IS FINAL

A computer virus has three parts. More generally, many contemporary types of malware also include one or more variants of each of these components:

• **Infection mechanism:** The means by which a virus spreads or propagates,

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, sometimes known as a **logic bomb**.

• **Payload:** What the virus does, besides spreading. The payload may involve damage

or may involve benign but noticeable activity.

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

• **Dormant phase:** The virus is idle. The virus will eventually be activated by some

event, such as a date, the presence of another program or file, or the capacity of

the disk exceeding some limit. Not all viruses have this stage.

• **Propagation phase:** The virus places a copy of itself into other programs or into

certain system areas on the disk. The copy may not be identical to the propagating

version; viruses often morph to evade detection. Each infected program will

now contain a clone of the virus, which will itself enter a propagation phase.

• **Triggering phase:** The virus is activated to perform the function for which it was

intended. As with the dormant phase, the triggering phase can be caused by a

variety of system events, including a count of the number of times that this copy

of the virus has made copies of itself.

Execution phase: The function is performed. The function may be harmless,

such as a message on the screen, or damaging, such as the destruction of programs and data files.

b)

Database Encryption Scheme

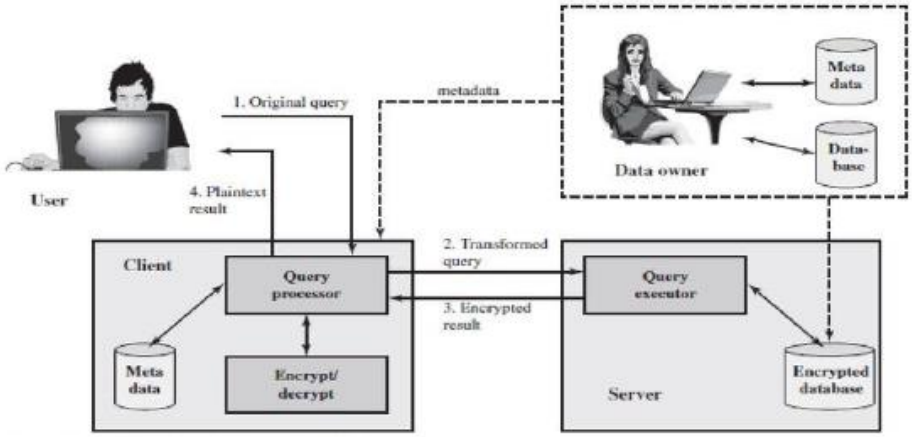
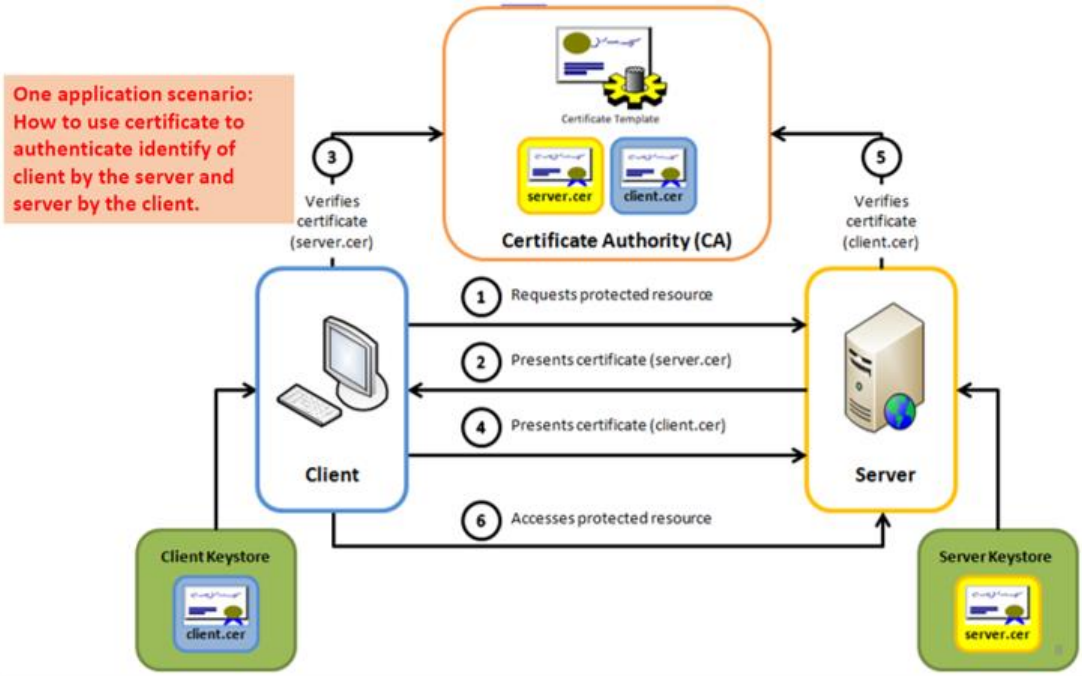
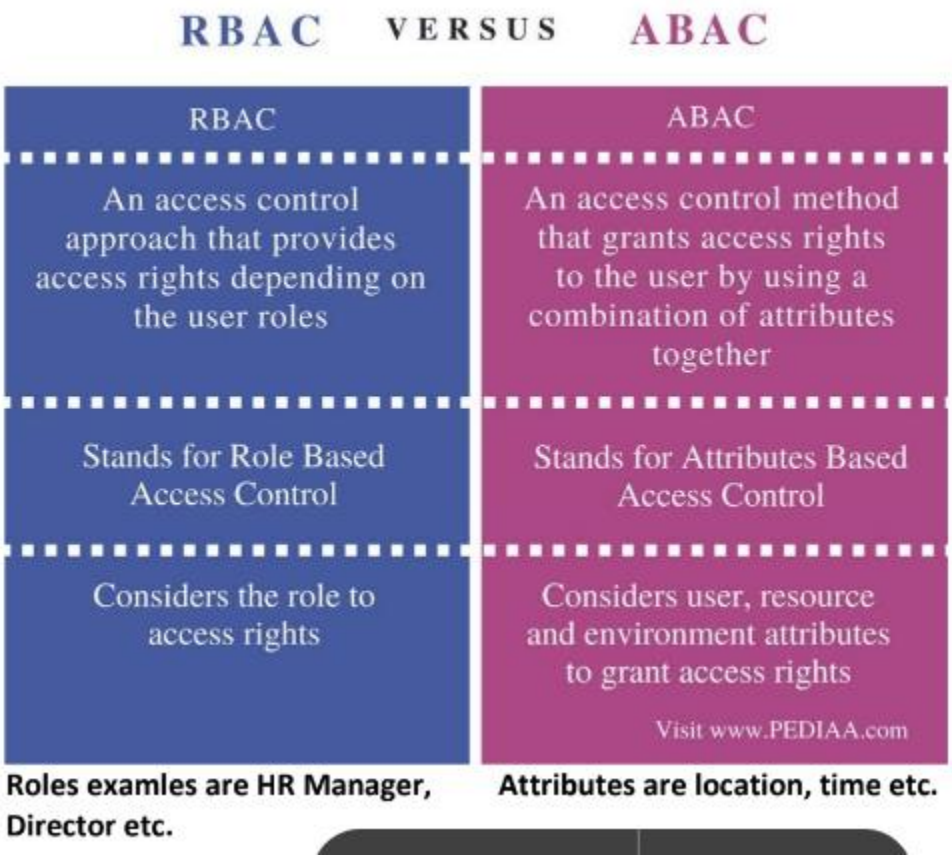


