M.S. Ramaiah Institute of Technology
(Autonomous Institute, Affiliated to VTU)
Department of Computer Science and Engineering

Course Name: Cryptography and Network Security

Course Code - CSE643

Credits - 3:0:0

Term: March 2022 – July 2022

UNIT - 5

Prepared by: Dr. Sangeetha. V
Assistant Professor

Textbooks

1. Behrouz A. Forouzan, Debdeep Mukhopadhyay: Cryptography and Network Security, 2nd Edition, Special Indian Edition, Tata McGrawHill, 2011.

2. William Stallings, Cryptography and Network Security, Fifth Edition, Prentice Hall of India, 2005.

Reference Book:

1. Josef Pieprzyk, Thomas Hardjono, Jennifer Serberry Fundamentals of Computer Security, Springer, ISBN 978-3-662-07324-7.

RAMAIAH
Institute of Technology

Unit V (Text 2) – System Security

Intruders:

(Chapter 20.1 to 20.2)

- Intruders
- Intrusion Detection

Malicious Software: (Chapter 21.1 – 21.2)

- Types of Malicious Software
- Viruses

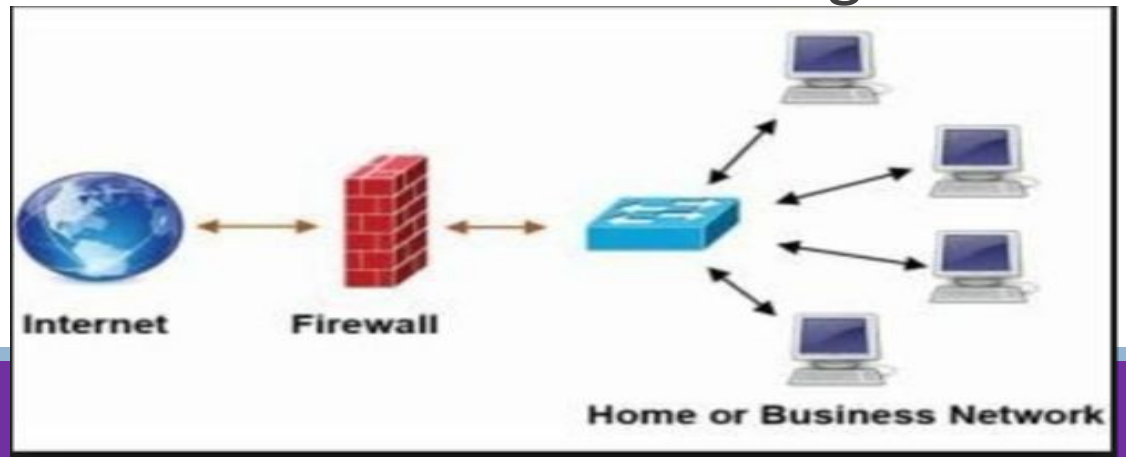
Firewalls : (Chapter 22.1,22.2,22.3

- The need for Firewalls
- Firewall Characteristics
- Types of Firewalls

RAMAIAH
Institute of Technology

Introduction

- A **firewall** is a network security system that monitors and controls incoming and outgoing network traffic based on predetermined security rules.
- It relies on the source, the destination addresses, and the ports.
- A firewall can deny any traffic that does not meet the specific criteria.
- Firewall performs actions such as blocking and filtering of traffic

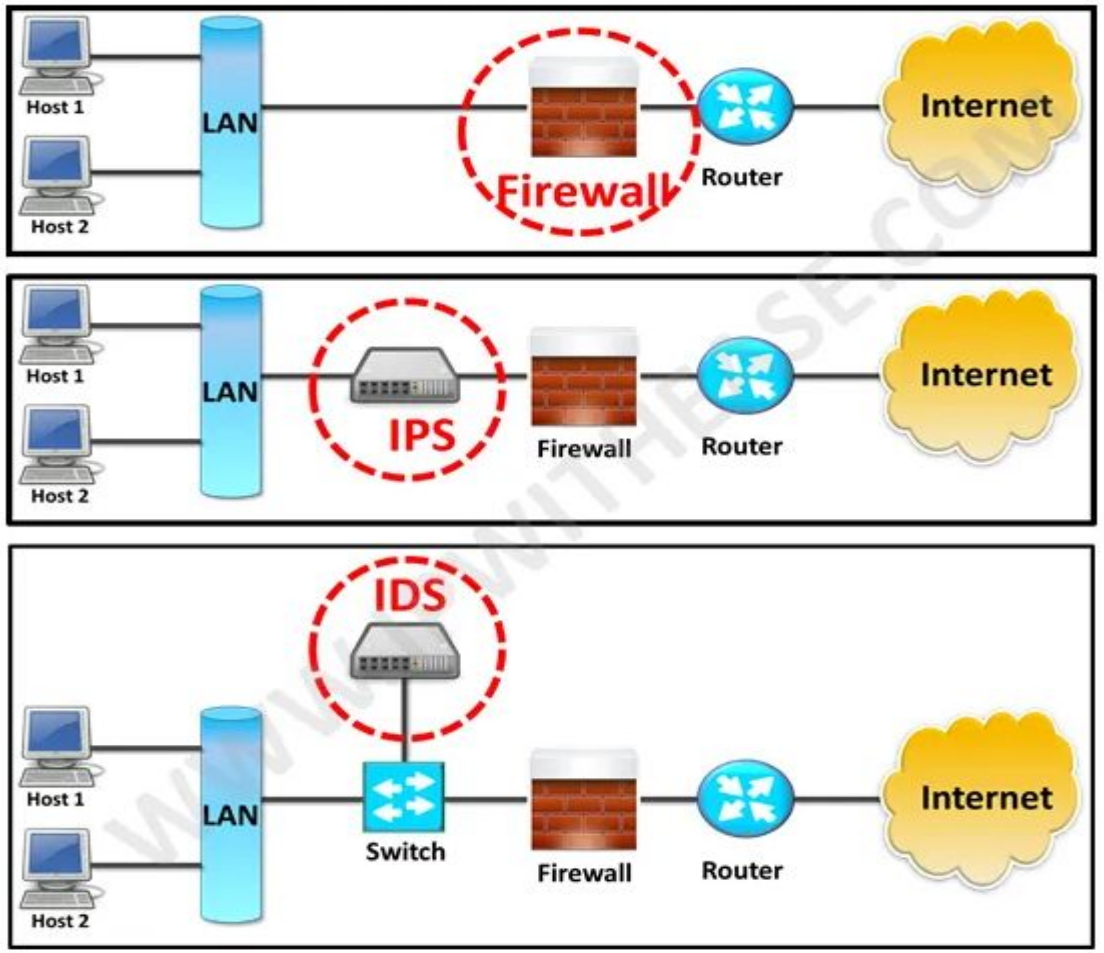


Introduction

- **Intrusion Detection System(IDS)** is a passive device which watches packets of data traversing the network, comparing with signature patterns and setting off an alarm on detection on suspicious activity.
- Monitors **network traffic** for suspicious activity and issues alerts when such activity is discovered.
- **Intrusion Detection System(IDS)** is a passive device which watches packets of data traversing the network, comparing with signature patterns and setting off an alarm on detection on suspicious activity.
- Monitors **network traffic** for suspicious activity and issues alerts when such activity is discovered.

