# **Malwares**

## 

## Malwares By Type:

1. Virus: A virus can spread from one computer to another, infecting files and programs. It can corrupt or delete data, disrupt system operations, and often requires user action to propagate, such as opening an infected file.
2. Trojan horse (Remote Access Trojan): Disguised as legitimate software, a Trojan horse tricks users into installing it. Once installed, it allows attackers to remotely access and control the infected computer, potentially stealing data or installing additional malware.
3. Spyware: This malware secretly monitors and collects personal information, such as passwords, browsing habits, and credit card numbers, without the user's knowledge. The stolen data is then sent to the attacker for malicious use.
4. Worm: Unlike viruses, worms can self-replicate and spread independently across networks. They consume system resources, slow down computers, and can also carry payloads that cause further damage, such as deleting files.
5. Adware: Adware displays intrusive advertisements on the computer, often in the form of pop-ups. While not always harmful, it can degrade system performance and lead to privacy issues by tracking user behavior to target ads.
6. Scareware: Scareware attempts to frighten users into believing their computer is infected or has another critical issue. It then persuades them to purchase fake or unnecessary software to "resolve" the fabricated problems.
7. Bot: A bot is an automated program that performs tasks over the internet. When controlled by attackers, bots can be used for malicious activities, such as launching denial-of-service attacks, sending spam, or distributing other malware.
8. Ransomware: Ransomware encrypts files on the infected system, rendering them inaccessible to the user. The attacker demands a ransom payment, typically in cryptocurrency, in exchange for the decryption key needed to restore access to the files.
9. Cryptominer: This malware exploits the infected computer's processing power to mine cryptocurrencies like Bitcoin without the user's consent. This activity can significantly slow down the computer and increase electricity usage.

## By Malicious Behavior:

1. Stealing information: This type of malware is designed to secretly collect and transmit personal data, such as passwords, credit card numbers, or sensitive business information, to the attacker.

Examples: Spyware like keyloggers record keystrokes to steal login credentials; Trojan horse can include data-stealing components to capture sensitive information.

1. Creating a vulnerability: Malware that creates vulnerabilities modifies system settings or exploits existing flaws to weaken the system’s defenses. This allows other malware or attackers to easily breach the system in the future.

Examples: Trojan horse can create backdoors in the system, allowing attackers to access it later; Worm can exploit vulnerabilities to spread itself and create further security weaknesses.

1. Denying service: Malware aimed at denying service floods a network or server with excessive traffic, causing it to crash or become inaccessible to legitimate users.

Examples: Bot can be used in botnet attacks to perform Distributed Denial-of-Service (DDoS) attacks; Worm can replicate so extensively that it consumes network resources and causes denial of service.

1. Executing commands from the C&C: Malware controlled by a Command and Control (C&C) center receives instructions from a remote attacker. This allows the attacker to execute arbitrary commands on the infected system, such as stealing data, installing more malware, or launching attacks on other systems.

Examples: Trojan horse (Remote Access Trojan) can receive and execute commands from a C&C server; Bot can be directed by a C&C server to perform specific tasks like sending spam or launching attacks.

1. Deceiving the User: Deceptive malware tricks users into performing actions that compromise their security, such as clicking on malicious links, downloading infected files, or providing personal information.

Examples: Scareware tricks users into thinking their system is infected and persuades them to buy fake antivirus software; Adware often disguises itself as a legitimate tool but bombards users with ads and may lead them to malicious sites.

1. Annoying the User: This malware aims to disrupt the user experience by displaying unwanted ads, generating pop-ups, or slowing down system performance.

Examples: Adware displays intrusive ads and pop-ups, often slowing down the system; Scareware annoys users with fake alerts and prompts to purchase unnecessary software.

1. Stealing computing resources: This malware hijacks the computer’s processing power and resources for purposes that benefit the attacker, such as mining cryptocurrency (cryptominer) or sending spam emails.

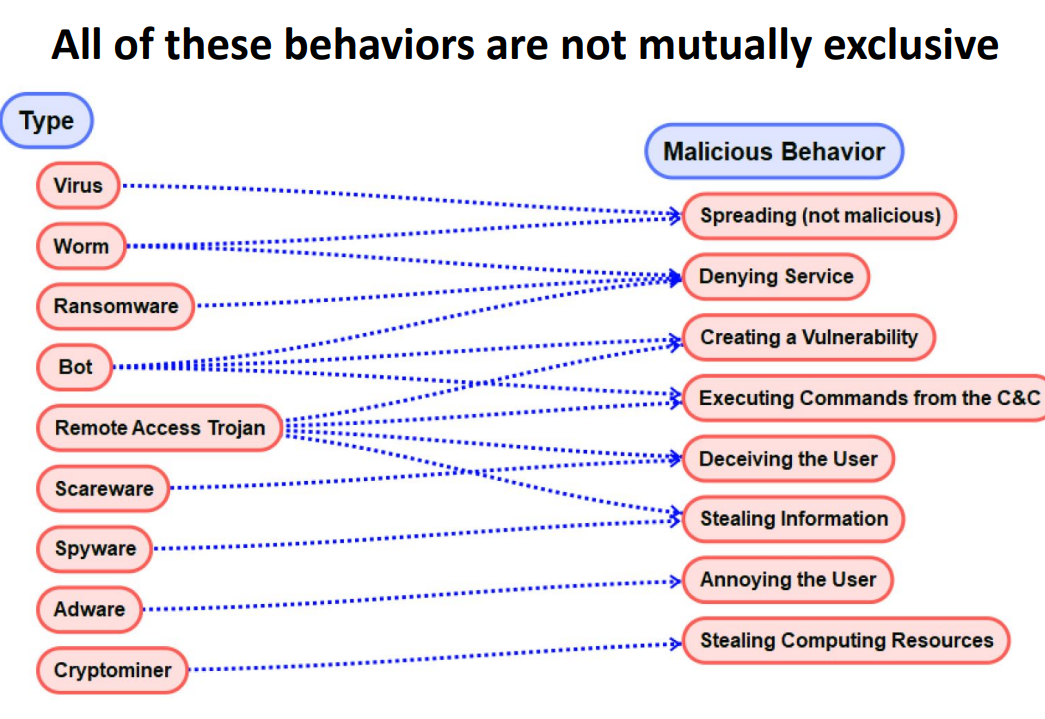
Examples: Cryptominer uses the computer's processing power to mine cryptocurrency without the user's consent; Bot can hijack resources to perform automated tasks like sending spam or participating in DDoS attacks.

