UNIT – V

E-MAIL SECURITY, IP SECURITY AND CASE STUDIES

1. E-Mail Security

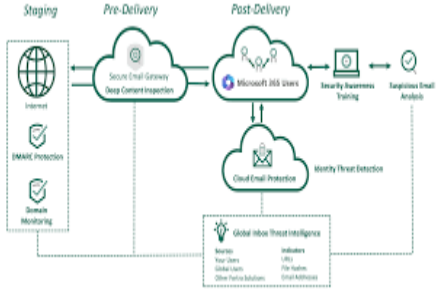
Need for Email Security

- Protect confidential information
- Prevent tampering
- Verify sender identity

Email Security Services

1. Confidentiality
2. Integrity
3. Authentication
4. Non-repudiation

Email Security Architecture



2. Pretty Good Privacy (PGP)

Pretty Good Privacy (PGP) is a widely used email security system developed by Phil Zimmermann in 1991. It provides secure communication by combining encryption, digital signatures, hashing, and compressiontechniques.PGP is primarily used to protect email messages and files from unauthorized access and modification.

Objectives of PGP

- Secure email communication
- Protect confidentiality of messages
- Verify sender identity
- Ensure message integrity

Services Provided by PGP

1. Confidentiality

Confidentiality ensures that only the intended recipient can read the message.

PGP Provides Confidentiality

- PGP encrypts the message using a **symmetric session key**.
- The session key is then encrypted using the recipient's **public key**.
- Only the recipient can decrypt the session key using their **private key**.

Process

Sender
|
Encrypt Message
|
Session Key
|

Encrypt Session Key
using Receiver's Public Key
|
Send Encrypted Message

Benefits

- Prevents unauthorized access
- Protects sensitive information
- Ensures privacy

Example

Sending confidential banking information through email.

2. Authentication

Authentication verifies the identity of the sender and confirms that the message actually came from the claimed sender.

PGP Provides Authentication

- A hash value of the message is generated.
- The hash is encrypted using the sender's private key.
- This encrypted hash becomes the **digital signature**.
- The receiver verifies the signature using the sender's public key.

Process

Message
|
Generate Hash
|
Encrypt Hash
with Sender's Private Key
|
Digital Signature

Benefits

- Verifies sender identity
- Prevents impersonation
- Provides non-repudiation

Example

A company verifies that an email was sent by its manager and not by an attacker.

3. Compression

Compression reduces the size of the message before encryption.

PGP Provides Compression

- The message is compressed before encryption.
- Compressed data requires less storage and transmission time.
- Compression also increases security by reducing predictable patterns.

Process

Original Message
|
Compression

|
Compressed Message
|
Encryption

Benefits

- Reduces message size
- Faster transmission
- Efficient storage

Example: Large email attachments are compressed before encryption and transmission.

5. **E-mail Compatibility:**E-mail compatibility ensures that encrypted messages can be transmitted through standard email systems.

PGP Provides Compatibility

- PGP converts binary encrypted data into ASCII text using **Radix-64 Encoding** (Base64).
- This allows encrypted messages to travel through email servers that support only text data.

Process

Encrypted Binary Data
|
V
Radix-64 Encoding
|
ASCII Text
|
Email Transmission

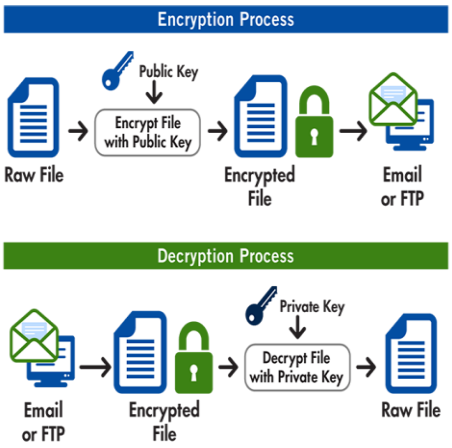
Benefits

- Compatible with email systems
- Supports secure transmission over the Internet
- Easy integration with email applications

Example

Encrypted emails can be sent through Gmail, Outlook, and other mail servers without corruption.

