UNIT-V

Syllabus

Web Security: *SSL , TLS and Secure Electronic Transactions*
Intruders, *Intrusion Detection, Password Management*
Malicious Software - Types, Viruses, Virus Countermeasures, Worms,
Firewalls *- Characteristics, Types of Firewalls, Placement of Firewalls,*
Firewall Configuration, Trusted systems.

Introduction

- Unauthorized intrusion into a computer system or network is one of the most serious threats to computer security.
- Intrusion detection systems have been developed to provide early warning of an intrusion so that defensive action can be taken to prevent or minimize damage.
- Intrusion detection involves detecting unusual patterns of activity or patterns of activity that are known to correlate with intrusions.
- One important element of intrusion prevention is password management, with the goal of preventing unauthorized users from having access to the passwords of others.

Intruders

One of the two most publicized threats to security is the intruder (the other is viruses), often referred to as a hacker or cracker.
Three classes of intruders:
• **Masquerader:** An individual who is not authorized to use the computer and who penetrates a system's access controls to exploit a legitimate user's account •
Misfeasor: A legitimate user who accesses data, programs, or resources for which such access is not authorized, or who is authorized for such access but misuses his or her privileges
• **Clandestine user:** An individual who seizes supervisory control of the system and uses this control to evade auditing and access controls or to suppress audit collection.

✍ The masquerader is likely to be an outsider; the misfeasor generally is an insider; and the clandestine user can be either an outsider or an insider.

Following examples of intrusion:
• Performing a remote root compromise of an e-mail server
• Defacing a Web server
• Guessing and cracking passwords
• Copying a database containing credit card numbers
• Viewing sensitive data, including payroll records and medical information, without authorization
• Running a packet sniffer on a workstation to capture usernames and passwords
• Using a permission error on an anonymous FTP server to distribute pirated software and music files
• Dialing into an unsecured modem and gaining internal network access
• Posing as an executive, calling the help desk, resetting the executive's e-mail password, and learning the new password

• Using an unattended, logged-in workstation without permission.

Intruder Behavior Patterns

✍ The techniques and behavior patterns of intruders are constantly shifting, to exploit newly discovered weaknesses and to evade detection and countermeasures. Even so, intruders typically follow one of a number of recognizable behavior patterns, and these patterns typically differ from those of ordinary users.

✍ Three broad examples of intruder behavior patterns

HACKERS Traditionally, those who hack into computers do so for the thrill of it or for status.

Some Examples of *Hacker* Patterns of Behavior

1. Select the target using IP lookup tools such as NSLookup, Dig, and others.
2. Map network for accessible services using tools such as NMAP.
3. Identify potentially vulnerable services (in this case, pcAnywhere).
4. Brute force (guess) pcAnywhere password.
5. Install remote administration tool called DameWare.
6. Wait for administrator to log on and capture his password.
7. Use that password to access remainder of network.

CRIMINALS Organized groups of hackers have become a widespread and common threat to Internet-based systems.

✍ These groups can be in the employ of a corporation or government but often are loosely affiliated gangs of hackers.
✍ Typically, these gangs are young, often Eastern European, Russian, or southeast Asian hackers who do business on the Web [ANTE06].
✍ They meet in underground forums with names like DarkMarket.org and theftservices.com to trade tips and data and coordinate attacks.
✍ A common target is a credit card file at an e-commerce server.
✍ Attackers attempt to gain root access.
✍ The card numbers are used by organized crime gangs to purchase expensive items and are then posted to carder sites, where others can access and use the account numbers; this obscures usage patterns and complicates investigation.

Some Examples of ***CRIMINALS*** Patterns of Behavior

1. Act quickly and precisely to make their activities harder to detect.
2. Exploit perimeter through vulnerable ports.
3. Use Trojan horses (hidden software) to leave back doors for reentry.
4. Use sniffers to capture passwords.
5. Do not stick around until noticed.
6. Make few or no mistakes.

INSIDER ATTACKS Insider attacks are among the most difficult to detect and prevent. Employees already have access and knowledge about the structure and content of corporate databases. Insider attacks can be motivated by revenge or simply a feeling of entitlement.

Some Examples of ***INSIDER ATTACKS*** Patterns of Behavior

1. Create network accounts for themselves and their friends.

2. Access accounts and applications they wouldn't normally use for their daily jobs.
3. E-mail former and prospective employers.
4. Conduct furtive instant-messaging chats.
5. Visit Web sites that cater to disgruntled employees, such as f'dcompany.com.
6. Perform large downloads and file copying.
7. Access the network during off hours.

Intrusion Techniques

- The objective of the intruder is to gain access to a system or to increase the range of privileges accessible on a system.
- Most initial attacks use system or software vulnerabilities that allow a user to execute code that opens a back door into the system.
- Alternatively, the intruder attempts to acquire information that should have been protected.
- In some cases, this information is in the form of a user password. Typically, a system must maintain a file that associates a password with each authorized user.
- If such a file is stored with no protection, then it is an easy matter to gain access to it and learn passwords.
- The password file can be protected in one of two ways:

One-way function: The system stores only the value of a function based on the user's password. When the user presents a password, the system transforms that password and compares it with the stored value. In practice, the system usually performs a one-way transformation (not reversible) in which the password is used to generate a key for the one-way function and in which a fixed-length output is produced.
Access control: Access to the password file is limited to one or a very few accounts.

- If one or both of these countermeasures are in place, some effort is needed for a potential intruder to learn passwords.

The following techniques for learning passwords:
1. Try default passwords used with standard accounts that are shipped with the system. Many administrators do not bother to change these defaults.
2. Exhaustively try all short passwords (those of one to three characters).
3. Try words in the system's online dictionary or a list of likely passwords. Examples of the latter are readily available on hacker bulletin boards.
4. Collect information about users, such as their full names, the names of their spouse and children, pictures in their office, and books in their office that are related to hobbies.
5. Try users' phone numbers, Social Security numbers, and room numbers.
6. Try all legitimate license plate numbers for this state.
7. Use a Trojan horse (described in Chapter 10) to bypass restrictions on access.
8. Tap the line between a remote user and the host system.