Figure 5.9 **A Database Encryption Scheme**

- **Data owner:** organization that produces data to be made available for controlled release
- **User:** human entity that presents queries to the system
- **Client:** frontend that transforms user queries into queries on the encrypted data stored on the server


One application scenario: How to use certificate to authenticate identify of client by the server and server by the client.
Certificate Template
server.cer
client.cer
Certificate Authority (CA)
3
Verifies certificate (server.cer)
5
Verifies certificate (client.cer)
1 Requests protected resource
2 Presents certificate (server.cer)
4 Presents certificate (client.cer)
6 Accesses protected resource
Client
Server
Client Keystore
client.cer
Server Keystore
server.cer


RBAC VERSUS ABAC
RBAC
ABAC
An access control approach that provides access rights depending on the user roles
An access control method that grants access rights to the user by using a combination of attributes together
Stands for Role Based Access Control
Stands for Attributes Based Access Control
Considers the role to access rights
Considers user, resource and environment attributes to grant access rights
Visit www.PEDIAA.com
Roles examles are HR Manager, Director etc.
Attributes are location, time etc.

A Model for Digital User Authentication

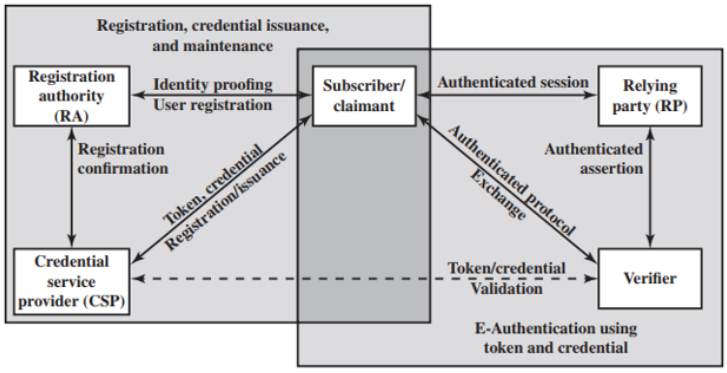
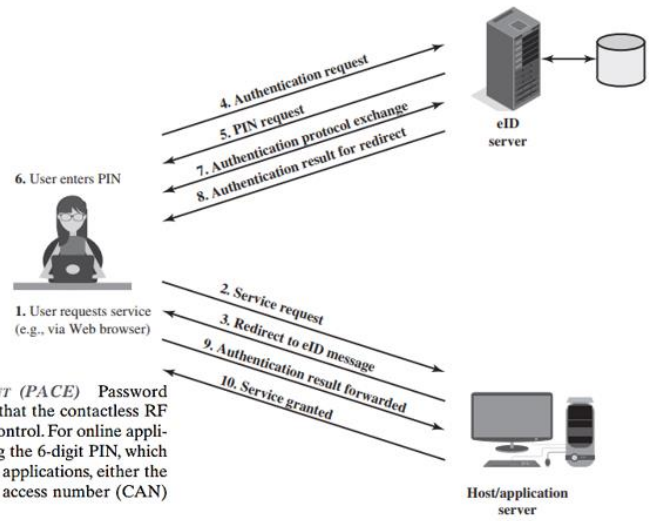