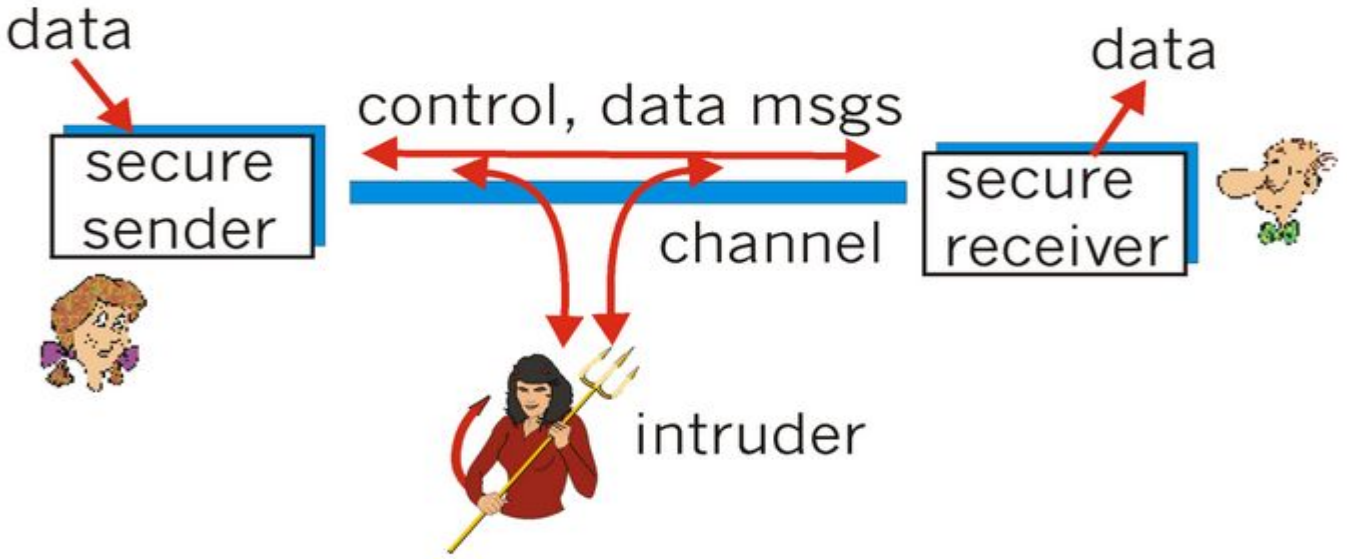
Introduction

PARAMETER	FIREWALL	IPS	IDS
Abbreviation for	-	Intrusion Prevention System	Intrusion Detection System
Philosophy	Firewall is a network security device that filters incoming and outgoing network traffic based on predetermined rules	IPS is a device that inspects traffic, detects it, classifies and then proactively stops malicious traffic from attack.	An intrusion detection system (IDS) is a device or software application that monitors a traffic for malicious activity or policy violations and sends alert on detection.
Principle of working	Filters traffic based on IP address and port numbers	inspects real time traffic and looks for traffic patterns or signatures of attack and then prevents the attacks on detection	Detects real time traffic and looks for traffic patterns or signatures of attack and them generates alerts



Intruders

- Intruders are the **attackers who attempt to breach the security of a network**.
- They attack the network in order to get unauthorized access.



Intruders

- Intruders are often referred to as hackers and are the most harmful factors contributing to the vulnerability of security.
- They have immense knowledge and an in-depth understanding of technology and security.
- Intruders steal the confidential information of the users.
- The stolen information is then sold to third-party, which aim at misusing the information for their own personal or professional gains.

Intruders

- Three classes of intruders
  - Masquerader
  - Misfeasor
  - Clandestine user
- Intruder Behavior Patterns
  - Hackers
  - Criminals
  - Insider Attacks
- Intrusion Techniques and Detection

RAMAIAH
Institute of Technology

Intruders

Masquerader:

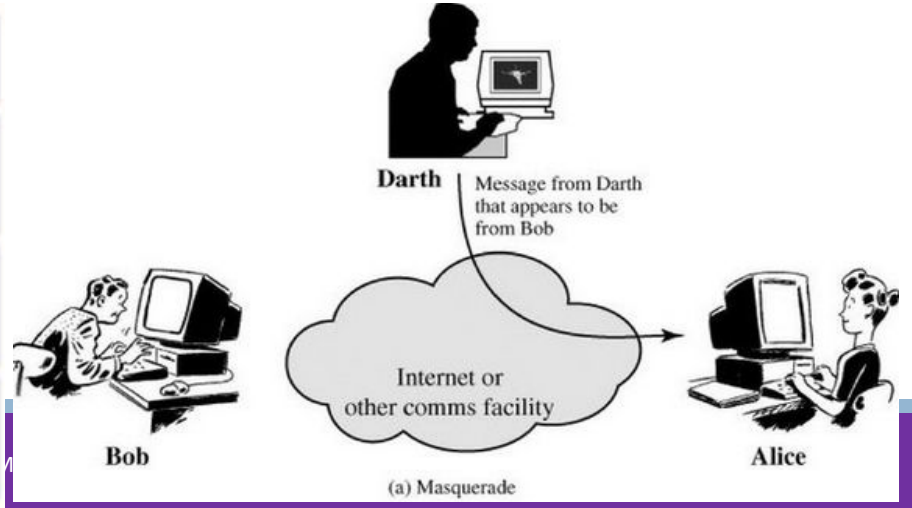
- The category of individuals that are not authorized to use the system but still exploit user's privacy and confidential information by possessing techniques that give them control over the system, such category of intruders is referred to as Masquerader.
- The masquerader is likely to be an outsider

Attackers infect a computer within a corporate network with malware.

The attackers install a keylogger on the system and steal sensitive credentials.

The attackers can now exploit the stolen sensitive credentials and impersonate the victim whose credentials are stolen.

Attackers can easily log in using the victim's stolen credentials, take the victim's identity and launch a cyberattack.



Intruders

- **Misfeasor:**
  - The category of individuals that are authorized to use the system, but misuse the granted access and privilege.
  - These are individuals that take undue advantage of the permissions and access given to them, such category of intruders is referred to as Misfeasor.
  - Misfeasor generally is an insider

Intruders

- **Clandestine user:**
  - The category of individuals that have supervision control over the system and misuse the authoritative power given to them.
  - The misconduct of power is often done by superlative authorities for financial gains, such a category of intruders is referred to as Clandestine User.
  - Clandestine user can be either an outsider or insider

Intruders

Intruder attacks range from the benign to the serious.

- At the benign end of the scale, there are many people who simply wish to explore internets and see what is out there.
- At the serious end are individuals who are attempting to read privileged data, perform unauthorized modifications to data, or disrupt the system.

Intruders

Lists the following examples of intrusion:

- remote root compromise
- web server defacement
- guessing / cracking passwords
- copying viewing sensitive data / databases
- running a packet sniffer
- distributing pirated software
- using an unsecured modem to access net
- impersonating a user to reset password
- using an unattended workstation

Intruder Behavior Patterns

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
- Criminals
- Insider Attacks

Intruder Behavior Patterns

Hackers

- motivated by thrill of access and status
  - hacking community a strong meritocracy
  - status is determined by level of competence
- benign intruders might be tolerable
  - do consume resources and may slow performance
  - can't know in advance whether benign or malign
- IDS / IPS / VPNs can help counter
- awareness led to establishment of CERTs
  - collect / disseminate vulnerability info / responses

Intruder Behavior Patterns

Some Examples of Intruder Patterns of Behavior - Hacker

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.

Intrusion Techniques

Criminal Enterprise

- organized groups of hackers now a threat
  - corporation / government / loosely affiliated gangs
  - typically young
  - often Eastern European or Russian hackers
  - often target credit cards on e-commerce server
- criminal hackers usually have specific targets
- once penetrated act quickly and get out
- IDS / IPS help but less effective
- sensitive data needs strong protection

Intruder Behavior Patterns

Some Examples of Intruder Patterns of Behavior – Criminal Enterprise

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.

Intruder Behavior Patterns

Insider Attacks

- among most difficult to detect and prevent
- employees have access & systems knowledge
- may be motivated by revenge / entitlement
  - when employment terminated
    - taking customer data when move to competitor
- IDS / IPS may help but also need:
  - least privilege, monitor logs, strong authentication, termination process to block access & mirror data

Intruder Behavior Patterns

Some Examples of Intruder Patterns of Behavior – Internal Threat

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.
- 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.

Intrusion Techniques

The password file can be protected in one of two ways:

1. **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.
2. **Access control :** Access to the password file is limited to one or a very few accounts.

Intrusion Detection

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.

Intrusion Detection

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 to bypass restrictions on access.

Intrusion Detection

Two principal countermeasures: **Detection and prevention**.

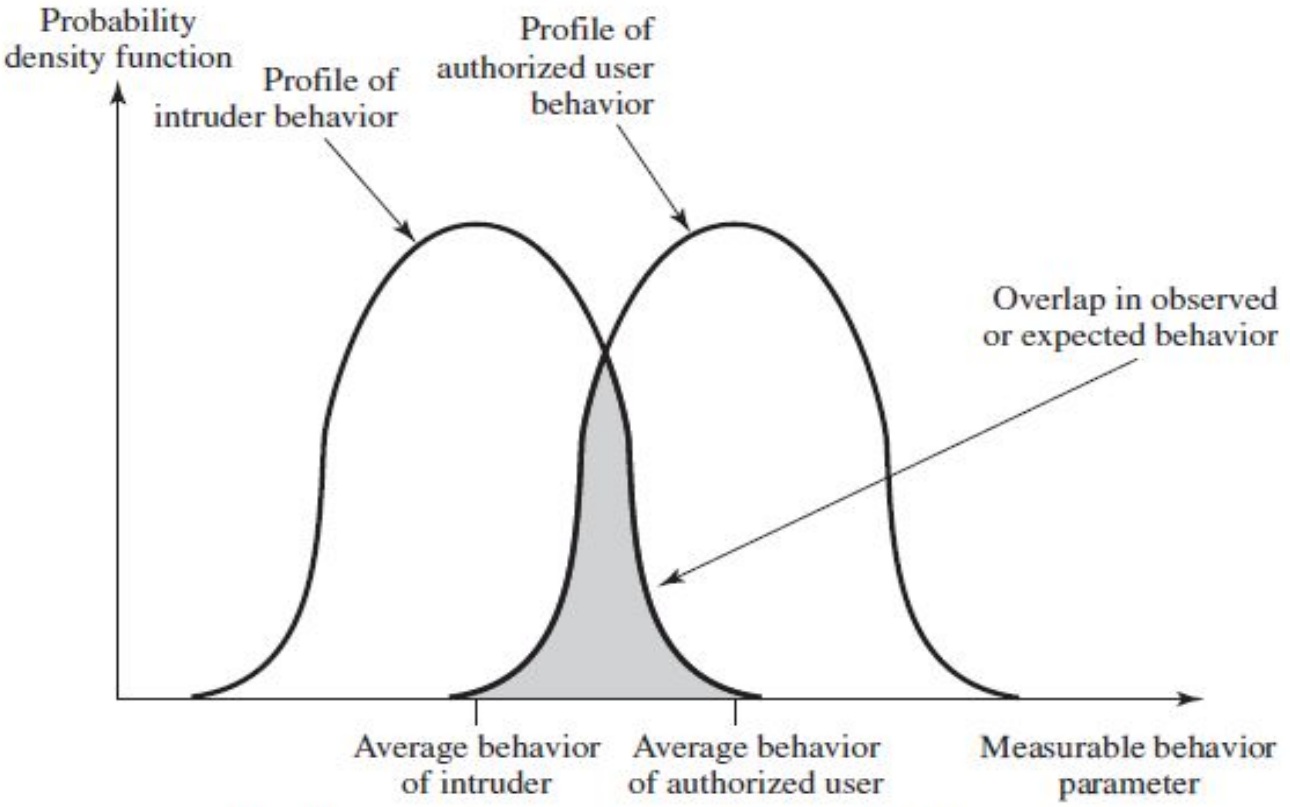
- Detection is concerned with learning of an attack, either before or after its success.
- Prevention is a challenging security goal and an uphill battle at all times.
- The difficulty stems from the fact that the defender must attempt to thwart all possible attacks, whereas the attacker is free to try to find the weakest link in the defense chain and attack at that point.