INTRUSION DETECTION

Intrusion detection is based on the assumption that the behavior of the intruder differs from that of a legitimate user in ways that can be quantified.

The following approaches to intrusion detection:

1. **Statistical anomaly detection:** Involves the collection of data relating to the behavior of legitimate users over a period of time. Then statistical tests are applied to observed behavior to determine with a high level of confidence whether that behavior is not legitimate user behavior.

a. **Threshold detection:** This approach involves defining thresholds, independent of user, for the frequency of occurrence of various events.

b. **Profile based:** A profile of the activity of each user is developed and used to detect changes in the behavior of individual accounts.

2. **Rule-based detection:** Involves an attempt to define a set of rules that can be used to decide that a given behavior is that of an intruder.
a. Anomaly detection: Rules are developed to detect deviation from previous usage patterns.
b. Penetration identification: An expert system approach that searches for suspicious behavior.

Audit Records

A fundamental tool for intrusion detection is the audit record. Some record of ongoing activity by users must be maintained as input to an intrusion detection system. Basically, two plans are used:

• **Native audit records:** Virtually all multiuser operating systems include accounting software that collects information on user activity.

- The advantage of using this information is that no additional collection software is needed.
- The disadvantage is that the native audit records may not contain the needed information or may not contain it in a convenient form.

• **Detection-specific audit records:** A collection facility can be implemented that generates audit records containing only that information required by the intrusion detection system.

- One advantage of such an approach is that it could be made vendor independent and ported to a variety of systems.
- The disadvantage is the extra overhead involved in having, in effect, two accounting packages running on a machine.

- A good example of detection-specific audit records is one developed by Dorothy Denning . Each audit record contains the following fields:

• **Subject:** Initiators of actions. A subject is typically a terminal user but might also be a process acting on behalf of users or groups of users.

• **Action:** Operation performed by the subject on or with an object; for example, login, read, perform I/O, execute.

• **Object:** Receptors of actions. Examples include files, programs, messages, records, terminals, printers, and user- or program-created structures. When a subject is the recipient of an action, such as electronic mail, then that subject is considered an object.

• **Exception-Condition:** Denotes which, if any, exception condition is raised on return.

• **Resource-Usage:** A list of quantitative elements in which each element gives the amount used of some resource (e.g., number of lines printed or displayed, number of records read or written, processor time, I/O units used, session elapsed time).

• **Time-Stamp:** Unique time-and-date stamp identifying when the action took place.
Consider the command

COPY GAME.EXE TO <Libray>GAME.EXE

issued by Smith to copy an executable file GAME from the current directory to the <Library> directory.The following audit records may be generated:

Smith	execute	<Library>COPY.EXE	0	CPU = 00002	11058721678
Smith	read	<Smith>GAME.EXE	0	RECORDS = 0	11058721679
Smith	execute	<Library>COPY.EXE	write-viol	RECORDS = 0	11058721680

Statistical Anomaly Detection

Statistical anomaly detection techniques fall into two broad categories: threshold detection and profile-based systems.

Threshold detection involves counting the number of occurrences of a specific event type over an interval of time. If the count surpasses what is considered a reasonable number that one might expect to occur, then intrusion is assumed. Threshold analysis, by itself, is a crude and ineffective detector of even moderately sophisticated attacks.

Profile-based anomaly detection focuses on characterizing the past behavior of individual users or related groups of users and then detecting significant deviations. A profile may consist of a set of parameters, so that deviation on just a single parameter may not be sufficient in itself to signal an alert.

The foundation of this approach is an analysis of audit records.
Examples of metrics that are useful for profile-based intrusion detection are the following:

• **Counter:** A nonnegative integer that may be incremented but not decremented until it is reset by management action. Typically, a count of certain event types is kept over a particular period of time. Examples include the number of logins by a single user during an hour, the number of times a given command is executed during a single user session, and the number of password failures during a minute.

• **Gauge:** A nonnegative integer that may be incremented or decremented. Typically, a gauge is used to measure the current value of some entity. Examples include the number of logical connections assigned to a user application and the number of outgoing messages queued for a user process.

• **Interval timer:** The length of time between two related events. An example is the length of time between successive logins to an account.

• **Resource utilization:** Quantity of resources consumed during a specified period. Examples include the number of pages printed during a user session and total time consumed by a program execution.

Rule-Based Intrusion Detection

Rule-based techniques detect intrusion by observing events in the system and applying a set of rules that lead to a decision regarding whether a given pattern of activity is or is not suspicious.

Rule-based anomaly detection is similar in terms of its approach and strengths to statistical anomaly detection. With the rule-based approach, historical audit records are analyzed to identify usage patterns and to generate automatically rules that describe those patterns. Rules may represent past behavior patterns of users, programs, privileges, time slots, terminals, and so on. Current behavior is then observed, and each transaction is matched against the set of rules to determine if it conforms to any historically observed pattern of behavior.

Rule-based penetration identification takes a very different approach to intrusion detection. The key feature of such systems is the use of rules for identifying known penetrations or penetrations that would exploit known weaknesses. Typically, the rules used in these systems are specific to the machine and operating system. The most fruitful approach to developing such rules is to analyze attack tools and scripts collected on the Internet. These rules can be supplemented with rules generated by knowledgeable security personnel.

✍ Heuristic rules that can be used to assign degrees of suspicion to activities. Example heuristics are the following:

1. Users should not read files in other users' personal directories.
2. Users must not write other users' files.
3. Users who log in after hours often access the same files they used earlier.
4. Users do not generally open disk devices directly but rely on higher-level operating system utilities.
5. Users should not be logged in more than once to the same system.
6. Users do not make copies of system programs.