1. Spreading (not malicious): Some malware primarily focuses on spreading itself without causing direct harm to the system.

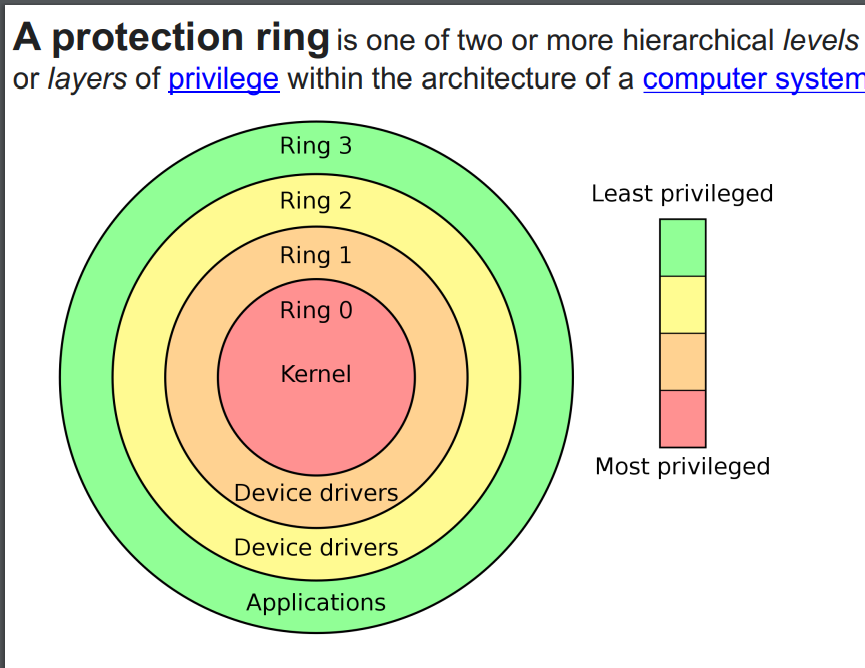
Examples: Worm spreads across networks to demonstrate a vulnerability or test security defenses, though it may not carry a damaging payload; some Virus variants spread to other files or systems mainly to replicate rather than to cause direct harm.

## Difference bw Ransomware and Cryptominers:





## By Privilege:



1. **User Mode (Ring 3)**

Description: This is the level where everyday programs run. These programs have limited access to the computer’s resources and need to ask the operating system for permission to use things like the hard drive or the internet.

Security: This separation keeps the system safe. It prevents regular programs from directly touching critical parts of the computer, which helps protect sensitive data and system stability.

Examples: Programs like web browsers, text editors, games, and other software you use daily.

1. **Kernel Mode (Ring 0)**

Description: This is the level where the core of the operating system (the kernel) operates. The kernel has full control over the computer’s hardware and system resources.

Security: Running at this level allows the operating system to manage everything on the computer securely and efficiently, making sure that all hardware and memory are used correctly and safely.

Examples: The heart of the operating system, such as the core components of Windows or Linux, as well as drivers that make devices like printers and network cards work.

1. **Hypervisor Mode (Ring -1)**

Description: This level is for hypervisors, which allow multiple operating systems to run on one physical machine. It sits below the operating systems and manages them.

Security: The hypervisor ensures that different virtual machines (VMs) are isolated from each other, so if one VM gets hacked, the others remain safe.

Examples: Virtualization software like VMware ESXi, Microsoft Hyper-V, and Xen, which let you run multiple operating systems on a single computer.

1. **Hardware Mode (Ring -3)**

Description: This is the deepest level, used by special firmware embedded in the hardware. It operates below even the hypervisor and operating system.

Security: This mode provides a foundational layer of security and control, independent of the software running on the machine. It can perform critical tasks and enforce security even if the higher-level software is compromised.

Examples: Built-in hardware management systems like Intel Management Engine (ME), AMD Secure Processor, and the firmware you see when the computer starts up (BIOS/UEFI).

In summary, the lower the number (or more negative), the more control and access the mode has over the computer. User mode is where your regular programs run with limited access, kernel mode is where the core operating system runs with full access, hypervisor mode manages multiple operating systems, and hardware mode provides the deepest level of control and security.

## Lifecycle of Malware:

**STAGE 1: Reconnaissance**

Reconnaissance is the initial stage where attackers gather information about the target system or network. The aim is to understand the environment and identify potential vulnerabilities.

**Passive Reconnaissance**: Collecting data without directly interacting with the target, such as through social media, public websites, and domain registration information.

**Active Reconnaissance:** Directly engaging with the target to obtain information, such as network scanning, port scanning, and vulnerability scanning.

**STAGE 2: Weaponization & Delivery**

In this stage, attackers create the malware payload and determine the method of delivery to the target.

**Weaponization**: Combining the malware payload with an exploit into a deliverable package (e.g., a malicious document, a piece of software, or a script).

**Delivery**: Sending the weaponized payload to the target. Common delivery methods include phishing emails, malicious websites, infected USB drives, and compromised software updates.

**STAGE 3: Exploitation**

Exploitation occurs when the delivered malware takes advantage of a vulnerability to execute code on the target system.

Exploiting software vulnerabilities in operating systems, applications, or network services.

Using social engineering techniques to trick users into executing malicious code.

Leveraging zero-day vulnerabilities that are unknown to the software vendor.

**STAGE 4: Installation**

During the installation stage, the malware payload is installed on the target system, establishing a foothold for the attacker.

Installing backdoors, rootkits, or other types of persistent malware.

Modifying system files or configurations to maintain access.

Ensuring the malware can survive system reboots and avoid detection.

**STAGE 5: Command and Control**

In this stage, the installed malware connects to the attacker's command and control (C2) server to receive instructions and send back data.

Establishing a communication channel with the C2 server, often using HTTP, HTTPS, or other protocols.

Exfiltrating data from the target system to the C2 server.

Receiving commands from the attacker, such as updating malware, executing further attacks, or gathering specific information.

**STAGE 6: Actions on the Objective**

This final stage involves the attacker achieving their goals, which can vary widely depending on the intent of the attack.