Intrusion Detection

- Intrusion prevention system will fail.
- A system's second line of defense is 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.

Intrusion Detection

Figure suggests the nature of the task confronting the designer of an intrusion detection system



Profiles of Behavior of Intruders and Authorized Users

Intrusion Detection

- Three classes of intruders
  - Masquerader
  - Misfeasor
  - Clandestine user
- Intruder Behavior Patterns
  - Hackers
  - Criminals
  - Insider Atttacks
- Intrusion Techniques and Detection

Intrusion Detection System

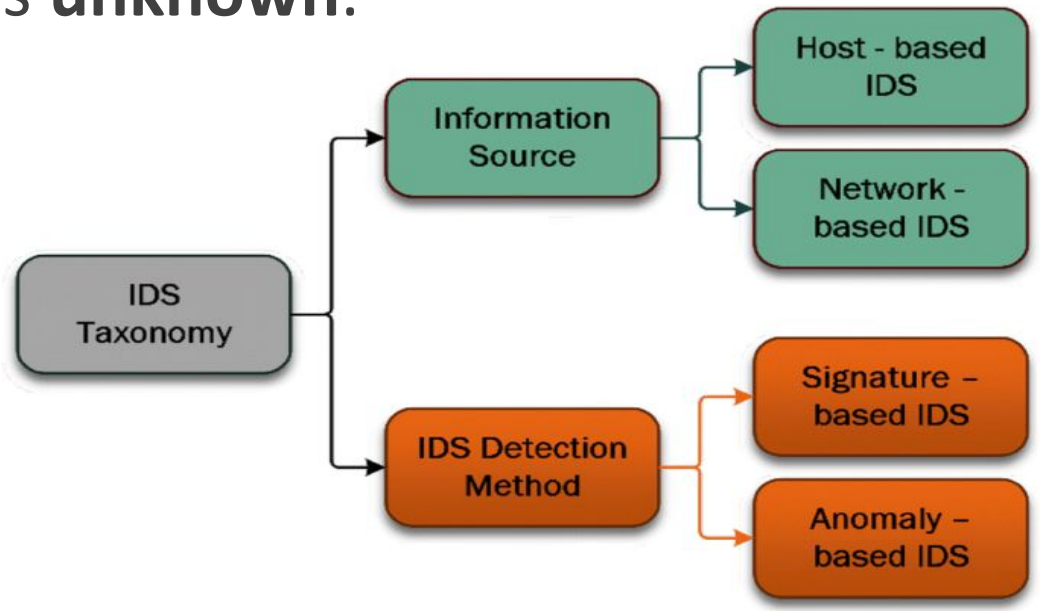
An Intrusion Detection System (IDS) is a system that monitors network traffic for suspicious activity and issues alerts when such activity is discovered.

It is a software application that scans a network or a system for the harmful activity or policy breaching.

Intrusion Detection System

Signature-based detection is typically used for identifying **known threats.**

Anomaly-based intrusion detection systems can alert suspicious behavior that is **unknown**.



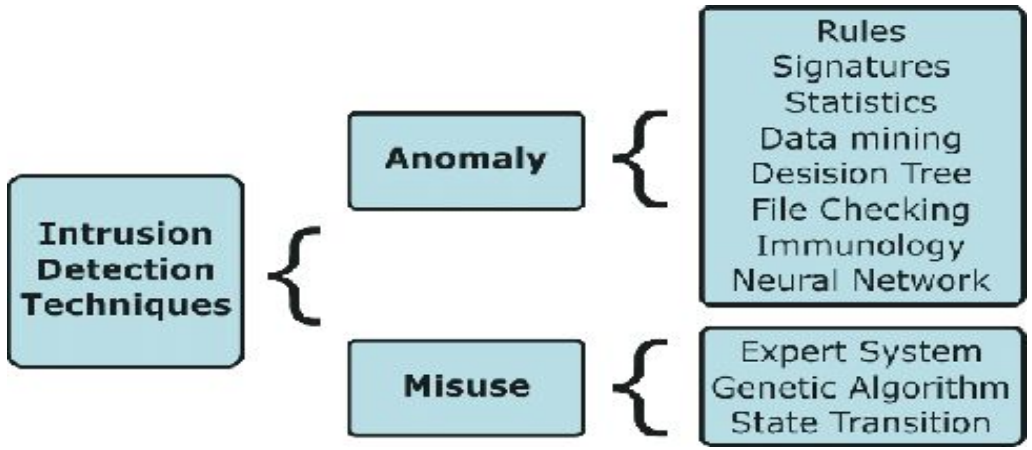
Intrusion Detection System

- **Anomaly-based Method:**
  Anomaly-based IDS detects unknown malware attacks as new malware are developed rapidly.
- It utilizes machine learning to train the detection system to recognize a normalized baseline.
- The baseline represents how the system normally behaves, and then all network activity is compared to that baseline,
- And it declares suspicious if it is not found in model.

Intrusion Detection

Two approaches to Anomaly-based intrusion detection systems:

1. Statistical anomaly detection
2. Rule-based detection



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 legitimate or not.

Statistical anomaly detection techniques fall into two broad categories:

- Threshold detection
- Profile-based systems

Statistical Anomaly Detection

Threshold detection

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.

Statistical Anomaly Detection

Threshold detection

- Threshold analysis is a crude and ineffective detector of even moderately sophisticated attacks.
- Thresholds generates a lot of false positives or a lot of false negatives.

Threshold detectors may be useful in conjunction with more sophisticated techniques.

Statistical Anomaly Detection

Profile based

- 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 of the activity of each user is developed and used to detect changes in the Behaviour of individual account.
- It 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.

Statistical Anomaly Detection

Profile based

The foundation of this approach is an analysis of audit records.

The audit records provide input to the intrusion detection function in two ways.

- First, the designer must decide on a number of quantitative metrics that can be used to measure user behavior.
- Second, current audit records are the input used to detect intrusion.

Statistical Anomaly Detection

Metrics that are useful for profile-based intrusion detection are:

Counter: A nonnegative integer that may be incremented but not decremented until it is reset by management action.

Gauge: A nonnegative integer that may be incremented or decremented.

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.

Statistical Anomaly Detection

Various statistical tests to determine whether current activity fits within acceptable limits.

- Mean and standard deviation
- Multivariate
- Markov process
- Time series
- Operational

Statistical Anomaly Detection

Mean and standard deviation

- This gives a reflection of the average behavior and its variability.
- The use of mean and standard deviation is applicable to a wide variety of counters, timers, and resource measures.
- But these measures are typically too crude for intrusion detection purposes.

Statistical Anomaly Detection

Multivariate model is based on correlations between two or more variables.

Intruder behavior may be characterized with greater confidence by considering such correlations (for example, processor time and resource usage, or login frequency and session elapsed time).

Statistical Anomaly Detection

A **Markov process** model is used to establish transition probabilities among various states.

As an example, this model might be used to look at transitions between certain commands.

A **time series** model focuses on time intervals, looking for sequences of events that happen too rapidly or too slowly.

A variety of statistical tests can be applied to characterize abnormal timing.

Statistical Anomaly Detection

Operational model is based on a judgment of what is considered abnormal, rather than an automated analysis of past audit records.

Typically, fixed limits are defined and intrusion is suspected for an observation that is outside the limits.

This type of approach works best where intruder behavior can be deduced from certain types of activities.