Distributed Intrusion Detection

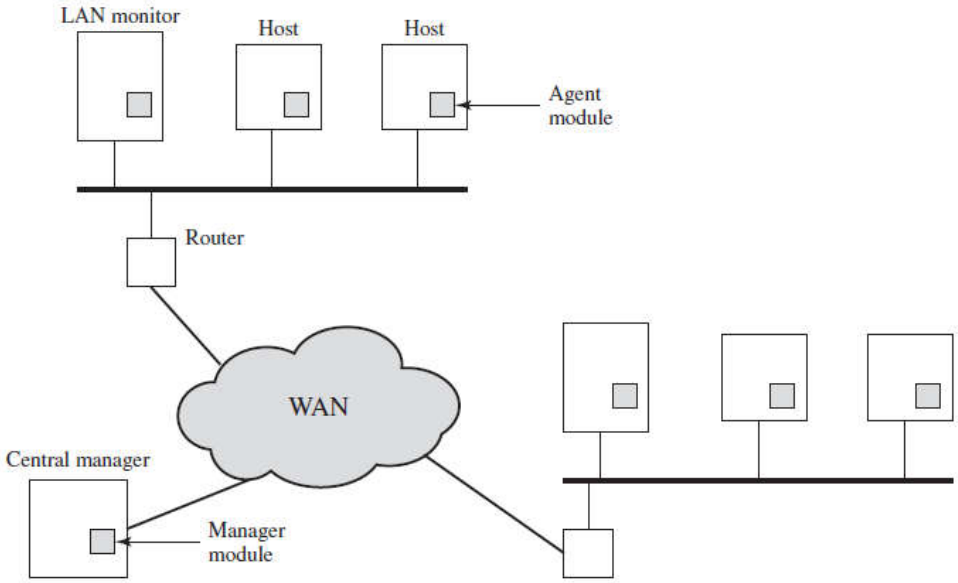
The typical organization, however, needs to defend a distributed collection of hosts supported by a LAN or internetwork. Although it is possible to mount a defense by using stand-alone intrusion detection systems on each host, a more effective defense can be achieved by coordination and cooperation among intrusion detection systems across the network.

The following major issues in the design of a distributed intrusion detection system

- A distributed intrusion detection system may need to deal with different audit record formats.
- One or more nodes in the network will serve as collection and analysis points for the data from the systems on the network. Thus, either raw audit data or summary data must be transmitted across the network. Therefore, there is a requirement to assure the integrity and confidentiality of these data. Integrity is required to prevent an

intruder from masking his or her activities by altering the transmitted audit information.

- Either a centralized or decentralized architecture can be used. With a centralized architecture, there is a single central point of collection and analysis of all audit data. This eases the task of correlating incoming reports but creates a potential bottleneck and single point of failure. With a decentralized architecture, there are more than one analysis centers, but these must coordinate their activities and exchange information.



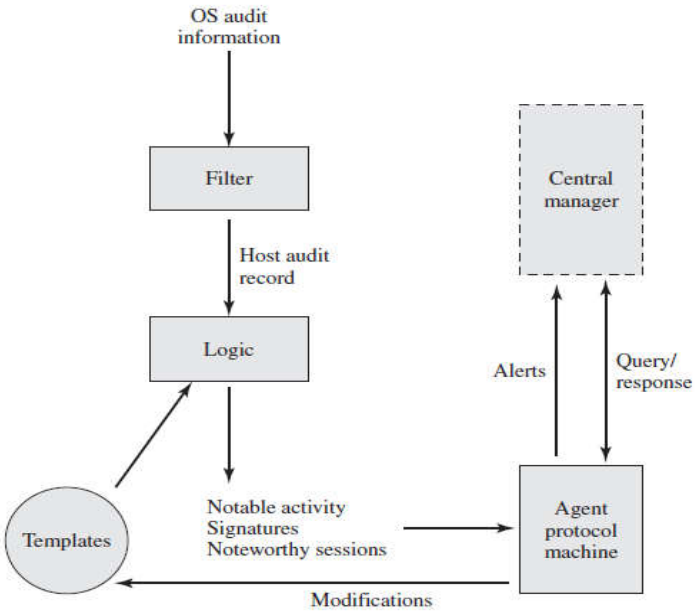
Architecture for Distributed Intrusion Detection

Architecture, which consists of three main components:

• **Host agent module:** An audit collection module operating as a background process on a monitored system. Its purpose is to collect data on security related events on the host and transmit these to the central manager.

• **LAN monitor agent module:** Operates in the same fashion as a host agent module except that it analyzes LAN traffic and reports the results to the central manager.

• **Central manager module:** Receives reports from LAN monitor and host agents and processes and correlates these reports to detect intrusion.



Agent Architecture

- The agent captures each audit record produced by the native audit collection system.
- A filter is applied that retains only those records that are of security interest.
- These records are then reformatted into a standardized format referred to as the host audit record (HAR).
- Next, a template-driven logic module analyzes the records for suspicious activity.
- At the lowest level, the agent scans for notable events that are of interest independent of any past events.
- Examples include failed file accesses, accessing system files, and changing a file's access control.
- At the next higher level, the agent looks for sequences of events, such as known attack patterns (signatures).
- Finally, the agent looks for anomalous behavior of an individual user based on a historical profile of that user, such as number of programs executed, number of files accessed, and the like.

MALICIOUS SOFTWARE

- Malicious software is software that is intentionally included or inserted in a system for a harmful purpose.
- A virus is a piece of software that can "infect" other programs by modifying them; the modification includes a copy of the virus program, which can then go on to infect other programs.
- A worm is a program that can replicate itself and send copies from computer to computer across network connections. Upon arrival, the worm may be activated to replicate and propagate again. In addition to propagation, the worm usually performs some unwanted function.

Types of malicious software

Malicious software can be divided into two categories: those that need a host program, and those that are independent.

Virus---- Malware that, when executed, tries to replicate itself into other executable code; when it succeeds the code is said to be infected. When the infected code is executed, the virus also executes.

Worm----A computer program that can run independently and can propagate a complete working version of itself onto other hosts on a network.

Logic bomb---- A program inserted into software by an intruder. A logic bomb lies dormant until a predefined condition is met; the program then triggers an unauthorized act.

