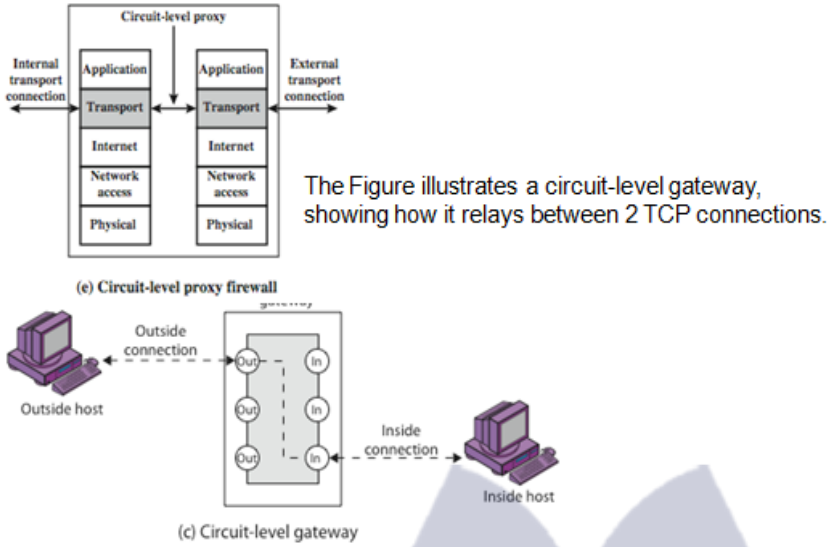
(b) Application-level gateway

The Figure illustrates an application-level gateway only supports a specific list of application services

4. Circuit Level Gateway

- Relays two TCP connections
- Imposes security by limiting which such connections are allowed
- Once created usually relays traffic without examining contents
- Typically used when trust internal users by allowing general outbound connections
- SOCKS is commonly used

Firewalls - Circuit Level Gateway



- A circuit-level gateway relays two TCP connections, one between itself and an inside TCP user, and the other between itself and a TCP user on an outside host.
- Once the two connections are established, it relays TCP data from one connection to the other without examining its contents.

Firewalls – Circuit Level Gateway

- One of the most common circuit-level gateways is SOCKS, defined in RFC 1928.
- It consists of a SOCKS server on the firewall, and a SOCKS library & SOCKS-aware applications on internal clients.
- When a TCP-based client wishes to establish a connection to an object that is reachable only via a firewall (such determination is left up to the implementation), it must open a TCP connection to the appropriate SOCKS port on the SOCKS server system.
- If the connection request succeeds, the client enters a negotiation for the authentication method to be used, authenticates with the chosen method, and then sends a relay request.
- The SOCKS server evaluates the request and either establishes the appropriate connection or denies it. UDP exchanges are handled in a similar fashion.

5. Bastion Host

- Highly secure host system
- Runs circuit / application level gateways
- Or provides externally accessible services
- Potentially exposed to "hostile" elements
- Hence is secured to withstand this
- May support 2 or more net connections
- May be trusted to enforce policy of trusted separation between these net connections
    - Relaying traffic only according to policy
- Common characteristics of a bastion host:
- • Executes a secure version of its O/S, making it a trusted system
- • Has only essential services installed on the bastion host
- • May require additional authentication before a user may access to proxy services
- • Configured to use only subset of standard commands, access only specific hosts
- • Maintains detailed audit information by logging all traffic

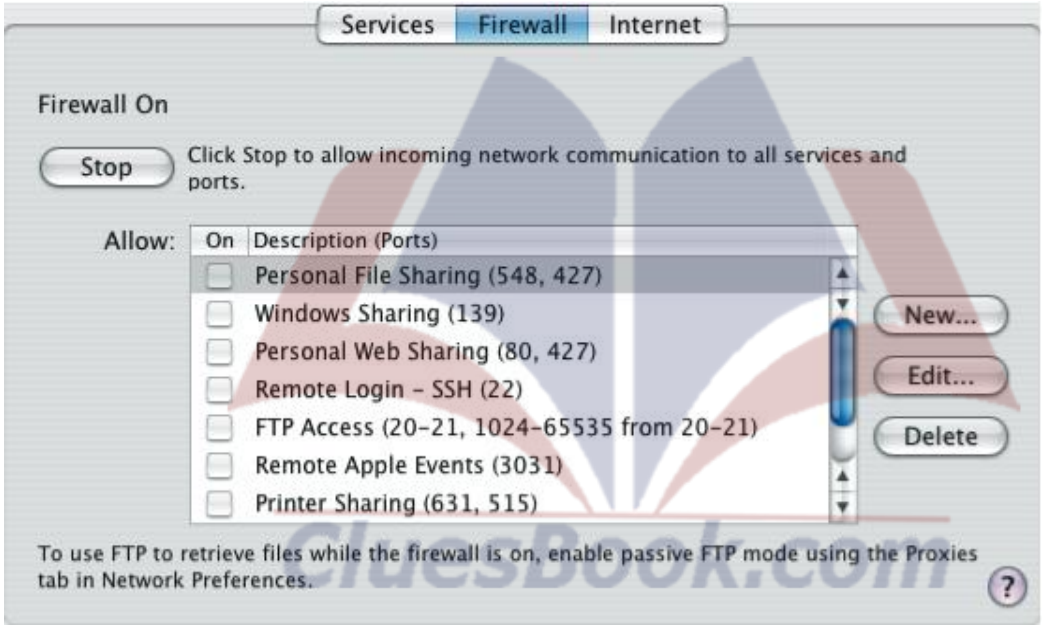
6. Host-Based Firewalls

- S/W module used to secure individual host
    - Available in many operating systems
    - Or can be provided as an add-on package
- Often used on servers
- Advantages:
    - Can tailor filtering rules to host environment
    - Protection is provided independent of topology
    - Provides an additional layer of protection

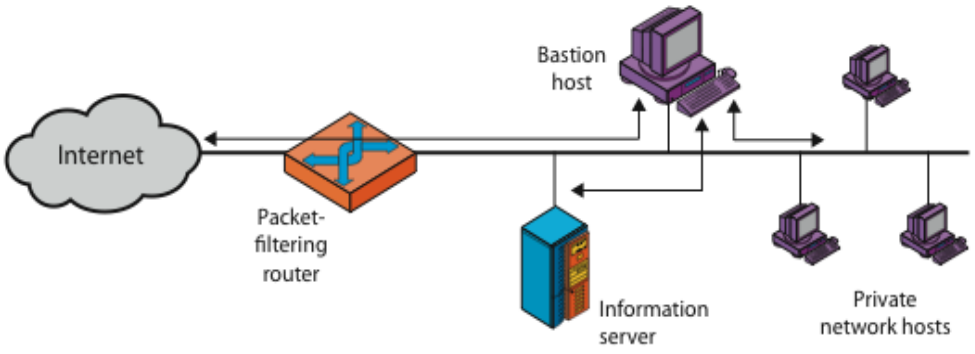
7. Personal Firewalls

- Controls traffic between PC/workstation and Internet or enterprise network
- A software module on personal computer
- Or in home/office DSL/cable/ISP router
- Typically much less complex than other firewall types
- Primary role to deny unauthorized remote access to the computer, and

Monitor outgoing activity for malware



4.3.3 Firewall Configurations – Screened Host Firewall (Single Homed Bastion Host)



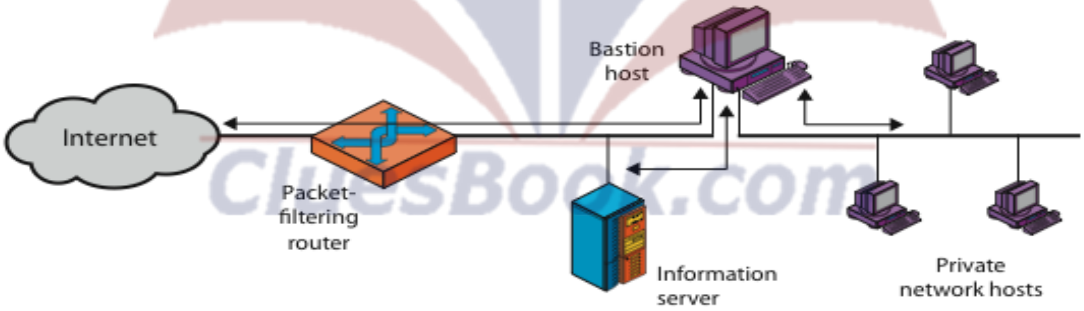
(a) Screened host firewall system (single-homed bastion host)

The firewall consists of two systems:

- A packet-filtering router - allows Internet packets to/from bastion only
- A bastion host - performs authentication and proxy functions

This configuration has greater security, as it implements both packet-level & application-level filtering, forces an intruder to generally penetrate two separate systems to compromise internal security, & also affords flexibility in providing direct Internet access to specific internal servers (eg web) if desired.

Firewall Configurations – Screened Host Firewall (Dual Homed Bastion Host)

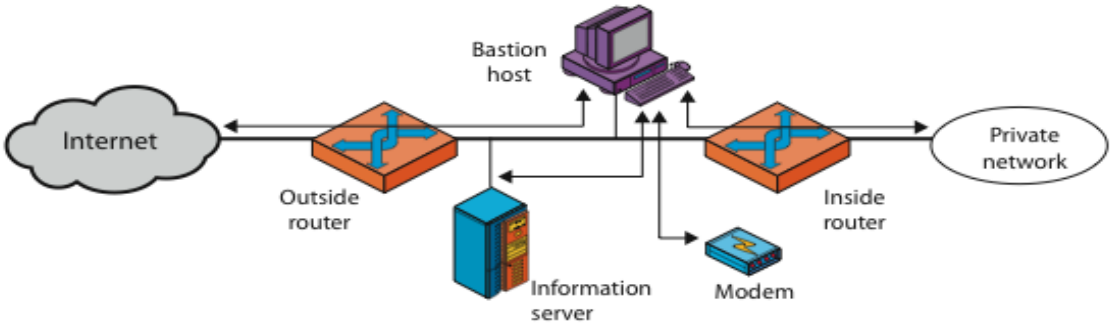


(b) Screened host firewall system (dual-homed bastion host)

This configuration physically separates the external and internal networks, ensuring two systems must be compromised to breach security.

The advantages of dual layers of security are also present here. Again, an information server or other hosts can be allowed direct communication with the router if this is in accord with the security policy, but are now separated from the internal network.

Firewall Configurations – Screened Subnet Firewall



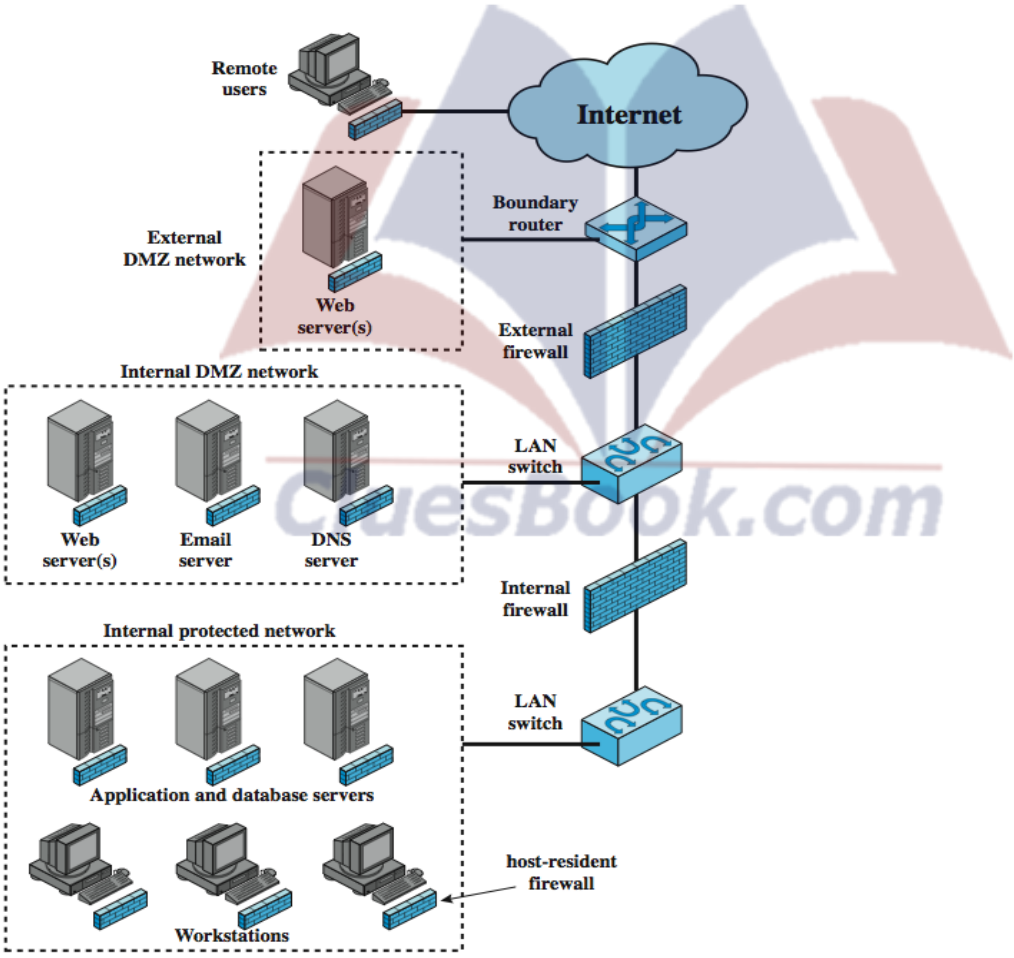
(c) Screened-subnet firewall system

- Most secure configuration
- It has two packet-filtering routers, one between the bastion host and the Internet and the other between the bastion host and the internal network, creating an isolated subnetwork.
- This may consist of simply the bastion host but may also include one or more information servers and modems for dial-in capability.
- Typically, both the Internet and the internal network have access to hosts on the screened subnet, but traffic across the screened subnet is blocked.
- This configuration offers several advantages:
    - There are now three levels of defense to thwart intruders
    - The outside router advertises only the existence of the screened subnet to the Internet; therefore the internal network is invisible to the Internet
    - Similarly, the inside router advertises only the existence of the screened subnet to the internal network; hence systems on the inside network cannot construct direct routes to the Internet

