Experiment No 11

Azlaan Shaikh
211248
Cryptography and Security Lab
Computer Engg.
M.H Saboo Siddik College of Engineering

***Aim*— Create a Mind map on any topic related to Syllabus**

I. INTRODUCTION

The OSI Security Architecture serves as a structured defense mechanism for safeguarding our digital information as it travels through the vast realm of the internet. Think of it like a team of guardians, with each member handling a specific layer of security – from the physical aspects, like cables, to the more user-centric applications.

Cryptography, the secret code language, is the hero of the story here. It's the method of encoding our messages so that only the intended recipient can understand. The OSI Security Architecture employs this cryptographic technique in each layer, ensuring that our data remains secure and protected from unauthorized access.

What makes this system particularly powerful is its ability to address different types of security challenges at each level. It's akin to having a well-thought-out plan for every possible scenario, making it a comprehensive defense strategy.

In essence, the OSI Security Architecture acts as a guide, providing a structured approach to maintaining the security of our online interactions. It adapts to the ever-changing landscape, making it a reliable companion not just during college but throughout our digital journeys. It's the guardian of our online world, ensuring the safety and integrity of our data.

II. THE OSI SECURITY ARCHITECTURE

To effectively evaluate an organization's security needs and make informed decisions about security products and policies, security managers require a systematic method for defining security requirements and understanding potential approaches. This challenge becomes even more complex in the context of decentralized data processing environments and the use of local and wide area networks.

One established framework for this purpose is ITU-T Recommendation X.800, which outlines the Security Architecture for OSI. This systematic approach proves valuable for managers in structuring the provision of security measures. Moreover, being an international standard, the OSI security architecture has influenced computer and communications vendors to incorporate security features in their products and services aligned with this well-defined structure of services and mechanisms.From our standpoint, the OSI security architecture offers a beneficial, though abstract, overview of many concepts relevant to our discussion. It specifically focuses on security attacks, mechanisms, and services. In essence, it provides a structured understanding of the key concepts we explore in this context.

- **Security attack:** Any action that compromises the security of information owned by an organization.
- **Security attack**: Any action that compromises the security of information owned by an organization.
- **Security service:** A processing or communication service that enhances the security of the data processing systems and the information transfers of an organization. The services are intended to counter security attacks, and they make use of one or more security mechanisms to provide the service.

In the world of security, people often use "threat" and "attack" like they mean the same thing. According to RFC 4949, a threat is like a potential danger that could mess up security. It happens when there's something, like a situation or an action, that could breach security and cause harm.

Now, an attack is more intentional. It's like a purposeful strike on a system's security that comes from a smart threat. This means someone is deliberately trying to break through security using specific methods or tricks. The goal is to get past security services and go against the system's security rules.

III. SECURITY ATTACKS

In the world of security attacks, there are two main types: passive attacks and active attacks (check out Figure 1.1). Passive attacks are like snoops – they try to listen in or peek at the system without messing with its stuff. On the flip side, active attacks are troublemakers – they want to mess with the system's resources or how they work.

A. Passive Attacks

These attacks (shown in Figure 1) are all about eavesdropping or spying on transmissions. The bad guys want to grab information that's moving through the system. There are two types of passive attacks: one where they spill the beans on message contents and another where they analyze the traffic. Both are sneaky ways to understand what's going on without causing a ruckus in the system's day-to-day operations.