Figure 3.1 **The NIST SP 800-63-3 E-Authentication Architectural Model**

- **User Registration**: Applicant registers with the system via a trusted RA.
- **Credential Issuance**: CSP issues an electronic credential linking identity to a token.
- **Claimant & Verifier**:
  - **Claimant**: User to be authenticated.
  - **Verifier**: Confirms claimant's identity.
- **Assertions & RP**: Verifier sends identity assertion to RP. RP makes access or authorization decisions.

Electronic Identity Cards



Password Authenticated Connection Establishment (PACE) Password Authenticated Connection Establishment (PACE) ensures that the contactless RF chip in the eID card cannot be read without explicit access control. For online applications, access to the card is established by the user entering the 6-digit PIN, which should only be known to the holder of the card. For offline applications, either the MRZ printed on the back of the card or the six-digit card access number (CAN) printed on the front is used.

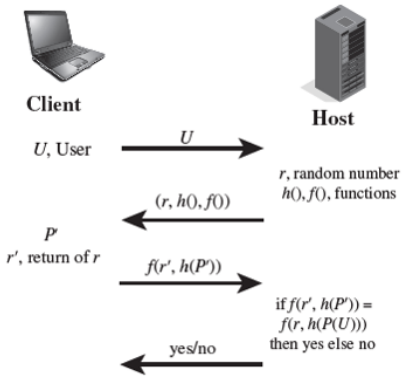
Figure 3.7 **User Authentication with eID**

Protocol for a password verification

- Similar approach for token and biometric verification **(Figure 3.13b,c and d self reading from book)**

In this example, a user first transmits his or her identity to the remote host. The host generates a random number r, often called a **nonce**, and returns this nonce to the user. In addition, the host specifies two functions, $h()$ and $f()$, to be used in the response. This transmission from host to user is the challenge. The user's response is the quantity $f(r', h(P'))$, where $r' = r$ and P' is the user's password. The function h is a hash function, so the response consists of the hash function of the user's password combined with the random number using the function f.

The host stores the hash function of each registered user's password, depicted as $h(P(U))$ for user U. When the response arrives, the host compares the incoming $f(r', h(P'))$ to the calculated $f(r, h(P(U)))$. If the quantities match, the user is authenticated.



(a) Protocol for a password

Figure 3.13 **Basic Challenge-Response Protocols for Remote User Authentication**
Source: Based on [OGOR03].

- An access control mechanism mediates between
  - a user (or a process executing on behalf of a user) and system resources
  - such as applications, operating systems, firewalls, routers, files, and databases