DMZ

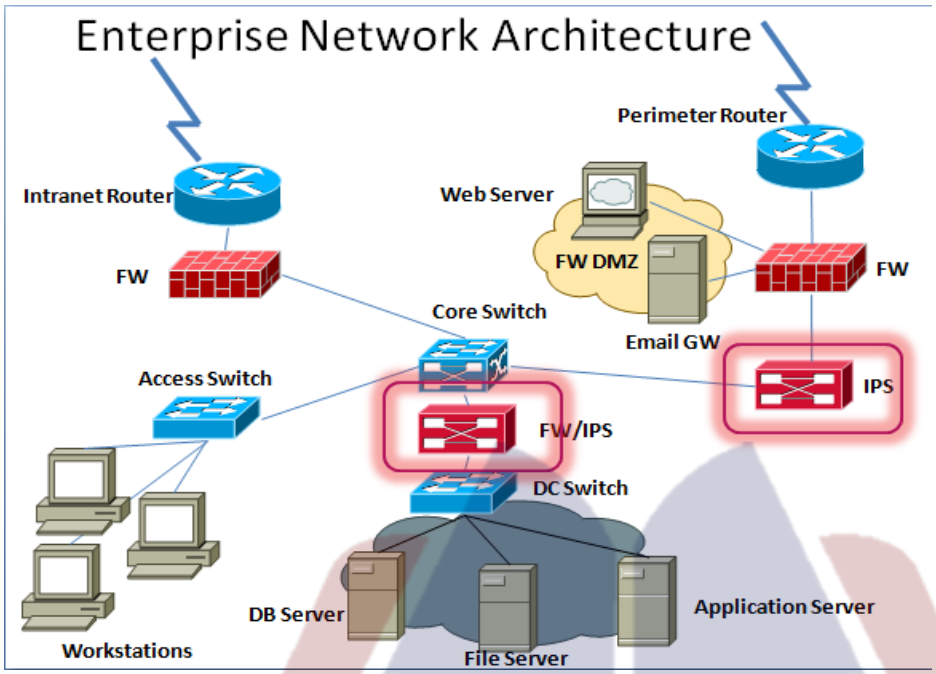
- "Screened subnet", also known as a demilitarized zone (DMZ), located between an internal and an external firewall.
- Typically, the systems in the DMZ require or foster external connectivity, such as a corporate Web site, an e-mail server, or a DNS (domain name system) server.
- The external firewall provides a measure of access control and protection for the DMZ systems consistent with their need for external connectivity. The external firewall also provides a basic level of protection for the remainder of the enterprise network.

Distributed Firewalls



Lecture 34

Intrusion Detection & Prevention Systems (IDPS)



4.4.1 What Is Intrusion Detection ?

Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of possible incidents, which are violations or imminent threats of violation of computer security policies, acceptable use policies, or standard security practices.

Incidents have many causes, such as malware (e.g., worms, spyware), attackers gaining unauthorized access to systems from the Internet, and authorized users of systems who misuse their privileges or attempt to gain additional privileges for which they are not authorized.

Although many incidents are malicious in nature, many others are not; for example, a person might mistype the address of a computer and accidentally attempt to connect to a different system without authorization.

IDS & IPS

An intrusion detection system (IDS) is software that automates the intrusion detection process. An intrusion prevention system (IPS) is software that has all the capabilities of an intrusion detection system and can also attempt to stop possible incidents.

IDS and IPS technologies offer many of the same capabilities, and administrators can usually disable prevention features in IPS products, causing them to function as IDSs. Accordingly, for brevity the term intrusion detection and prevention systems (IDPS) is used throughout the rest of this section to refer to both IDS and IPS technologies

4.4.2 Uses Of IDPS Technologies

IDPSs are primarily focused on identifying possible incidents. For example, an IDPS could detect when an attacker has successfully compromised a system by exploiting a vulnerability in the system. The IDPS could then report the incident to security administrators, who could quickly initiate incident response actions to minimize the damage caused by the incident.

The IDPS could also log information that could be used by the incident handlers. Many IDPSs can also be configured to recognize violations of security policies. For example, some IDPSs can be configured with firewall rule set-like settings, allowing them to identify network traffic that violates the organization's security or acceptable use policies.

An IDPS might be able to block reconnaissance and notify security administrators, who can take actions if needed to alter other security controls to prevent related incidents. Because reconnaissance activity is so frequent on the Internet, reconnaissance detection is often performed primarily on protected internal networks.

In addition to identifying incidents and supporting incident response efforts, organizations have found other uses for IDPSs, including the following:

Other Uses Of IDPS Technologies

1. Identifying security policy problems. An IDPS can provide some degree of quality control for security policy implementation, such as duplicating firewall rule sets and alerting when it sees network traffic that should have been blocked by the firewall but was not because of a firewall configuration error.

2. Documenting the existing threat to an organization. IDPSs log information about the threats that they detect. Understanding the frequency and characteristics of attacks against an organization's computing resources is helpful in identifying the appropriate security measures for protecting the resources.

3. Deterring individuals from violating security policies. If individuals are aware that their actions are being monitored by IDPS technologies for security policy violations, they may be less likely to commit such violations because of the risk of detection.

Because of the increasing dependence on information systems and the prevalence and potential impact of intrusions against those systems, IDPSs have become a necessary addition to the security infrastructure of nearly every organization.

4.4.3 Key Functions Of IDPS Technologies

There are many types of IDPS technologies, which are differentiated primarily by the types of events that they can recognize and the methodologies that they use to identify incidents. In addition to monitoring and analyzing events to identify undesirable activity, all types of IDPS technologies typically perform the following functions:

1. Recording information related to observed events. Information is usually recorded locally, and might also be sent to separate systems such as centralized logging servers, security information and event management (SIEM) solutions, and enterprise management systems.

2. Notifying security administrators of important observed events. This notification, known as an alert, occurs through any of several methods, including the following:

E-mails, pages, messages on the IDPS user interface, Simple Network Management Protocol (SNMP) traps, syslog messages, and user-defined programs and scripts. A notification message typically includes only basic information regarding an event; administrators need to access the IDPS for additional information.

3. Producing reports. Reports summarize the monitored events or provide details on particular events of interest.

Some IDPSs are also able to change their security profile when a new threat is detected. For example, an IDPS might be able to collect more detailed information for a particular session after malicious activity is detected within that session. An IDPS might also alter the settings for when certain alerts are triggered or what priority should be assigned to subsequent alerts after a particular threat is detected.

Lecture 35

IPS Technologies

IPS technologies are differentiated from IDS technologies by one characteristic: IPS technologies can respond to a detected threat by attempting to prevent it from succeeding. They use several response techniques, which can be divided into the following groups

4.4.4 IPS Response Techniques

1. **The IPS stops the attack itself.**

Examples of how this could be done are as follows:

-Terminate the network connection or user session that is being used for the attack

-Block access to the target (or possibly other likely targets) from the offending user account, IP address, or other attacker attribute

-Block all access to the targeted host, service, application, or other resource.

2. The IPS changes the security environment.

The IPS could change the configuration of other security controls to disrupt an attack. Common examples are reconfiguring a network device (e.g., firewall, router, switch) to block access from the attacker or to the target, and altering a host-based firewall on a target to block incoming attacks. Some IPSs can even cause patches to be applied to a host if the IPS detects that the host has vulnerabilities.

3. The IPS changes the attack's content.

- Some IPS technologies can remove or replace malicious portions of an attack to make it benign. A simple example is an IPS removing an infected file attachment from an e-mail and then permitting the cleaned email to reach its recipient.
- A more complex example is an IPS that acts as a proxy and normalizes incoming requests, which means that the proxy repackages the payloads of the requests, discarding header information. This might cause certain attacks to be discarded as part of the normalization process.

Accuracy Of Detection

Another common attribute of IDPS technologies is that they cannot provide completely accurate detection. When an IDPS incorrectly identifies benign activity as being malicious, a false positive has occurred. When an IDPS fails to identify malicious activity, a false negative has occurred.

It is not possible to eliminate all false positives and negatives; in most cases, reducing the occurrences of one increases the occurrences of the other.

Many organizations choose to decrease false negatives at the cost of increasing false positives, which means that more malicious events are detected but more analysis resources are needed to differentiate false positives from true malicious events. Altering the configuration of an IDPS to improve its detection accuracy is known as tuning.

Evasion Techniques

Most IDPS technologies also offer features that compensate for the use of common evasion techniques. Evasion is modifying the format or timing of malicious activity so that its appearance changes but its effect is the same.

Attackers use evasion techniques to try to prevent IDPS technologies from detecting their attacks. For example, an attacker could encode text characters in a particular way, knowing that the target understands the encoding and hoping that any monitoring IDPSs do not.

Most IDPS technologies can overcome common evasion techniques by duplicating special processing performed by the targets. If the IDPS can “see” the activity in the same way that the target would, then evasion techniques will generally be unsuccessful at hiding attacks.

4.4.5 Common Detection Methodologies

IDPS technologies use many methodologies to detect incidents. The primary classes of detection methodologies are: signature-based, anomaly-based, and stateful protocol analysis, respectively.

Most IDPS technologies use multiple detection methodologies, either separately or integrated, to provide more broad and accurate detection.

1. Signature Based Detection

A signature is a pattern that corresponds to a known threat. Signature-based detection is the process of comparing signatures against observed events to identify possible incidents. Examples of signatures are as follows:

A telnet attempt with a username of “root”, which is a violation of an organization’s security policy

-An e-mail with a subject of “Free pictures!” and an attachment filename of “freepics.exe”, which are characteristics of a known form of malware

-An operating system log entry with a status code value of 645, which indicates that the host’s auditing has been disabled.

Signature-based detection is very effective at detecting known threats but largely ineffective at detecting previously unknown threats, threats disguised by the use of evasion techniques, and many variants of known threats.

For example, if an attacker modified the malware in the previous example to use a filename of “freepics2.exe”, a signature looking for “freepics.exe” would not match it

Signature-based detection is the simplest detection method because it just compares the current unit of activity, such as a packet or a log entry, to a list of signatures using string comparison operations.

Signature-based detection technologies have little understanding of many network or application protocols and cannot track and understand the state of complex communications. For example, they cannot pair a request with the corresponding response, such as knowing that a request to a Web server for a particular page generated a response status code of 403, meaning that the server refused to fill the request.

They also lack the ability to remember previous requests when processing the current request. This limitation prevents signature-based detection methods from detecting attacks that comprise multiple events if none of the events contains a clear indication of an attack.

Lecture 36

2. Anomaly Based Detection

Anomaly-based detection is the process of comparing definitions of what activity is considered normal against observed events to identify significant deviations. An IDPS using anomaly-based detection has profiles that represent the normal behavior of such things as users, hosts, network connections, or applications.

The profiles are developed by monitoring the characteristics of typical activity over a period of time. For example, a profile for a network might show that Web activity comprises an average of 13% of network bandwidth at the Internet border during typical workday hours.

The IDPS then uses statistical methods to compare the characteristics of current activity to thresholds related to the profile, such as detecting when Web activity comprises significantly more bandwidth than expected and alerting an administrator of the anomaly.

Profiles can be developed for many behavioral attributes, such as the number of e-mails sent by a user, the number of failed login attempts for a host, and the level of processor usage for a host in a given period of time.

An initial profile is generated over a period of time (typically days, sometimes weeks) sometimes called a training period. Profiles for anomaly-based detection can either be static or dynamic. Once generated, a static profile is unchanged unless the IDPS is specifically directed to generate a new profile.

A dynamic profile is adjusted constantly as additional events are observed. Because systems and networks change over time, the corresponding measures of normal behavior also change; a static profile will eventually become inaccurate, so it needs to be regenerated periodically.

Dynamic profiles do not have this problem, but they are susceptible to evasion attempts from attackers. For example, an attacker can perform small amounts of malicious activity occasionally, then slowly increase the frequency and quantity of activity.

If the rate of change is sufficiently slow, the IDPS might think the malicious activity is normal behavior and include it in its profile. Malicious activity might also be observed by an IDPS while it builds its initial profiles.

Inadvertently including malicious activity as part of a profile is a common problem with anomaly-based IDPS products. (In some cases, administrators can modify the profile to exclude activity in the profile that is known to be malicious.) Another problem with building profiles is that it can be very challenging in some cases to make them accurate, because computing activity can be so complex.

For example, if a particular maintenance activity that performs large file transfers occurs only once a month, it might not be observed during the training period; when the maintenance occurs, it is likely to be considered a significant deviation from the profile and trigger an alert.

Anomaly-based IDPS products often produce many false positives because of benign activity that deviates significantly from profiles, especially in more diverse or dynamic environments.