For example, a large number of login attempts over a short period suggests an attempted intrusion.

Rule-Based Intrusion 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.
- Rules may represent past behavior patterns of users, programs, privileges, time slots, terminals, and so on.

2 Types :

1. Anomaly detection
2. Penetration Identification

Intrusion Detection

a. Anomaly detection:

- Audit records are analyzed to identify the usage pattern and to automatically generate rules that describe those patterns.
- Current Behaviour is observed, and each transaction is matched against the set of rules to determine if it conforms to any historically observed pattern of behavior.
- This approach requires a large database of rules

Intrusion Detection

b. Penetration identification:

- Rules used in these systems are specific to the machine and operating system
- The most fruitful approach to develop 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

Intrusion Detection Tool

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:

1. Native audit records
2. Detection-specific audit records

Intrusion Detection Tool

Native audit records

- Virtually all multiuser operating systems include accounting software that collects information on user activity.
- Advantage:

No additional collection software is needed.

- Disadvantage:

May not contain the needed information or may not contain it in a convenient form.

Intrusion Detection Tool

Detection-specific audit records

- A collection facility can be implemented that generates audit records containing only that information required by the intrusion detection system.
- Advantage

  Vendor independent and can be ported to a variety of systems.
- Disadvantage

  Extra overhead involved with two accounting packages running on a machine.

Intrusion Detection Tool

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.

All activity arises through commands issued by subjects.

Subjects may be grouped into different access classes, and these classes may overlap.

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

Intrusion Detection Tool

Each audit record contains the following fields:

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.

For example, database actions may be audited for the database as a whole or at the record level.

Intrusion Detection Tool

Each audit record contains the following fields:

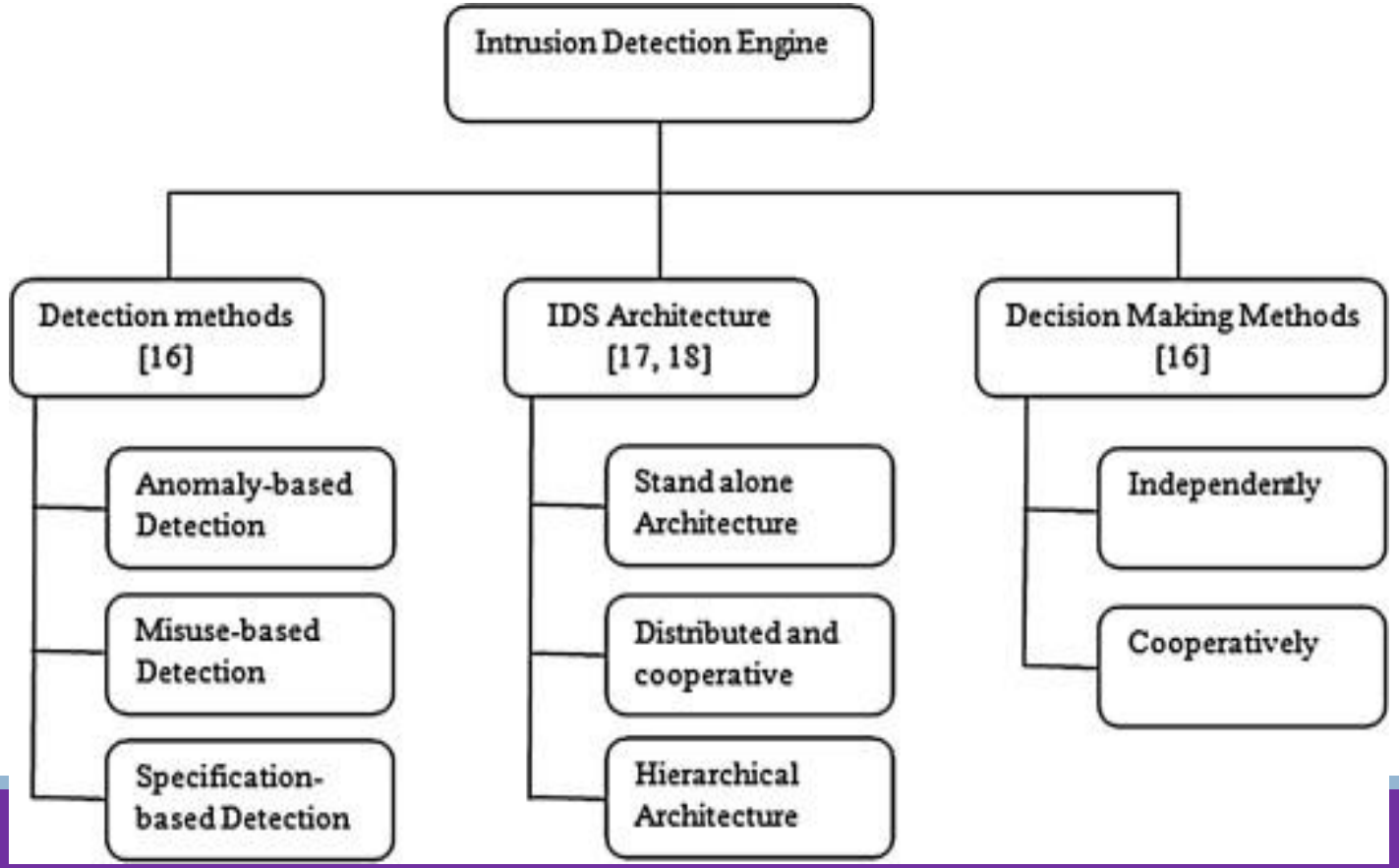
Exception-Condition: Denotes if any, exception condition is raised on return.

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.

IDS Architecture

Traditional system focus on single system stand alone



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.

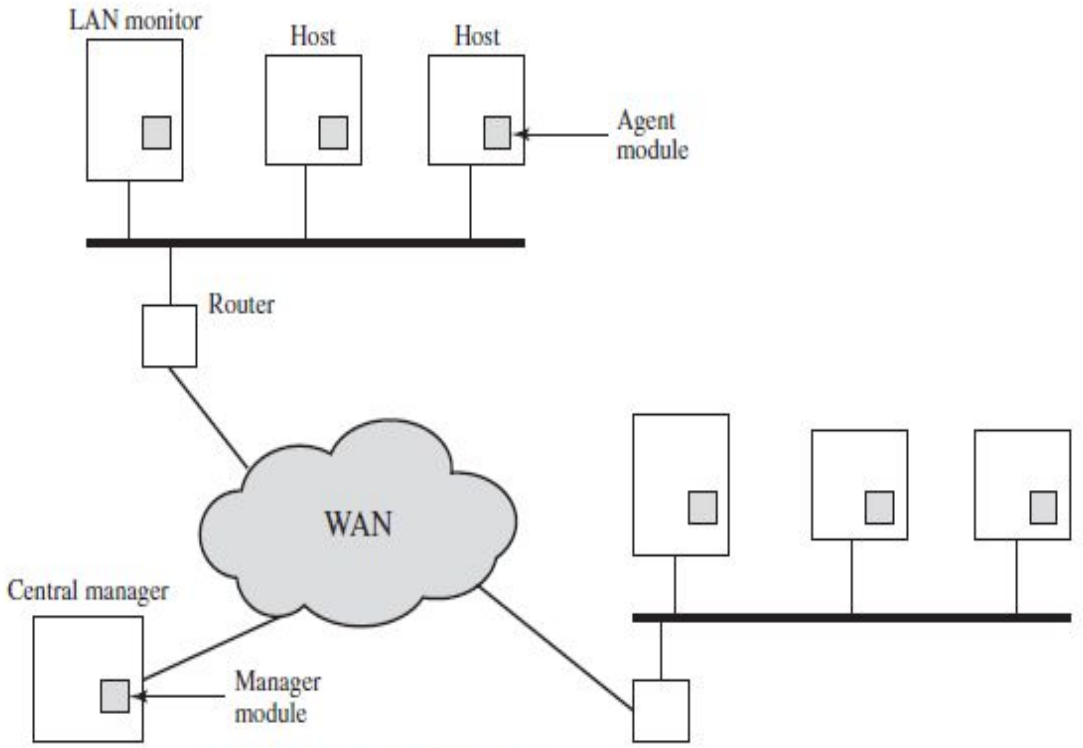
Distributed Intrusion Detection

Major issues in the design of a distributed intrusion detection system

1. A distributed intrusion detection system may need to deal with different audit record formats.
2. One or more nodes in the network will serve as collection and analysis points for the data from the systems on the network.
3. 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.
   - With a decentralized architecture, there are more than one analysis centers, but these must coordinate their activities and exchange information.