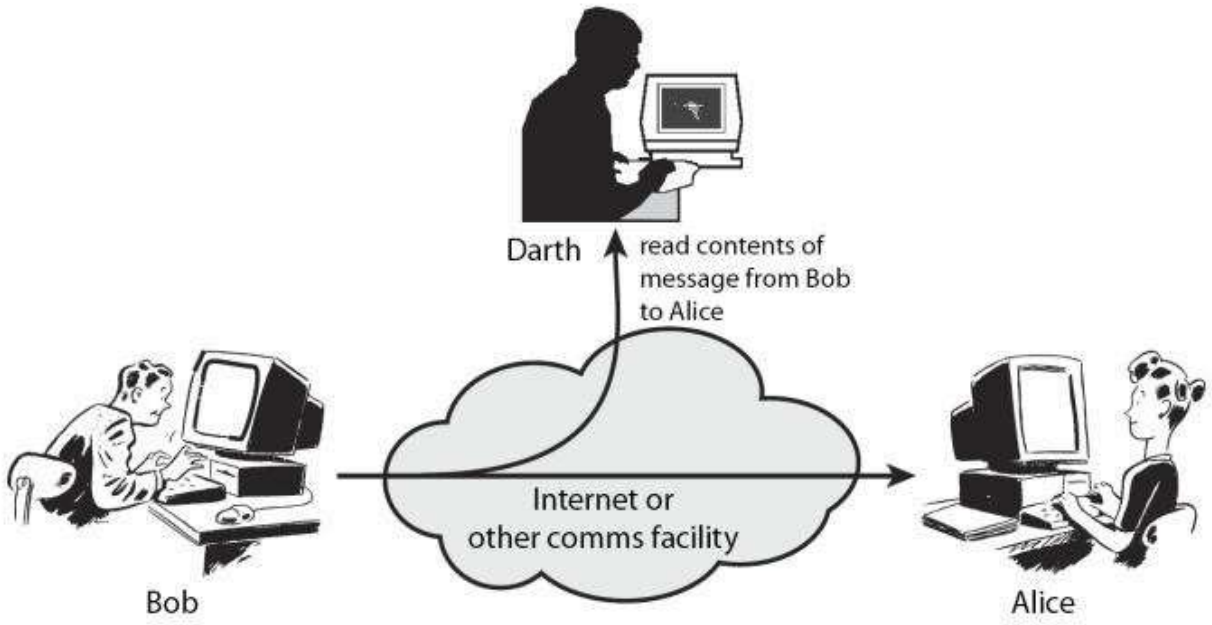


Figure 1(Passive Attack)

The **release of message** contents is easily understood. A telephone conversation, an electronic mail message, and a transferred file may contain sensitive or confidential information. We would like to prevent an opponent from learning the contents of these transmissions.

A second type of passive attack, **traffic analysis**, is subtler. Suppose that we had a way of masking the contents of messages or other information traffic so that opponents, even if they captured the message, could not extract the information from the message. The common technique for masking contents is encryption. If we had encryption protection in place, an opponent might still be able to observe the pattern of these messages. The opponent could determine the location and identity of communicating hosts and could observe the frequency and length of messages being exchanged. This information might be useful in guessing the nature of the communication that was taking place.

Passive attacks are very difficult to detect, because they do not involve any alteration of the data. Typically, the message traffic is sent and received in an apparently normal fashion, and neither the sender nor receiver is aware that a third party has read the messages or observed the traffic pattern. However, it is feasible to prevent the success of these attacks, usually by means of encryption. Thus, the emphasis in dealing with passive attacks is on prevention rather than detection.

B. Active Attack

Active attacks (see Figure 2) are a bit more hands-on. They involve messing with the data flow, like changing it or making up a fake one. There are four main types: masquerade, replay, message modification, and denial of service.