Another noteworthy problem with the use of anomaly-based detection techniques is that it is often difficult for analysts to determine why a particular alert was generated and to validate that an alert is accurate and not a false positive, because of the complexity of events and number of events that may have caused the alert to be generated.

3. Stateful Protocol Analysis

Stateful protocol analysis is the process of comparing predetermined profiles of generally accepted definitions of benign protocol activity for each protocol state against observed events to identify deviations.

Unlike anomaly-based detection, which uses host or network-specific profiles, stateful protocol analysis relies on vendor-developed universal profiles that specify how particular protocols should and should not be used. The “stateful” in stateful protocol analysis means that the IDPS is capable of understanding and tracking the state of network, transport, and application protocols that have a notion of state.

For example, when a user starts a File Transfer Protocol (FTP) session, the session is initially in the unauthenticated state. Unauthenticated users should only perform a few commands in this state, such as viewing help information or providing usernames and passwords.

An important part of understanding state is pairing requests with responses, so when an FTP authentication attempt occurs, the IDPS can determine if it was successful by finding the status code in the corresponding response.

Once the user has authenticated successfully, the session is in the authenticated state, and users are expected to perform any of several dozen commands. Performing most of these commands while in the unauthenticated state would be considered suspicious, but in the authenticated state performing most of them is considered benign.

Stateful protocol analysis can identify unexpected sequences of commands, such as issuing the same command repeatedly or issuing a command without first issuing a command upon which it is dependent.

Stateful protocol analysis methods use protocol models, which are typically based primarily on protocol standards from software vendors and standards bodies (e.g., Internet Engineering Task Force [IETF] Request for Comments [RFC]). The protocol models also typically take into account variances in each protocol’s implementation.

Many standards are not exhaustively complete in explaining the details of the protocol, which causes variations among implementations. Also, many vendors either violate standards or add proprietary features, some of which may replace features from the standards.

For proprietary protocols, complete details about the protocols are often not available, making it difficult for IDPS technologies to perform comprehensive, accurate analysis. As protocols are revised and vendors alter their protocol implementations, IDPS protocol models need to be updated to reflect those changes.

The primary drawback to stateful protocol analysis methods is that they are very resource-intensive because of the complexity of the analysis and the overhead involved in performing state tracking for many simultaneous sessions.

Another serious problem is that stateful protocol analysis methods cannot detect attacks that do not violate the characteristics of generally acceptable protocol behavior, such as performing many benign actions in a short period of time to cause a denial of service.

Yet another problem is that the protocol model used by an IDPS might conflict with the way the protocol is implemented in particular versions of specific applications and operating systems, or how different client and server implementations of the protocol interact.

Lecture 37

System & Server Security

Threats to Hosts

- **The Problem**
  - Some attacks inevitably reach host computers
  - So servers and other hosts must be hardened— a complex process that requires a diverse set of protections to be implemented on each host
- **What Is a Host?**
  - Anything with an IP address is a host (because it can be attacked)
  - Servers
  - Clients (including mobile telephones)
  - Routers (including home access routers) and sometimes switches
  - Firewalls

-

4.5.1 Elements of Host Hardening

- Backup
- Backup
- Backup
- Restrict physical access to hosts
- Install the operating system with secure configuration options
    - Change all default passwords, etc.
- Minimize the applications that run on the host
- Harden all remaining applications on the host
- Download and install patches for operating vulnerabilities
- Manage users and groups securely
- Manage access permissions for users and groups securely
- Encrypt data if appropriate
- Add a host firewall
- Read operating system log files regularly for suspicious activity
- Run vulnerability tests frequently

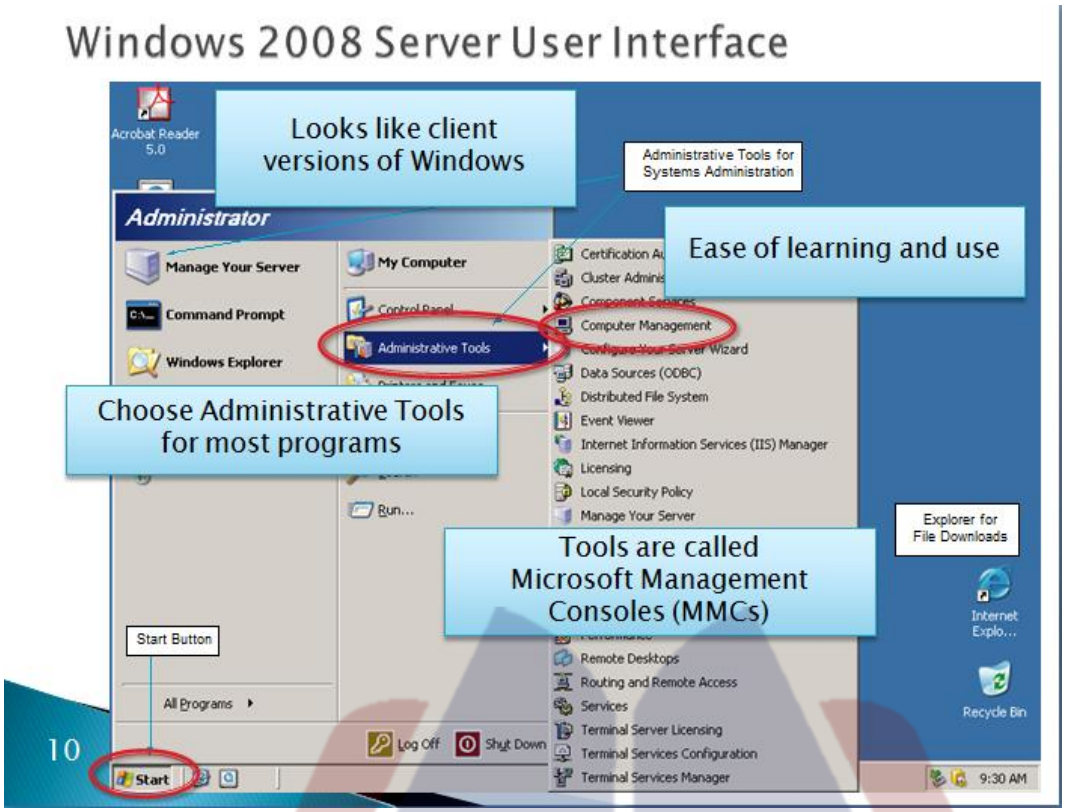
4.5.2 Security Baselines and Systems Administrators

- **Security Baselines Guide the Hardening Effort**
    - Specifications for how hardening should be done
    - Needed because it is easy to forget a step
    - Different baselines for different operating systems and versions
    - Different baselines for servers with different functions (webservers, mail servers, etc.)
    - Used by systems administrators (server administrators)
        - Usually do not manage the network

- **Security Baselines Guide the Hardening Effort**
  - Disk Images
    - Can also create a well-tested secure implementation for each operating system versions and server function
    - Save as a disk image
    - Load the new disk image on new servers

4.5.3 Windows Server Operating Systems

- **Windows Server**
  - The Microsoft Windows Server operating system
  - Windows NT, 2003, and 2008
- **Windows Server Security**
  - Intelligently minimize the number of running programs and utilities by asking questions during installation
  - Simple (and usually automatic) to get updates
  - Still many patches to apply, but this is true of other operating systems



UNIX Operating Systems

- **Many Versions of UNIX**
  - There are many commercial versions of UNIX for large servers
    - Compatible in the kernel (core part) of the operating system
      - Can generally run the same applications
    - But may run many different management utilities, making cross-learning difficult
  - LINUX is a version of UNIX created for PCs
    - Many different LINUX distributions
      - Distributions include the LINUX kernel plus application and programs, usually from the GNU project
      - Each distribution and version needs a different baseline to guide hardening

4.5.4 Vulnerabilities and Exploits

- **Vulnerabilities**
  - Security weaknesses that open a program to attack
  - An exploit takes advantage of a vulnerability
  - Vendors develop fixes
  - Zero-day exploits: exploits that occur before fixes are released
  - Exploits often follow the vendor release of fixes within days or even hours
  - Companies must apply fixes quickly
- **Fixes**
  - Work-arounds
    - Manual actions to be taken
    - Labor-intensive so expensive and error-prone
  - Patches:
    - Small programs that fix vulnerabilities
    - Usually easy to download and install
  - Service packs (groups of fixes in Windows)
  - Version upgrades

Lecture 38

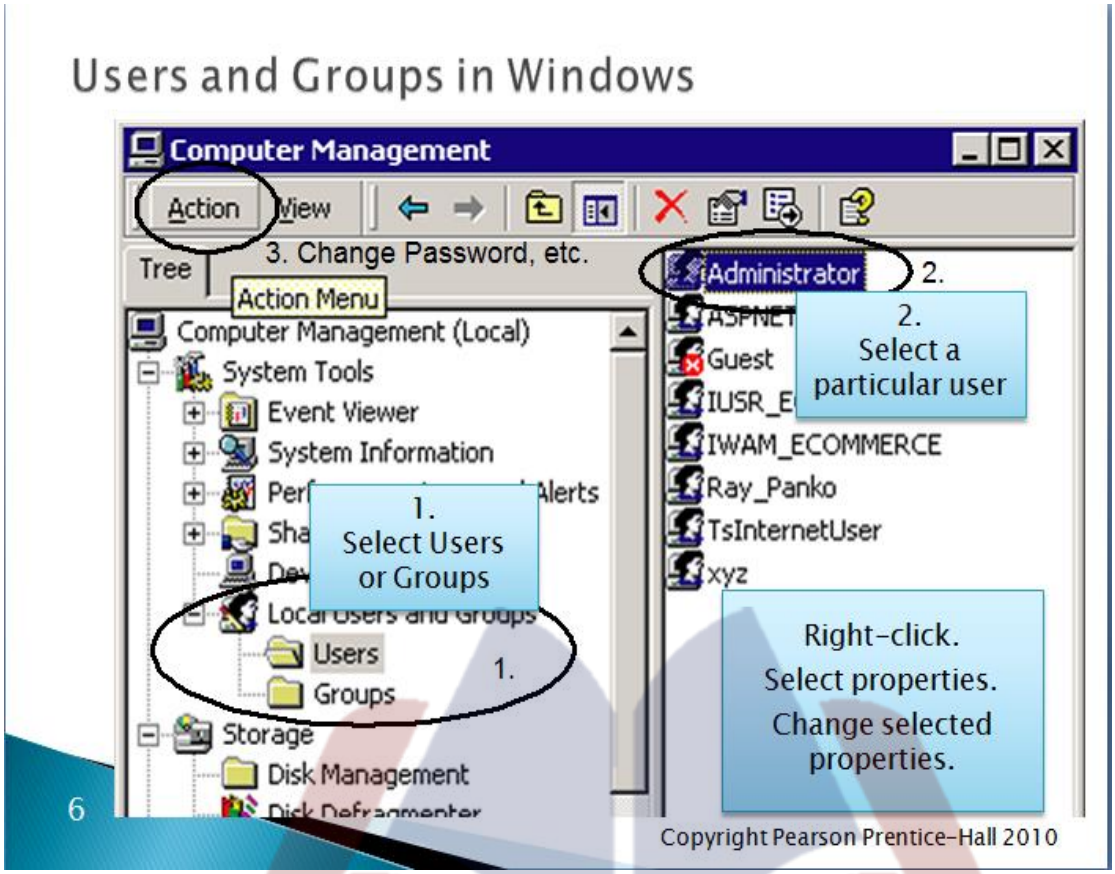
4.5.5 Applying Patching

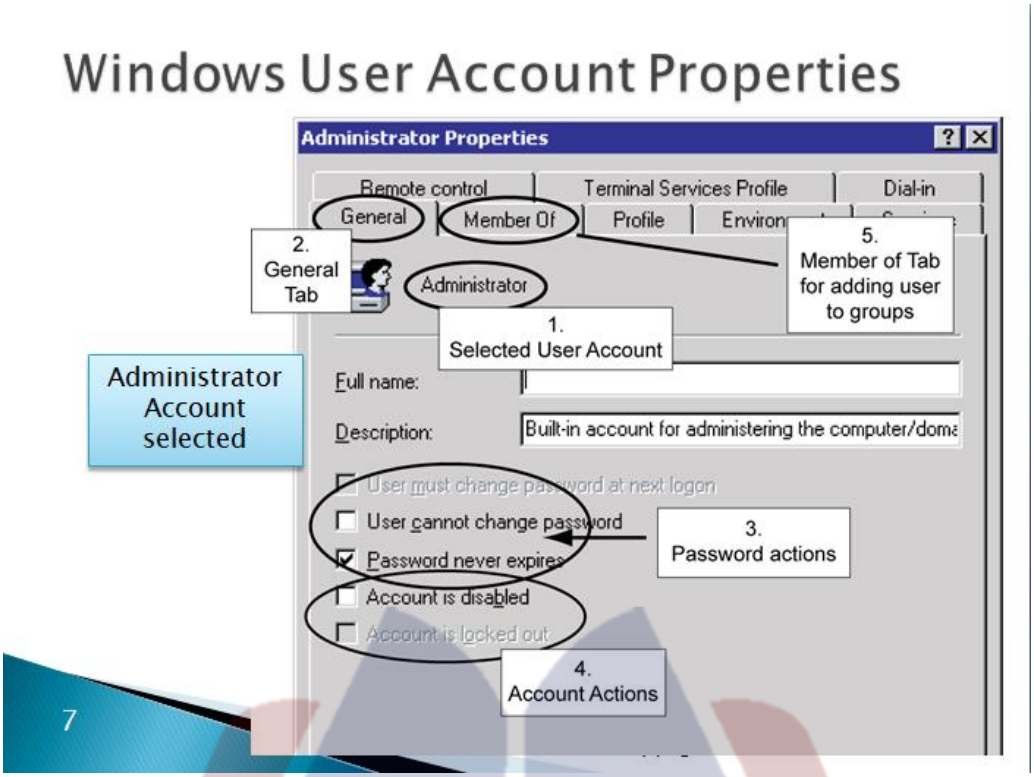
- **Problems with Patching**
  - Must find operating system patches
    - Windows Server does this automatically
    - LINUX versions often use rpm

