UNIT IV SECURITY PRACTICE & SYSTEM SECURITY

Authentication applications – Kerberos – X.509 Authentication services - Internet Firewalls for Trusted System: Roles of Firewalls – Firewall related terminology-Types of Firewalls - Firewall designs – SET for E-Commerce Transactions. Intruder – Intrusion detection system – Virus and related threats – Countermeasures – Firewalls design principles – Trusted systems – Practical implementation of cryptography and security.

PART-A

1. **Define Intruder and List Classes of Intruders. (APR/MAY 2011) (APR/MAY 2010)**

   An individual who gains, or attempts to gain, unauthorized access to a computer system or to gain unauthorized privileges on that system.

   - Masquerader
   - Misfeasor
   - Clandestine user

2. **Write short notes on Intrusion detection system and Malicious software. (NOV/DEC 2011) (APRIL/MAY 2015)**

   A set of automated tools desi ned to detect unauthorized access to a host system.

   Malicious software:

   Malicious software is software that is intentionally included or inserted in a system for a harmful purpose.

3. **What do you meant by Trojan horse? (APR/MAY 2011)**

   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.

4. **Define Firewall. (APRIL/MAY 2015)**

   A firewall is a device or set of devices designed to permit or deny network transmissions based upon a set of rules and is frequently used to protect networks from unauthorized access.

5. **What is worm? (NOV/DEC 2013) (APR/MAY 2015)**

A worm is a self-replicating virus that does not alter files but resides in active memory and duplicates itself. Worms use parts of an operating system that are automatic and usually invisible to the user.

6. List out the requirements of Kerberos. (APR/MAY 2011) (APR/MAY 2010)

- **Secure:** A network eavesdropper should not be able to obtain the necessary information to impersonate a user.
- **Reliable**: For all services that rely on Kerberos for access control, lack of availability of the Kerberos service means lack of availability of the supported services.
- **Transparent:** Ideally, the user should not be aware that authentication is taking place, beyond the requirement to enter a password.
- **Scalable:** The system should be capable of supporting large numbers of clients and servers. This suggests a modular, distributed architecture.

7. What are the two types of audit records? (MAY/JUNE 2012)

- Native audit records
- Detection specific audit records

8. Define Honeypot . (APR/MAY 2010)

A decoy system desig ed to lure a potential attacker away from critical systems. It is a form of intrusion detection.

9. What is meant by polymorphic viruses? (APR/MAY 2008)

A virus that mutates with every infection, making detection by the "signature" of the virus impossible.

10. What are zombies? (May/June 2014)

Zombies are programs which secretly takes over another networked computer. Then uses it to indirectly launch attacks. It is often used to launch distributed denial of service (DDoS) attacks. It exploits known flaws in network systems.

1. KERBEROS

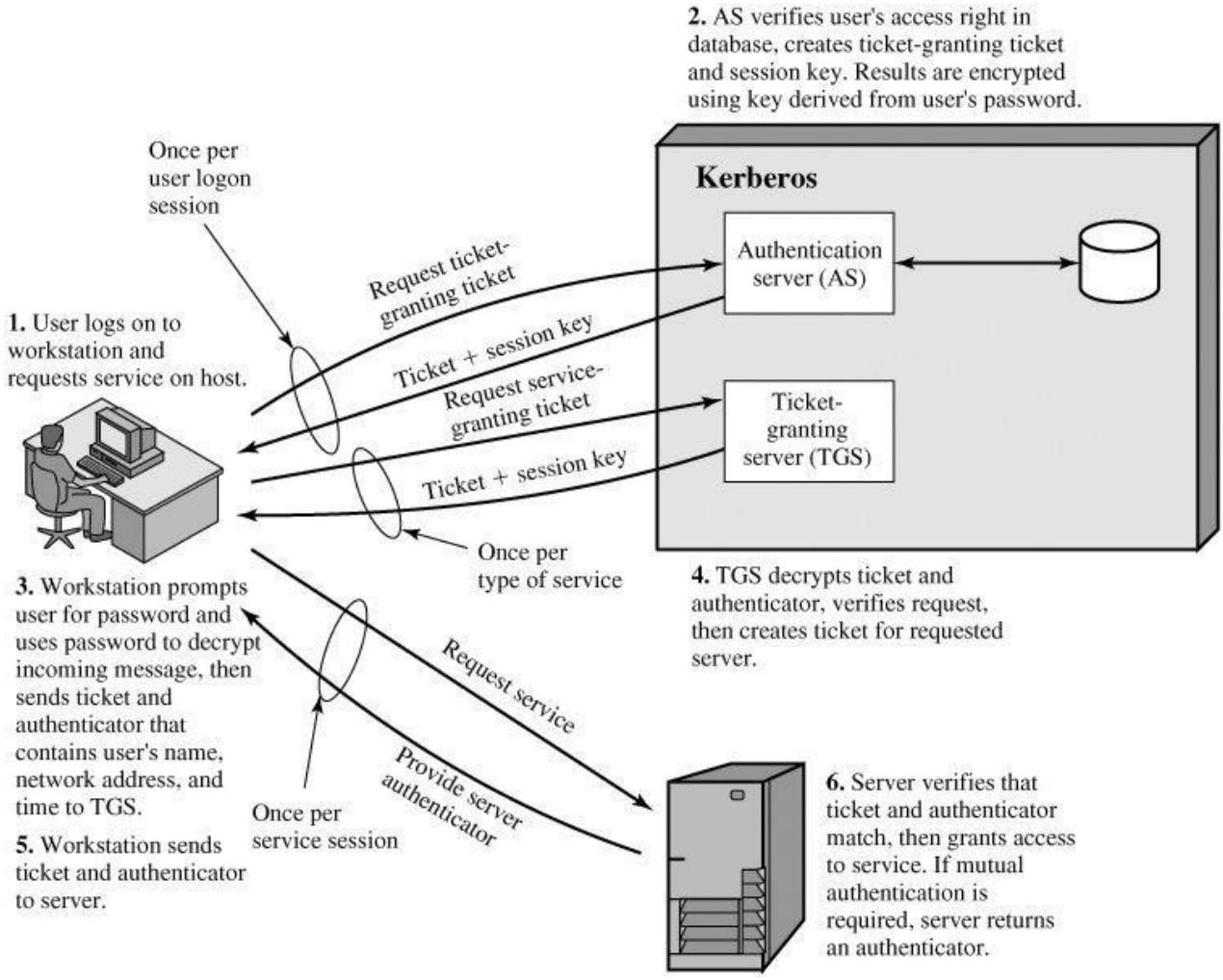
Kerberos is an authentication service developed as part of Project Athena at MIT. The problem that Kerberos addresses is this: Assume an open distributed environment in which users at workstations wish to access services on servers distributed throughout the network.