**Data Exfiltration**: Stealing sensitive data such as intellectual property, financial information, or personal data.

**Destruction:** Deleting files, corrupting data, or disrupting operations.

## Components of Malware:

1. **Payloads**

The payload is the part of the malware that actually does the harmful things the attacker wants. This could be stealing your data, encrypting your files, breaking into your system, or causing other damage.

1. **Propagation Mechanisms**

These are the ways malware spreads from one computer to another. It can spread through email attachments, links in messages, USB drives, shared files on a network, or by exploiting software vulnerabilities.

1. **Evasion Techniques**

These are tricks malware uses to avoid being detected by antivirus programs and other security tools. This can include hiding its code, changing its appearance, or encrypting its communication.

1. **Persistence Mechanisms**

These are methods the malware uses to stay on your system even after you restart your computer or try to remove it. It can change system settings, create special entries in your computer’s registry, or install itself to start automatically when your computer boots up.

1. **Command and Control Communication**

Many types of malware need to communicate with a server controlled by the attacker. This server sends instructions to the malware, receives stolen data, or updates the malware with new features.

1. **Data Exfiltration Modules**

These are parts of the malware specifically designed to steal information from your computer and send it to the attacker. This could include things like passwords, bank information, or company secrets.

1. **Self-Propagation Mechanisms**

Some malware, like worms, can spread by themselves to other computers without any help from the user. They can infect many computers quickly by exploiting vulnerabilities.

## **Samy Worm**

The **Samy malware attack**, also known as the **Samy worm**, was a cross-site scripting (XSS) worm that specifically targeted MySpace, a popular social networking site, in 2005. Here’s an overview of the attack and its significance:

### **Overview of the Samy Worm:**

1. **Creator**:
   * The worm was created by Samy Kamkar, a security researcher and hacker.
2. **Objective**:
   * The main goal was to gain notoriety by creating a worm that would spread rapidly across MySpace profiles. The payload of the worm was to add the phrase “but most of all, Samy is my hero” to affected profiles.
3. **How It Worked**:
   * **Cross-Site Scripting (XSS)**: The worm exploited an XSS vulnerability in MySpace’s code. XSS vulnerabilities allow attackers to inject malicious scripts into web pages viewed by other users.
   * **Self-Replicating Payload**: The injected script was designed to copy itself to the profile of anyone who viewed an infected profile, making it a worm.
   * **Social Engineering**: The worm relied on the fact that people frequently visited each other's profiles, facilitating rapid spread.
4. **Spread**:
   * The worm spread exponentially. Each time an infected profile was viewed, the worm’s script would execute, infecting the viewer's profile.
   * In less than 24 hours, over one million MySpace profiles were infected.

### **Workflow of the Samy Worm:**

1. **Infection**:
   * Samy Kamkar injects a malicious script into his own MySpace profile using an XSS vulnerability.
2. **Propagation**:
   * When another user views Samy’s profile, the script runs, copying itself to the viewer's profile.
3. **Further Spread**:
   * The script continues to propagate as each newly infected profile is viewed by more users, leading to exponential spread.
4. **Payload**:
   * Alongside propagation, the worm also changes the profile to include the message “but most of all, Samy is my hero”.

In summary, the Samy worm was a highly effective and rapid-spreading XSS attack that targeted MySpace, infecting over a million profiles in a short time and bringing significant attention to the vulnerabilities and potential impacts of XSS attacks on social networking sites.

# **Reverse Engineering of a Malware:**

**1. Acquire a Sample**

Get a copy of the malware. This could be done by downloading it from the internet or receiving it from someone else.

Example:

You find a suspicious file online or get it from a security researcher who wants you to analyze it.

**2. Obtain a Disassembler or Decompiler**

Get tools that can turn the malware’s binary code (which is hard to understand) back into more readable code.

Example:

Download a tool like IDA Pro (disassembler) or Ghidra (decompiler).

**3. Analyze the Code**

Use the disassembler or decompiler to look at the malware’s code. This helps you understand what the malware is programmed to do.

Example:

Open the malware file in IDA Pro and look at the functions and instructions to see how it operates.

**4. Create a Sandbox Environment**

Set up a controlled and isolated environment where you can safely run the malware without it affecting your real system.

Example:

Use virtual machine software (like VirtualBox) to create a fake computer where you can test the malware.

**5. Run the Malware in the Sandbox**

Execute the malware in the sandbox and watch what it does. Take notes on its behavior, such as what files it creates or modifies, what network connections it makes, etc.

Example:

Start the malware in your virtual machine and monitor its actions using tools like Process Monitor and Wireshark.

**6. Generate a Report of Your Findings**

Write down everything you discovered about the malware. This helps others understand the malware and take action against it.

Example:

Create a document detailing how the malware works, what harm it can cause, and how to protect against it.

# Reverse Shell:

A reverse shell is a type of network connection that allows a remote user to gain control over a computer. Unlike a traditional shell, where the remote user initiates a connection to the target computer, a reverse shell works by having the target computer initiate the connection back to the remote user. This technique is often used to bypass firewalls and security measures that block incoming connections but allow outgoing connections.

# Shell Code:

Shellcode is a small piece of code used as the payload in the exploitation of a software vulnerability. Its primary purpose is to execute a command, usually to open a shell (hence the name), which allows the attacker to take control of the target system. Shellcode is typically written in machine code, which is a low-level programming language that can be directly executed by the CPU.

How It Works:

Discovery of Vulnerability: The attacker identifies a software vulnerability, such as a buffer overflow.

Exploit Development: The attacker writes shellcode designed to be executed when the vulnerability is exploited.

Payload Delivery: The shellcode is injected into the target system through the exploit.

Execution: Once the exploit triggers, the shellcode runs, performing its intended action, such as spawning a shell or executing a specific command.

# Windows API:

The Windows API is like a set of tools and instructions that software can use to talk to the Windows operating system. These tools let programs do things like open files, manage processes, and connect to the internet.

How Malware Uses Windows API

Malware (bad software) uses these tools to perform harmful actions on a computer. Here’s how it works:

Manipulating Files: Malware can use Windows API functions to read, write, and delete files on your computer.

Controlling Processes: Malware can start, stop, or control programs running on your computer.

Connecting to Networks: Malware can use the internet functions to send and receive data, such as stealing your personal information and sending it to hackers.