- …
- Companies get overwhelmed by number of patches
  - Use many programs; vendors release many patches per product
  - Especially a problem for a firm's many application programs
- Cost of patch installation
  - Each patch takes some time and labor costs
  - Usually lack the resources to apply all
- Prioritization
  - Prioritize patches by criticality
  - May not apply all patches, if risk analysis does not justify them
- Risks of patch installation
  - Reduced functionality
  - Freeze machines, do other damage—sometimes with no uninstall possible
  - Should test on a test system before deployment on servers

4.5.6 Managing Users and Groups

- **Accounts**
  - Every user must have an account
- **Groups**
  - Individual accounts can be consolidated into groups
  - Can assign security measures to groups
  - Inherited by each group's individual members
  - Reduces cost compared to assigning to individuals
  - Reduces errors





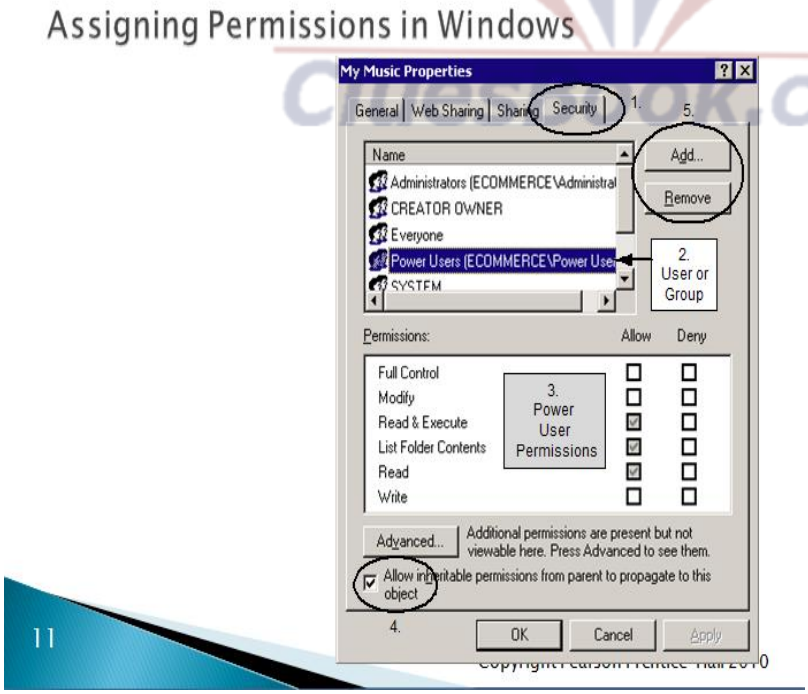
The Super User Account

- **Super User Account**
  - Every operating system has a super user account
  - The owner of this account can do anything
  - Called Administrator in Windows
  - Called root in UNIX
- **Hacking Root**
  - Goal is to take over the super user account
  - Will then “own the box”
  - Generically called hacking root
- **Appropriate Use of a Super User Account**

- Log in as an ordinary user
- Switch to super user only when needed
  - In Windows, the command is RunAs
  - In UNIX, the command is su (switch user)
- Quickly revert to ordinary account when super user privileges are no longer needed

Managing Permissions in Windows

- **Permissions**
  - Specify what the user or group can do to files, directories, and subdirectories
- **Assigning Permissions in Windows (See Fig.)**
  - Right click on file or directory
  - Select Properties, then Security tab
  - Select a user or group
  - Select the 6 standard permissions (permit or deny)
  - For more fine-grained control, 13 special permissions



4.5.7 Vulnerability Testing

- **Mistakes Will Be Made in Hardening**
  - So do vulnerability testing
- **Run Vulnerability Testing Software on Another Computer**
  - Run the software against the hosts to be tested
  - Interpret the reports about problems found on the server
    - This requires extensive security expertise
  - Fix them
- **Get Permission for Vulnerability Testing**
  - Looks like an attack
    - Must get prior written agreement
  - Vulnerability testing plan
    - An exact list of testing activities
    - Approval in writing to cover the tester
    - Supervisor must agree, in writing, to hold the tester blameless if there is damage
    - Tester must not diverge from the plan

4.5.8 Protecting Notebook Computers

- **Threats**
  - Loss or theft
  - Loss of capital investment
  - Loss of data that was not backed up
  - Loss of trade secrets
  - Loss of private information, leading to lawsuits
- **Backup**

- Before taking the notebook out
- Frequently during use outside the firm

- **Use a Strong Password**
  - If attackers bypass the operating system password, they get open access to encrypted data
  - The loss of login passwords is a major concern

4.5.9 Centralized PC Security Management

- **Network Access Control (NAC)**
  - Goal is to reduce the danger created by computers with malware
  - Control their access to the network

Stage 1: Initial Health Check

- Checks the "health" of the computer before allowing it into the network
- Choices:
  - Accept it
  - Reject it
  - Quarantine and pass it to a remediation server; retest after remediation

- **Stage 2: Ongoing Traffic Monitoring**
  - If traffic after admission indicates malware on the client, drop or remediate
  - Not all NAC systems do this

lecture 39

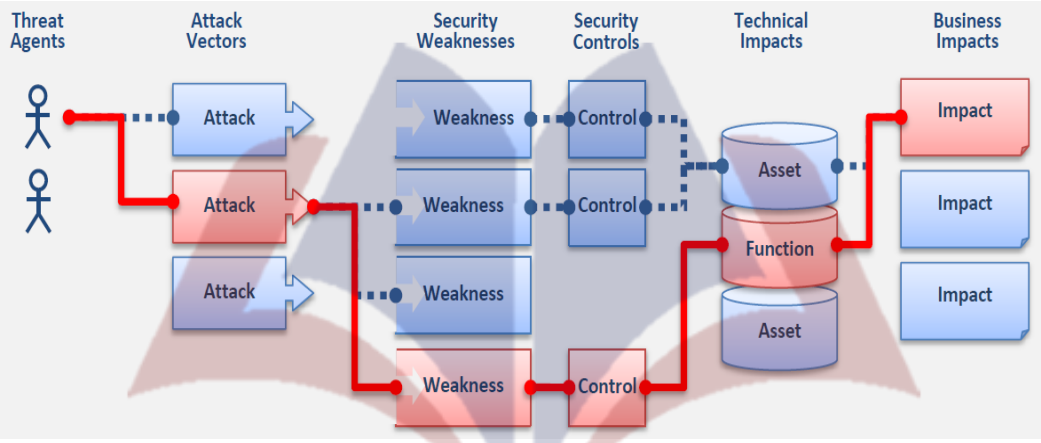
Web Application Security

The importance of applications and application security

4.6.1 Application Security Risks

Attackers can potentially use many different paths through your application to do harm to your business or organization. Each of these paths represents a risk that may, or may not, be serious enough to warrant attention.

What Are Application Security Risks ?



Application Security Risks

- Sometimes, these paths are trivial to find and exploit and sometimes they are extremely difficult.
- Similarly, the harm that is caused may range from nothing, all the way through putting you out of business.
- To determine the risk to your organization, you can evaluate the likelihood associated with each threat agent, attack vector, and security weakness and combine it with an estimate of the technical and business impact to your organization.
- Together, these factors determine the overall risk.

4.6.2 Applications Security Program

- Application security is no longer a choice. Between increasing attacks and regulatory pressures, organizations must establish an effective capability for securing their applications.
- Given the staggering number of applications and lines of code already in production, many organizations are struggling to get a handle on the enormous volume of vulnerabilities.
- OWASP recommends that organizations establish an application security program to gain insight and improve security across their application portfolio
- Achieving application security requires many different parts of an organization to work together efficiently, including security and audit, software development, and business and executive management.
- It requires security to be visible, so that all the different players can see and understand the organization's application security posture.
- It also requires focus on the activities and outcomes that actually help improve enterprise security by reducing risk in the most cost effective manner.

Some of the key activities in effective application security programs include:

Get Started

- Establish an application security program and drive adoption.
- Conduct a capability gap analysis comparing your organization to your peers to define key improvement areas and an execution plan.
- Gain management approval and establish an application security awareness campaign for the entire IT organization.

Risk Based Portfolio Approach

- Identify and prioritize your application portfolio from an inherent risk perspective.
- Create an application risk profiling model to measure and prioritize the applications in your portfolio. Establish assurance guidelines to properly define coverage and level of rigor required.
- Establish a common risk rating model with a consistent set of likelihood and impact factors reflective of your organization's tolerance for risk.

Enable With A Strong Foundation

- Establish a set of focused policies and standards that provide an application security baseline for all development teams to adhere to.

- Define a common set of reusable security controls that complement these policies and standards and provide design and development guidance on their use.
- Establish an application security training curriculum that is required and targeted to different development roles and topics.

Integrate Security Into Existing Processes

- Define and integrate security implementation and verification activities into existing development and operational processes.
- Activities include Threat Modeling, Secure Design & Review, Secure Code & Review, Pen Testing, Remediation, etc.
- Provide subject matter experts and support services for development and project teams to be successful.

Provide Management Visibility

- Manage with metrics. Drive improvement and funding decisions based on the metrics and analysis data captured. Metrics include adherence to security practices / activities, vulnerabilities introduced, vulnerabilities mitigated, application coverage, etc.
- Analyze data from the implementation and verification activities to look for root cause and vulnerability patterns to drive strategic and systemic improvements across the enterprise.

4.6.3 Application Security Threats

- **Executing Commands with the Privileges of a Compromised Application**
  - If an attacker takes over an application, the attacker can execute commands with the privileges of that application
  - Many applications run with super user (root) privileges

4.6.4 Hardening Applications

- **Basics**
  - Physical Security
  - Backup
  - Harden the Operating System
  - Etc.

- **Minimize Applications**
  - Main applications
  - Subsidiary applications
  - Be guided by security baselines
- **Create Secure Application Program Configurations**
  - Use baselines to go beyond default installation configurations for high-value targets
  - Avoid blank passwords or well-known default passwords
- **Install Patches for All Applications**
- **Minimize the Permissions of Applications**
  - If an attack compromises an application with low permissions, will not own the computer
- **Add Application Layer Authentication, Authorizations, and Auditing**
  - More specific to the needs of the application than general operating system logins
  - Can lead to different permissions for different users
- **Implement Cryptographic Systems**

  For communication with users

Securing Custom Applications

- **Custom Applications**
  - Written by a firm's programmers
  - Not likely to be well trained in secure coding
- **The Key Principle**
  - Never trust user input
  - Filter user input for inappropriate content

Lecture 40

4.6.5 Open Web Application Security Project (OWASP)

- The Open Web Application Security Project (OWASP) is an open community dedicated to enabling organizations to develop, purchase, and maintain applications that can be trusted. At OWASP you'll find free and open …
  - Application security tools and standards
  - Complete books on application security testing, secure code development, and security code review
  - Standard security controls and libraries
  - Local chapters worldwide
  - Cutting edge research
  - Extensive conferences worldwide
  - Mailing lists

  And more … all at www.owasp.org

4.6.6 OWASP Top 10

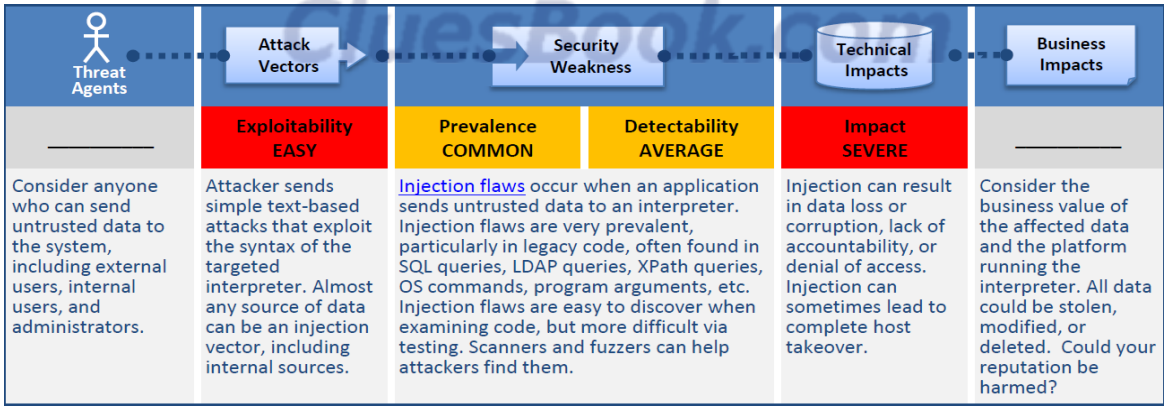
- The primary aim of the OWASP Top 10 is to educate developers, designers, architects, managers, and organizations about the consequences of the most important web application security weaknesses.
- The Top 10 provides basic techniques to protect against these high risk problem areas – and also provides guidance on next steps
- The goal of the Top 10 project is to raise awareness about application security by identifying some of the most critical risks facing organizations.
- The Top 10 project is referenced by many standards, books, tools, and organizations, including MITRE, PCI DSS, DISA, FTC, and many more.
- The OWASP Top 10 was first released in 2003, minor updates were made in 2004 and 2007, and this is the 2010 release.
- This release of the OWASP Top 10 marks this project's eighth year of raising awareness of the importance of application security risks.
-