Kerberos listed the following requirements:

- ✓ **Secure:** A network eavesdropper should not be able to obtain the necessary information to impersonate a user.
- ✓ **Reliable**: For all services that rely on Kerbe os for access control, lack of availability of the Kerberos service means lack of availability of the supported services.

Transparent: Ideally, the user should not be aware that authentication is taking place, beyond the requirement to enter a password.

- ✓ **Scalable:** The system should be capable of supporting large numbers of clients and servers. This suggests a modular, distributed architecture.



Overview of Kerberos

Kerberos Version 4

✓ Version 4 of Kerberos m kes use of DES, in a rather elaborate protocol, to provide the authentication service.

✓ An alternative is to use an authentication server (AS) that knows the passwords of all users and stores these in a centralized database.

(1) C → AS: $ID_C||P_C||ID_V$

(2) AS →C:Ticket

(3) C → V: $ID_C||$Ticket Ticket

= $E(K_v, [ID_C||AD_C||ID_V])$

Where C = client, AS= authentication server, V=server ,ID_C= identifier of user on C

ID_V= identifier of V, P_C = password of user on C, AD_C= network address of C

K_V= secret encryption key shared by AS and V

Kerberos Realms and Multiple Kerberi

1. The Kerberos server must have the user ID and hashed passwords of all participating users in its database. All users are registered with the Kerberos server.
2. The Kerberos server must share a secret key with each server. All servers are registered with the Kerberos server.

Such an environment is referred to as a **Kerberos realm**.

Request for Service in Another Realm

(1) C → AS: $ID_c||ID_{tgs}||TS_1$

(2) AS → C:$E(K_c, [K_{c,tgs}||ID_{tgs}||TS_2||$L fetime$_2||Ticket_{tgs}])$

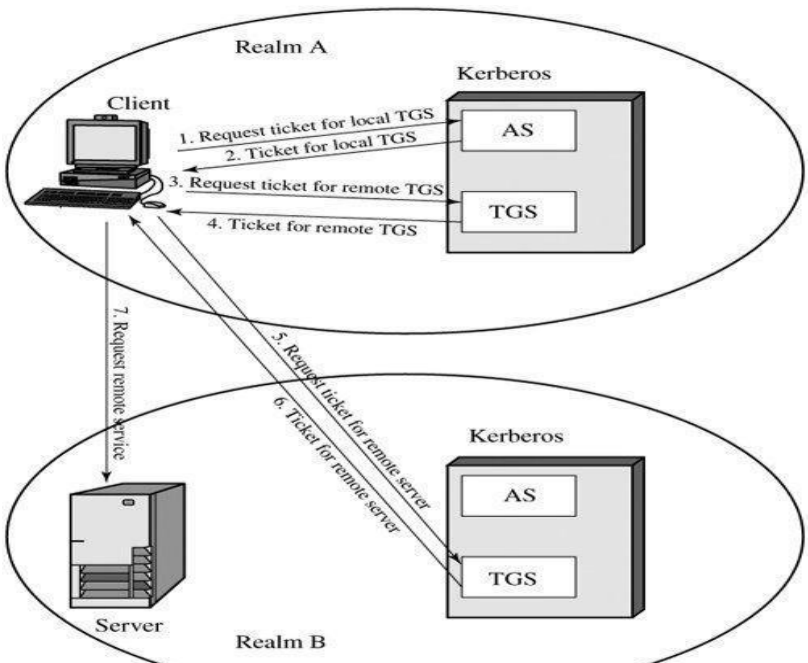
(3) C → TGS:$ID_{tgsrem}||Ticket_{tgs}||$Auth nticator$_c$

(4) TGS → C: $E(K_{c,tgs}, [K_{c,tgsrem}||ID_{tgsrem}||TS_4||Ticket_{tgsrem}])$

(5) C → TGS_{rem}: $ID_{vrem}||Ticket_{tgsrem}||Authenticator_c$

(6) TGS_{rem} → C: $E(K_{c,tgsrem}, [K_{c,vrem}||ID_{vrem}||TS_6||Ticket_{vrem}])$