Distributed Intrusion Detection

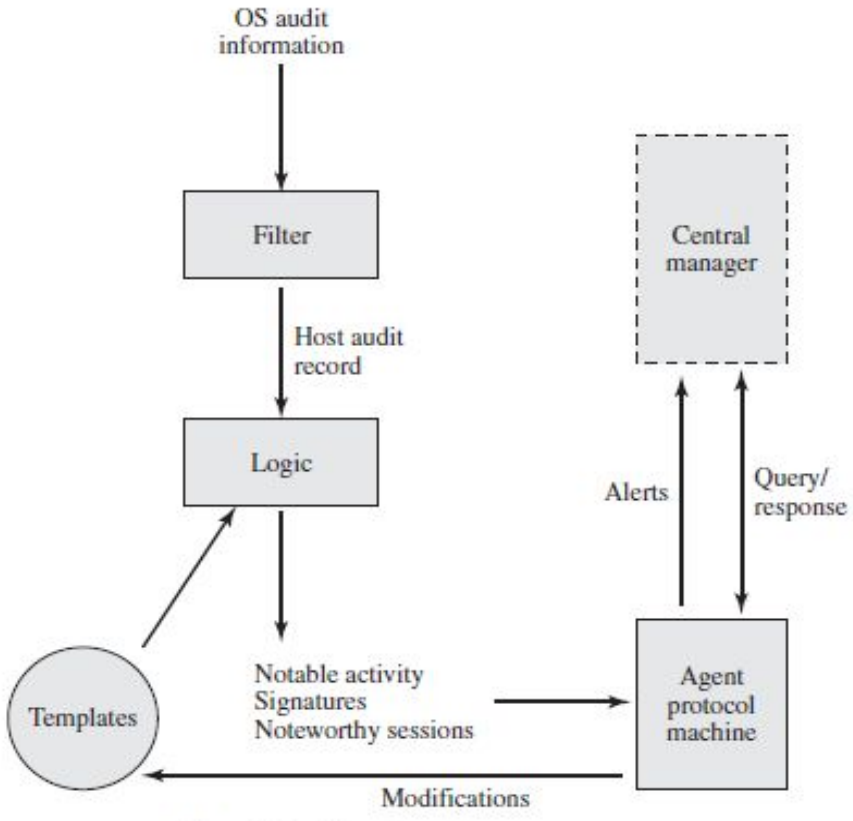
1. **Host agent module:** It collects data on security related events on the host and transmit these to the central manager.
2. **LAN monitor agent module:** It analyzes LAN traffic and reports the results to the central manager.
3. **Central manager module:** Receives reports from LAN monitor and host agents and processes and correlates these reports to detect intrusion.



Architecture for Distributed Intrusion Detection

Distributed Intrusion Detection

- 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).



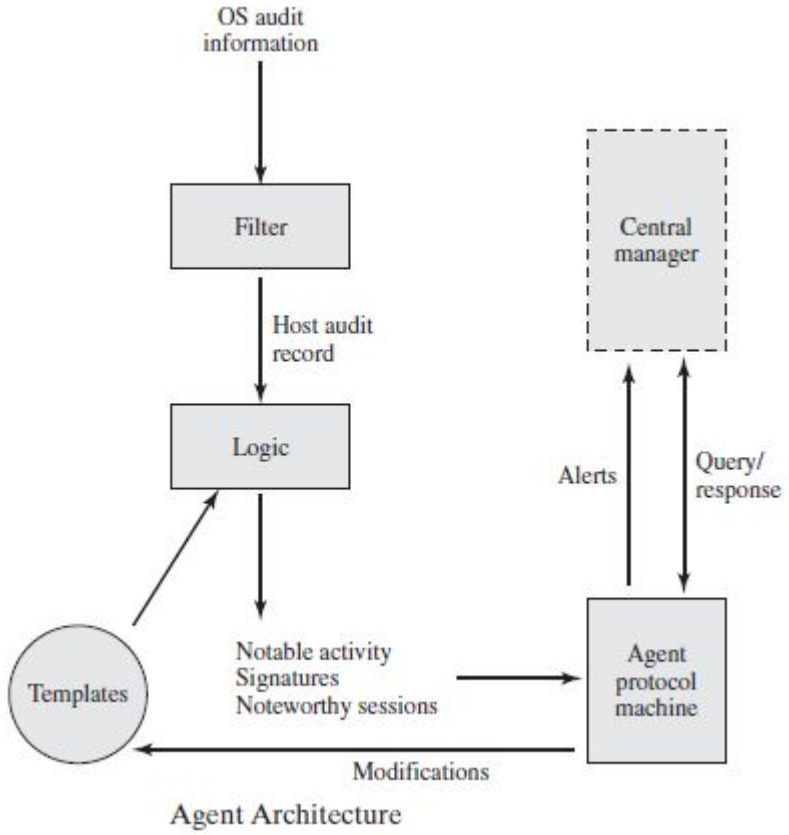
Agent Architecture

Distributed Intrusion Detection

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



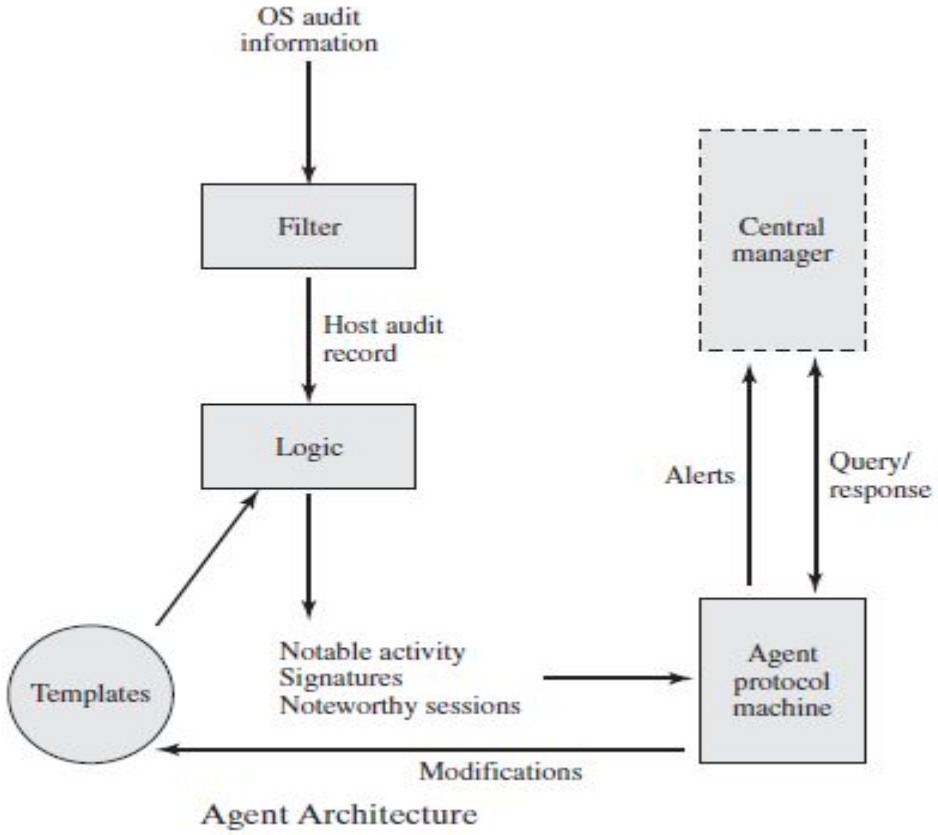
Agent Architecture

Distributed Intrusion Detection

Agent Protocol machine

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.



Agent Architecture

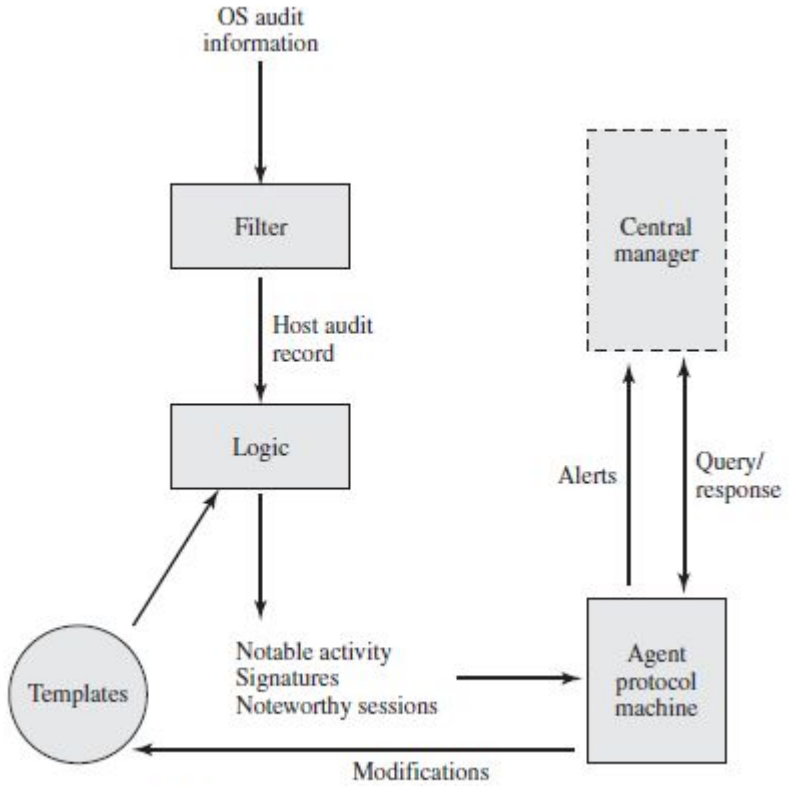
Distributed Intrusion Detection

Agent Protocol machine

When suspicious activity is detected, an alert is sent to the central manager.