OWASP Top 10
Application Security Risks

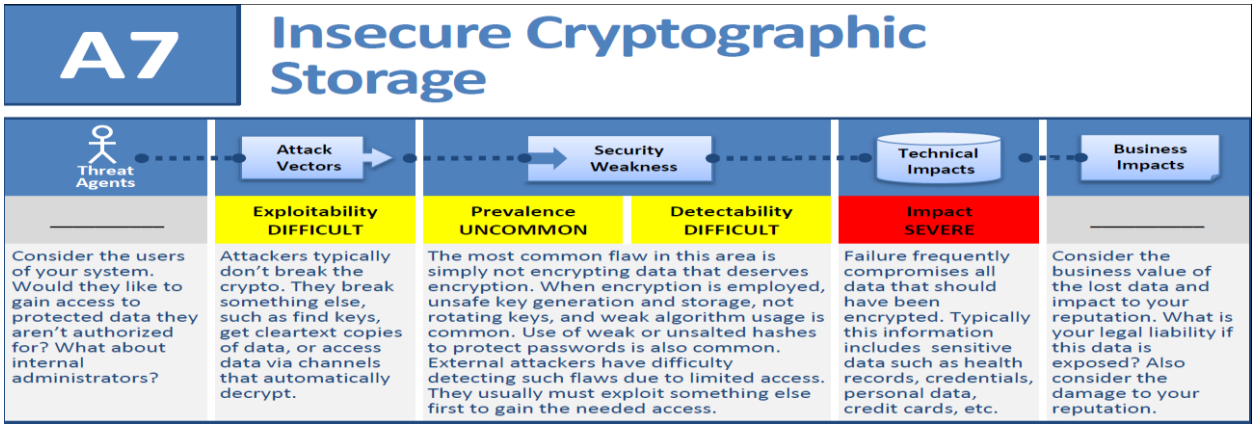
- A1: Injection flaws,
  - Such as SQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing unauthorized data.
- **A2: Cross Site Scripting (XSS)**
  - XSS flaws occur whenever an application takes untrusted data and sends it to a web browser without proper validation. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.
- **A3: Broken Authentication & Session Management**
  - Application functions related to authentication and session management are often not implemented correctly, allowing attackers to compromise passwords, keys, session tokens, or exploit other implementation flaws to assume other users' identities.
- **A4: Insecure Direct Object References**
  - A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, or database key. Without an access control check or other protection, attackers can manipulate these references to access unauthorized data.
- **A5: Cross-Site Request Forgery**
  - A CSRF attack forces a logged-on victim's browser to send a forged HTTP request, including the victim's session cookie and any other automatically included authentication information, to a vulnerable web application. This allows the attacker to force the victim's browser to generate requests the vulnerable application thinks are legitimate requests from the victim.
- **A6: Security Misconfiguration**
  - Good security requires having a secure configuration defined and deployed for the application, frameworks, application server, web server, database server, and platform. All these settings should be defined, implemented, and maintained as many are not shipped with secure defaults. This includes keeping all software up to date, including all code libraries used by the application.

- **A7: Insecure Cryptographic Storage**
  - Many web applications do not properly protect sensitive data, such as credit cards, SSNs, and authentication credentials, with appropriate encryption or hashing. Attackers may steal or modify such weakly protected data to conduct identity theft, credit card fraud, or other crimes.
- **A8: Failure To Restrict URL Access**
  - Many web applications check URL access rights before rendering protected links and buttons. However, applications need to perform similar access control checks each time these pages are accessed, or attackers will be able to forge URLs to access these hidden pages anyway.
- **A9: Insufficient Transport Layer Protection**
  - Applications frequently fail to authenticate, encrypt, and protect the confidentiality and integrity of sensitive network traffic. When they do, they sometimes support weak algorithms, use expired or invalid certificates, or do not use them correctly.
- **A10: Unvalidated Redirects & Forwards**
  - Web applications frequently redirect and forward users to other pages and websites, and use untrusted data to determine the destination pages. Without proper validation, attackers can redirect victims to phishing or malware sites, or use forwards to access unauthorized pages.

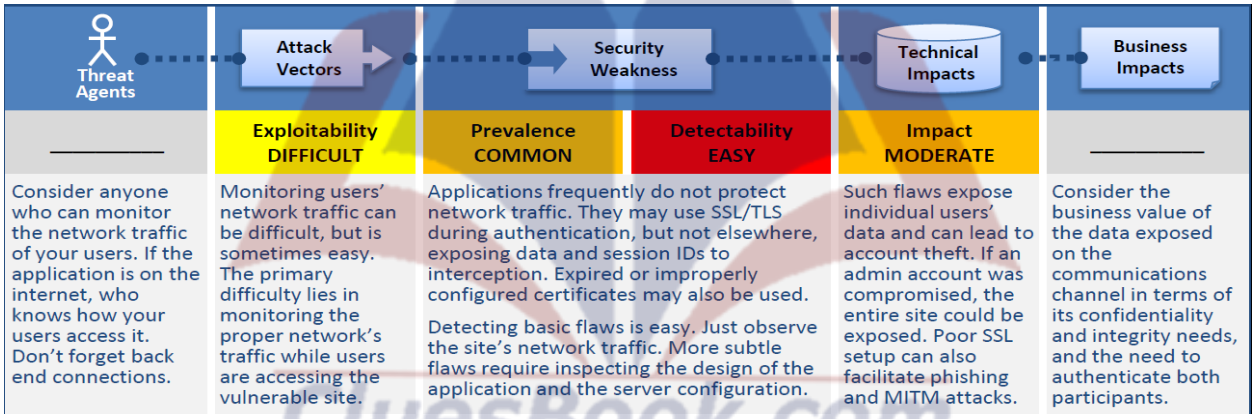
A1: Injection



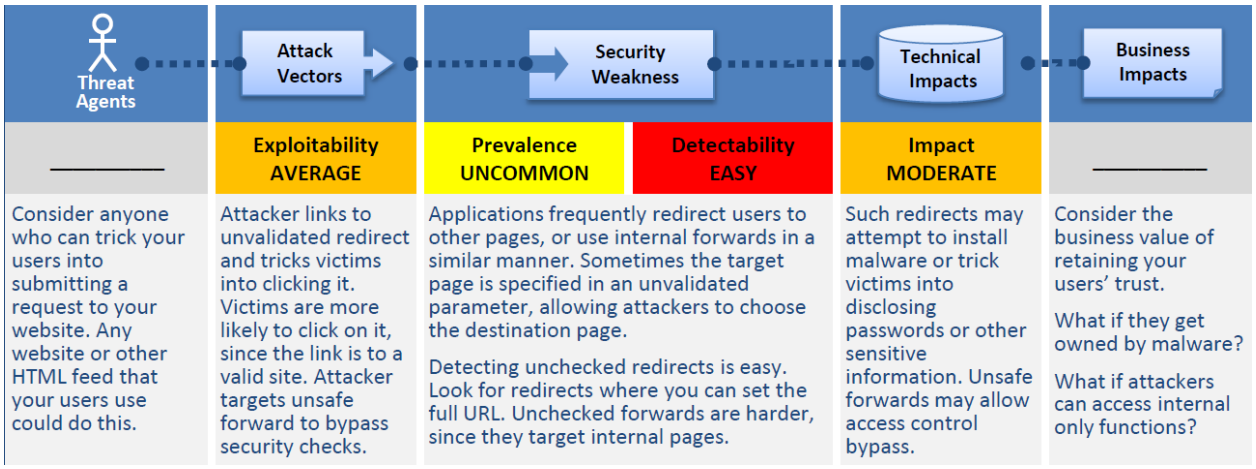
Insecure Cryptographic Storage



A9: Insufficient Transport Layer Protection



A10: Unvalidated Redirects & Forwards



Don't Stop At 10

- Don't stop at 10. There are hundreds of issues that could affect the overall security of a web application as discussed in the OWASP Developer's Guide.
- This is essential reading for anyone developing web applications today.
- Guidance on how to effectively find vulnerabilities in web applications are provided in the OWASP Testing Guide and OWASP Code Review Guide.

Application Security Verification Standard (ASVS)

- When you're ready to move on and focus on establishing strong application security controls, OWASP recently produced the Application Security Verification Standard (ASVS) as a guide to organizations and application reviewers on what to verify.

4.6.7 SDLC

- Secure web applications are only possible when a secure software development lifecycle is used. For guidance on how to implement a secure SDLC, OWASP recently released the Open Software Assurance Maturity Model (SAMM).

Lecture 40

Information Security Testing & Assessment

4.7.1 Information Security Assessments

- An information security *assessment is the process of determining how effectively an entity being assessed (e.g., host, system, network, procedure, person—known as the assessment object) meets specific security objectives*

4.7.2 Technical Assessment Techniques

- Dozens of technical security testing and examination techniques exist that can be used to assess the security posture of systems and networks.
- The most commonly used techniques are grouped into the following three categories:
  - 1. Review techniques
  - 2. Target identification and analysis techniques
  - 3. Target vulnerability validation techniques

Testing Viewpoints: External

- External security testing is conducted from outside the organization's security perimeter. This offers the ability to view the environment's security posture as it appears outside the security perimeter—usually as seen from the Internet—with the goal of revealing vulnerabilities that could be exploited by an external attacker.

Testing Viewpoints: Internal

- For internal security testing, assessors work from the internal network and assume the identity of a trusted insider or an attacker who has penetrated the perimeter defenses.
- This kind of testing can reveal vulnerabilities that could be exploited, and demonstrates the potential damage this type of attacker could cause.
- Internal security testing also focuses on system-level security and configuration—including application and service configuration, authentication, access control, and system hardening.

Testing Viewpoints: Overt

- Overt security testing, also known as white hat testing, involves performing external and/or internal testing with the knowledge and consent of the organization's IT staff, enabling comprehensive evaluation of the network or system security posture.
- Because the IT staff is fully aware of and involved in the testing, it may be able to provide guidance to limit the testing's impact.

Testing Viewpoints: Covert

- Covert security testing, also known as black hat testing, takes an adversarial approach by performing testing without the knowledge of the organization's IT staff but with the full knowledge and permission of upper management.

- Some organizations designate a trusted third party to ensure that the target organization does not initiate response measures associated with the attack without first verifying that an attack is indeed underway (e.g., that the activity being detected does not originate from a test).

4.7.3 Vulnerability Scanning

- Vulnerability scanning identifies hosts and host attributes (e.g., operating systems, applications, open ports), but it also attempts to identify vulnerabilities rather than relying on human interpretation of the scanning results.
- Vulnerability scanning can help identify outdated software versions, missing patches, and mis-configurations, and validate compliance with or deviations from an organization's security policy.
- This is done by identifying the operating systems and major software applications running on the hosts and matching them with information on known vulnerabilities stored in the scanners' vulnerability databases.

Vulnerability Scanning Functionality

- Vulnerability scanners can:
  - Check compliance with host application usage and security policies
  - Provide information on targets for penetration testing
  - Provide information on how to mitigate discovered vulnerabilities.

Vulnerability Scanning Mechanisms

- Vulnerability scanners can be run against a host either locally or from the network. Some network-based scanners have administrator-level credentials on individual hosts and can extract vulnerability information from hosts using those credentials.
- Other network-based scanners do not have such credentials and must rely on conducting scanning of networks to locate hosts and then scan those hosts for vulnerabilities.
- In such cases, network-based scanning is primarily used to perform network discovery and identify open ports and related vulnerabilities—in most cases, it is not limited by the OS of the targeted systems.

Vulnerability Scanning

- Assessors can also request that personal or host-based firewalls be configured to permit traffic from test system IP addresses during the assessment period.
- These steps will give assessors increased insight into the network, but do not accurately reflect the capabilities of an external attacker—although they may offer a better indication of the capabilities available to a malicious insider or an external attacker with access to another host on the internal network.
- Assessors can also perform scanning on individual hosts.
- A vulnerability scanner is a relatively fast and easy way to quantify an organization's exposure to surface vulnerabilities.
- A surface vulnerability is a weakness that exists in isolation, independent from other vulnerabilities. The system's behaviors and outputs in response to attack patterns submitted by the scanner are compared against those that characterize the signatures of known vulnerabilities, and the tool reports any matches that are found.
- Besides signature-based scanning, some vulnerability scanners attempt to simulate the reconnaissance attack patterns used to probe for exposed, exploitable vulnerabilities, and report the vulnerabilities found when these techniques are successful.

Vulnerability Scanning Challenges

- vulnerabilities is that they rarely exist in isolation. For example, there could be several low-risk vulnerabilities that present a higher risk when combined.
- Scanners are unable to detect vulnerabilities that are revealed only as the result of potentially unending combinations of attack patterns.
- The tool may assign a low risk to each vulnerability, leaving the assessor falsely confident in the security measures in place. A more reliable way of identifying the risk of vulnerabilities in aggregate is through penetration testing,
- Another problem with identifying the risk level of vulnerabilities is that vulnerability scanners often use their own proprietary methods for defining the levels.
- For example, one scanner might use the levels low, medium, and high, while another scanner might use the levels informational, low, medium, high, and critical.
- This makes it difficult to compare findings among multiple scanners.
- Network-based vulnerability scanning has some significant weaknesses. As with network sniffing and discovery, this type of scanning uncovers vulnerabilities only for active systems.