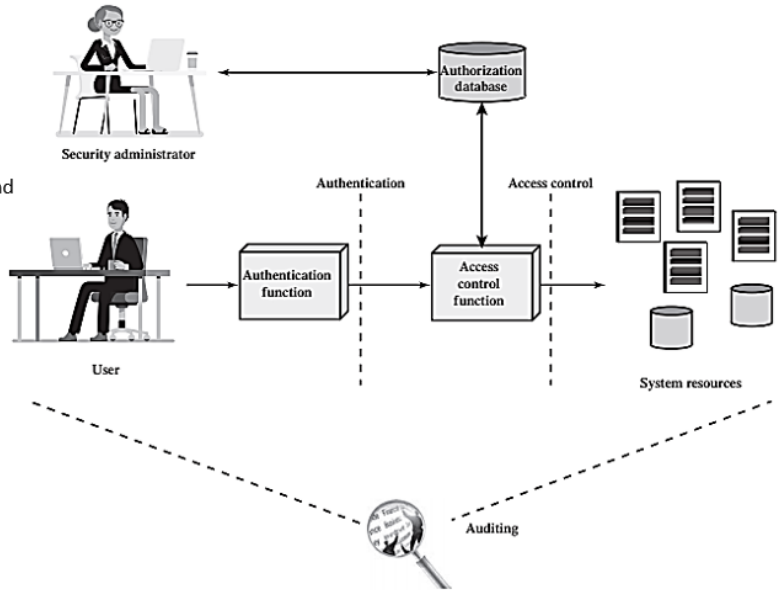


Figure 4.1 **Relationship Among Access Control and Other Security Functions**

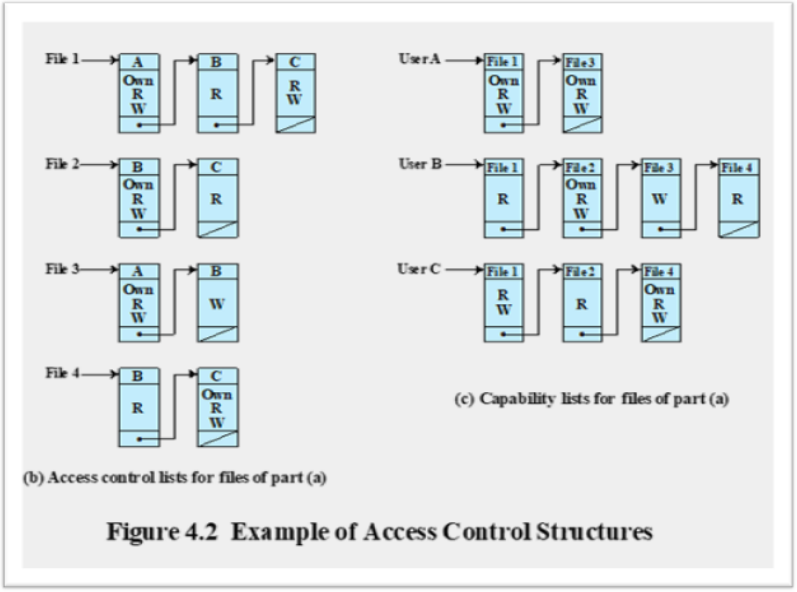


Figure 4.2 Example of Access Control Structures

Differences:

- **ACLs** are object-centric. They are focused on what each object (file) allows different users to do.
- **Capabilities** are subject-centric. They focus on what each user or process is allowed to do with different objects.

OBJECTS

SUBJECTS	File 1	File 2	File 3	File 4
User A	Own Read Write		Own Read Write	
User B	Read	Own Read Write	Write	Read
User C	Read Write	Read		Own Read Write

(a) Access matrix

- ABAC is a logical access control model that is distinguishable because it controls access to objects by evaluating rules against the attributes of entities (subject and object), operations, and the environment relevant to a request.
- ABAC relies upon the evaluation of attributes of the subject, attributes of the object, and a formal relationship or access control rule defining the allowable operations for subject object attribute combinations in a given environment.
- All ABAC solutions contain these basic core capabilities to evaluate attributes and enforce rules or relationships between those attributes.

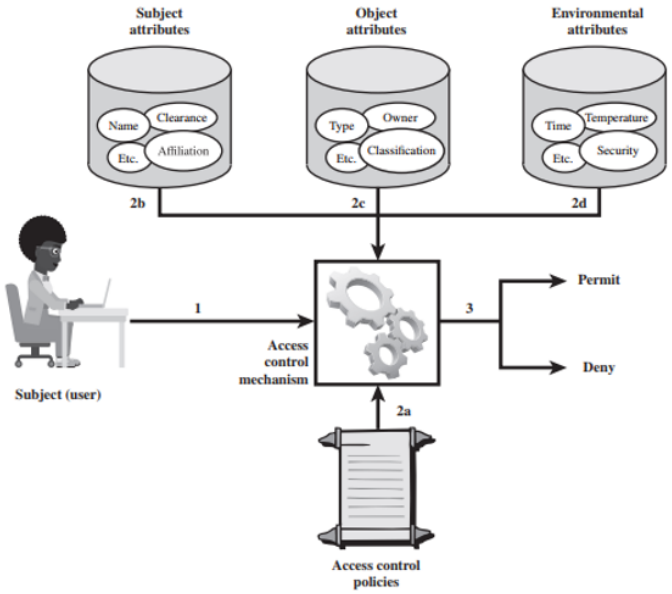


Figure 4.10 **ABAC Scenario**

6.3 PROPAGATION—INFECTED CONTENT—VIRUSES

A typical virus goes through the following four phases during its lifetime:

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

Types of firewall

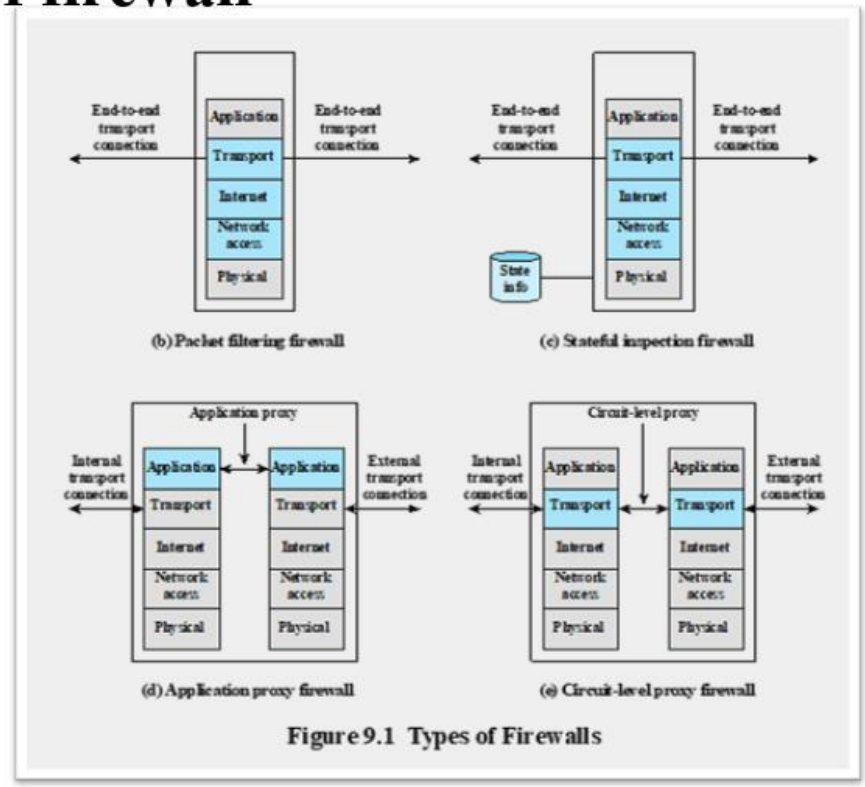


Figure 9.1 Types of Firewalls

time will be wasted analyzing the false alarms.

In general, if the actual numbers of intrusions is low compared to the number of legitimate uses of a system, then the false alarm rate will be high unless the test is extremely discriminating. This is an example of a phenomenon known as the *baserate fallacy*.

For example, If among 10 malicious events, 9 will detected as a malicious, which mean we have 1 false negative.

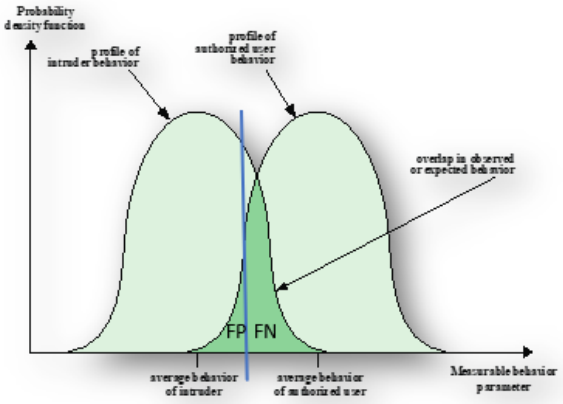


Figure 8.1 Profiles of Behavior of Intruders and Authorized Users

- A **false positive (FP)** *state* ***is when the IDS identifies an activity as an attack but the activity is acceptable behavior.*** A false positive is a false alarm.
- A **false negative (FN)** state is the most serious and dangerous state. ***This is when the IDS identifies an activity as acceptable when the activity is actually an attack.*** **That is, a false negative is when the IDS fails to catch an attack.**

Figure 8.3 shows details of the agent module architecture.

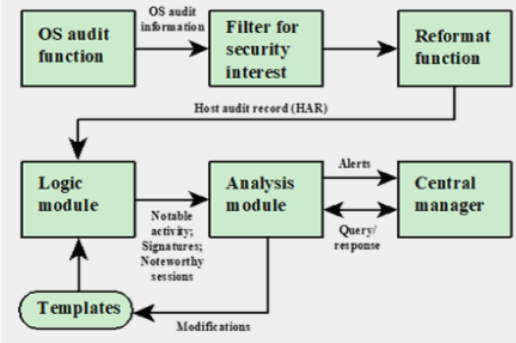


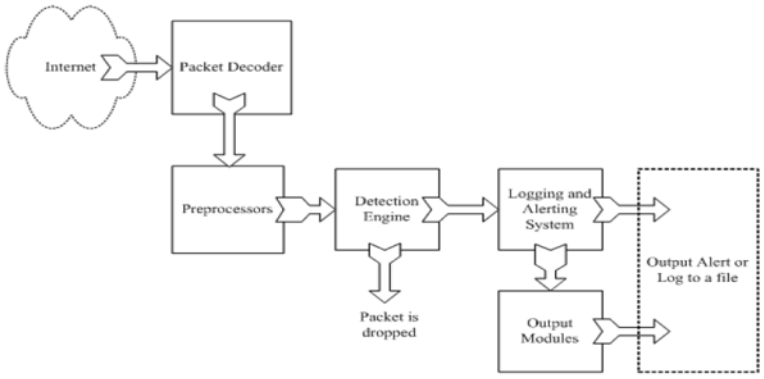
Figure 8.3 Agent Architecture

Figure 8.3 shows the general approach that is taken.

- The agent captures each audit record produced by the ***native audit collection system.***
- *A **filter*** is applied that retains only those records that are of security interest.
- These records are then ***reformatted*** into a standardized format referred to as the host audit record (HAR).
- Next, a ***template-driven*** *logic module* analyzes the records for suspicious activity.
- ***At the lowest level,*** the agent scans for notable events that are of interest independent of any past events.