# Fork Bomb:

A fork bomb is a type of denial-of-service (DoS) attack against a computer system. It works by creating a large number of processes very quickly to exhaust the system’s resources, making it unable to perform any useful work. Here’s a simple explanation:

The fork bomb rapidly creates copies of itself, with each new copy also creating more copies. This quickly overwhelms the system's CPU and memory, leading to a slowdown or complete freeze.

# **XSS:**

Cross-Site Scripting (XSS) is a type of security vulnerability typically found in web applications. It allows attackers to inject malicious scripts into webpages viewed by other users. These scripts can steal sensitive data, such as cookies, session tokens, or other information stored by the user's browser, and can even control the user's browser remotely.

# **Types of XSS:**

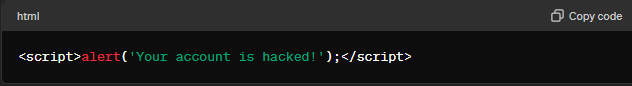
## **Stored XSS:**

Stored XSS (also known as persistent XSS) occurs when the malicious script is permanently stored on the target server, such as in a database, a message forum, visitor log, comment field, etc. The victim retrieves the malicious script from the server when they open the relevant webpage.

**Example of Stored XSS:**

A social media platform allows users to post status updates. The platform does not properly sanitize user inputs before storing them in the database.

* Attacker's Action: The attacker posts a status update with the following content:



* Stored in Database: The status update, including the malicious script, is stored in the platform’s database.
* User Views Status: When any user, including the attacker, views the timeline, the platform retrieves the status update from the database and displays it as part of the webpage.
* Script Execution: The user's browser executes the malicious script within the context of the social media platform’s page. This results in an alert box popping up with the message 'Your account is hacked!'.

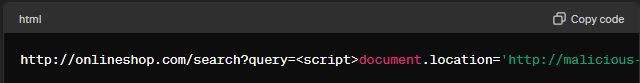
## **Reflected XSS:**

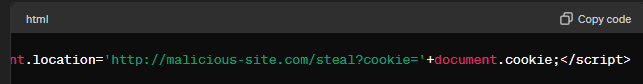
Reflected XSS (also known as non-persistent XSS) occurs when the malicious script is reflected off a web server, such as in an error message, search result, or any other response that includes some or all of the input sent to the server as part of the request. This type of XSS is typically delivered via email or another web server and tricked into clicking a malicious link.

Example of Reflected XSS:

An online shopping site has a search feature that displays user queries in the search results page without proper sanitization.

Attacker Crafts URL: The attacker creates a URL designed to execute a script:

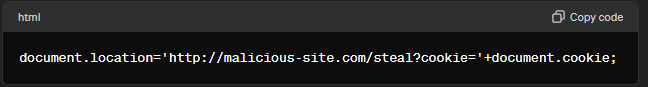


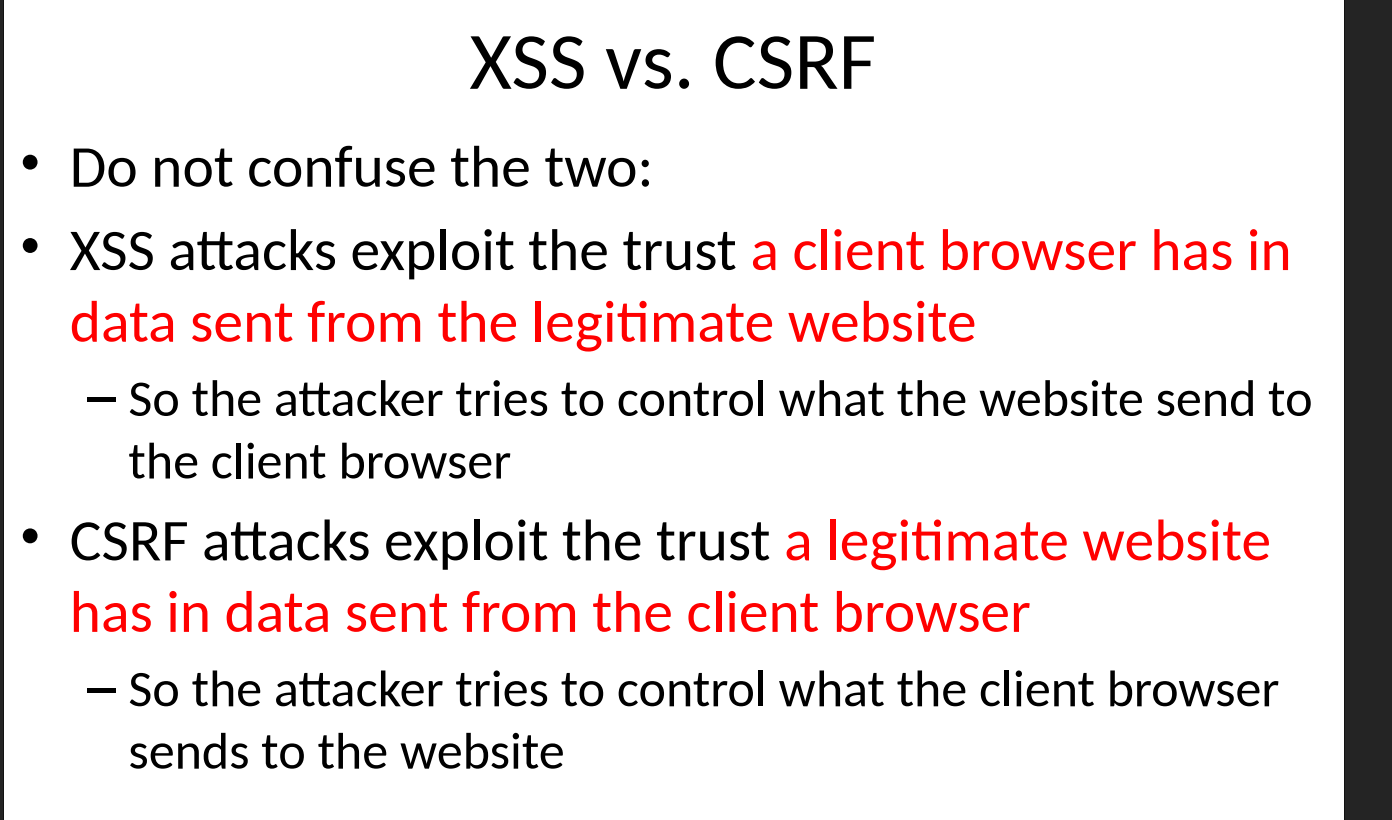


Sends Malicious Link: The attacker sends this URL to a victim via email or social media, disguised as a link to a product search.