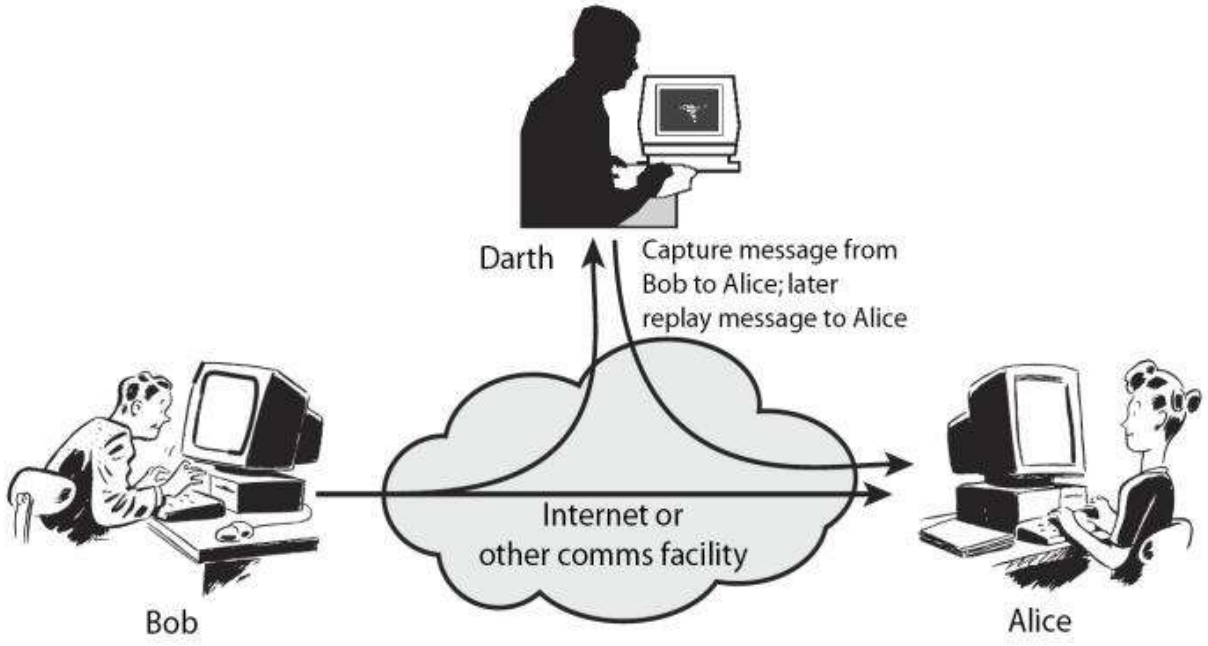


Figure 2(Active Attack)

- **Masquerade:** Imagine someone pretending to be someone else (like a character in a play). They might also use tricks from other active attacks. For instance, they can copy and replay authentication sequences to act like someone with more privileges.
- **Replay:** This is like playing back a recorded message. Someone grabs a piece of data, like a message, and plays it again to cause trouble.

- **Message Modification:** It's like changing a sentence in a message or delaying or mixing up messages to cause problems. For example, a message saying "Let John Smith read confidential files" could be changed to "Let Fred Brown read confidential files."
- **Denial of Service:** This one's about stopping or slowing down normal communication. It could be targeted, like blocking messages to a specific place, or broader, like flooding a network with too many messages to make it slow down.

Active attacks are a bit trickier than passive ones. While it's hard to prevent them completely because there are so many ways they can happen, we focus on spotting them and fixing any mess they cause. If we can detect them early, it might even stop them from happening in the first place.

IV. SECURITY SERVICES

X.800 says a security service is something provided by a layer in communication systems. Its job is to make sure the systems or data transfers are safe. For a clearer picture, RFC 4949 defines it as a service offered by a system to protect its resources in a specific way. These services follow security policies and use security mechanisms to get the job done. It's like having a guard service that protects the important stuff in a system.

X.800 divides these services into five categories and fourteen specific services.

A. Authentication

The authentication service is like a trust checker. If it's a single message, like a warning, its job is to make sure the recipient knows it's genuinely from the claimed source. For ongoing interactions, like when a terminal connects to a host, it has two jobs. First, it checks that both entities are who they say they are when the connection starts. Second, it ensures the connection is secure, so no outsider can pretend to be one of the legit parties and do sneaky stuff without permission. It's basically making sure everyone is who they claim to be and nothing fishy is happening during the interaction.

Two specific authentication services are defined in X.800:

- **Peer entity authentication:** Provides for the corroboration of the identity of a peer entity in an association. Two entities are considered peers if they implement the same protocol in different systems; e.g., two TCP mod-ules in two communicating systems. Peer entity authentication is provided for use at the establishment of, or at times during the data transfer phase of, a connection. It attempts to provide confidence that an entity is not performing either a masquerade or an unauthorized replay of a previous connection.
- **Data origin authentication:** Provides for the corroboration of the source of a data unit. It does not provide protection against the duplication or modification of data units. This type of service supports applications like electronic mail, where there are no prior interactions between the communicating entities.

B. Access Control

In network security, access control is like setting up rules to control who can get into host systems and apps through communication links. To make this work, every entity trying to get in needs to be recognized, or authenticated. This way, we can customize access rights based on who they are. It's all about making sure the right people get in and have the right permissions.