PGP Architecture



PGP Working

Sender Receiver
Message
|
Hash
|
Sign with Private Key
|
Encrypt using Public Key
|
----------Internet---------->
|
Decrypt with Private Key
|
Verify Signature

Advantages

- Strong security
- Digital signatures
- Email encryption

3. S/MIME (Secure MIME)

Introduction

S/MIME provides secure email communication using public key cryptography.

Services

- Authentication
- Encryption
- Digital Signatures

S/MIME Architecture

Email
|
S/MIME

|
Certificate
|
Secure Mail

Features

- Certificate based security
- Widely supported
- Secure communication

4. IP Security (IPSec)

Introduction

IPSec is a suite of protocols that secures IP communications.

IPSec Services

- Authentication
- Confidentiality
- Integrity
- Access Control

IPSec Architecture

Application
|
TCP
|
IPSec
|
IP
|
Network

5. Authentication Header (AH)

Purpose

Provides:

- Authentication
- Integrity
- Anti-replay protection

AH Format

Characteristics

- No encryption
- Integrity protection only

6. Encapsulating Security Payload (ESP)

1. Encryption

Encryption is the process of converting readable data (plaintext) into an unreadable form (ciphertext) using a cryptographic algorithm and key.

Purpose

- Protect data from unauthorized access.
- Ensure secure communication over networks.

Working

Plaintext
|
Encryption Algorithm + Key
|
V
Ciphertext
|
Decryption Key
|
Plaintext

Example: When you send a password through a secure website (HTTPS), the password is encrypted before transmission.

Advantages

- Protects sensitive information.
- Prevents eavesdropping.
- Ensures data privacy.
-

Common Encryption Algorithms

- AES (Advanced Encryption Standard)
- DES (Data Encryption Standard)
- RSA (Rivest-Shamir-Adleman)

2. Authentication: Authentication is the process of verifying the identity of a user, device, or system before granting access.

Purpose

- Ensure only authorized users can access resources.
- Prevent unauthorized access and impersonation.

Working

User
|
Login Request
|
Authentication System
|
Verify Credentials
|
Access Granted/Denied

Authentication Methods

1. Password-based Authentication
2. OTP (One-Time Password)
3. Biometric Authentication
4. Digital Certificates
5. Multi-Factor Authentication (MFA)

Example: Entering a username and password to log in to an email account.

Advantages

- Enhances security.
- Protects user accounts.
- Prevents identity theft.

3. Confidentiality: Confidentiality ensures that information is accessible only to authorized individuals and remains hidden from unauthorized users.

Purpose

- Protect private and sensitive information.
- Prevent information disclosure.

Working

Sender
|
Encrypted Data
|
Network
|
Authorized Receiver

Unauthorized User

|
Cannot Read Data

Techniques Used:

1. Encryption: Encryption is the process of converting readable data (plaintext) into an unreadable form (ciphertext) using an encryption algorithm and a secret key.

Purpose

- Protect sensitive information during storage and transmission.
- Prevent unauthorized users from reading data.

Working

Plaintext
|
Encryption Key
|
V
Ciphertext
|
Decryption Key
|
V
Plaintext

Example: When a user enters banking details on a secure website (HTTPS), the information is encrypted before being transmitted.

Advantages

- Protects confidentiality.
- Prevents eavesdropping.
- Secures online transactions.

2. Access Control: Access Control is a security mechanism that regulates who can access specific resources and what actions they can perform.

Purpose

- Restrict unauthorized access.
- Ensure that users access only permitted resources.

Types of Access Control

1. **Discretionary Access Control (DAC)**
2. **Mandatory Access Control (MAC)**
3. **Role-Based Access Control (RBAC)**

Working

User Request
|
Authentication
|

Authorization Check
|

| Access Granted |
| Access Denied |

Example: In a college management system, faculty members can update student marks, while students can only view their marks.

Advantages

- Protects sensitive resources.
- Reduces insider threats.
- Enhances security management.

3. Password Protection: Password Protection is a security technique that uses passwords to verify user identity and prevent unauthorized access.

Purpose

- Authenticate users.
- Protect accounts and systems.