Victim Clicks Link: The victim clicks the link, which takes them to the search results page on the online shopping site.

Script Execution: The search feature reflects the query parameter back to the user without sanitization, resulting in the script execution. The malicious script steals the user's cookies and sends them to the attacker's server:





## **Prevention Methods:**

To prevent both types of XSS, developers should:

1. Sanitize Input: Ensure all user inputs are properly sanitized and encoded. This can be done by removing or escaping characters that can be interpreted as part of a script.
2. Use Secure Coding Practices: Employ frameworks and libraries that inherently protect against XSS by default.
3. Content Security Policy (CSP): Implement CSP headers to restrict the sources from which scripts can be loaded.
4. Output Encoding: Encode data before displaying it in the browser to ensure that any executable code is rendered as text.
5. Validation: Validate input on both the client side and server side, ensuring that only the expected types of data are accepted.

By understanding the mechanisms of XSS and employing best practices for secure coding, developers can significantly reduce the risk of such vulnerabilities in their applications.

# **History of Cyber Crimes:**

1960:

* Transistors based comp, vacuum tubes
* Physical damage of data storage
* 1969: Canada, student riot caused fire and destroy data hosted in the uni

1970s:

* Mainframes intro
* Phy dam + frauds
* Illegal use of comp sys
* There was a group MOD (Masters of Deception), stole passwords & technical data from telephone companies, big agencies and unis

1980s:

* PC introduced
* Softwares and network concepts
* Software piracy
* Networks introduce hue to attackers k lie easy hgya k wo physically present na ho, remotely attack kren
* 1983: 19 year old boy of University of california, used his PC to break into a Defense Department.

1990s:

* Internet introduced hue.
* Jo locally attacks the wo globally bn gae ab
* Between 1995 and 1998, SKY-TV service was repeatedly hacked. The main motivation for the hackers, primarily in Germany, was to watch reruns of "Star Trek," which SKY-TV did not have the rights to broadcast in that region. The hackers broke SKY-TV's encryption to gain unauthorized access to this content.
* On 26 March 1999, the Melissa worm (a massmailing macro virus) infected a document on a victim's computer, then automatically sent that document and a copy of the virus spread via e-mail to other people

2000s (21st century):

* Cloud computing wgera bh intro hgyu
* • Phishing • Botnet Attacks • Automated attacks • more difficult for law enforcement to handle and investigate
* Feb 2000 m Mafia Boy n DoS attack kra high profiles p like Dell Yahoo eBay and CNN etc
* In June 2012 LinkedIn and eHarmony were attacked, compromising 65 million password hashes. – 30,000 passwords were cracked and 1.5 million EHarmony passwords were posted online.
* In May 2017, 74 countries logged a ransomware cybercrime, called "WannaCry"

# **Cyber Crime:**

Cyber crime refers to illegal activities that involve computers, networks, or digital devices. These crimes can target individuals, organizations, or governments and often involve exploiting vulnerabilities in software, hardware, or human behavior.

## **Types of Cyber Crime:**

1. Hacking: Unauthorized access to computer systems to steal, modify, or destroy data.
2. Phishing: Deceptive attempts to obtain sensitive information like passwords and credit card numbers by pretending to be a trustworthy entity.
3. Malware: Malicious software, such as viruses, worms, and ransomware, designed to damage or disrupt systems.
4. Identity Theft: Stealing personal information to impersonate someone else, often for financial gain.
5. Fraud: Deceitful practices like online scams to obtain money or goods unlawfully.
6. Denial of Service (DoS) Attacks: Overloading a system with traffic to make it unavailable to users.
7. Cyberbullying: Using digital communication tools to harass or intimidate individuals.

# Cyber Security:

Cyber security involves protecting computers, networks, and data from unauthorized access, damage, or theft. It encompasses a wide range of practices, technologies, and policies designed to safeguard information and ensure the integrity, confidentiality, and availability of data.

Examples:

1. Using firewalls and intrusion detection systems to protect a corporate network.
2. Encrypting sensitive data to prevent unauthorized access during transmission.
3. Implementing strong passwords and multi-factor authentication to secure user accounts.
4. Regularly updating software to patch vulnerabilities and prevent exploits.
5. Conducting security audits and penetration tests to identify and mitigate potential risks.

# **Security Threat/Attack:**

It refers to any action that compromises the security of information owned by an organisation.

1. Active Attack
2. Passive Attack

## **Active Attack:**

Active attacks involve deliberate attempts by attackers to alter system resources or affect their operation. These attacks aim to disrupt, manipulate, or tamper with data and systems.

Characteristics:

* Directly modifies data, system resources, or network traffic.
* More likely to be detected due to noticeable effects on the system.
* Often requires real-time interaction with the target.

Types of Active Attacks:

1. Masquerade: An attacker pretends to be an authorized user to gain access to a system or data.
2. Repudiation: An attacker falsely denies performing an action, such as sending a message.
3. Denial of Service (DoS): Overloading a system or network to make it unavailable to legitimate users.

## Passive Attack:

Passive attacks involve eavesdropping or monitoring of data transmission without altering the data. The main goal is to gain information without being detected.

Characteristics:

* Does not alter data or system resources.
* Harder to detect as it does not affect the system’s normal operations.
* Typically involves gathering intelligence over time.

Types of Passive Attacks:

1. Eavesdropping: Intercepting and listening to private communications without consent.
2. Traffic Analysis: Monitoring and analyzing patterns in data traffic to infer information, such as the frequency and timing of communication.
3. Spyware:....

# Security Mechanism:

Security mechanisms are specific tools, techniques, or processes used to implement security within a system. They are the practical methods employed to safeguard data and systems from threats.

Examples:

1. Encryption: Converts data into a coded format to prevent unauthorized access.
2. Firewalls: Filters incoming and outgoing network traffic based on security rules.
3. Intrusion Detection Systems (IDS): Monitors network traffic for suspicious activities.
4. Authentication Methods: Processes like passwords, biometric scans, and multi-factor authentication.
5. Access Control: Restricts who can access certain resources based on policies.

# Security Service:

Security services are broader functions or capabilities that ensure the overall protection of information and systems. They are the objectives or goals that are achieved through the use of various security mechanisms.