AUTHENTICATION	DATA INTEGRITY
The assurance that the communicating entity is the one that it claims to be.	The assurance that data received are exactly as sent by an authorized entity (i.e., contain no modification, insertion, deletion, or replay).
Peer Entity Authentication Used in association with a logical connection to provide confidence in the identity of the entities connected.	**Connection Integrity with Recovery** Provides for the integrity of all user data on a connection and detects any modification, insertion, deletion, or replay of any data within an entire data sequence, with recovery attempted.
Data-Origin Authentication In a connectionless transfer, provides assurance that the source of received data is as claimed.	**Connection Integrity without Recovery** As above, but provides only detection without recovery.
ACCESS CONTROL	**Selective-Field Connection Integrity** Provides for the integrity of selected fields within the user data of a data block transferred over a connection and takes the form of determination of whether the selected fields have been modified, inserted, deleted, or replayed.
The prevention of unauthorized use of a resource (i.e., this service controls who can have access to a resource, under what conditions access can occur, and what those accessing the resource are allowed to do).	**Connectionless Integrity** Provides for the integrity of a single connectionless data block and may take the form of detection of data modification. Additionally, a limited form of replay detection may be provided.
DATA CONFIDENTIALITY	**Selective-Field Connectionless Integrity** Provides for the integrity of selected fields within a single connectionless data block; takes the form of determination of whether the selected fields have been modified.
The protection of data from unauthorized disclosure.	**NONREPUDIATION**
Connection Confidentiality The protection of all user data on a connection.	Provides protection against denial by one of the entities involved in a communication of having participated in all or part of the communication.
Connectionless Confidentiality The protection of all user data in a single data block	**Nonrepudiation, Origin** Proof that the message was sent by the specified party.
Selective-Field Confidentiality The confidentiality of selected fields within the user data on a connection or in a single data block.	**Nonrepudiation, Destination** Proof that the message was received by the specified party.
Traffic-Flow Confidentiality The protection of the information that might be derived from observation of traffic flows.	

Table 1

C. Data Confidentiality

Confidentiality is like a shield for data when it's being sent, especially from snoopy attacks. It comes in different levels. The most comprehensive one protects all data sent between two users over time. For instance, when there's a connection between two systems, this level stops any user data from leaking out. There are also more specific forms of this service, like safeguarding a single message or certain parts of a message. However, these detailed protections are not as practical as the broader one and might even be trickier and costlier to put in place. It's essentially about keeping data safe from prying eyes during its journey.

D. Data Integrity

Just like with keeping things confidential, we also want to make sure our data stays intact. This can apply to a series of messages, a single message, or just certain parts of a message. But, the most practical way is to protect the entire flow of data.

For a connected flow of messages, we have an integrity service making sure messages arrive as they were sent – no copying, adding, changing, mixing up, or playing them again. It even covers the destruction of data. So, it's like a guardian against messing with the message flow or trying to stop it altogether.

On the other hand, a service dealing with individual messages (without considering the bigger picture) mostly protects against changes to the messages.

We can split these services into those with and without recovery. Since integrity services are about detecting active attacks, we focus on spotting issues rather than stopping them upfront. If an integrity problem is detected, the service usually reports it, and then we need some other software or human help to fix it. But there are also ways to automatically recover from data integrity issues, which is generally the more appealing option. It's all about making sure our data stays true and intact during its journey.

E. Nonrepudiation

Nonrepudiation prevents either sender or receiver from denying a transmitted message. Thus, when a message is sent, the receiver can prove that the alleged sender in fact sent the message. Similarly, when a message is received, the sender can prove that the alleged receiver in fact received the message.

V. SECURITY MECHANISMS

In Table 1.3 of X.800, there are different security mechanisms. Some are specific to certain protocol layers like TCP or application-layer protocols, while others aren't tied to any particular layer or service. We'll delve into the details of these mechanisms later.

One interesting point from X.800 is how it defines encipherment. It makes a distinction between two types: reversible encipherment (where you can undo the encryption) and irreversible encipherment (where you can't reverse the process). It's like the difference between locking something in a box that you can open later versus using a shredder where you can't get the original back.