- This generally covers surface vulnerabilities, and is unable to address the overall risk level of a scanned network.
- Although the process itself is highly automated, vulnerability scanners can have a high false positive error rate (i.e., reporting vulnerabilities when none exist). An individual with expertise in networking and OS security should interpret the results.

Vulnerability Scanning

- Another significant limitation of vulnerability scanners is that, like virus scanners and IDSs, they rely on a repository of signatures.
- This requires the assessors to update these signatures frequently to enable the scanner to recognize the latest vulnerabilities.
- Before running any scanner, an assessor should install the latest updates to its vulnerability database.
- Some vulnerability scanner databases are updated more regularly than others—this update frequency should be a major consideration when selecting a vulnerability scanner.
- Most vulnerability scanners allow the assessor to perform different levels of scanning that vary in terms of thoroughness.
- While more comprehensive scanning may detect a greater number of vulnerabilities, it can slow the overall scanning process.
- Less comprehensive scanning can take less time, but identifies only well-known vulnerabilities.
- It is generally recommended that assessors conduct a thorough vulnerability scan if resources permit.
- Vulnerability scanning is a somewhat labor-intensive activity that requires a high degree of human involvement to interpret results.
- It may also disrupt network operations by taking up bandwidth and slowing response times.
- Nevertheless, vulnerability scanning is extremely important in ensuring that vulnerabilities are mitigated before they are discovered and exploited by adversaries.
- As with all pattern-matching and signature-based tools, application vulnerability scanners typically have high false positive rates.

- Assessors should configure and calibrate their scanners to minimize both false positives and false negatives to the greatest possible extent, and meaningfully interpret results to identify the real vulnerabilities.

Lecture 42

4.7.4 Target Vulnerability Validation Techniques

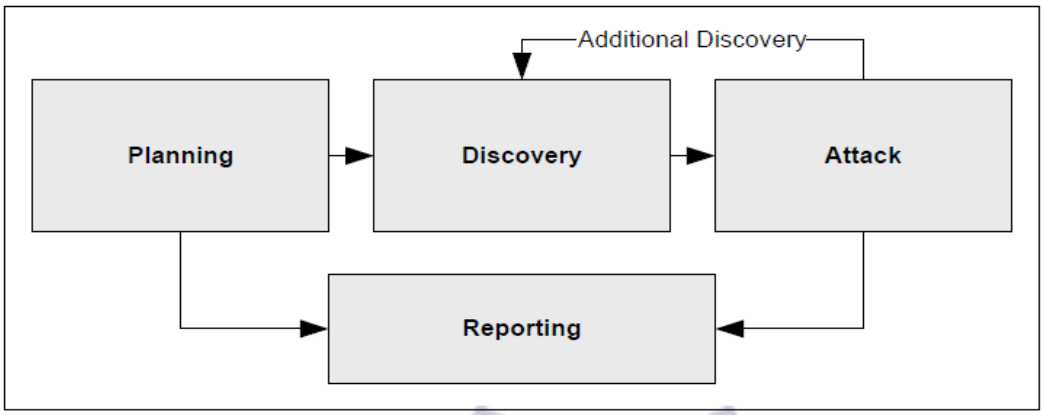
- These techniques use information produced from target identification and analysis to further explore the existence of potential vulnerabilities.
- The objective is to prove that a vulnerability exists, and to demonstrate the security exposures that occur when it is exploited.
- Target vulnerability validation involves the greatest amount of risk in assessments, since these techniques have more potential to impact the target system or network than other techniques.

4.7.5 Target Vulnerability Validation Techniques: Penetration Testing

- Penetration testing is security testing in which assessors mimic real-world attacks to identify methods for circumventing the security features of an application, system, or network.
- It often involves launching real attacks on real systems and data that use tools and techniques commonly used by attackers.
- Most penetration tests involve looking for combinations of vulnerabilities on one or more systems that can be used to gain more access than could be achieved through a single vulnerability.
- Penetration testing can also be useful for determining:
    - How well the system tolerates real world-style attack patterns
    - The likely level of sophistication an attacker needs to successfully compromise the system
    - Additional countermeasures that could mitigate threats against the system
    - Defenders' ability to detect attacks and respond appropriately
- Penetration testing can be invaluable, but it is labor-intensive and requires great expertise to minimize the risk to targeted systems.
- Systems may be damaged or otherwise rendered inoperable during the course of penetration testing, even though the organization benefits in knowing how a system could be rendered inoperable by an intruder.
- Although experienced penetration testers can mitigate this risk, it can never be fully eliminated.

- Penetration testing should be performed only after careful consideration, notification, and planning.

Penetration Testing Phases



Planning (1)

- The Figure represents the four phases of penetration testing.
- In the planning phase, rules are identified, management approval is finalized and documented, and testing goals are set.
- The planning phase sets the groundwork for a successful penetration test. No actual testing occurs in this phase.

Discovery (2)

- The discovery phase of penetration testing includes two parts.
- The first part is the start of actual testing, and covers information gathering and scanning. Network port and service identification, is conducted to identify potential targets.
- In addition to port and service identification, other techniques are used to gather information on the targeted network:
- Host name and IP address information can be gathered through many methods, including DNS interrogation, InterNIC (WHOIS) queries, and network sniffing (generally only during internal tests)
- Employee names and contact information can be obtained by searching the organization's Web servers or directory servers

- System information, such as names and shares can be found through methods such as NetBIOS enumeration (generally only during internal tests) and Network Information System (NIS) (generally only during internal tests) Application and service information, such as version numbers, can be recorded through banner grabbing.
- In some cases, techniques such as dumpster diving and physical walkthroughs of facilities may be used to collect additional information on the targeted network, and may also uncover additional information to be used during the penetration tests, such as passwords written on paper.
- The second part of the discovery phase is vulnerability analysis, which involves comparing the services, applications, and operating systems of scanned hosts against vulnerability databases (a process that is automatic for vulnerability scanners) and the testers' own knowledge of vulnerabilities.

Executing (3)

- Executing an attack is at the heart of any penetration test. The Figure represents the individual steps of the attack phase—the process of verifying previously identified potential vulnerabilities by attempting to exploit them.
- If an attack is successful, the vulnerability is verified and safeguards are identified to mitigate the associated security exposure.
- In many cases, exploits that are executed do not grant the maximum level of potential access to an attacker.
- They may instead result in the testers learning more about the targeted network and its potential vulnerabilities, or induce a change in the state of the targeted network's security.
- Some exploits enable testers to escalate their privileges on the system or network to gain access to additional resources.
- If this occurs, additional analysis and testing are required to determine the true level of risk for the network, such as identifying the types of information that can be gleaned, changed, or removed from the system.
- In the event an attack on a specific vulnerability proves impossible, the tester should attempt to exploit another discovered vulnerability.

Penetration Testing Phases

- If testers are able to exploit a vulnerability, they can install more tools on the target system or network to facilitate the testing process. These tools are used to gain access to additional

systems or resources on the network, and obtain access to information about the network or organization.

- Testing and analysis on multiple systems should be conducted during a penetration test to determine the level of access an adversary could gain.
- This process is represented in the feedback loop in the Figure between the attack and discovery phase of a penetration test.

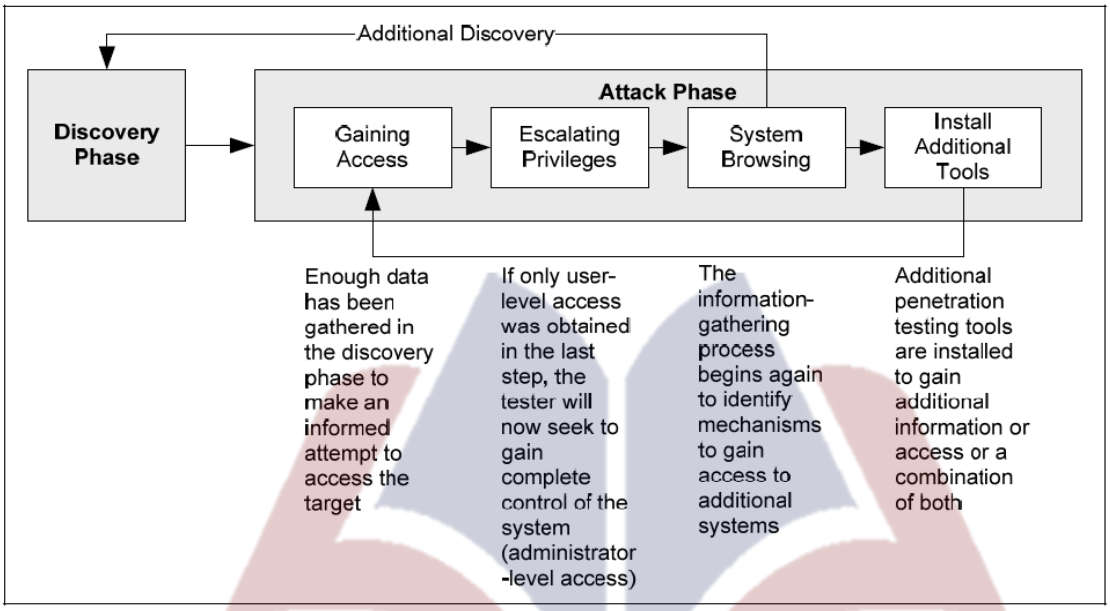


FIGURE: Attack Phase Steps with Loopback to Discovery Phase

- While vulnerability scanners check only for the possible existence of a vulnerability, the attack phase of a penetration test exploits the vulnerability to confirm its existence. Most vulnerabilities exploited by penetration testing fall into the following categories:

4.7.6 Possible Vulnerabilities Exploited By Penetration Testing

- Mis-configurations
- Kernel Flaws
- Buffer overflows
- Insufficient input validation
- Incorrect file and directory permissions

Reporting (4)

- The reporting phase occurs simultaneously with the other three phases of the penetration test (see Figure).
- In the discovery and attack phases, written logs are usually kept and periodic reports are made to system administrators and/or management.
- At the conclusion of the test, a report is generally developed to describe identified vulnerabilities, present a risk rating, and give guidance on how to mitigate the discovered weaknesses.

4.7.6 Penetration Testing Logistics

- Penetration test scenarios should focus on locating and targeting exploitable defects in the design and implementation of an application, system, or network. Tests should reproduce both the most likely and most damaging attack patterns—including worst-case scenarios such as malicious actions by administrators.
- Since a penetration test scenario can be designed to simulate an inside attack, an outside attack, or both, external and internal security testing methods are considered. If both internal and external testing is to be performed, the external testing usually occurs first.
- Penetration testing also poses a high risk to the organization's networks and systems because it uses real exploits and attacks against production systems and data.
- Because of its high cost and potential impact, penetration testing of an organization's network and systems on an annual basis may be sufficient. Also, penetration testing can be designed to stop when the tester reaches a point when an additional action will cause damage.
- The results of penetration testing should be taken seriously, and any vulnerabilities discovered should be mitigated.
- Results, when available, should be presented to the organization's managers. Organizations should consider conducting less labor-intensive testing activities on a regular basis to ensure that they are maintaining their required security posture.
- A well-designed program of regularly scheduled network and vulnerability scanning, interspersed with periodic penetration testing, can help prevent many types of attacks and reduce the potential impact of successful ones.

lecture 43

Security Features On Switches

4.8.1 Layer 2 Security

- The data-link layer (Layer 2 of the OSI model) provides the functional and procedural means to transfer data between network entities with interoperability and interconnectivity to other layers, but from a security perspective, the data-link layer presents its own challenges.
- Network security is only as strong as the weakest link, and Layer 2 is no exception.
- Applying first-class security measures to the upper layers (Layers 3 and higher) does not benefit your network if Layer 2 is compromised.
- Switches offer a wide range of security features at Layer 2 to protect the network traffic flow and the devices themselves.

4.8.2 Layer 2 Terminology

- Switch
- Data Link Layer
- MAC (Media Access Control) Address
- Switch Port
- Address Resolution Protocol (ARP)
- Dynamic Host Configuration Protocol (DHCP)
- Spanning Tree Protocol (STP)
- Virtual LAN (VLAN)

4.8.3 Types Of Layer 2 Attacks

- **CAM Table Overflow—MAC Attack**
  - Switches do not have unlimited memory; hence, the CAM table has a fixed allocated memory space.
  - This makes the switch vulnerable to exploitation from sniffing by flooding the switch with a large number of randomly generated invalid source and destination MAC addresses, until the CAM table fills up and no new entries can be accepted.

- When this happens, the switch cannot handle any further frames and acts in a hub mode, in which it broadcasts all received frames to all the ports on the switch, essentially turning it into one big broadcast domain.

- **MAC Spoofing Attack**
  - MAC spoofing is a technique used to spoof source MAC addresses to impersonate other hosts or devices in a network.
  - This is different from an ARP spoofing attack. In ARP spoofing, the switch is misguided by poisoning the ARP cache, whereas with MAC spoofing, the switch is confused to believe two ports have the same MAC address, thereby forcing the switch to attempt to forward frames destined for the trusted host to the attacker.

- **ARP Spoofing Attack**
  - An ARP spoofing attack is a method in which an intruder attempts to disguise its source MAC address by impersonating another host on the network.
  - In ARP spoofing, the switch is misguided by poisoning the ARP cache. ARP spoofing is generally motivated to aid in making other DoS and MITM-type attacks possible.