Examples:

1. Confidentiality: Ensures that information is only accessible to authorized individuals.
2. Integrity: Ensures that information is accurate and has not been tampered with.
3. Availability: Ensures that information and resources are accessible to authorized users when needed.
4. Authentication: Verifies the identities of users and systems.
5. Authorization: Grants permissions to users and systems based on their identities and roles.
6. Non-repudiation: Ensures that the origin and receipt of data cannot be denied.

# Relation between Threats - Services - Mechanisms:

An online shopping platform wants to ensure its customers can shop securely. The platform needs to protect customer data, ensure transactions are legitimate, and maintain system availability.

1. **Addressing Phishing Attacks:**

Threat: Phishing attacks aim to steal user credentials.

Service: Authentication ensures that only legitimate users can access accounts.

Mechanism: Implementing Two-Factor Authentication (2FA) adds an extra layer of security, making it harder for attackers to gain access even if they obtain login credentials through phishing.

1. **Preventing Man-in-the-Middle Attacks:**

Threat: Man-in-the-Middle attacks intercept communications between the customer and the platform.

Service: Confidentiality ensures that customer data is protected during transmission.

Mechanism: Encryption secures the data being transmitted, making it unreadable to anyone who intercepts it.

1. **Ensuring Availability During DDoS Attacks:**

Threat: DDoS attacks overload the platform's servers, making the website unavailable.

Service: Availability ensures that the platform remains accessible to customers.

Mechanism: Load Balancers distribute traffic across multiple servers, preventing any single server from being overwhelmed and maintaining the platform’s availability.

# 

# **NTLM (NT LAN Manager)**

**Background and History:**

NTLM, which stands for NT LAN Manager, is a suite of security protocols developed by Microsoft to provide authentication, integrity, and confidentiality for users.

It was introduced in the early versions of Windows NT and served as the primary authentication protocol before being largely replaced by Kerberos in more recent versions of Windows.

Despite being succeeded by Kerberos, NTLM is still supported for backward compatibility with older systems and applications.

**Purpose:** The primary purpose of NTLM is to authenticate users and systems within a Windows-based network. It provides mechanisms for:

* **User Authentication:** Verifying the identity of users attempting to access resources.
* **Message Integrity:** Ensuring that data has not been altered during transmission.
* **Confidentiality:** Protecting data from unauthorized access during transmission.

**Limitations:**

1. **Security Vulnerabilities:** NTLM is susceptible to several types of attacks, including:
   * **Pass-the-Hash Attack:** Attackers can use stolen password hashes to authenticate without knowing the actual password.
   * **Replay Attacks:** Attackers can capture and reuse valid authentication messages to gain unauthorized access.
   * **Man-in-the-Middle Attacks:** Without mutual authentication, attackers can intercept and manipulate communications between clients and servers.
2. **No Mutual Authentication:** NTLM only authenticates the client to the server, not the server to the client.
3. **Weak Encryption:** NTLM uses relatively weak cryptographic methods (e.g., MD4 hash function), making it vulnerable to modern cryptographic attacks.
4. **Limited Scalability:** NTLM is less efficient in large environments with frequent authentication requests compared to Kerberos.

**Current Status:**

Although NTLM has been largely replaced by Kerberos in modern Windows environments, it is still supported for backward compatibility.

Many legacy systems and applications still rely on NTLM for authentication.

However, Microsoft recommends using Kerberos wherever possible due to its enhanced security features and better performance.NTLM as the primary authentication protocol in modern network environments due to several key advantages and improvements. Here are the main reasons why:

### **Reasons NTLM is Still in Use**

* **Legacy System Support:** Needed for compatibility with older systems and applications.
* **Simplicity:** Easier to set up and use in small or straightforward environments.
* **No KDC Required:** Can be used in environments without a Key Distribution Center.
* **Third-Party Applications:** Some applications only support NTLM.
* **Practical Constraints:** Transitioning to Kerberos may be resource-intensive or complex.

### **Comparison of Kerberos and NTLM**

| **Feature** | **Kerberos** | **NTLM** |
| --- | --- | --- |
| **Authentication** | Mutual authentication (client and server) | Only client authentication |
| **Encryption** | Stronger encryption using tickets and keys | Weaker encryption using hashes |
| **Ticket Lifespan** | Limited lifespan of tickets | No ticket concept, relies on password hash |
| **Scalability** | Efficient in large, complex environments | Less efficient, especially in large networks |
| **Centralized** | Requires a Key Distribution Center (KDC) | No KDC required |
| **Compatibility** | Modern systems and applications | Older systems and applications |
| **Complexity** | More complex to set up and manage | Simpler setup and use |
| **Support** | Widely supported across different platforms | Primarily supported in Windows environments |

# **Kerberos:**

1. Developed at MIT to protect network services provided by a Project Athena - a joint project of **MIT, Digital Equipment Corporation, and IBM** to produce a campus-wide distributed computing environment for educational use

2. The protocol is based on the earlier Needham– Schroeder symmetric key protocol

3. **Steve Miller and Clifford Neuman,** the primary designers of Kerberos **version 4**, published that version in the late **1980s**