Characteristics of a Strong Password

- Minimum 8–12 characters
- Combination of uppercase and lowercase letters
- Numbers and special symbols
- Difficult to guess

Working

User
|
Enter Password
|
System Verification
|
| Correct Password|
| Access Granted |

Example: Logging into an email account using a username and password.

Advantages

- Simple and cost-effective.
- Provides basic security.
- Prevents unauthorized access.

Best Practices

- Change passwords regularly.
- Avoid sharing passwords.
- Use Multi-Factor Authentication (MFA).

4. Firewalls: A Firewall is a network security device or software that monitors and controls incoming and outgoing network traffic based on predefined security rules.

Purpose

- Block unauthorized access.
- Prevent cyberattacks and malware infections.
- Protect internal networks.

Types of Firewalls

1. Packet Filtering Firewall
2. Stateful Inspection Firewall
3. Proxy Firewall
4. Next-Generation Firewall (NGFW)

Example: A company firewall blocks suspicious traffic from unknown IP addresses while allowing legitimate employees to access company resources.

Advantages

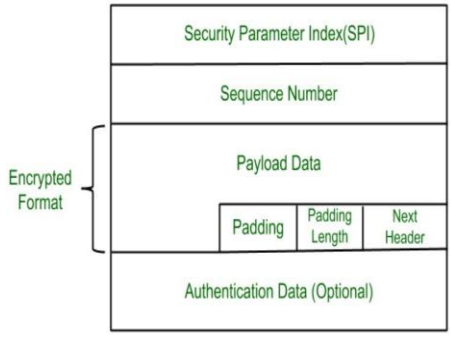
- Protects against unauthorized access.
- Filters malicious traffic.
- Enhances network security.

Example: Medical records can only be viewed by authorized doctors and patients.

Advantages

- Protects privacy.
- Prevents data leakage.
- Maintains trust in information systems.

ESP Format



ESP Working

Original Data
|
Encryption
|
ESP Header Added
|
Transmission

7. Security Associations (SA)

Definition

A Security Association is a relationship between sender and receiver defining security parameters.

Components

- SPI
- Encryption Algorithm
- Authentication Algorithm
- Keys

SA Diagram

Sender <------SA------> Receiver

8. Internet Key Exchange (IKE)

Introduction

IKE automatically exchanges cryptographic keys.

IKE Phases

Phase 1
Establish Secure Channel
|
Phase 2
Generate IPSec Keys

Advantages

- Automatic key management
- Strong authentication
- Secure communication

Secure Multiparty Computation (SMC)

Definition

Allows multiple parties to jointly compute a function without revealing private inputs.

Party A ----\
\
Party B ------> Secure Computation
/
Party C ----/

Applications

- Voting
- Healthcare
- Financial Analysis

Virtual Elections

Process

Voter
|

Authentication
|
Encrypted Vote
|
Vote Server
|
Counting
|
Result

Security Requirements

- Confidentiality
- Integrity
- Authentication
- Non-repudiation

Single Sign-On (SSO)

Definition

Allows users to log in once and access multiple applications.

Architecture

User
|
Login Once
|
SSO Server
| \
App1 App2 App3

Benefits

- Easy access
- Better security
- Reduced password management

Secure Inter-Branch Payment Transactions

Working

Branch A
|

Encrypted Transaction
|
Central Server
|
Branch B

1. Confidentiality

Confidentiality ensures that information is accessible only to authorized users and remains hidden from unauthorized individuals.

Purpose

- Protect sensitive information from disclosure.
- Prevent unauthorized access to data.

Techniques Used

- Encryption (AES, DES, RSA)
- Password Protection
- Access Control Mechanisms
- Firewalls

Example: When a customer enters credit card details on an online shopping website, encryption ensures that only the intended recipient can read the information.

Diagram

Authorized User
|
V
Encrypted Data
|
Network
|
V
Authorized Receiver

Unauthorized User
|
Access Denied

Benefits

- Protects privacy
- Prevents information leakage
- Secures confidential data

2. Integrity: Integrity ensures that data remains accurate, complete, and unaltered during storage

or transmission.

Purpose

- Detect unauthorized modifications.
- Maintain data accuracy and consistency.