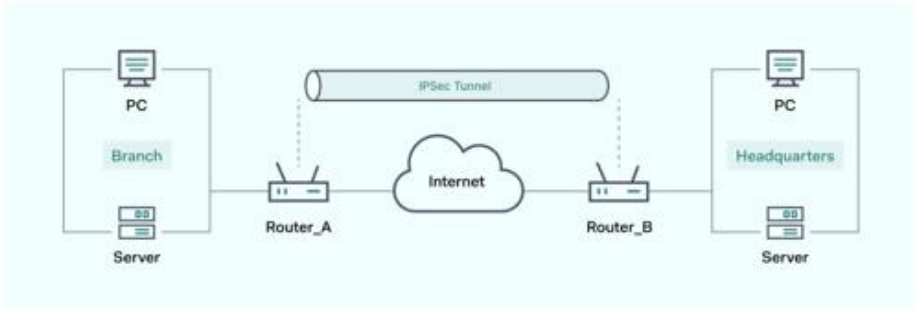
Example: Snort

http://www.snort.org/



From: Rafeeq Ur Rehman, *Intrusion Detection Systems with Snort: Advanced IDS Techniques with Snort, Apache, MySQL, PHP, and ACID.*

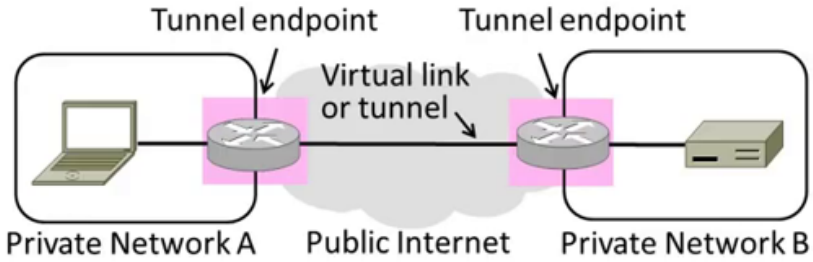
IP Security Protocol (IPSec)

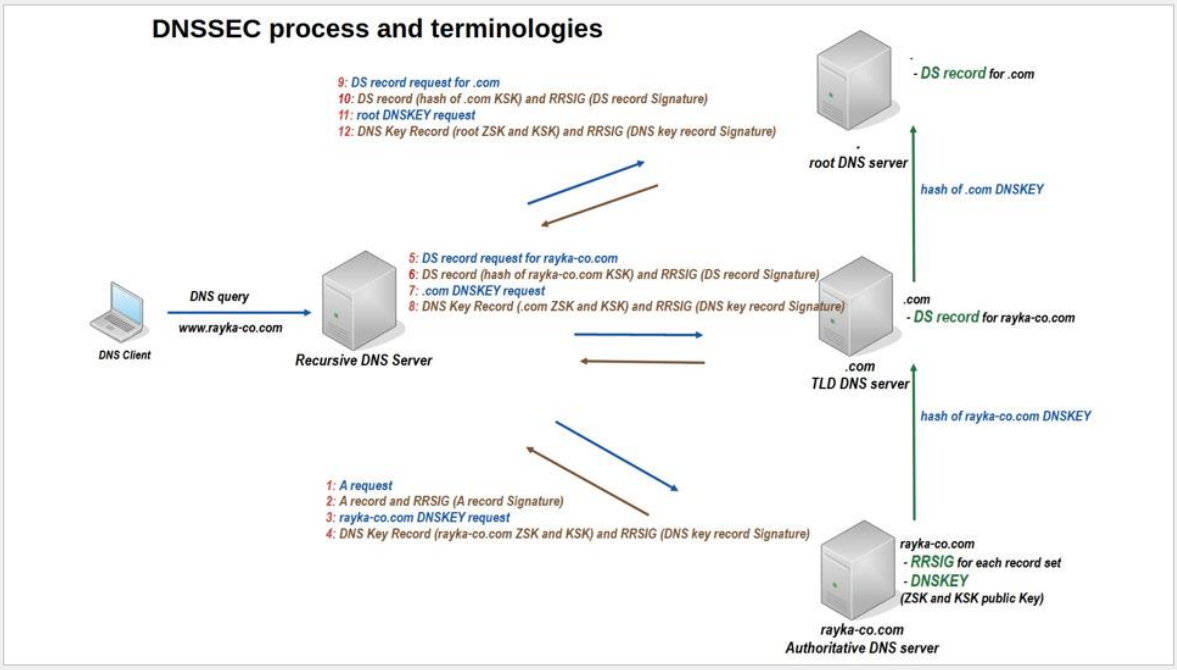


IPsec framework

Tunneling

- How can we build a virtual link? With tunneling!
  - Hosts in private network send to each other normally
  - To cross virtual link (tunnel), endpoints encapsulate packet




DNSSEC process and terminologies
9: DS record request for .com
10: DS record (hash of .com KSK) and RRSIG (DS record Signature)
11: root DNSKEY request
12: DNS Key Record (root ZSK and KSK) and RRSIG (DNS key record Signature)
.
- DS record for .com
.
root DNS server
hash of .com DNSKEY
5: DS record request for rayka-co.com
6: DS record (hash of rayka-co.com KSK) and RRSIG (DS record Signature)
7: .com DNSKEY request
8: DNS Key Record (.com ZSK and KSK) and RRSIG (DNS key record Signature)
DNS query
www.rayka-co.com
DNS Client
Recursive DNS Server
.com
- DS record for rayka-co.com
.com
TLD DNS server
hash of rayka-co.com DNSKEY
1: A request
2: A record and RRSIG (A record Signature)
3: rayka-co.com DNSKEY request
4: DNS Key Record (rayka-co.com ZSK and KSK) and RRSIG (DNS key record Signature)
rayka-co.com
- RRSIG for each record set
- DNSKEY
(ZSK and KSK public Key)
rayka-co.com
Authoritative DNS server