4. Current **version 5** was designed by **John Kohl**

5. Now version 4 is **no longer supported** by MIT

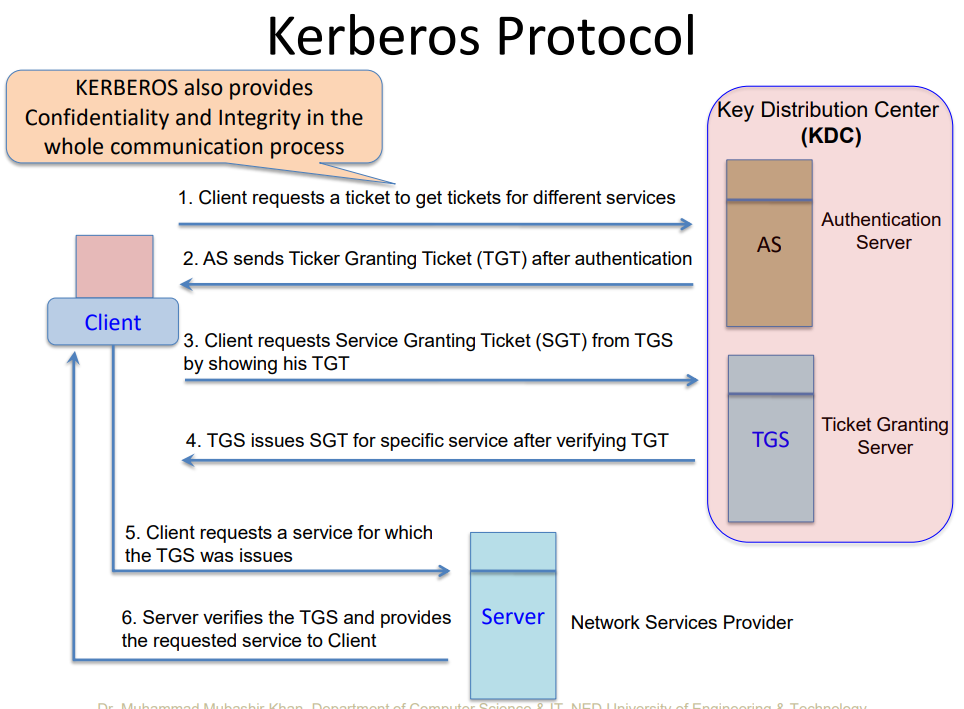
6. Uses **UDP port 88** by default

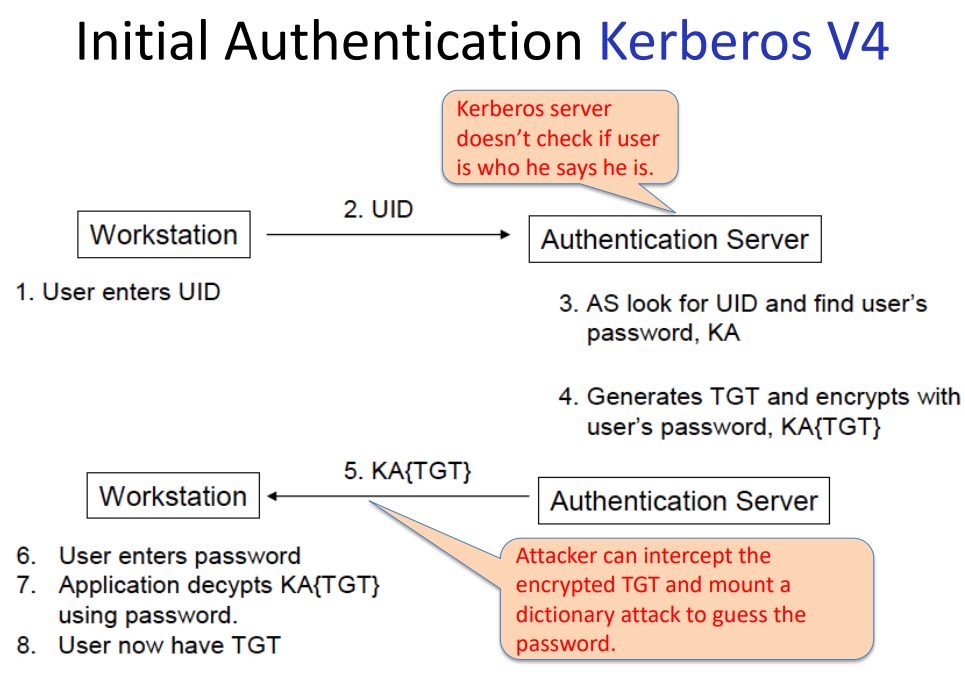
7. It is a Trusted key server system

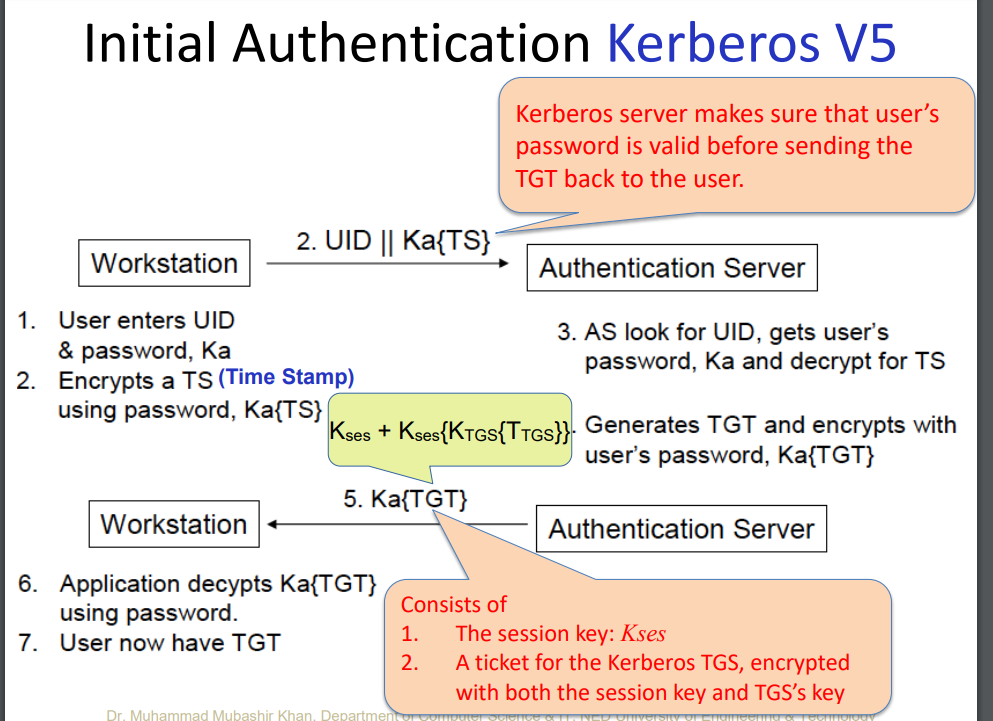
8. “ Kerberos provides a centralized authentication server whose function is to authenticate users to servers and servers to users. In Kerberos Authentication server and database is used for client authentication. Kerberos runs as a third-party trusted server known as the Key Distribution Center (KDC). “

The main components of Kerberos are:

* **Authentication Server (AS):**   
   The Authentication Server performs the initial authentication and ticket for Ticket Granting Service.
* **Ticket Granting Server (TGS):**   
   The Ticket Granting Server issues the ticket for the Server







### **1. Client Requests a Ticket-Granting Ticket (TGT):**

* **Client (user) asks for access**: The user sends a request to the Authentication Server (AS) with their ID and a timestamp, optionally encrypted with their password.
* **AS checks the password**: The AS looks up the user's password (stored securely) and uses it to check if the request is legitimate.
* **AS sends back a TGT**: If everything checks out, the AS creates a TGT (a special ticket) and a session key (a temporary key for this session). The TGT is encrypted with the Ticket-Granting Server’s (TGS) key, and the session key is encrypted with the user’s password.