SPECIFIC SECURITY MECHANISMS	PERVASIVE SECURITY MECHANISMS
May be incorporated into the appropriate protocol layer in order to provide some of the OSI security services.	Mechanisms that are not specific to any particular OSI security service or protocol layer.
Encipherment The use of mathematical algorithms to transform data into a form that is not readily intelligible. The transformation and subsequent recovery of the data depend on an algorithm and zero or more encryption keys.	**Trusted Functionality** That which is perceived to be correct with respect to some criteria (e.g., as established by a security policy).
Digital Signature Data appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery (e.g., by the recipient).	**Security Label** The marking bound to a resource (which may be a data unit) that names or designates the security attributes of that resource.
Access Control A variety of mechanisms that enforce access rights to resources.	**Event Detection** Detection of security-relevant events.
Data Integrity A variety of mechanisms used to assure the integrity of a data unit or stream of data units.	**Security Audit Trail** Data collected and potentially used to facilitate a security audit, which is an independent review and examination of system records and activities.
	Security Recovery Deals with requests from mechanisms, such as event handling and management functions, and takes recovery actions.

CONTINUED

Authentication Exchange A mechanism intended to ensure the identity of an entity by means of information exchange. **Traffic Padding** The insertion of bits into gaps in a data stream to frustrate traffic analysis attempts. **Routing Control** Enables selection of particular physically secure routes for certain data and allows routing changes, especially when a breach of security is suspected. **Notarization** The use of a trusted third party to assure certain properties of a data exchange.	

Table 2(Security Mechanism)

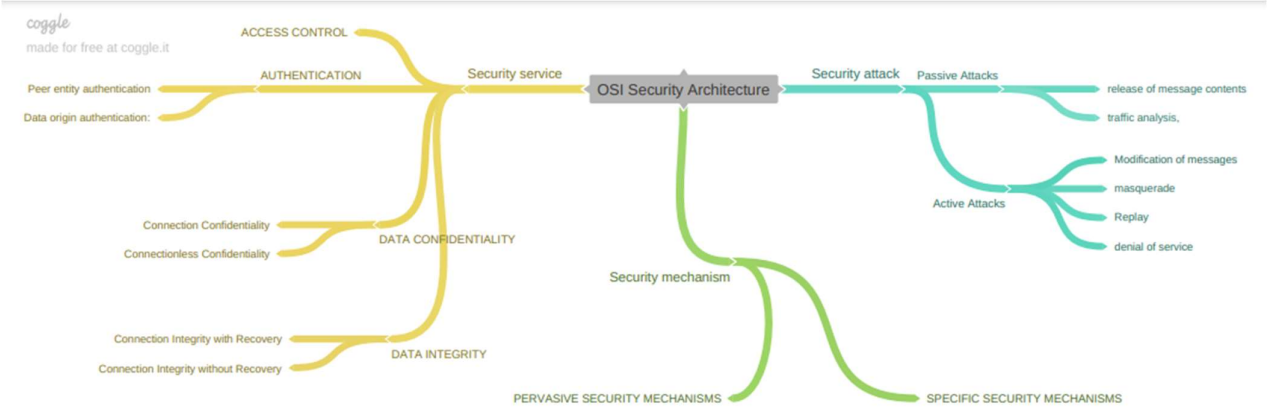
A. Persuasive Security Mechanisms

Persuasive security mechanisms aim to influence users' behavior and choices to enhance overall security. Instead of relying solely on technical barriers, persuasive mechanisms encourage individuals to adopt secure practices willingly. Examples include user awareness campaigns, security training programs, and initiatives that highlight the benefits of following security protocols. The goal is to create a security-conscious culture and promote responsible behaviors among users.

B. Specific Security Mechanisms

Specific security mechanisms are detailed techniques and tools designed to address specific security concerns in a system or network. These mechanisms are implemented to enforce security policies and protect against various types of attacks or vulnerabilities. Examples of specific security mechanisms include firewalls, encryption, intrusion detection systems, access control systems, and antivirus software. Each mechanism serves a particular purpose, contributing to the overall security posture by addressing specific aspects of security, such as confidentiality, integrity, and availability.

VI. MIND MAP



REFERENCES

[1] Stallings, William. Cryptography and Network Security: Principles and Practice. 7th ed. Pearson Education Limited, 2017.J.

[2] Security Services by Brainkart (Online: https://www.brainkart.com/article/Security-Services_8382/)

[3] Security Mechanism by Brainkart (Online: https://www.brainkart.com/article/Security-Mechanisms_8383/)

[4] Cryptography and Network Security Principles by GeekForGeek