- **Spanning-Tree Attacks**
  - Spanning Tree Protocol attacks are methods whereby the intruder assumes the identity of a root bridge in the topology by broadcasting forged Bridge Protocol Data Unit (BPDU) messages in an attempt to force spanning-tree recalculations and thereby disrupt the network data flow.
  - Spanning Tree Protocol attacks can be mitigated using the BPDU Guard and the ROOT Guard features available on Cisco Catalyst switches.
  - These features are designed to enforce the placement of the root bridge in the spanning-tree topology and can also be used to prevent rogue switch network extensions.

- **DHCP Spoofing and Starvation Attacks**
  - DHCP spoofing and starvation attacks are methods to exhaust the DHCP address pool on the DHCP server, resulting in resource starvation where no DHCP addresses are available to be assigned to legitimate users.

4.8.4 Port-Level Traffic Controls

- Port-based traffic control features can be used to provide protection at the port level.
- Catalyst switches offer:

- Storm Control
- Protected Ports
- Private Virtual Local Area Network (PVLAN)
- Port Blocking
- Port Security features

Storm Control

- A LAN storm typically occurs when hostile packets are flooded on the LAN segment, creating unnecessary and excessive traffic resulting in network performance degradation.
- Several factors can cause a storm on a network; examples include errors in the protocol-stack implementation or a loophole that is exploited in a device configuration.
- The Storm Control feature prevents regular network traffic from being disrupted by a broadcast, multicast, or unicast packet storm on any of the physical interfaces.
- The traffic storm control (also known as a traffic suppression feature) monitors inbound packets over a 1-second interval and compares it to the configured storm-control suppression level by using one of the following methods to measure activity:
- The percentage of total available bandwidth of the port allocated for the broadcast, multicast, or unicast traffic
- Traffic rate over a 1-second interval in packets per second at which broadcast, multicast, or unicast packets are received on an interface
- With either method, the port blocks traffic when a threshold is reached, filtering out all subsequent packets.
- As the port remains in a blocked state, the traffic continues to be dropped until the traffic rate drops below the suppression level, at which point the port resumes normal traffic forwarding.
- The storm-control action {shutdown | trap} command is used to specify the action to be taken when a storm is detected.
- By default, the storm traffic is suppressed when no action is configured.

Protected Ports (PVLAN Edge)

- In some network environments, there is a requirement for no traffic to be seen or forwarded between host(s) on the same LAN segment, thereby preventing interhost communications.
- The PVLAN edge feature provisions this isolation by creating a firewall-like barrier, thereby blocking any unicast, broadcast, or multicast traffic among the protected ports on the switch.
- Note that the significance of the protected port feature is limited to the local switch, and there is no provision in the PVLAN edge feature to isolate traffic between two "protected" ports located on different switches.
- For this purpose, the PVLAN feature can be used.
- The PVLAN edge offers the following features:
    - The switch will not forward traffic (unicast, multicast, or broadcast) between ports that are configured as protected. Data traffic must be routed via a Layer 3 device between the protected ports.
    - Control traffic, such as routing protocol updates, is an exception and will be forwarded between protected ports.
    - Forwarding behavior between a protected port and a nonprotected port proceeds normally per default behavior.

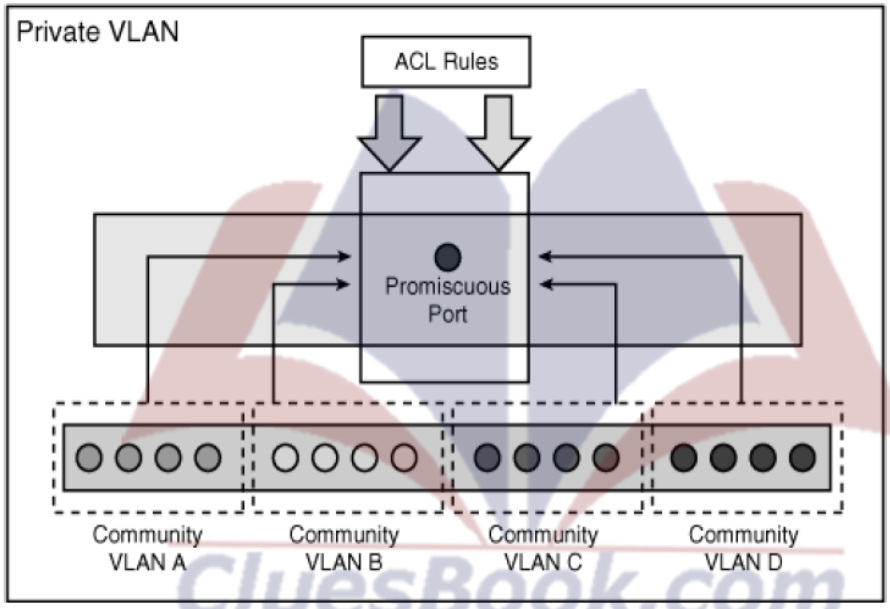
```
Switch(config)# interface Fastethernet0/1
Switch(config-if)# switchport protected
Switch(config-if)# end

Switch# show interfaces FastEthernet 0/1 switchport
Name: Fa0/1
Switchport: Enabled
Administrative Mode: static access
...
Protected: true
```

Private VLAN (PVLAN)

- As discussed in the "Protected Ports (PVLAN Edge") section, the PVLAN feature prevents interhost communications providing port-based security among adjacent ports within a VLAN across one or more switches.

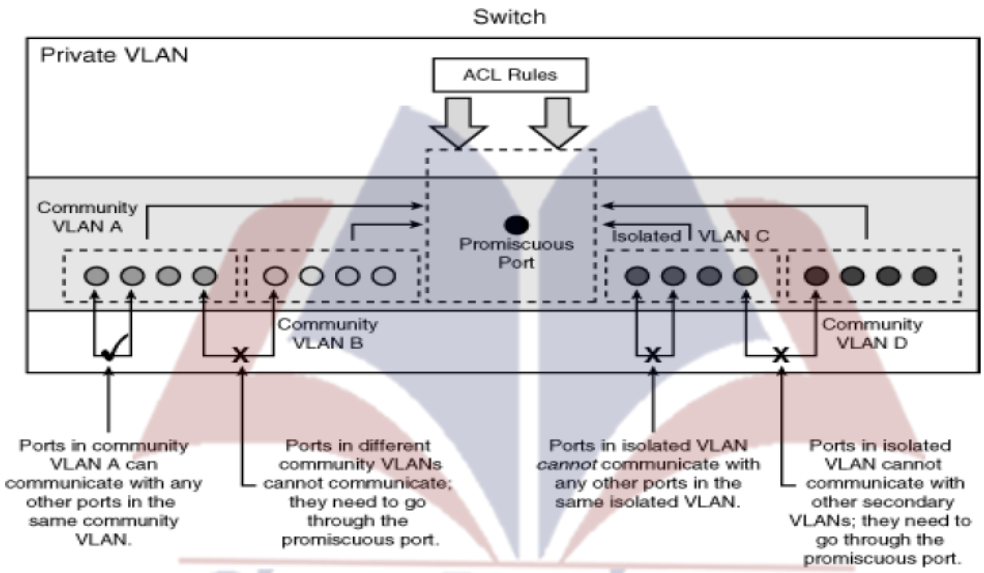
- PVLAN provides Layer 2 isolation to quarantine hosts from one another among ports within the same PVLAN.
- Access ports in a PVLAN are allowed to communicate only with the certain designated router ports. In most cases, this is the default gateway IP address.
- Private VLANs and normal VLANs can coexist on the same switch.
- To prevent interhost and interserver communication, PVLAN can be used efficiently because the number of subnets or VLANs is greatly reduced, although the segmented approach within a single network segment is still achieved.
- The number is reduced because there is no need to create extra subnet/VLANs.



- Promiscuous
  - A promiscuous port can communicate with all interfaces, including the isolated and community ports within a PVLAN. The function of the promiscuous port is to move traffic between ports in community or isolated VLANs.
  - It can use access lists to identify which traffic can pass between these VLANs.
  - Only one promiscuous port is allowed per single PVLAN, and it serves all the community and isolated VLANs in the Private VLAN.
- Isolated

  - An isolated PVLAN port has complete Layer 2 segregation from all the other ports within the same PVLAN, but not from the promiscuous ports.
  - Traffic from the isolated port is forwarded only to the promiscuous ports and none other.

- Community
  - Community ports are logically combined groups of ports in a common community and can pass traffic among themselves and with promiscuous ports.
  - Ports are separated at Layer 2 from all other interfaces in other communities or isolated ports within their PVLAN.



Port Blocking

- When a packet arrives at the switch, the switch performs a lookup for the destination MAC address in the MAC address table to determine which port it will use to send the packet out to send on.
- If no entry is found in the MAC address table, the switch will broadcast (flood) unknown unicast or multicast traffic out to all the ports in the same VLAN(broadcast domain).
- Forwarding an unknown unicast or multicast traffic to a protected port could raise security issues.

- Unknown unicast or multicast traffic can be blocked from being forwarded by using the port blocking feature.

```
Switch(config)# interface Fastethernet0/1
Switch(config-if)# switchport block multicast
Switch(config-if)# switchport block unicast
Switch(config-if)# end
Switch# show interfaces FastEthernet 0/1 switchport
Name: Fa0/1
Switchport: Enabled
Administrative Mode: static access
...
Protected: true
Unknown unicast blocked: enabled
Unknown multicast blocked: enabled
Appliance trust: none
```

Port Security

- Port security is a dynamic feature that prevents unauthorized access to a switch port.
- The port security feature can be used to restrict input to an interface by identifying and limiting the MAC addresses of the hosts that are allowed to access the port.
- When secure MAC addresses are assigned to a secure port, the switch does not forward packets with source MAC addresses outside the defined group of addresses.
- To understand this process, think of the analogy of a secure car park facility, where a spot is reserved and marked with a particular car registration number so that no other car is allowed to park at that spot.
- Similarly, a switch port is configured with the secure MAC address of a host, and no other host can connect to that port with any other MAC address.
- Port security can be implemented in the following three ways:
    - Static secure MAC addresses are manually configured and stored in the MAC address table and in the configuration.
    - Dynamic secure MAC addresses are dynamically learned, stored in the MAC address table, but removed when the switch is reloaded or powered down.
    - Sticky secure MAC addresses are the combination of items 1 and 2 in this list. They can be learned dynamically or configured statically and are stored in the MAC address table

and in the configuration. When the switch reloads, the interface does not need to dynamically discover the MAC addresses if they are saved in the configuration file.

- The example shows how to configure a static secure MAC address on a port and enable sticky learning

```
Switch(config)# interface Fastethernet0/1
Switch(config-if)# switchport mode access
Switch(config-if)# switchport port-security
Switch(config-if)# switchport port-security mac-address 0009.6B90.F4FE
Switch(config-if)# switchport port-security mac-address sticky
Switch(config-if)# end
```

The example shows how to configure a maximum of 10 secure MAC addresses on VLAN 5 on port interface FastEthernet 0/2.

```
Switch(config)# interface Fastethernet0/2
Switch(config-if)# switchport mode access
Switch(config-if)# switchport port-security maximum 10 vlan 5
Switch(config-if)# end
```

Access Control Lists On Switches

- The switch supports the following four types of ACLs for traffic filtering:
  - Router ACL
  - Port ACL
  - VLAN ACL
  - MAC ACL
- Router ACLs
  - As the name implies, Router ACLs are similar to the IOS ACL and can be used to filter network traffic on the switched virtual interfaces (SVI).
  - SVI interfaces are Layer 3 interfaces on VLANs, on Layer 3 physical interfaces, and on Layer 3 EtherChannel interfaces.
- Port ACLs:
  - Port ACLs are similar to Router ACLs but are supported on physical interfaces and configured on Layer 2 interfaces on a switch.

  - Port ACL supports only inbound traffic filtering.
  - Processing of the Port ACL is similar to that of the Router ACLs; the switch examines ACLs associated with features configured on a given interface and permits or denies packet forwarding based on packet-matching criteria in the ACL.
  - The main benefit with Port ACL is that it can filter IP traffic (using IP access lists) and non-IP traffic (using MAC access list). Both types of filtering can be achieved—that is, a Layer 2 interface can have both an IP access list and a MAC access list applied to it at the same time.

- Port ACLs:
  - The main benefit with Port ACL is that it can filter IP traffic (using IP access lists) and non-IP traffic (using MAC access list).
  - Both types of filtering can be achieved—that is, a Layer 2 interface can have both an IP access list and a MAC access list applied to it at the same time.
- VLAN ACLs:
  - VLAN ACL (also called VLAN map) provides packet filtering for all types of traffic that are bridged within a VLAN or routed into or out of the VLAN.
  - Unlike Router ACL, VACL is not defined by a direction (input or output).
  - All packets entering the VLAN (bridged or routed) are checked against the VACL.
  - VACLs are processed in hardware, so there is no performance penalty in processing them. Therefore, they are also referred to as wire-speed ACLs.

- MAC ACLs:
  - MAC ACL, also known as Ethernet ACL, can filter non-IP traffic on a VLAN and on a physical Layer 2 interface by using MAC addresses in an ACL.
  - MAC ACL supports only inbound traffic filtering.

Lecture 44

Spanning Tree Protocol Features

4.8.5 Spanning Tree Protocol Features

- Spanning Tree Protocol (STP) resolves redundant topologies into loop-free, treelike topologies.
- When switches are interconnected via multiple paths, STP prevents loops from being formed.
- An STP loop (or forwarding loops) can occur when the entire network fails because of a hardware failure, a configuration issue, or a network attack.
- STP loops can be costly, causing major network outages. The following STP features can be used to improve the stability of the Layer 2 networks.