The central manager includes an expert system that can draw inferences from received data.

The manager may also query individual systems for copies of HARs to correlate with those from other agents.



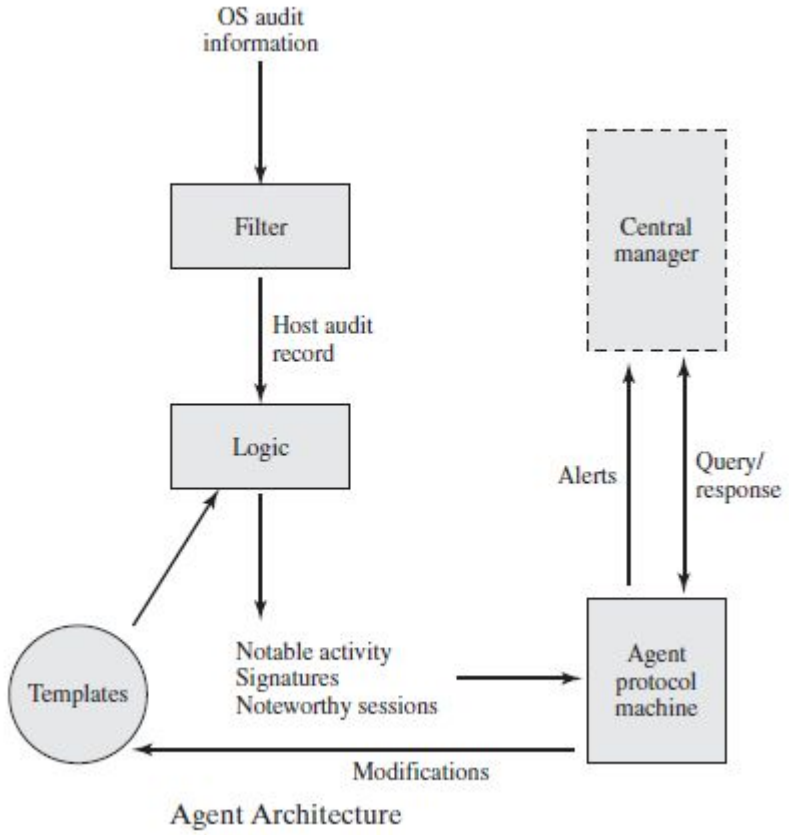
Agent Architecture

Distributed Intrusion Detection

The LAN monitor agent also supplies information to the central manager.

The LAN monitor agent audits host-host connections, services used, and volume of traffic.

It searches for significant events, such as sudden changes in network load, the use of security-related services, and network activities such as *rlogin*.



Agent Architecture

Honeypots

Honeypot is a system, which sole purpose is to attract potential intruders and record their activity, to further analyze and investigate security breaches.

They are helpful in learning activities of an intruder who has gained control over the Honeypot.

Honeypots

Honeypots are decoy systems that are designed to lure a potential attacker away from critical systems.

Honeypots are designed to

- divert an attacker from accessing critical systems
- collect information about the attacker's activity
- encourage the attacker to stay on the system long enough for administrators to respond

Honeypots

These systems are filled with fabricated information designed to appear valuable but that a legitimate user of the system wouldn't access.

Thus, any access to the honeypot is suspect.

The system is instrumented with sensitive monitors and event loggers that detect these accesses and collect information about the attacker's activities.

Intrusion Detection Exchange Format

To facilitate the development of distributed intrusion detection systems that can function across a wide range of platforms and environments, standards are needed to support interoperability.

Such standards are the focus of the IETF Intrusion Detection Working Group.

The purpose of the working group is to define data formats and exchange procedures for sharing information of interest to intrusion detection and response systems and to management systems that may need to interact with them.

Intrusion Detection Exchange Format

The outputs of this working group include:

1. A requirements document, which describes the high-level functional requirements for communication between intrusion detection systems and with management systems, including the rationale for those requirements. Scenarios will be used to illustrate the requirements.

2. A common intrusion language specification, which describes data formats that satisfy the requirements.

3. A framework document, which identifies existing protocols best used for communication between intrusion detection systems, and describes how the devised data formats relate to them.

RAMAIAH
Institute of Technology

Unit V (Text 2) – System Security

Intruders:

(Chapter 20.1 to 20.2)

- Intruders
- Intrusion Detection

Malicious Software: (Chapter 21.1 – 21.2)

- Types of Malicious Software
- Viruses

Firewalls : (Chapter 22.1,22.2,22.3

- The need for Firewalls
- Firewall Characteristics
- Types of Firewalls

TYPES OF MALICIOUS SOFTWARE

Malicious software can be divided into two categories:

1. Those that need a host program
2. Those that are independent

TYPES OF MALICIOUS SOFTWARE

Malicious software can be divided into two categories:

1. Those that need a host program
2. Those that are independent

Some of the key categories of malicious software are Viruses, logic bombs, Trojan Horses and backdoors

TYPES OF MALICIOUS SOFTWARE

Name	Description
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.

TYPES OF MALICIOUS SOFTWARE

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.

TYPES OF MALICIOUS SOFTWARE

Flooders	Used to attack networked computer systems with a large volume of traffic to carry out a denial-of-service (DoS) attack.
Keyloggers	Captures keystrokes on a compromised system.
Rootkit	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**.

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 horse 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.

Multiple-Threat Malware

- A **multipartite** virus infects in multiple ways.
- Typically, the multipartite virus is capable of infecting multiple types of files, so that virus eradication must deal with all of the possible sites of infection.
- A **blended attack** uses multiple methods of infection or transmission, to maximize the speed of contagion and the severity of the attack.

Multiple-Threat Malware

Some writers characterize a blended attack as a package that includes multiple types of malware.

An example of a blended attack is the Nimda attack, erroneously referred to as simply a worm.

Nimda uses four distribution methods:

- E-mail
- Windows shares
- Web servers
- Web clients

VIRUSES

A computer virus has three parts

- **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.

VIRUSES

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.

VIRUSES

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.

VIRUS STRUCTURE

A very general depiction of virus structure is shown in Figure.

In this case, the virus code,V, is prepended to infected programs, and it is assumed that the entry point to the program, when invoked, is the first line of the program.

```
    program V :=

{goto main;
    1234567;

    subroutine infect-executable :=
        {loop:
        file := get-random-executable-file;
        if (first-line-of-file = 1234567)
          then goto loop
          else prepend V to file; }

    subroutine do-damage :=
        {whatever damage is to be done}

    subroutine trigger-pulled :=
        {return true if some condition holds}

main:   main-program :=
        {infect-executable;
        if trigger-pulled then do-damage;
        goto next;}
next:

}
```

A Simple Virus

VIRUS STRUCTURE

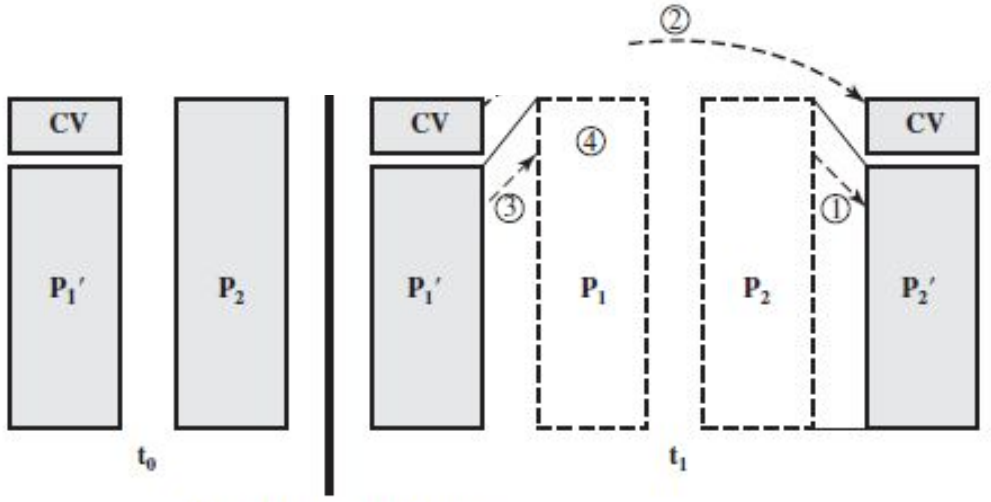
```
    program CV :=

{goto main;
    01234567;

    subroutine infect-executable :=
            {loop:
                file := get-random-executable-file;
            if (first-line-of-file = 01234567) then goto loop;
        (1)  compress file;
        (2)  prepend CV to file;
        }

main:   main-program :=
            {if ask-permission then infect-executable;
        (3)  uncompress rest-of-file;
        (4)  run uncompressed file;}
        }
```

Logic for a Compression Virus

RAMAIAH
Institute of Technology

VIRUS STRUCTURE

1. For each uninfected file P_2 that is found, the virus first compresses that file to produce P'_2, which is shorter than the original program by the size of the virus.
2. A copy of the virus is prepended to the compressed program.
3. The compressed version of the original infected program, P'_1, is uncompressed.
4. The uncompressed original program is executed.