(7) C → V_{rem}:$Ticket_{vrem}||Authenticator_c$



Differences between Versions 4 and 5

1. **Encryption system dependence:** Version 4 requires the use of DES. In version 5, ciphertext is tagged with an encryption type identifier so that any encryption technique may be used.
2. **Internet protocol dependence**: Version 4 requires the use of Internet Protocol (IP) addresses. Version 5 network addresses are tagged with type and length, allowing any network address type to be used.
3. **Message byte ordering:** In version 4, the sender of a message employs a byte ordering of its own choosing and tags the message to indicate least significant byte in lowest address or most significant byte in lowest address.
4. **Ticket lifetime:** Lifetime values in version 4 are encoded n an 8-bit quantity in units of five minutes. Thus, the maximum lifetime that can be expressed is 2^8 x 5 = 1280 minutes, or a little over 21 hours. In version 5, tickets include an explicit start time and end time, allowing tickets with arbitrary lifetimes.
5. **Authentication forwarding**: Version 4 does not allow credentials issued to one client to be forwarded to some other host and us d by some other client. Version 5 provides this capability.
6. **Interrealm authentication**: In version 4, interoperability among N realms requires on the order of N^2 Kerberos-to-Kerberos relationships. Version 5 supports a method that requires fewer relationships.

INTRUSION DETECTION

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
    - ✓ Attempts to copy the password file
    - ✓ Suspicious remote procedure call
    - ✓ Attempts to connect

Intrusion Techniques

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

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.
3. Try words in the system's online dictionary or a l st 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, a d 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.
8. Tap the line between a remote user a d the host system.

Intrusion Detection

1. If an intrusion is detected quickly enough, the intruder can be identified and ejected from the system before any damage is done or any data are compromised.
2. An effective intrusion detection system can serve as a deterrent, so acting to prevent intrusions.
3. Intrusion detection nables the collection of information about intrusion techniques that can be used to strengthen the intrusion prevention facility.

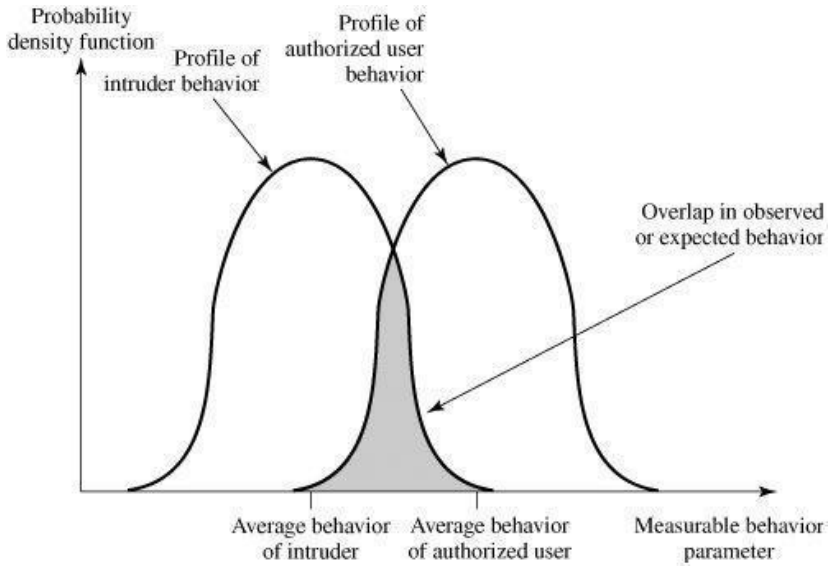


Figure: Profiles of Behavior of Intruders and Authorized Users

Approaches to intrusion detection:

1. **Statistical anomaly detection:** Involves the collection of data relating to the behavior of legitimate users over a p riod of time. Then statistical tests are applied to observed behavior to determi e 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:** I volves 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, t o plans are used:

- **Native audit records:** Virtually all multiuser operating systems include accounting software that collects information on user activity.
- Detection-specific audit records: A collection facility can be implemented that generates audit records containing only that information required by the intrusion detection system.

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

Statistical Anomaly Detection

Examples of metrics that are useful for profile-bas d intrusion detection are the following:

- ✓ **Counter:** A nonnegative integer 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.
- **Resource utilization:** Quantity of resources consumed during a specified period.

Lists the following approaches that may be taken:

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

Rule-Based Intrusion Detection

1. Users should not read files in other users' personal directories.
2. Users must not rite other users' files.
3. Users ho 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

- ✓ A distributed intrusion detection system may need to deal with different audit record formats.
- ✓ r more nodes in the network will serve as collection and analysis points for the data from the systems on the network.
- ✓ Either a centralized or decentralized architecture can be used.