Techniques Used

- Hash Functions (SHA-256, MD5)
- Digital Signatures
- Message Authentication Codes (MAC)

Example: During an online banking transaction, integrity mechanisms ensure that the transferred amount is not altered by an attacker.

Diagram

Sender
|
Generate Hash
|
Send Data + Hash
|
Network
|
Receiver
|
Verify Hash
|
Data Integrity Confirmed

Benefits

- Prevents data tampering
- Ensures reliable communication
- Maintains data accuracy

3. Authentication: Authentication is the process of verifying the identity of a user, device, or system before granting access.

Purpose

- Confirm that users are who they claim to be.
- Prevent unauthorized access.

Techniques Used

- Usernames and Passwords
- Biometrics (Fingerprint, Face Recognition)
- One-Time Passwords (OTP)
- Digital Certificates

Example: A user enters a username and password to access an email account. The system verifies the credentials before granting access.

Diagram

User
|
Login Request
|
Authentication Server
|
Verify Credentials
|
Access Granted

Benefits

- Prevents impersonation
- Enhances system security
- Ensures authorized access only

Cross-Site Scripting (XSS)

Definition

A web vulnerability where attackers inject malicious scripts into web pages.

XSS Attack

Attacker
|
Malicious Script
|
Web Page
|
Victim Browser

Prevention

- Input Validation
- Output Encoding
- Secure Cookies
- Content Security Policy

Types of Cross-Site Scripting (XSS)

Cross-Site Scripting (XSS) is a web security vulnerability that allows attackers to inject malicious scripts into web pages viewed by other users. These scripts usually execute in the victim's browser and can steal cookies, session IDs, or sensitive information.

1. Stored XSS (Persistent XSS)

Stored XSS occurs when malicious scripts are permanently stored on the target server, such as in a database, comment section, forum post, or user profile. Whenever a user accesses the affected page, the malicious script is automatically executed.

Works

1. Attacker submits malicious JavaScript code.
2. The application stores the script in its database.
3. Other users visit the webpage.
4. The browser executes the stored script.

Example
An attacker posts the following script in a comment section:
<script>alert('XSS Attack');</script>
Every user viewing the comment will execute the script.
Impact

- Cookie theft
- Session hijacking
- User impersonation
- Data theft

Diagram
Attacker
|
Malicious Script
|
Web Server Database
|
Victim Visits Page
|
Script Executes
Prevention

- Input validation
- Output encoding
- Content Security Policy (CSP)
- Sanitizing user inputs

2. Reflected XSS (Non-Persistent XSS)
Reflected XSS occurs when malicious code is included in a URL or request and immediately reflected back by the web application without proper validation.
Works

1. Attacker creates a malicious URL.
2. Victim clicks the URL.
3. Server reflects the malicious script in the response.
4. Browser executes the script.

Example:https://example.com/search?q=<script>alert('XSS')</script>
If the application displays the search query without filtering, the script runs in the victim's browser.
Impact

- Session hijacking
- Credential theft
- Redirecting users to malicious websites

Diagram
Attacker
|

Malicious URL
|
Victim Clicks Link
|
Web Server Reflects Script
|
Victim Browser Executes Script

Prevention

- Validate user input
- Encode output data
- Use secure frameworks
- Avoid displaying raw user input

3. DOM-Based XS: DOM-Based XSS occurs entirely on the client side when JavaScript modifies the Document Object Model (DOM) using untrusted user input.

Works

1. User input is read by client-side JavaScript.
2. JavaScript inserts the input into the webpage.
3. Malicious script executes in the browser.
4. No malicious code is sent to the server.

Example

```
document.getElementById("output").innerHTML=
location.hash.substring(1);
```

URL:
https://example.com/#<script>alert('XSS')</script>

The browser executes the script because it is inserted directly into the page.

Impact

- Data theft
- Session hijacking
- Browser manipulation

Diagram

Attacker URL
|
Victim Browser
|
JavaScript Reads URL
|
Updates DOM
|
Script Executes

Prevention

- Avoid using innerHTML
- Use textContent instead
- Validate and sanitize client-side inputs
- Implement Content Security Policy (CSP)