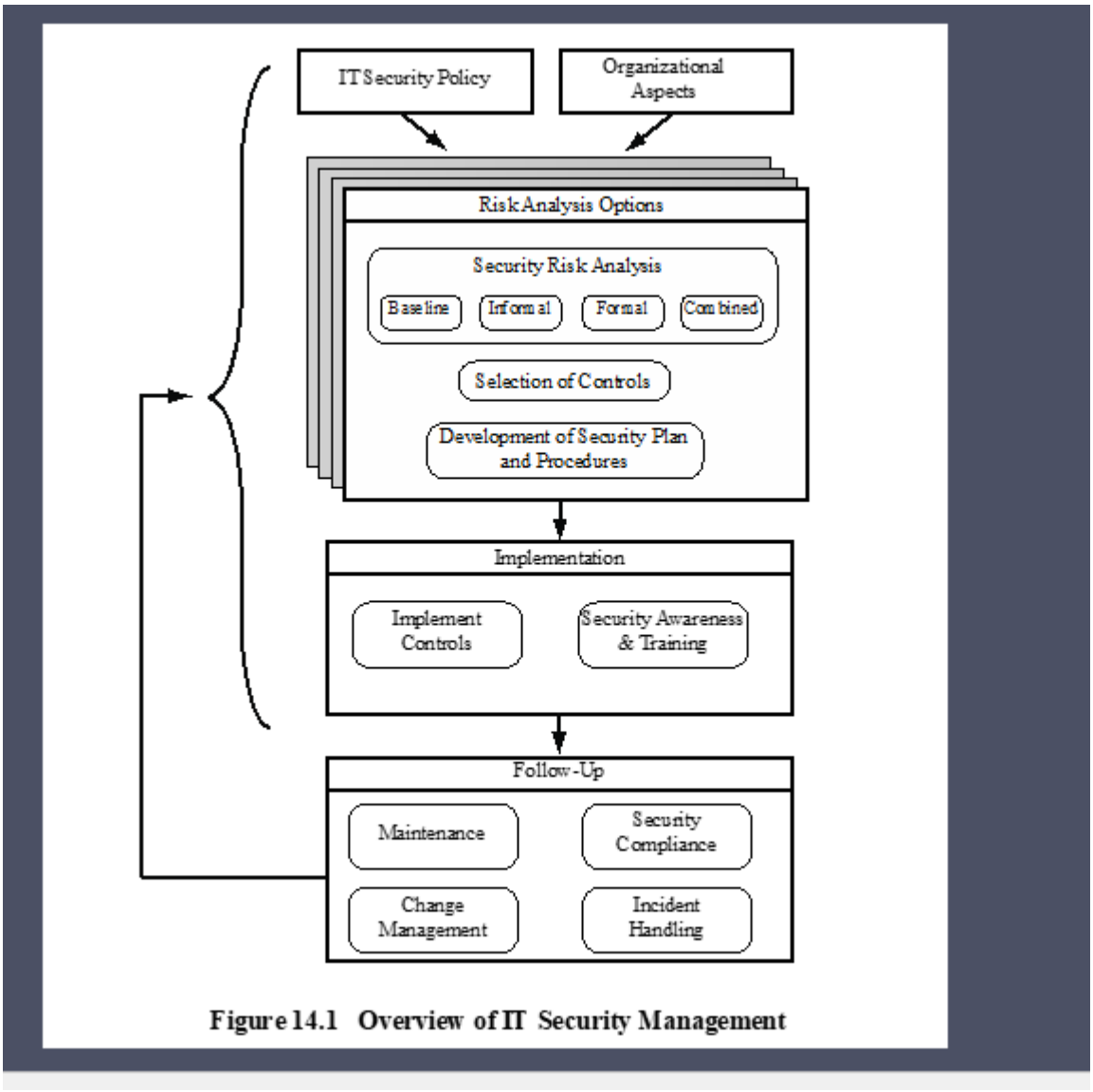


Figure 14.1 Overview of IT Security Management

c) Explain signature based and anomaly-based detection. How Anomaly based detection is better than signature-based detection? **(3)**

Solution:

Signature Based Detection:

- Uses a set of known malicious data patterns or attack rules that are compared with current behavior
- Also known as misuse detection
- Can only identify known attacks for which it has patterns or rules

Anomaly Based Detection:

- Involves the collection of data relating to the behavior of legitimate users over a period of time.
- Current observed behavior is analyzed to determine whether this behavior is that of a legitimate user or that of an intruder.