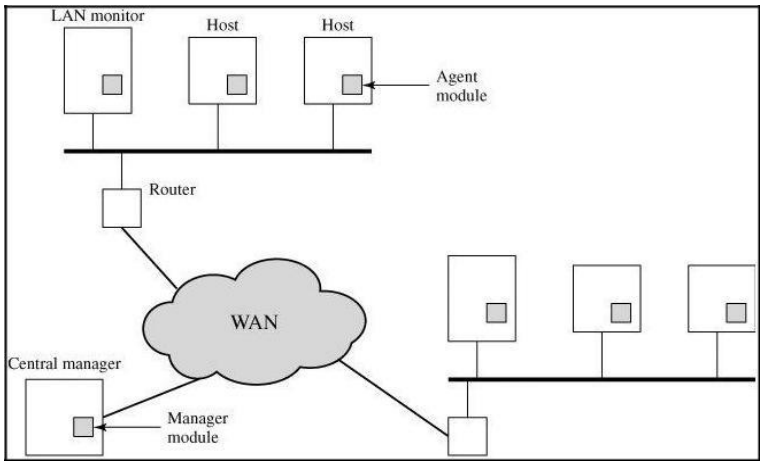


Figure: Architecture for Distributed Intrusion Detection

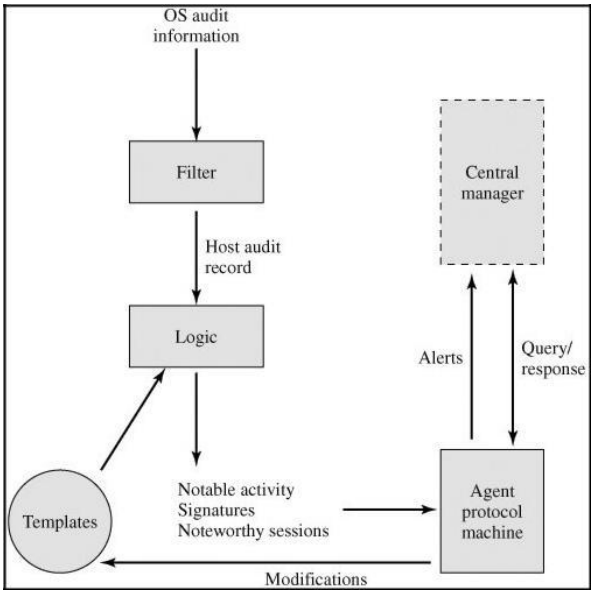


Figure : Agent Architecture

Honeypots

A relatively recent innovation in intrusion detection technology is the honeypot. 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

Password Management

Password Protection

The front line of defense against intruders is the password system. Virtually all multiuser systems require that a user provide not only a name or identifier (ID) but also a password.

- ✓ The ID determines whether the user is authorized to gain access to a system.
- ✓ The ID determines the privileges accorded to the user.
- ✓ The ID is used in what is referred to as discretionary access control.

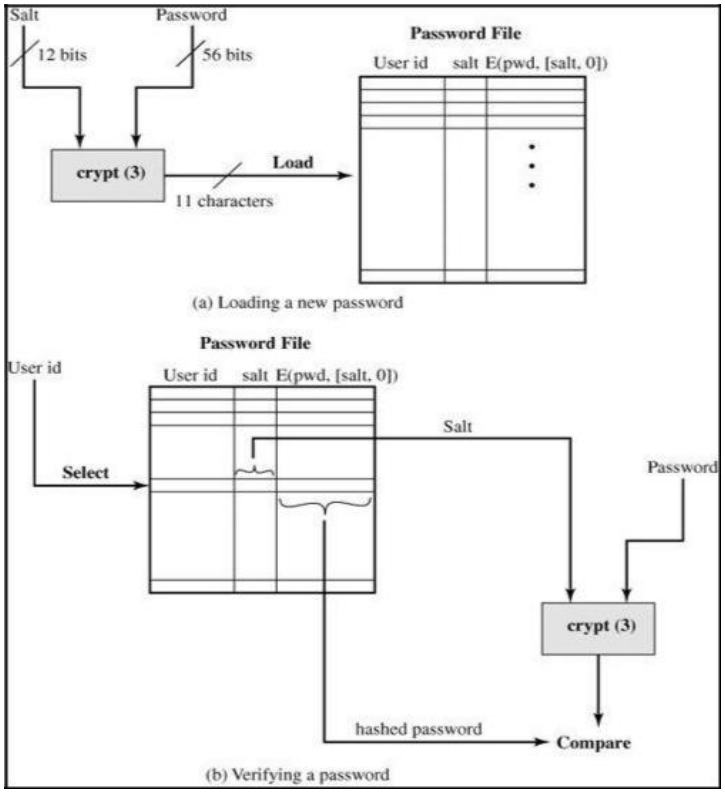


Figure: UNIX Password Scheme

The salt serves three purposes:

- It prevents duplicate passwords from being visible in the password file.
- It effectively increases the length of the password without requiring the user to remember two additional characters.
- It prevents the use of a hardware implementation of DES

Access Control

- Many systems, including most UNIX systems, are susceptible to unanticipated break-ins.
- An accident of protection might render the password file readable, thus compromising all the accounts.
- Some of the users have accounts on other mach nes n other protection domains, and they use the same password.

Password Selection Strategies

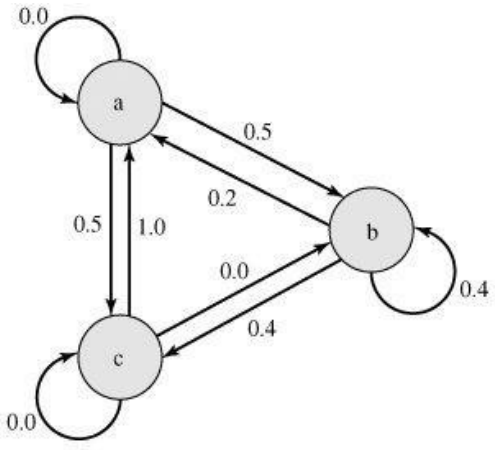
Four basic techniques are in use:

- User education
- Computer-generated passwords
- Reactive password checking
- Proactive password checking

The first approach is a simple system for rule enforcement. For example, the following rules could be enforced:

- All passwords must be at least eight characters long.
- In the first eight characters, the passwords must include at least one each of uppercase, lowercase, numeric digits, and punctuation marks.
- Space: The dictionary must be very large to be effective.
- Time: The time required to search a large dictionary may itself be large.