A Compression Virus

Viruses Classification

A virus **classification by target** includes the following categories:

1. **Boot sector infector:** Infects a master boot record or boot record and spreads when a system is booted from the disk containing the virus.
2. **File infector:** Infects files that the operating system or shell consider to be executable.
3. **Macro virus:** Infects files with macro code that is interpreted by an application.

Viruses Classification

A virus classification by concealment strategy includes the following categories:

1.Encrypted virus: A typical approach is as follows.A portion of the virus creates a random encryption key and encrypts the remainder of the virus.The key is stored with the virus.When an infected program is invoked, the virus uses the stored random key to decrypt the virus.When the virus replicates, a different random key is selected. Because the bulk of the virus is encrypted with a different key for each instance, there is no constant bit pattern to observe.

Viruses Classification

2.Stealth virus: A form of virus explicitly designed to hide itself from detection by antivirus software. Thus, the entire virus, not just a payload is hidden.

3.Polymorphic virus: A virus that mutates with every infection, making detection by the “signature” of the virus impossible.

Viruses Classification

4. Metamorphic virus: As with a polymorphic virus, a metamorphic virus mutates with every infection.

The difference is that a metamorphic virus rewrites itself completely at each iteration, increasing the difficulty of detection.

Metamorphic viruses may change their behavior as well as their appearance.

Virus Kits

Macro viruses are particularly threatening for a number of reasons:

1. A macro virus is platform independent. Many macro viruses infect Microsoft Word documents or other Microsoft Office documents. Any hardware platform and operating system that supports these applications can be infected.

2. Macro viruses infect documents, not executable portions of code. Most of the information introduced onto a computer system is in the form of a document rather than a program.

3. Macro viruses are easily spread.A very common method is by electronic mail.

4. Because macro viruses infect user documents rather than system programs, traditional file system access controls are of limited use in preventing their spread.

E-Mail Viruses

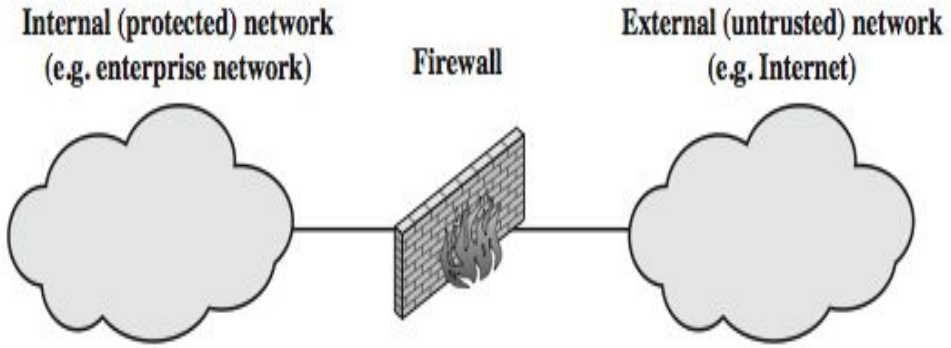
The first rapidly spreading e-mail viruses, such as Melissa, made use of a Microsoft Word macro embedded in an attachment. If the recipient opens the e-mail attachment, the Word macro is activated. Then

1. The e-mail virus sends itself to everyone on the mailing list in the user's e-mail package.

2. The virus does local damage on the user's system.

Firewalls

- 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.



THE NEED FOR FIREWALLS

Information systems in corporations, government agencies, and other organizations have undergone a steady evolution.

The following are notable developments:

- Centralized data processing system, with a central mainframe supporting a number of directly connected terminals
- Local area networks (LANs) interconnecting PCs and terminals to each other and the mainframe

THE NEED FOR FIREWALLS

- Premises network, consisting of a number of LANs, interconnecting PCs, servers, and perhaps a mainframe or two
- Enterprise-wide network, consisting of multiple, geographically distributed premises networks interconnected by a private wide area network (WAN)
- Internet connectivity, in which the various premises networks all hook into the Internet and may or may not also be connected by a private WAN

FIREWALL CHARACTERISTICS

Lists the following design goals for a firewall:

- All traffic from inside to outside, and vice versa, must pass through the firewall.
- Only authorized traffic, as defined by the local security policy, will be allowed to pass.
- Trusted computer systems are suitable for hosting a firewall and often required in government applications.

FIREWALL CHARACTERISTICS

Four general techniques that firewalls use to control access and enforce the site's security policy.

1. **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.
2. **Direction control:** Determines the direction in which particular service requests may be initiated and allowed to flow through the firewall.

FIREWALL CHARACTERISTICS

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.

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.

FIREWALL CHARACTERISTICS

What to expect from a firewall?

The following capabilities are within the scope of a firewall:

- 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 attack.
- A firewall provides a location for monitoring security-related events. Audits and alarms can be implemented on the firewall system.

FIREWALL CHARACTERISTICS

What to expect from a firewall?

The following capabilities are within the scope of a firewall:

- 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.
- A firewall can serve as the platform for IPsec.

Firewalls limitations

- The firewall cannot protect against attacks that bypass the firewall.
- The firewall may not protect fully against internal threats, such as a disgruntled employee or an employee who unwittingly cooperates with an external attacker.
- 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.
- 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.

It can operate as a positive filter, allowing to pass only packets that meet specific criteria, or as a negative filter, rejecting any packet that meets certain criteria.

Depending on the type of firewall, it may examine one or more protocol headers in each packet, the payload of each packet, or the pattern generated by a sequence of packets.

In this section, we look at the principal types of firewalls.

TYPES OF FIREWALLS

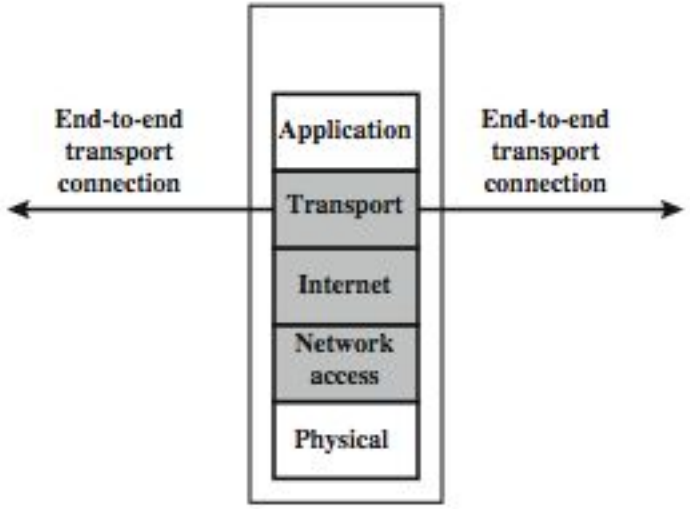
Principal types of firewalls

1. Packet Filtering Firewall
2. Stateful Inspection Firewalls
3. Application-Level Gateway
4. Circuit-Level Gateway

RAMAIAH
Institute of Technology

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



RAMAIAH
Institute of Technology

Packet Filtering Firewall

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

Packet Filtering Firewall

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.

RAMAIAH
Institute of Technology

Packet Filtering Firewall

Table gives some examples of packet filtering rule sets.

Table 20.1 Packet-Filtering Examples

A

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

B

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

C

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

D

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

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 Firewall - Attacks

IP address spoofing

- fake source address to be trusted
- add filters on router to block

source routing attacks

- attacker sets a route other than default
- block source routed packets

tiny fragment attacks

- split header info over several tiny packets
- either discard or reassemble before check