***Trojan horse*----** A computer program that appears to have a useful function, but also has a hidden and potentially malicious function that evades security mechanisms, sometimes by exploiting legitimate authorizations of a system entity that invokes the Trojan horse program.

***Backdoor(trapdoor)*-----**Any mechanism that bypasses a normal security check; it may allow unauthorized access to functionality.

Mobile code ----Software (e.g., script, macro, or other portable instruction) that can be shipped unchanged to a heterogeneous collection of platforms and execute with identical semantics.

***Exploits*----** Code specific to a single vulnerability or set of vulnerabilities.

Downloaders -----Program that installs other items on a machine that is under attack. Usually, a downloader is sent in an e-mail.

Auto-rooter---- Malicious hacker tools used to break into new machines remotely.

Kit (virus generator) -----Set of tools for generating new viruses automatically.

Spammer----programs Used to send large volumes of unwanted e-mail.

Flooders---- Used to attack networked computer systems with a large volume of traffic to carry out a denial-of-service (DoS) attack.

Key loggers---- Captures keystrokes on a compromised system.

Root kit----- Set of hacker tools used after attacker has broken into a computer system and gained root-level access.

Zombie, bot------ Program activated on an infected machine that is activated to launch attacks on other machines.

Spyware------ Software that collects information from a computer and transmits it to another system.

Adware----- Advertising that is integrated into software. It can result in pop-up ads or redirection of a browser to a commercial site.

Backdoor

- A **backdoor**, also known as a **trapdoor**, is a secret entry point into a program that allows someone who is aware of the backdoor to gain access without going through the usual security access procedures.
- Programmers have used backdoors legitimately for many years to debug and test programs; such a backdoor is called a **maintenance hook**.

- This usually is done when the programmer is developing an application that has an authentication procedure, or a long setup, requiring the user to enter many different values to run the application.
- To debug the program, the developer may wish to gain special privileges or to avoid all the necessary setup and authentication.
- The programmer may also want to ensure that there is a method of activating the program should something be wrong with the authentication procedure that is being built into the application.
- The backdoor is code that recognizes some special sequence of input or is triggered by being run from a certain user ID or by an unlikely sequence of events.
- Backdoors become threats when unscrupulous programmers use them to gain unauthorized access.
- It is difficult to implement operating system controls for backdoors.

Logic Bomb

- One of the oldest types of program threat, predating viruses and worms, is the logic bomb.
- The logic bomb is code embedded in some legitimate program that is set to "explode" when certain conditions are met.
- Examples of conditions that can be used as triggers for a logic bomb are the presence or absence of certain files, a particular day of the week or date, or a particular user running the application .
- Once triggered, a bomb may alter or delete data or entire files, cause a machine halt, or do some other damage

Trojan Horses

- A Trojan horse1 is a useful, or apparently useful, program or command procedure containing hidden code that, when invoked, performs some unwanted or harmful function.
- Trojan horse programs can be used to accomplish functions indirectly that an unauthorized user could not accomplish directly.
- For example, to gain access to the files of another user on a shared system, a user could create a Trojan horse program that, when executed, changes the invoking user's file permissions so that the files are readable by any user.
- The author could then induce users to run the program by placing it in a common directory and naming it such that it appears to be a useful utility program or application.
- An example is a program that ostensibly produces a listing of the user's files in a desirable format.
- Another common motivation for the Trojan horse is data destruction.
- The program appears to be performing a useful function (e.g., a calculator program), but it may also be quietly deleting the user's files.
- Trojan horses fit into one of three models:
- Continuing to perform the function of the original program and additionally performing a separate malicious activity
- Continuing to perform the function of the original program but modifying the function to perform malicious activity.
- Performing a malicious function that completely replaces the function of the original program.

Viruses

- A computer virus is a piece of software that can "infect" other programs by modifying them; the modification includes injecting the original program with a routine to make copies of the virus program, which can then go on to infect other programs.
- Virus can do anything that other programs do. The difference is that a virus attaches itself to another program and executes secretly when the host program is run.
- Once a virus is executing, it can perform any function, such as erasing files and programs that is allowed by the privileges of the current user.
- A computer virus has three parts [AYCO06]:
  - **Infection mechanism:** The means by which a virus spreads, enabling it to replicate. The mechanism is also referred to as the **infection vector**.
  - **Trigger:** The event or condition that determines when the payload is activated or delivered.
  - **Payload:** What the virus does, besides spreading. The payload may involve damage or may involve benign but noticeable activity.

✍ During its lifetime, a typical virus goes through the following four phases:

- **Dormant phase:** The virus is idle. The virus will eventually be activated by some event, such as a date, the presence of another program or file, or the capacity of the disk exceeding some limit. Not all viruses have this stage.
- **Propagation phase:** The virus places a copy of itself into other programs or into certain system areas on the disk. The copy may not be identical to the propagating version; viruses often morph to evade detection. Each infected program will now contain a clone of the virus, which will itself enter a propagation phase.
- **Triggering phase:** The virus is activated to perform the function for which it was intended. As with the dormant phase, the triggering phase can be caused by a variety of system events, including a count of the number of times that this copy of the virus has made copies of itself.
- **Execution phase:** The function is performed. The function may be harmless, such as a message on the screen, or damaging, such as the destruction of programs and data files.

FIREWALL

- A firewall forms a barrier through which the traffic going in each direction must pass. A firewall security policy dictates which traffic is authorized to pass in each direction.
- A firewall may be designed to operate as a filter at the level of IP packets, or may operate at a higher protocol layer.
- 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.

Firewall Characteristics

The following design goals for a firewall:

1. All traffic from inside to outside, and vice versa, must pass through the firewall. This is achieved by physically blocking all access to the local network except via the firewall. Various configurations are possible, as explained later in this chapter.

2. Only authorized traffic, as defined by the local security policy, will be allowed to pass. Various types of firewalls are used, which implement various types of security policies, as explained later in this chapter.

3. The firewall itself is immune to penetration. This implies the use of a hardened system with a secured operating system. Trusted computer systems are suitable for hosting a firewall and often required in government applications.