Two techniques for developing an effective and efficient proactive password checker that is based on rejecting words on a list show promise. One of these develops a Markov model for the generation of guessable passwords



M = {3, {a, b, c} , T, 1 } where

$$T = \begin{bmatrix} 0.0 & 0.5 & 0.5 \\ 0.2 & 0.4 & 0.4 \\ 1.0 & 0.0 & 0.0 \end{bmatrix}$$

e.g., string probably from this language: abbcacaba

e.g., string probably not from this language: aacccbaaa

Figure: An example Markov Model

3. VIRUSES

Malicious Programs

Terminology of Malicious Programs

Name	Description
Virus	Attaches itself to a program and propagates copies of itself to other programs
Worm	Program that propagates copies of itself to other computers
Logic bomb	Triggers action when condition occurs
Trojan horse	Program that contains unexpected additio al functionality
Backdoor (trapdoor)	Program modification that allows unauthorized access to functionality
Exploits	Code specific to a single vuln rability or set of vulnerabilities
Downloaders	Program that installs other tems 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 la ge 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
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

Terminology of Malicious Programs

Name	Description
Zombie	Program activated on an infected machine that is activated to launch attacks on other machines

Backdoor

✓ A backdoor, also known as a trapdoor, is a secret entry point into a program that allows someone that is aware of the backdoor to gain access w thout going through the usual security access procedures.

✓ Programmers have used backdoors legitimately for many years to debug and test programs.

✓ This usually is done when the programmer is develop ng an application that has an authentication procedure, or a long setup, requiring the user to enter many different values to run the application.

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 horse is a useful, or apparently useful, program or command procedure containing hidden code that, when invoked, performs some un anted 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, changed the invoking user's file permissions so that the files are readable by any user.

Zombie

✓ A zombie is a program that secretly takes over another Internet-attached computer and then uses that computer to launch attacks that are difficult to trace to the zombie's creator.

- Zombies are used in denial-of-service attacks, typically against targeted Web sites.

✓ The zombie is planted on hundreds of computers belonging to unsuspecting third parties, and then used to overwhelm the target Web site by launching an overwhelming onslaught of Internet traffic.

The Nature of Viruses

✓ A virus is a piece of software that can "infect" oth r programs by modifying them; the modification includes a copy of the virus program, which can then go on to infect other programs.

✓

During its lifetime, a typical virus goes throu h 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 an identical copy of itself into other programs or into cert in system areas on the disk. 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.

Virus Structure

✓ A virus can be prepended or postpended to an executable program, or it can be embedded in some other fashion. The key to its operation is that the infected program, when invoked, will first execute the virus code and then execute the original code of the program.

✓ 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}
```

Figure: A Simple Virus