RAMAIAH
Institute of Technology

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

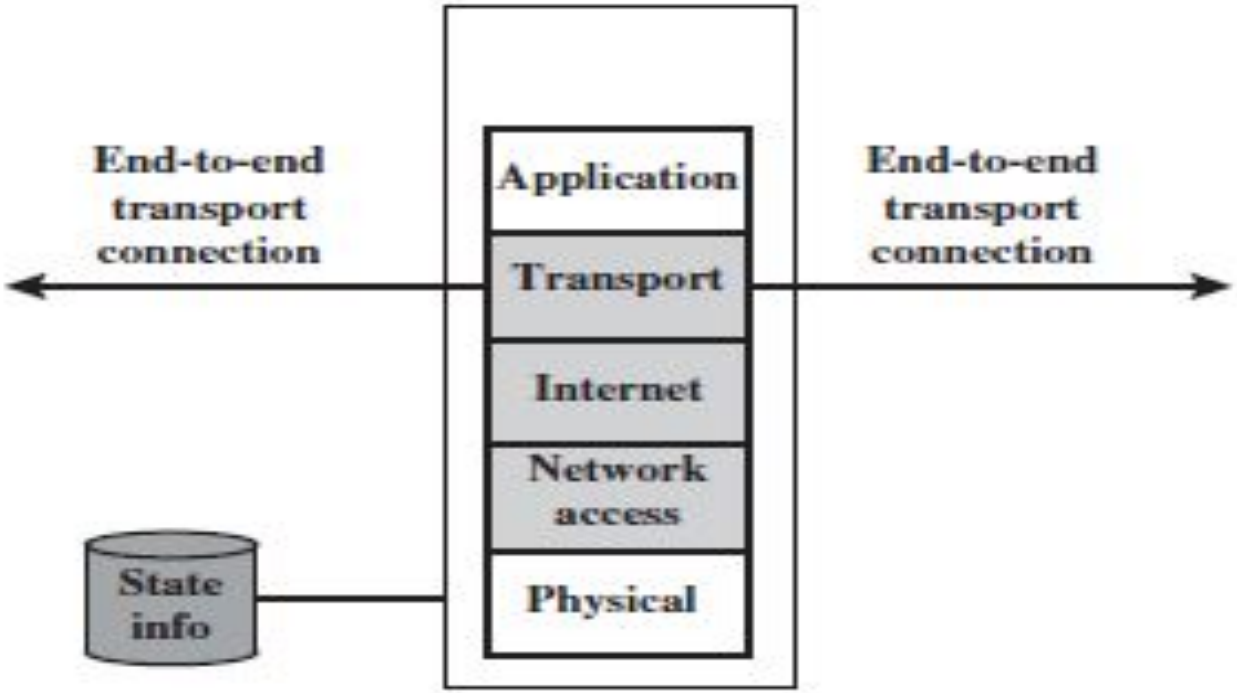
Hence are better able to detect bogus packets out of context may even inspect limited application data

Stateful Packet Filters

Table 22.2 Example Stateful Firewall Connection State Table [WACK02]

Source Address	Source Port	Destination Address	Destination Port	Connection State
192.168.1.100	1030	210.22.88.29	80	Established
192.168.1.102	1031	216.32.42.123	80	Established
192.168.1.101	1033	173.66.32.122	25	Established
192.168.1.106	1035	177.231.32.12	79	Established
223.43.21.231	1990	192.168.1.6	80	Established
2122.22.123.32	2112	192.168.1.6	80	Established
210.922.212.18	3321	192.168.1.6	80	Established
24.102.32.23	1025	192.168.1.6	80	Established
223.21.22.12	1046	192.168.1.6	80	Established

Stateful Packet Filters

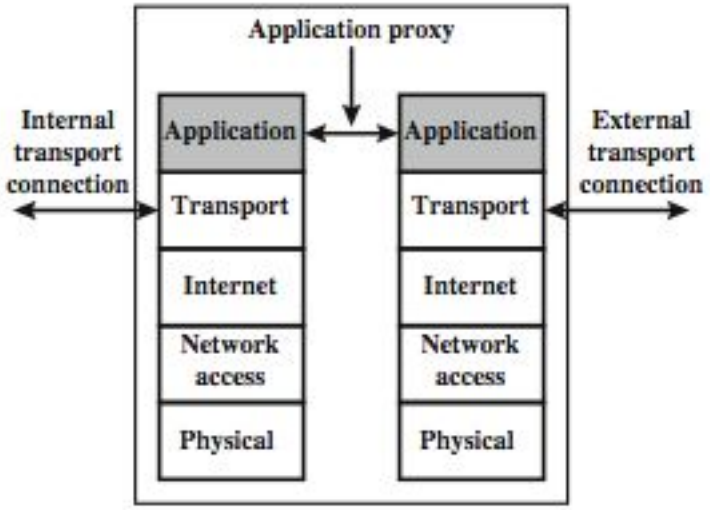


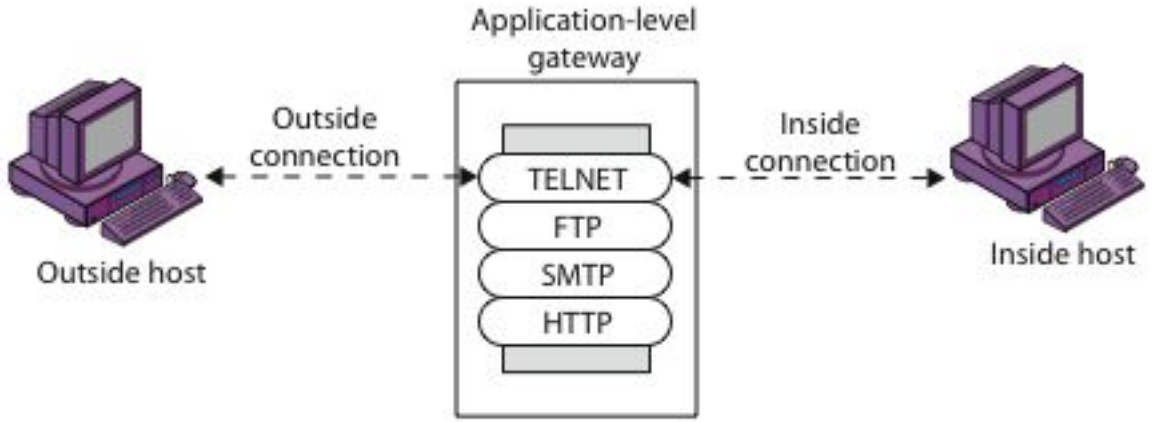
(c) Stateful inspection firewall

Application Level Gateway (or Proxy)

- have application specific gateway / proxy
- has full access to protocol
  - user requests service from proxy
  - proxy validates request as legal
  - then actions request and returns result to user
  - can log / audit traffic at application level
- need separate proxies for each service
  - some services naturally support proxying
  - others are more problematic

Application Level Gateway (or Proxy)





(b) Application-level gateway

RAMAIAH
Institute of Technology

Circuit Level Gateway

relays two TCP connections

imposes security by limiting which such connections are allowed

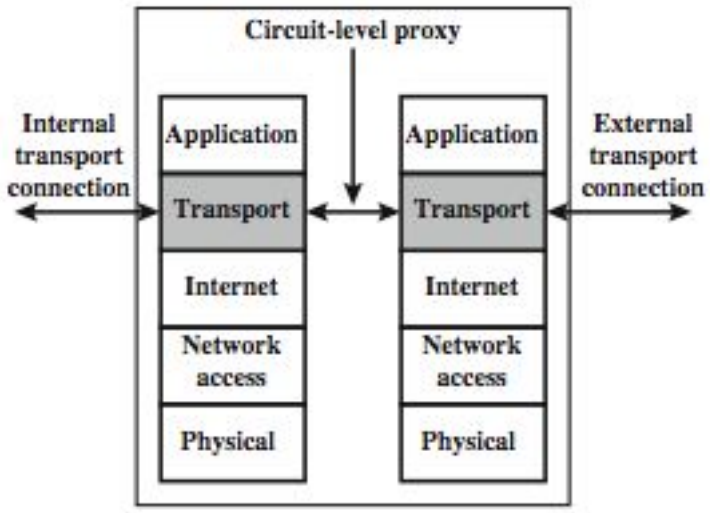
once created usually relays traffic without examining contents

typically used when trust internal users by allowing general outbound connections

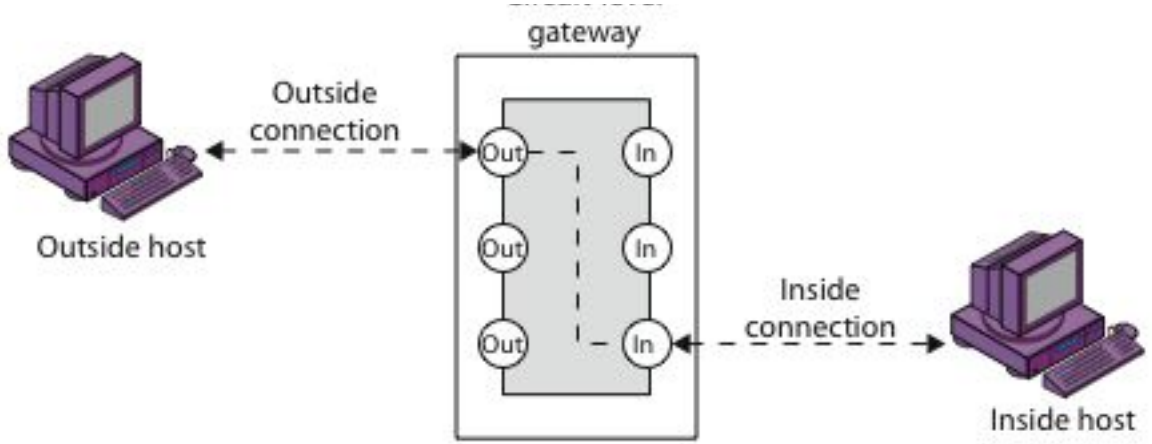
SOCKS is commonly used

RAMAIAH
Institute of Technology

Circuit Level Gateway



(e) Circuit-level proxy firewall



(c) Circuit-level gateway

Thank you