Firewalls focused primarily on following control

Service control: Determines the types of Internet services that can be accessed, inbound or outbound. The firewall may filter traffic on the basis of IP address, protocol, or port number; may provide proxy software that receives and interprets each service request before passing it on; or may host the server software itself, such as a Web or mail service.

Direction control: Determines the direction in which particular service requests may be initiated and allowed to flow through the firewall.

User control: Controls access to a service according to which user is attempting to access it. This feature is typically applied to users inside the firewall perimeter (local users). It may also be applied to incoming traffic from external users; the latter requires some form of secure authentication technology, such as is provided in IPSec.

Behavior control: Controls how particular services are used. For example, the firewall may filter e-mail to eliminate spam, or it may enable external access to only a portion of the information on a local Web server.

The following capabilities are within the scope of a firewall:

1. A firewall defines a single choke point that keeps unauthorized users out of the protected network, prohibits potentially vulnerable services from entering or leaving the network, and provides protection from various kinds of IP spoofing and routing attacks
2. A firewall provides a location for monitoring security-related events. Audits and alarms can be implemented on the firewall system.
3. A firewall is a convenient platform for several Internet functions that are not security related. These include a network address translator, which maps local addresses to Internet addresses, and a network management function that audits or logs Internet usage.
4. A firewall can serve as the platform for IPSec. Using the tunnel mode capability described in Chapter 8, the firewall can be used to implement virtual private networks.

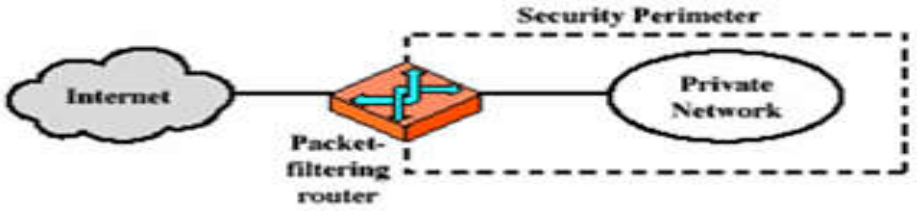
Firewalls have their limitations, including the following:

1. The firewall cannot protect against attacks that bypass the firewall. Internal systems may have dial-out capability to connect to an ISP. An internal LAN may support a modem pool that provides dial-in capability for traveling employees and telecommuters.
2. The firewall may not protect fully against internal threats, such as a disgruntled employee or an employee who unwittingly cooperates with an external attacker.

3. An improperly secured wireless LAN may be accessed from outside the organization. An internal firewall that separates portions of an enterprise network cannot guard against wireless communications between local systems on different sides of the internal firewall.
4. A laptop, PDA, or portable storage device may be used and infected outside the corporate network, and then attached and used internally.

Types of Firewalls

A firewall may act as a packet filter.



(a) Packet-filtering router

Packet Filtering Firewall

- A packet filtering firewall applies a set of rules to each incoming and outgoing IP packet and then forwards or discards the packet .
- The firewall is typically configured to filter packets going in both directions (from and to the internal network).
- Filtering rules are based on information contained in a network packet:

• **Source IP address:** The IP address of the system that originated the IP packet (e.g., 192.178.1.1)
• **Destination IP address:** The IP address of the system the IP packet is trying to reach (e.g., 192.168.1.2)
• **Source and destination transport-level address:** The transport-level (e.g., TCP or UDP) port number, which defines applications such as SNMP or TELNET
• **IP protocol field:** Defines the transport protocol
• **Interface:** For a firewall with three or more ports, which interface of the firewall the packet came from or which interface of the firewall the packet is destined for.

- The packet filter is typically set up as a list of rules based on matches to fields in the IP or TCP header.
- If there is a match to one of the rules, that rule is invoked to determine whether to forward or discard the packet.
- If there is no match to any rule, then a default action is taken.
- Two default policies are possible:
    - **Default = discard:** That which is not expressly permitted is prohibited.
    - **Default = forward:** That which is not expressly prohibited is permitted.

The default discard policy is more conservative. Initially, everything is blocked, and services must be added on a case-by-case basis.

Rule Set A

action	ourhost	port	theirhost	port	comment
block	*	*	SPIGOT	*	we don't trust these people
allow	OUR-GW	25	*	*	connection to our SMTP port

Rule Set B

action	ourhost	port	theirhost	port	comment
block	*	*	*	*	default

Rule Set C

action	ourhost	port	theirhost	port	comment
allow	*	*	*	25	connection to their SMTP port

Rule Set D

action	src	port	dest	port	flags	comment
allow	{our hosts}	*	*	25		our packets to their SMTP port
allow	*	25	*	*	ACK	their replies

Rule Set E

action	src	port	dest	port	flags	comment
allow	{our hosts}	*	*	*		our outgoing calls
allow	*	*	*	*	ACK	replies to our calls
allow	*	*	*	>1024		traffic to nonservers

Packet-Filtering Examples

A. Inbound mail is allowed (port 25 is for SMTP incoming), but only to a gateway host. However, packets from a particular external host, SPIGOT, are blocked because that host has a history of sending massive files in e-mail messages.

B. This is an explicit statement of the default policy. All rulesets include this rule implicitly as the last rule.

C. This ruleset is intended to specify that any inside host can send mail to the outside. A TCP packet with a destination port of 25 is routed to the SMTP server on the destination machine.

D. This ruleset achieves the intended result that was not achieved in C. The rules take advantage of a feature of TCP connections. Once a connection is set up, the ACK flag of a TCP segment is set to acknowledge segments sent from the other side. Thus, this ruleset states that it allows IP packets where the source IP address is one of a list of designated internal hosts and the destination TCP port number is 25. It also allows incoming packets with a source port number of 25 that include the ACK flag in the TCP segment. Note that we explicitly designate source and destination systems to define these rules explicitly.