```
     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;}
          }
```

Figure: A Compression Virus

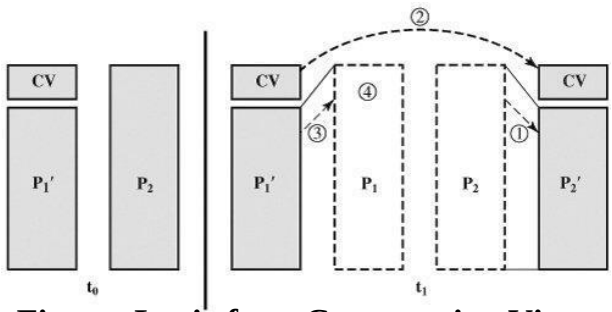


Figure: Logic for a Compression Virus

Initial Infection

✓ Once a virus has gained entry to a system by infecting a single program, it is in a position to infect some or all other executable files on that system when the infected program executes.

Types of Viruses

- Parasitic virus: The traditional and still most common form of virus. A parasitic virus attaches itself to executable files and replicates, when the infected program is executed, by finding other executable files to infect.

- Memory-resident virus: Lodges in main memory as part of a resident system program. From that point on, the virus infects every program that executes.
- Boot sector virus: Infects a master boot record or boot record and spreads when a system is booted from the disk containing the virus.
- Stealth virus: A form of virus explicitly designed to hide itself from detection by antivirus software.
- Polymorphic virus: A virus that mutates with every infection, making detection by the "signature" of the virus impossible.
- Metamorphic virus: As with a polymorphic virus, a metamorphic virus mutates with every infection. The difference is that a metamorph c v rus rewrites itself completely at each iteration, increasing the difficulty of detection. Metamorphic viruses my change their behavior as well as their appeara ce.

Macro Viruses

1. A macro virus is platform independent.
2. Macro viruses infect documents, not executable portions of code.
3. Macro viruses are easily spread. A very common method is by electronic mail.

E-mail Viruses

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.

Worms

✓ 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.

✓ An e-mail virus has some of the characteristics of a worm, because it propagates itself from system to system. However, we can still classify it as a virus because it requires a human to move it forward.

✓ A worm actively seeks out more machines to infect and each machine that is infected serves as an automated launching pad for attacks on other machines.

Examples include the following:

- **Electronic mail facility**: A worm mails a copy of itself to other systems.
- **Remote execution capability**: A worm executes a copy of itself on another system.
- **Remote login capability**: A worm logs onto a remote system as a user and then uses commands to copy itself from one system to the other.

The new copy of the worm program is then run on the remote system where, in addition to any functions that it performs at that system, it continues to spread in the same fashion.

1. Search for other systems to infect by examining host tables or s milar repositories of remote system addresses.

2. Establish a connection with a remote system.
3. Copy itself to the remote system and cause the copy to be run.

The Morris Worm

For each discovered host, the worm tried a umber of methods for gaining access:

1. It attempted to log on to a remote host as a legitimate user. To obtain the passwords, the worm ran a password-cracking program that tried
    a. Each user's account name a d simple permutations of it
    b. A list of 432 built-in passwords that Morris thought to be likely candidates
    c. All the words in the local system directory
2. It exploited a bug in the fi ger protocol, which reports the whereabouts of a remote user.
3. It exploited a trapdoor in the debug option of the remote process that receives and sends mail.

Recent Worm Attacks

Nimda spreads by multiple mechanisms:

- from client to client via e-mail
- from client to client via open network shares
- from Web server to client via browsing of compromised Web sites