- **Bridge Protocol Data Unit (BPDU) Guard**
  - Bridge protocol data units (BPDU) are data messages exchanged between bridges/switches using spanning tree protocol to detect loops in a network topology.
  - BPDU contains management and control data information that is used to determine the root bridge and establish the port roles—for example: root, designated, or blocked port.
  - The BPDU Guard feature is designed to keep the active topology predictable and to enhance switch network reliability by enforcing the STP domain borders.
  - The BPDU Guard feature provides a secure response to invalid configurations because you must manually put the interface back in service.
  - In a service-provider network environment, the BPDU Guard feature can be used to prevent an access port from participating in the spanning tree.

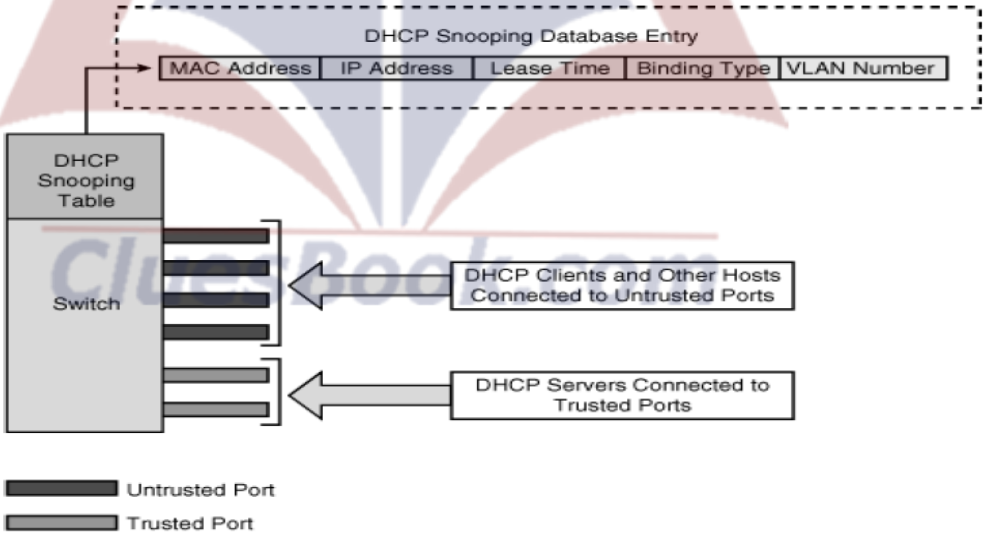
- **Root Guard**
  - In a switched network environment with shared administrative control or in a service provider (SP) environment where there are many connections to other switches (into customer networks), it is important to identify the correct placement of the root bridge.
  - If possible, it is also important to identify a specific predetermined location to achieve an optimal forwarding loop-free topology.
  - There is no mechanism in the standard STP to enforce the position of the root bridge, as any bridge in a network with a lower bridge ID can assume the role of the root bridge.

    Loop Guard

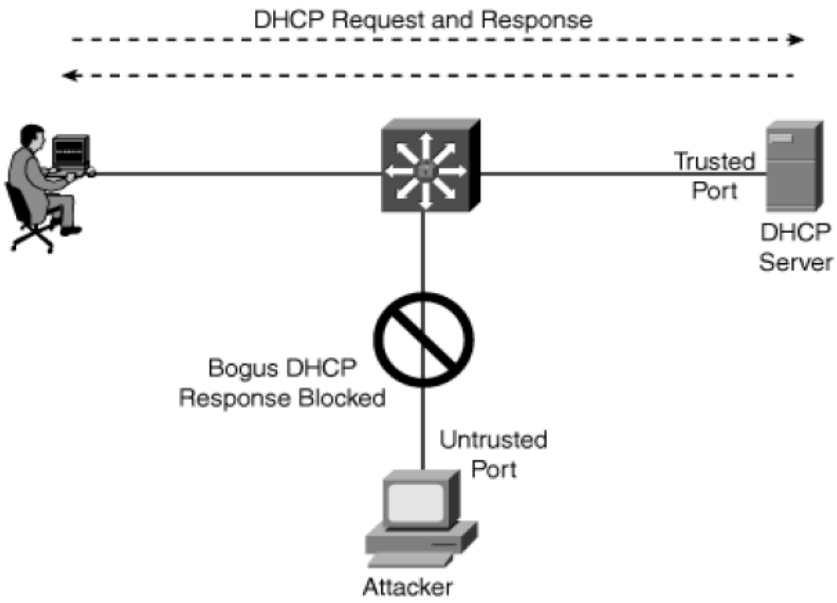
- The Loop Guard feature provides an additional layer of protection against the Layer 2 forwarding loops (STP loops) by preventing alternative or root ports from becoming designated ports because of a failure resulting in a unidirectional link.
- This feature works best when enabled on all switches across a network.

Dynamic Host Configuration Protocol (DHCP) Snooping

- The DHCP Snooping feature provides network protection from rogue DHCP servers.
- It creates a logical firewall between untrusted hosts and DHCP servers. The switch builds and maintains a DHCP snooping table (also called DHCP binding database), shown in the Figure.
- In addition, the switch uses this table to identify and filter untrusted messages from the network.
- The switch maintains a DHCP binding database that keeps track of DHCP addresses that are assigned to ports, as well as filtering DHCP messages from untrusted ports.
- For incoming packets received on untrusted ports, packets are dropped if the source MAC address does not match MAC in the binding table entry.



The Figure illustrates the DHCP Snooping feature in action, showing how the intruder is blocked on the untrusted port when it tries to intervene by injecting a bogus DHCP response packet to a legitimate conversation between the DHCP client and server.



- For DHCP Snooping to function correctly, all DHCP servers connected to the switch must be configured as trusted interfaces.
- A trusted interface can be configured by using the ip dhcp snooping trust interface configuration command.
- All other DHCP clients connected to the switch and other ports receiving traffic from outside the network or firewall should be configured as untrusted by using the no ip dhcp snooping trust interface configuration command.

IP Source Guard

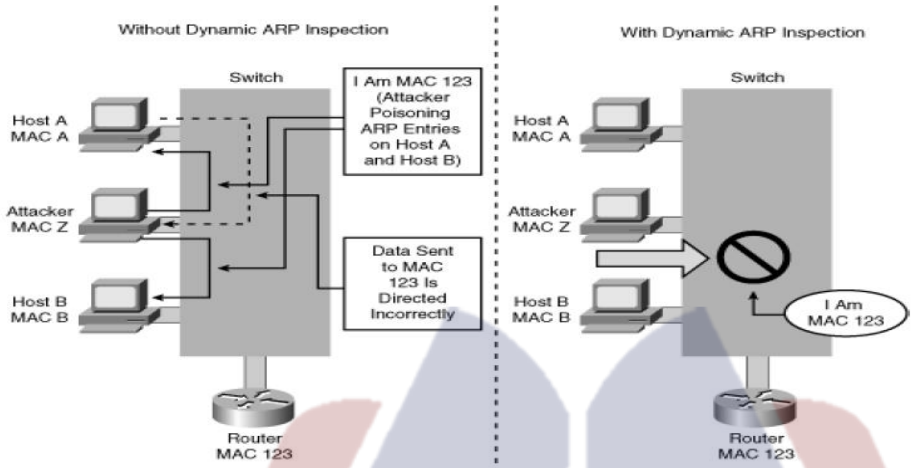
- IP Source Guard is a security feature that restricts IP traffic on untrusted Layer 2 ports by filtering traffic based on the DHCP snooping binding database or manually configured IP source bindings.
- This feature helps prevent IP spoofing attacks when a host tries to spoof and use the IP address of another host.
- Any IP traffic coming into the interface with a source IP address other than that assigned (via DHCP or static configuration) will be filtered out on the untrusted Layer 2 ports.
- The IP Source Guard feature is enabled in combination with the DHCP snooping feature on untrusted Layer 2 interfaces.
- It builds and maintains an IP source binding table that is learned by DHCP snooping or manually configured (static IP source bindings).

- An entry in the IP source binding table contains the IP address and the associated MAC and VLAN numbers.
- The IP Source Guard is supported on Layer 2 ports only, including access and trunk ports.

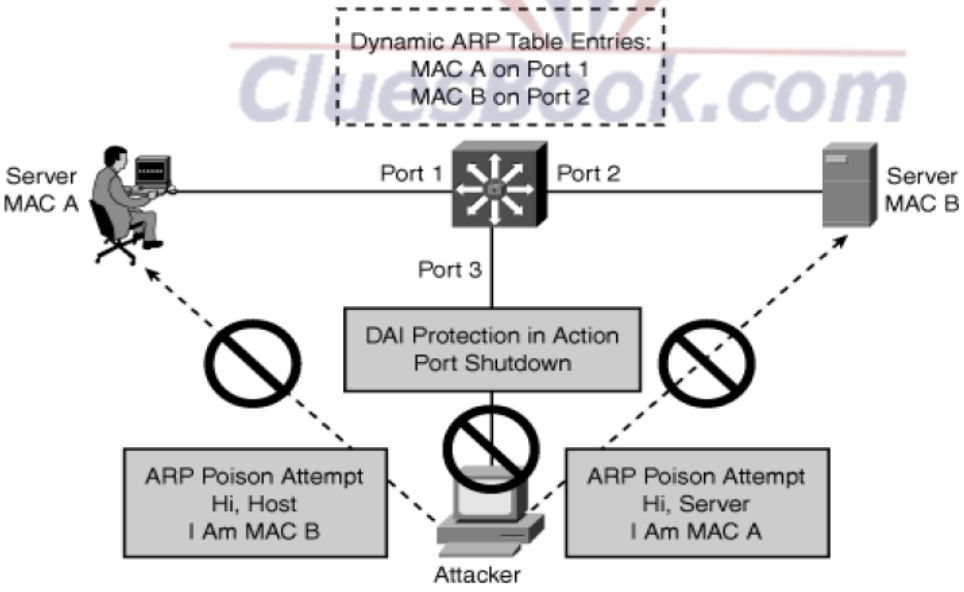
Dynamic ARP Inspection (DAI)

- Address Resolution Protocol (ARP) provides IP-to-MAC (32-bit IP address into a 48-bit Ethernet address) resolution.
- ARP operates at Layer 2 (the data-link layer) of the OSI model.
- ARP provides the translation mapping the IP address to the MAC address of the destination host using a lookup table (also known as the ARP cache).
- Several types of attacks can be launched against a host or devices connected to Layer 2 networks by "poisoning" the ARP caches.
- A malicious user could intercept traffic intended for other hosts on the LAN segment and poison the ARP caches of connected systems by broadcasting forged ARP responses.
- Several known ARP-based attacks can have a devastating impact on data privacy, confidentiality, and sensitive information.
- To block such attacks, the Layer 2 switch must have a mechanism to validate and ensure that only valid ARP requests and responses are forwarded.
- Dynamic ARP inspection is a security feature that validates ARP packets in a network.
- Dynamic ARP inspection determines the validity of packets by performing an IP-to-MAC address binding inspection stored in a trusted database, (the DHCP snooping binding database) before forwarding the packet to the appropriate destination.
- Dynamic ARP inspection will drop all ARP packets with invalid IP-to-MAC address bindings that fail the inspection.
- The DHCP snooping binding database is built when the DHCP snooping feature is enabled on the VLANs and on the switch.
- The Figure shows an example of an attacker attempting to spoof and hijack traffic for an important address (a default gateway in this example) by broadcasting to all hosts spoofing the MAC address of the router (using a gratuitous ARP).
- This will poison ARP cache entries (create an invalid ARP entry) on Host A and Host B, resulting in data being redirected to the wrong destination.

- Because of the poisoned entries, when Host A sends data destined for the router, it is incorrectly sent to the attacker instead. Dynamic ARP inspection locks down the IP-MAC mapping for hosts so that the attacking ARP is denied and logged.
- The dynamic ARP Inspection (DAI) feature safeguards the network from many of the commonly known man-in-the-middle (MITM) type attacks. Dynamic ARP Inspection ensures that only valid ARP requests and responses are forwarded.



The Figure illustrates the DAI feature in action and shows how the intruder is blocked on the untrusted port when it is trying to poison ARP entries.



4.8.6 Layer 2 Security Best Practices

- Manage the switches in a secure manner. For example, use SSH, authentication mechanism, access list, and set
- privilege levels.
- Restrict management access to the switch so that untrusted networks are not able to exploit management
- interfaces and protocols such as SNMP.
- Always use a dedicated VLAN ID for all trunk ports.
- Be skeptical; avoid using VLAN 1 for anything.
- Disable DTP on all non-trunking access ports.
- Deploy the Port Security feature to prevent unauthorized access from switching ports.
- Use the Private VLAN feature where applicable to segregate network traffic at Layer 2.
- Use MD5 authentication where applicable.
- Disable CDP where possible.
- Prevent denial-of-service attacks and other exploitation by disabling unused services and protocols.
- Shut down or disable all unused ports on the switch, and put them in a VLAN that is not used for normal operations.
- Use port security mechanisms to provide protection against a MAC flooding attack.
- Use port-level security features such as DHCP Snooping, IP Source Guard, and ARP security where applicable.
- Enable Spanning Tree Protocol features (for example, BPDU Guard, Loopguard, and Root Guard).
- Use Switch IOS ACLs and Wire-speed ACLs to filter undesirable traffic (IP and non-IP).