E. This ruleset is one approach to handling FTP connections. With FTP, two TCP connections are used: a control connection to set up the file transfer and a data connection for the actual file transfer. The data connection uses a different port number that is dynamically assigned for the transfer. Most servers, and hence most attack

targets, use low-numbered ports; most outgoing calls tend to use a higher-numbered port, typically above 1023. Thus, this ruleset allows
— Packets that originate internally
— Reply packets to a connection initiated by an internal machine
— Packets destined for a high-numbered port on an internal machine
This scheme requires that the systems be configured so that only the appropriate port numbers are in use.

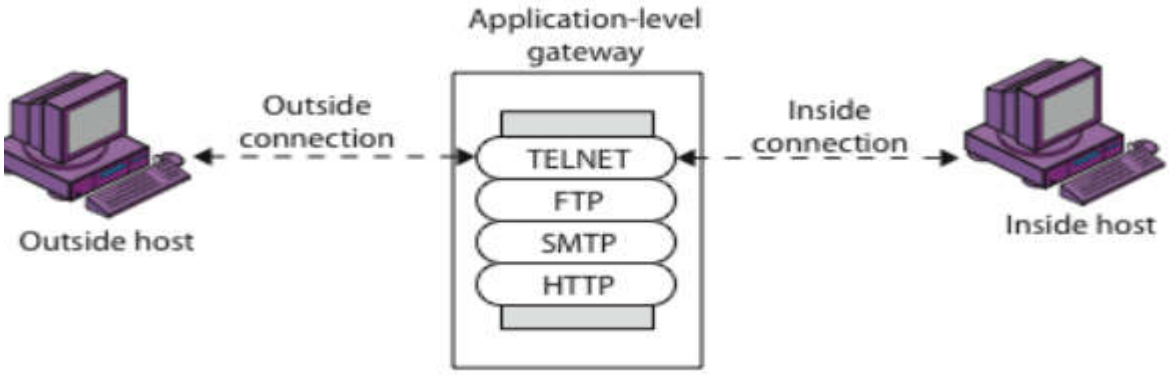
The following weaknesses of packet filter firewalls:
• Because packet filter firewalls do not examine upper-layer data, they cannot prevent attacks that employ application-specific vulnerabilities or functions. For example, a packet filter firewall cannot block specific application commands; if a packet filter firewall allows a given application, all functions available within that application will be permitted.
• Because of the limited information available to the firewall, the logging functionality present in packet filter firewalls is limited. Packet filter logs normally contain the same information used to make access control decisions (source address, destination address, and traffic type).
• Most packet filter firewalls do not support advanced user authentication schemes. Once again, this limitation is mostly due to the lack of upper-layer functionality by the firewall.
• Packet filter firewalls are generally vulnerable to attacks and exploits that take advantage of problems within the TCP/IP specification and protocol stack, such as *network layer address spoofing*. Many packet filter firewalls cannot detect a network packet in which the OSI Layer 3 addressing information has been altered. Spoofing attacks are generally employed by intruders to bypass the security controls implemented in a firewall platform.
• Finally, due to the small number of variables used in access control decisions, packet filter firewalls are susceptible to security breaches caused by improper configurations. In other words, it is easy to accidentally configure a packet.

Application level gate way:
Every connection between two networks is made via an application program called a proxy.

- Connection state is maintained and updated.
- Proxies are application or protocol specific
- Only protocols that have specific proxies configured are allowed through the firewall; all other traffic is rejected.

E.g., a gateway that is configured to be a web proxy will not allow any ftp, gopher, telnet or other traffic through.

- It filters packets on application data as well as on IP/TCP/UDP fields.
  Example: It allows selected internal users to telnet outside.



1. Require all telnet users to telnet through gateway.
2. For authorized users, gateway sets up telnet connection to destination host. Gateway relays data between 2 connections.
3. Router filter blocks all telnet connections not originating from gateway.

Application Gateway Weaknesses

- Very CPU intensive

There are two spliced connections between the end users, with the gateway at the splice point, and the gateway must examine and forward all traffic in both directions.

- Requires high performance host computer
- Expensive

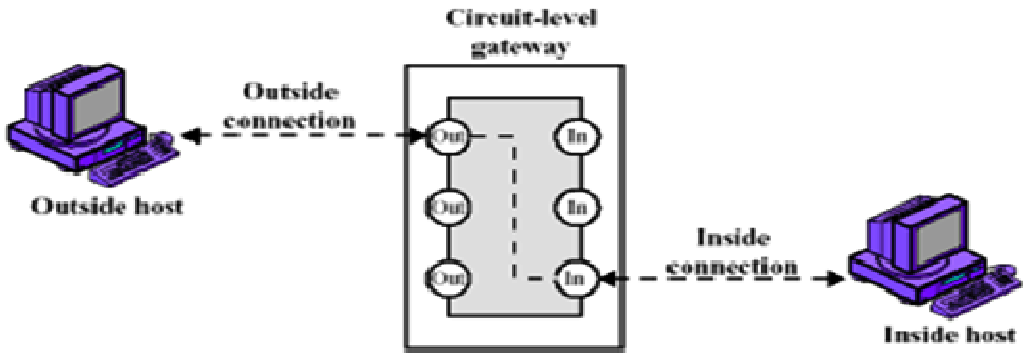
Circuit level Gateway:

Circuit level gateways work at the session layer of the OSI model, or the TCP layer of TCP/IP

❒ Monitor TCP handshaking between packets to determine whether a requested session is legitimate.

❒ Do not permit an end-to-end TCP connection

- Rather, the gateway sets up two TCP connections, one between itself and a TCP user on an inner host and one between itself and a TCP user on an outside host.
- Once the two connections are established, the gateway typically relays TCP segments from one to the other without examining the contents



• Traffic is filtered based on specified session rules, like: a session is initiated by a recognized computer;

- A circuit-level gateway sets up two connections:
  - One between itself and a TCP user on the inner host;
  - One between itself and a TCP user on the outer host;

Once connections are established and security criteria are met , both connections are linked by the gateway;

Pros:

- Circuit-level gateways are relatively inexpensive;
- Have the advantage of hiding information about the private network they protect.

Cons:

- Do not filter individual packets.

Bastion Host:

Bastion host is a system identified by the fire wall administrator as a critical point in the network's security.