- from client to Web server via active scanning for and exploitation of various Microsoft IIS 4.0 / 5.0 directory traversal vulnerabilities
- from client to Web server via scanning for the back doors left behind by the "Code Red II" worms

State of Worm Technology

The state of the art in worm technology includes the following:

- **Multiplatform**: Newer worms are not limited to Windows machines but can attack a variety of platforms, especially the popular varieties of UNIX.
- **Multiexploit:** New worms penetrate systems in a variety of ways, using exploits against Web servers, browsers, e-mail, file sharing, and other network-based applications.
- **Ultrafast spreading**: One technique to accelerate the spread of a worm is to conduct a prior Internet scan to accumulate Internet addresses of vulnerable machines.
- **Polymorphic**: To evade detection, skip past filters, and foil real-time analysis, worms adopt the virus polymorphic technique. Each copy of the worm has new code generated on the fly using functionally equivalent instructions and encryption techniques.

- **Metamorphic:** In addition to changing their appearance, metamorphic worms have a repertoire of behavior patterns that are unleashed at different stages of propagation.
- **Transport vehicles**: Because worms can rapidly compromise a large number of systems, they are id al for spreading other distributed attack tools, such as distributed denial of service zombies.
- **Zero-day exploit**: To achieve maximum surprise and distribution, a worm should exploit an unknown vulnerability that is only discovered by the general network community hen the worm is launched.

Virus Countermeasures

Antivirus Approaches

- **Detection:** Once the infection has occurred, determine that it has occurred and locate the virus.
- **Identification:** Once detection has been achieved, identify the specific virus that has infected a program.
- **Removal:** Once the specific virus has been identified, remove all traces of the virus from the infected program and restore it to its original state. Remove the virus from all infected systems so that the disease cannot spread further.

Four generations of antivirus software:

- **First generation**: simple scanners
- **Second generation**: heuristic scanners
- **Third generation**: activity traps
- **Fourth generation**: full-featured protection

Advanced Antivirus Techniques

Generic Decryption

- Generic decryption (GD) technology e ables the antivirus program to easily detect even the most complex polymorphic viruses, while maintaining fast scanning speeds.
- In order to detect such a structure, executable files are run through a GD scanner, which contains the following elements:
  - **CPU emulator**: A softw re-based virtual computer. Instructions in an executable file are interpret d by the emulator rather than executed on the underlying processor.
  - **Virus signature scanner**: A module that scans the target code looking for known virus signatures
  - **Emulation control module**: Controls the execution of the target code.

Digital Immune System

- Integrated mail systems: Systems such as Lotus Notes and Microsoft Outlook make it very simple to send anything to anyone and to work with objects that are received.
- Mobile-program systems: Capabilities such as Java and ActiveX allow programs to move on their own from one system to another.

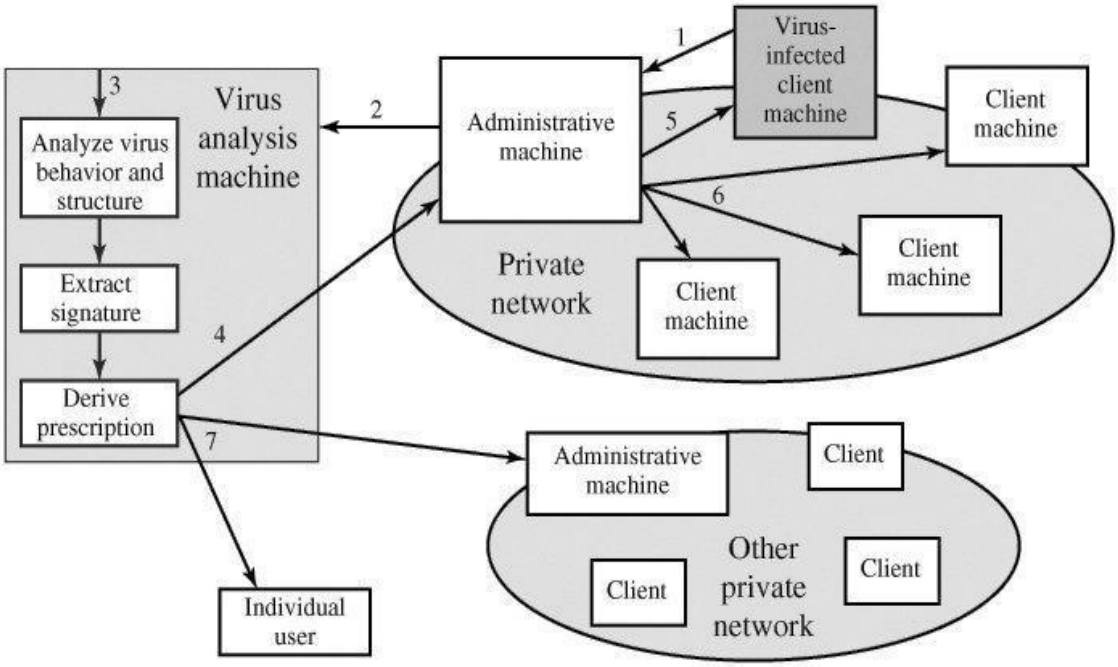


Figure: Digital Immune System

4. FIREWALL

Firewall Design Principles

- 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
- 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

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 section.
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 section.