Why better: Anomaly based detects novel attacks

b) Suppose you bought a new smartphone and are enthusiastic about game applications available for it. When you download and start to install one game application, you are asked to approve the access permissions granted to it. It wants permission to "Send SMS messages" and to "Access your address-book". What threat might the application pose to your smartphone, should you grant these permissions and proceed to install it? **(3)**

Solution:

If when you download and start to install some game app, you are asked to approve the access permissions "Send SMS messages" and to "Access your address-book", you should indeed be suspicious that a game wants these types of permissions, as it would not seem needed just for a game. Rather it could be malware that wants to collect details of all your contacts, and either return them to the attacker via SMS, or allow the code to send SMS messages to your contacts, perhaps enticing them to also download and install this malware. Such code is a trojan horse, since it contains covert functions as well as the advertised functionality.

Case Study

Scenario:
SecureTech Corp is a technology company with a large workforce. The company uses a central system to manage access to its internal applications, such as HR, finance, and project management tools. To ensure security and streamline access, SecureTech Corp is considering two different access control models: Role-Based Access Control (RBAC) and Attribute-Based Access Control (ABAC).

Company Setup:

1. Employees are divided into different roles, such as **Manager**, **Developer**, **HR Staff**, and **Finance Staff**.

2. Access needs differ based on not just roles but also specific attributes like **location, department, project assignment**, and **time of access**.

Access Requirements:

- **HR Staff** should access employee records only within their own department.
- **Finance Staff** can access salary details but only when connected from the office network.
- **Developers** working on a specific project can access that project's documentation but not other unrelated projects.
- **Managers** need access to performance reports for their direct reports, but only during business hours.

Problem:
The IT team needs to decide between using RBAC or ABAC to implement access controls.

ABAC (Attribute-Based Access Control) is a better fit than RBAC. Here's why:

- RBAC alone would require creating multiple roles to accommodate different requirements, such as network location and project assignment, which could quickly become complex and hard to manage.
- ABAC allows more flexibility by using attributes (like department, project, location, and time) to create rules, which can more easily handle specific conditions without creating excessive roles.

2. Implementation Using ABAC

Here's how the access rules would look for each group based on the attributes required:

Role	Attributes	Access Rule
HR Staff	`department`, `location`	Can access employee records **only within their department.**
Finance Staff	`network location`	Can access salary details **only when connected from the office network.**
Developer	`project`	Can access project documentation **only for projects they're assigned to.**
Manager	`direct reports`, `time`	Can access performance reports **for direct reports during business hours only.**

Scenario Summary

An online entertainment store streams movies to users. Access control policies are based on the **user's age** and the **movie's content rating**:

- **R-rated movies**: Only accessible to users aged **17 and older**.
- **PG-13 movies**: Accessible to users aged **13 and older**.
- **G-rated movies**: Accessible to **all users**.

Solution Using ABAC (Attribute-Based Access Control)

In ABAC, we'll still use **attributes** to define the rules based on the **user's age** and the **movie's content rating**. This method will dynamically evaluate access based on these attributes, without pre-defined roles.

Attribute	Condition	Access Rule
User Age	If **age >= 17**	Can access **R**, **PG-13**, and **G** movies.
User Age	If **age >= 13** and **< 17**	Can access **PG-13** and **G** movies only.
User Age	If **age < 13**	Can access **G** movies only.
Movie Rating	**R**, **PG-13**, or **G**	Defines movie content rating restriction.

Solution Using RBAC

In an RBAC model, we would define roles based on age restrictions to manage access:

Role	Age Restriction	Permissions
General	No restriction	Can access **G-rated movies** only.
Teen	13 and older	Can access **G** and **PG-13** movies.
Adult	17 and older	Can access **G**, **PG-13**, and **R** movies.

Explanation:

- **General** role: Users in this role can only watch **G-rated** movies, as it has no age restriction.
- **Teen** role: Users aged **13 and older** can watch **G** and **PG-13** movies.
- **Adult** role: Users aged **17 and older** can w[illegible]h all movies, including **R-rated**.

A: A risk is defined as the result of a system being secure but not secured sufficiently, thereby increasing the likelihood of a threat. A vulnerability is a weakness or breach in your network or equipment (e.g. modems, routers, access points). A threat is the actual means of causing an incident; for instance, a virus attack is deemed a threat.

Rootkit has no role in helping a hacker (bad guy) install software.
DISAGREE. Rootkits hide malware; they don't directly help with installing software.

A common approach for creating polymorphic viruses uses encryption technology and a mutation engine.
AGREE. Encryption makes virus detection harder, facilitating polymorphic viruses.

Message Authentication Code and digital signature both are verified with the same type of key.
DISAGREE. MAC uses a shared key, while digital signatures use asymmetric key pairs.

Botnets are networks of compromised computers that are controlled remotely by one or more cyber criminals. Cyber criminals infect a victim's computer with bots using phishing attacks and browser vulnerabilities.
AGREE. Botnets target unprotected computers to install malicious software via phishing and vulnerabilities.

Access control ensures that authorized users who have access to sensitive data will not misuse it.
DISAGREE. Access control only limits access, not user actions once inside.