- The bastion host serves as a platform for an application level or circuit level gate way.

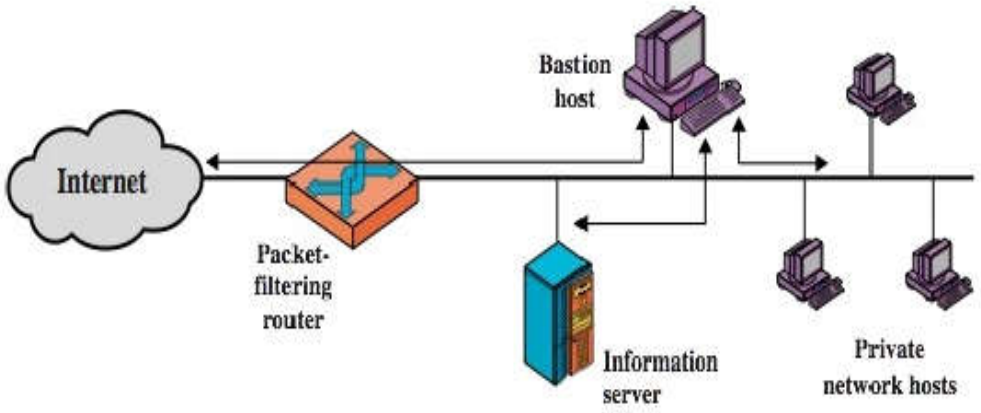
The bastion host hardware platform executes a secure OS.

- Only the services that the network administrator considers essential are installed on the bastion host.
- The bastion host may require authentication before a user is allowed to access to the proxy services.
- Each proxy is configured to allow access to only specific host systems.

- Each proxy logs all traffic, each connection and its duration.
- Each proxy is independent of other proxies on the bastion host and runs in a private secured directory.

Firewall configurations:

Screened Fire Wall Single-Homed Bastion:



It consists of two systems: a packet filtering router and a bastion host. The bastion host performs authentication and proxy functions. This configuration has greater security than simple a packet filter router or an application gateway alone, for two reasons.

1. This configuration implements both packet-level and application level filtering, allowing for considerable flexibility defining security policy.

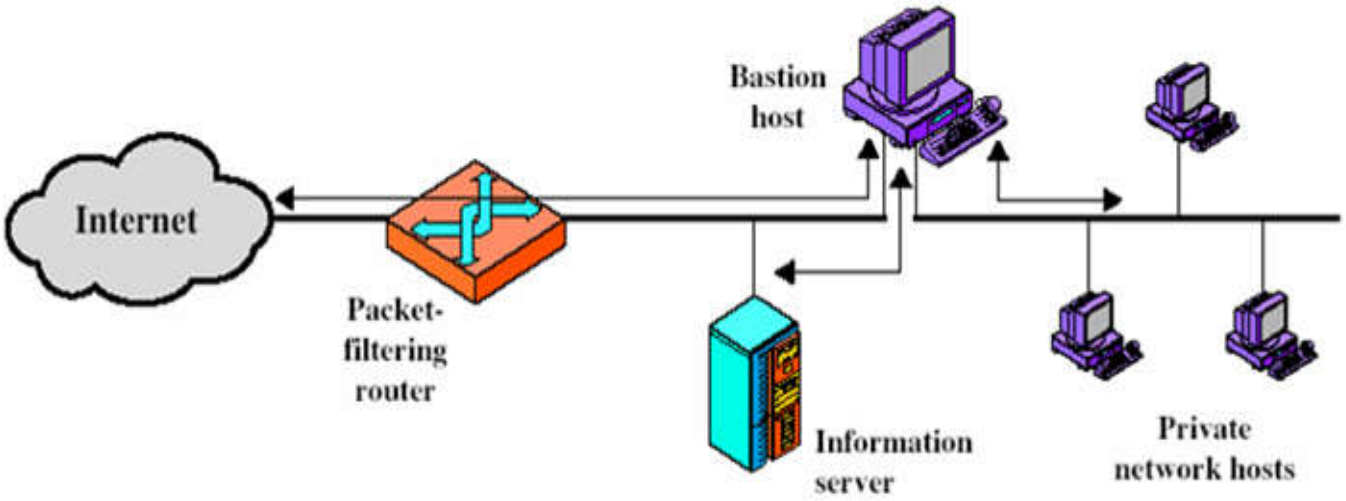
2. An intruder must generally penetrate two separate systems before of the internal network promised.

Screened firewall host System (Dual-homed bastion host):

In Screened Fire Wall Single-Homed Bastion, if the packet filtering router is completely compromised, traffic cloud floe directly through the router between the Internet and other hosts on the private network.

The Screened firewall host System (Dual-homed bastion host) physically prevents such a security breach.

Here an information server or previous other hosts can be allowed direct communication with the router if this is in accordance with the security policy.

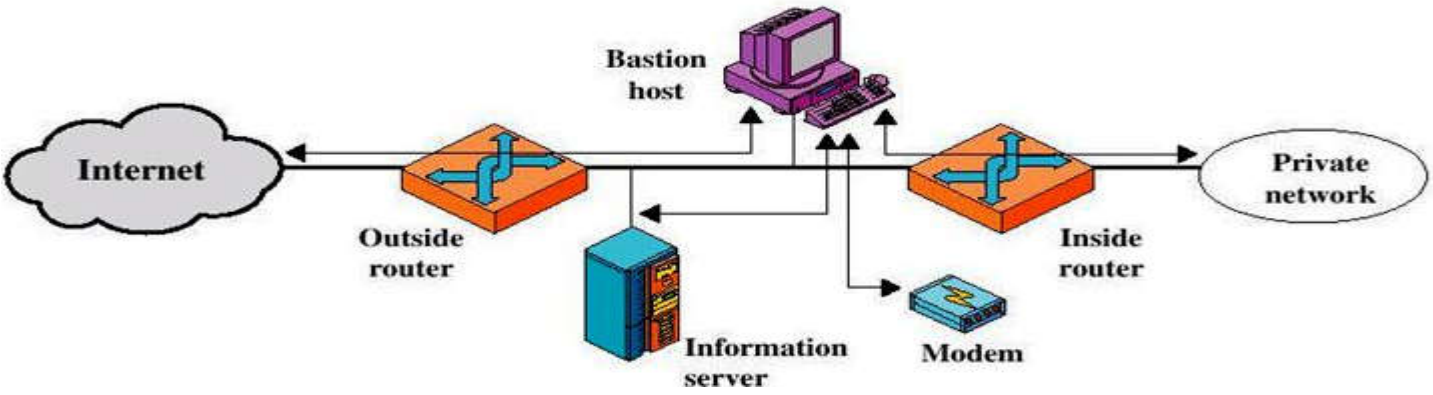


Screened subnet firewall:

A screened subnet firewall is a model that includes three important components for security. This type of setup is often used by enterprise systems that need additional protection from outside attacks.

A screened subnet firewall also called a 'triple-homed' setup.

This is one of the most secured firewall configurations. In this configuration, two packet filtering routers are used and the bastion host is positioned in between the two routers.



In a typical case, both the Internet and the internal users have access to the screened subnet, but the traffic flow between the two subnets (one is from bastion host to the internal network and the other is the sub-network between the two routers) is blocked.

Trusted systems:

Data access control:

The general model of access control as exercised by a file ort data base management systems is that of an access Matrix.

The basic elements of the model as follows:

Subject: An entity capable of accessing objects.

Object: Anything to which access is controlled.
Ex: Files, Portion of files, programs and segment of memory.

Access right: The way in which an object is accessed by a subject.
EX: read, write and execute.

The concept of trusted system:

An approach to protection data and resources is based on levels of security. This is commonly found in military, where information is categorized as unclassified (U), confidential (C), secret (S), top secret (TS), or beyond.

This concept is equally applicable in other areas, where information can be organized into categories and users can be granted clearances to access certain categories of data.

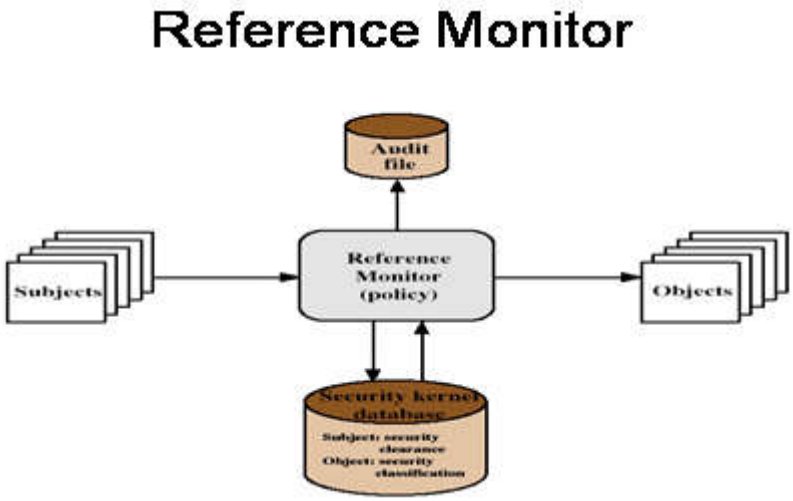
When multiple categories or levels of data are defined, the requirement is referred to as **multilevel security**. The general statement of the requirement for multilevel

security is that a subject at a high level may not convey information to a subject at a lower or non-comparable level unless that flow accurately reflects the will of an authorized user. For implementation purposes, this requirement is in two parts and is simply stated.

A multilevel secure system must enforce the following:

•**No read-up:** A subject can only read an object of less or equal security level. This is referred to in the literature as the simple security property

•**No write-down:** A subject can write into an object of greater or equal security level. This is referred to as the *-property (pronounced star property)



These two rules, if properly enforced, provide multilevel security. For a data processing system, the approach that has been taken, and has been the object of much research and development, is based on the reference monitor concept.

The reference monitor is a controlling element in the hardware and operating system of a computer that regulates the access of subjects to objects on the basis of security parameters of the subject and object.

The reference monitor has access to a file, known as the security kernel database that lists the access privilege (security clearance) of each subject and the protection attributes (classification level) of each object. The reference monitor enforces the security rules (no read-up, no write-down) and has the following properties:

•**Complete mediation:** The security rules are enforced on every access, not just, for example, when a file is opened (requires high performance overhead)

•**Isolation:** The reference monitor and database are protected from unauthorized modification (requires impossibility for attacker to change database)

•**Verifiability:** The reference monitor's correctness must be provable. That is, it must be possible to demonstrate mathematically that the reference monitor enforces the security rules and provided complete mediation and isolation If provided, system is referred to as a trusted system)

Secure System Vs Trusted System:

Secure system	Trusted system
Binary: it's secure, or it's not	Degrees of trustworthiness
Intrinsic to the system Based solely on the system	User decides on trustworthiness User makes a judgment based on system
Absolute: security doesn't depend on how or by whom the system is used	Relative: trustworthiness depends on the details of system use
Goal: absolute security	Characteristic: system can be viewed as trustworthy at any particular time

Previous paper questions

1 a) Describe different classes of Intruders. [8]
b) Explain malicious programs. [7]

2 a) Explain various approaches to Intrusion Detection. [8]
b) What is a firewall? Explain packet filter router. [7]

3 a) Explain password selection procedure in detail. [8]
b) Explain the capabilities and limitations of firewalls. [7]

4a) briefly explain the following:
i) Trapdoors ii) logic bomb iii) Trojan horse iv) Viruses [8]
b) Explain the concept of trusted systems. [7]

5) Write a short note on Intrusion Detection. [7]
b) Describe trusted system in detail.

6) What is a firewall? Explain different types of firewalls.
Explain the characteristics and capabilities of firewall?

7) Write short notes on
a) Parasitic virus
b) memory-resident virus
c) Boot sector virus
d) Stealth Virus
e) Polymorphic virus