3. The firewall itself is immune to penetration. This implies that use of a trusted system with a secure operating system.

Firewalls focused primarily on service control, but they have since evolved to provide all four:

- **Service control:** Determines the types of Internet services that can be accessed, inbound or outbound.
- **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.
- **Behavior control**: Controls how particular services are used. For example, the fire all 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.

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.
3. A firewall is a convenient platform for several Internet functions that are not security related.
4. A firewall can serve as the platform for IPSec.

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.
2. The firewall does not protect against internal threats, such as a disgruntled employee or an employee who unwittingly coop rates with an external attacker.
3. The firewall cannot protect against the transfer of virus-infected programs or files.

Types of Firewalls

Packet-Filtering Router

- A packet-filtering router applies a set of rules to each incoming and outgoing IP packet and then forwards or discards the packet.

- 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 router with three or more ports, which interface of the router the packet came from or which interface of the router the packet is destined for.

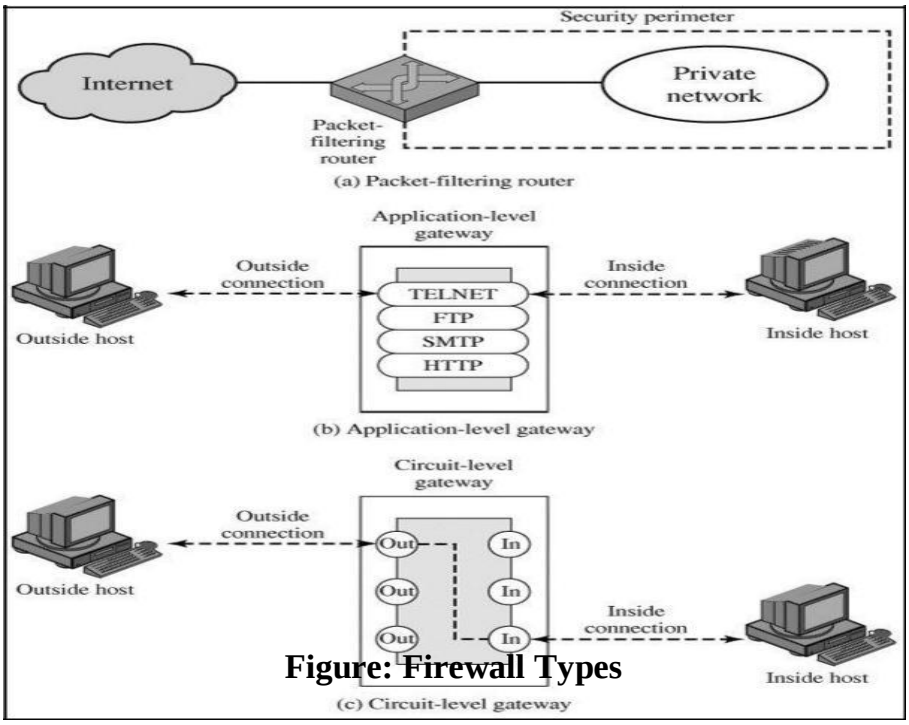


Figure: Firewall Types

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.

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

- **IP address spoofing**: The intruder transmits packets from the outside with a source IP address field containing an address of an internal host.
- **Source routing attacks**: The source station specifies the route that a packet should take as it crosses the Internet, in the hopes that this will bypass security measures that do not analyze the source routing information. The countermeasure is to discard all packets that use this option.

- **Tiny fragment attacks**: The intruder uses the IP fragmentation option to create extremely small fragments and force the TCP header information into a separate packet fragment.

Stateful Inspection Firewalls

Circuit-Level Gateway

✓ A circuit-level gateway does 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

Bastion Host

- The bastion host hardware platform executes a secure version of its operating system, making it a trusted system.

Firewall Configurations

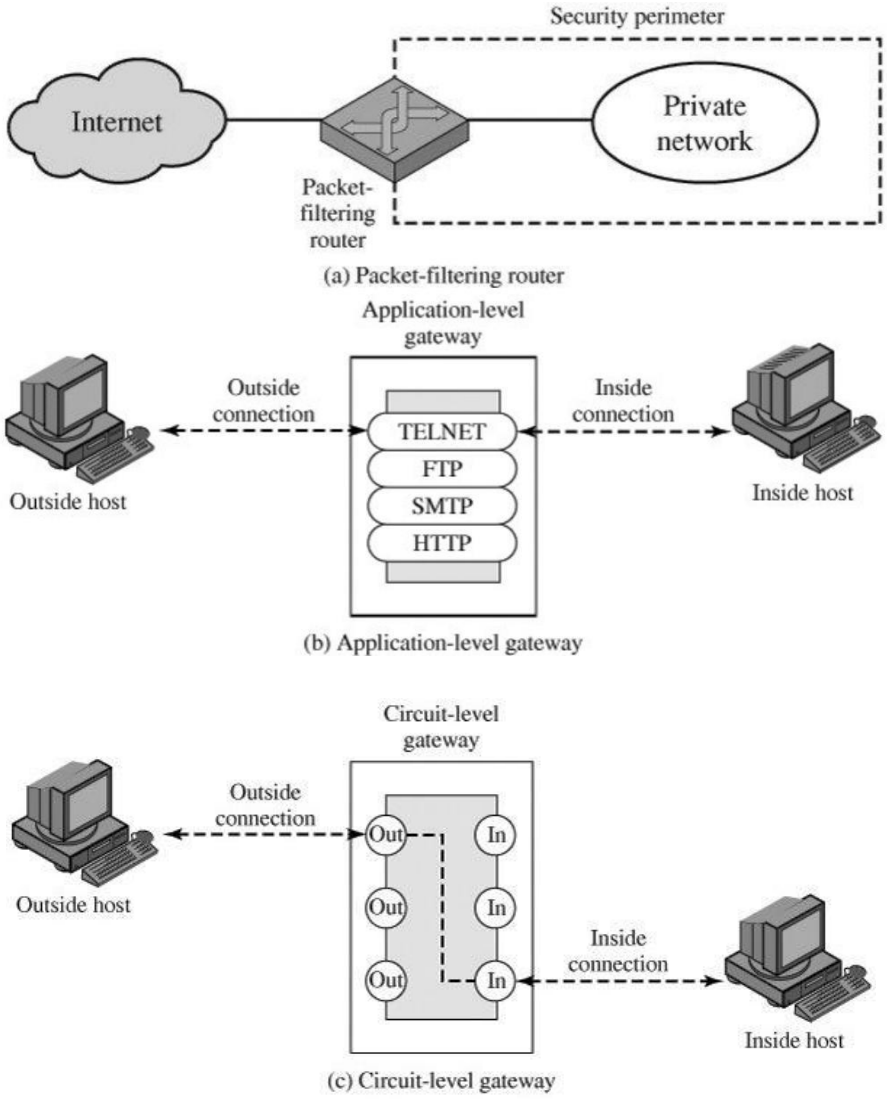


Figure: Firewall Configurations

Typically, the router is configured so that

1. For traffic from the Internet, only IP packets dested for the bastion host are allowed in.
2. For traffic from the internal network, only IP packets from the bastion host are allowed out.

5. TRUSTED SYSTEMS

The basic elements of the model are as follows:

- **Subject**: An entity capable of accessing objects. Generally, the concept of subject equates with that of process.
- **Object:** Anything to which access is controlled. Examples include files, portions of files, programs, and segments of memory.
- **Access right**: The way in which an object is accessed by a subject. Examples are read, write, and execute.

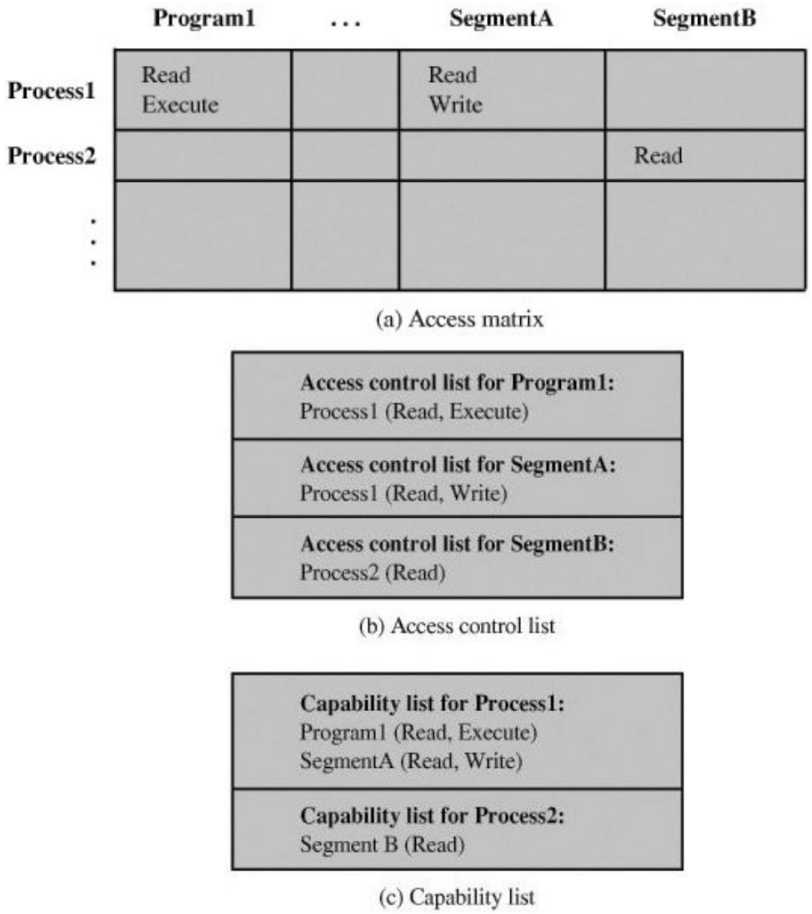


Figure: Access Control Structure

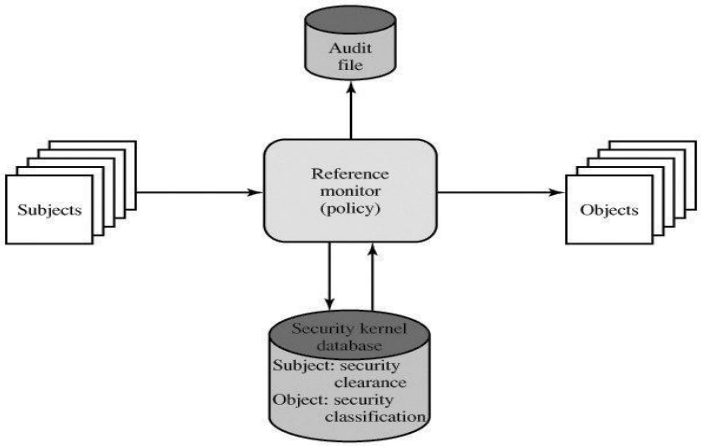


Figure: Reference Monitor Concept

Trojan Horse Defense

✓
One way to secure against Trojan horse attacks is the use of a secure, trusted operating system.

- In this case, a Trojan horse is used to get around the standard security mechanism used by most file management and operating systems: the access control list