### **2. Client Requests Service Ticket from TGS:**

* **Client decrypts TGT**: The client uses their password to decrypt the session key and uses this key to work with the TGT.
* **Client asks TGS for a service ticket**: The client sends the TGT and an authenticator (ID and timestamp encrypted with the session key) to the TGS.
* **TGS checks the request**: The TGS decrypts the TGT with its key, verifies the authenticator, and then creates a service ticket.

### **3. Ticket-Granting Server Response:**

* **TGS sends service ticket**: The TGS sends back the service ticket to the client. This ticket includes a new session key for the service the client wants to access, and it’s encrypted with the session key from the TGT.

### **4. Client Requests Access to the Service:**

* **Client sends service ticket**: The client sends the service ticket and a new authenticator (ID and timestamp) to the target service.

### **5. Service Response:**

* **Service checks the ticket**: The service decrypts the service ticket with its own key and verifies the authenticator.
* **Service grants access**: If everything checks out, the service allows the client access.

## 

## 

## **Client’s Long-Term Key:**

1. **Password to Encryption Key Conversion**:
   * **User Logs On**: When a user logs on to a workstation, the Kerberos client software on the workstation captures the user's password.
   * **Hashing Function**: The Kerberos client converts the user's password into an encryption key by inputting the password into a one-way hashing function. This derived key is known as the **client’s long-term key**.

## **Defense Against Attacks**

To defend against the risk of a session ticket or session key being compromised due to a long time interval, Kerberos employs two strategies:

### **1. Renewing Tickets at Short Intervals**

* **Concept**: Instead of allowing tickets to last for a long time, they are set to expire relatively quickly.
* **Process**: When a ticket expires, the client requests a new ticket.
* **Benefit**: Each new ticket comes with a new session key, so if a key gets compromised, it will only be useful for a short time. Regular renewal reduces the window of opportunity for an attacker.

### **2. Periodically Refreshing Session Keys**

* **Concept**: Even without issuing a completely new ticket, the session keys associated with the ticket can be refreshed or updated regularly.
* **Process**: The client and the server can agree to update the session key at set intervals while still using the same ticket.
* **Benefit**: This ensures that even if the session key is compromised, it will soon be replaced by a new key, thereby reducing the risk.

## **Drawbacks & Limitations:**

* **Single point of failure:**

It requires continuous availability of a central server. When the Kerberos server is down, new users cannot log in. This can be mitigated by using multiple Kerberos servers and (alternative) fallback authentication mechanisms.

* **Strict time requirements:**

If the host clock is not synchronized with the Kerberos server clock, the authentication will fail.

The default configuration per MIT requires that clock times be no more than five minutes apart.

* **Requires trust relationship:**

Not suitable for unknown/untrusted clients as in a typical Internet or cloud computer scenario, where the authentication provider typically does not have knowledge about the users client system.

# **IPSec**:

IPsec is a suite of protocols used to secure communications over the internet. It ensures that data sent between two devices (like computers or routers) is encrypted and protected from being intercepted or tampered with.

## **Key Features of IPsec:**

1. **Confidentiality**: IPsec encrypts the data, making it unreadable to anyone who intercepts it.
2. **Integrity**: It ensures the data hasn’t been altered during transmission.
3. **Authentication**: IPsec verifies that the data comes from a trusted source.

## **Modes of IPSec:**

## **1. Transport Mode**

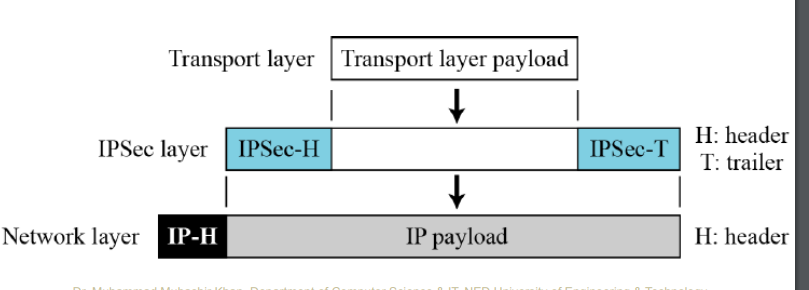
* Transport mode only encrypts the payload (the actual data) of the IP packet, leaving the IP header untouched.
* Typically used for end-to-end communication between two devices, such as two computers or servers.
* Ideal for scenarios where the devices themselves handle encryption and decryption, without needing to hide the IP addresses involved in the communication.

**How It Works:**

* The original IP header is kept intact.
* Only the data part of the packet (the payload) is encrypted and authenticated.
* Since the IP header is not encrypted, routers and other intermediate devices can still see and route the packet based on its destination.

**Example Scenario:**

* Secure communication between two computers on a private network. The computers encrypt and decrypt the data, while the network devices route the packets based on the original IP headers.



This diagram illustrates the encapsulation process in IPsec (Internet Protocol Security), highlighting the layers involved and the placement of headers and trailers.

1. **Transport Layer**:
   * The **Transport Layer Payload** represents the data that needs to be sent over the network. This typically includes TCP or UDP data.
2. **IPSec Layer**:
   * **IPSec-H (IPSec Header)**: This header is added to the transport layer payload by the IPsec protocol. It contains important security information, such as the Security Parameters Index (SPI), sequence number, and other control information.
   * **IPSec-T (IPSec Trailer)**: This trailer is appended to the end of the data packet and may contain additional information such as padding for alignment and an Integrity Check Value (ICV) for ensuring data integrity.
3. **Network Layer**:
   * **IP-H (IP Header)**: This header is added by the IP protocol. It contains information necessary for routing the packet to its destination, such as the source and destination IP addresses.
   * **IP Payload**: This is the combination of the IPSec-processed data (IPSec header, transport layer payload, and IPSec trailer).

## 

### **When It’s Acceptable:**

In secure, controlled environments like internal networks within a company or data centers, where physical and network access is tightly controlled, not encrypting the IP header can be acceptable and efficient.

For applications where performance is critical and the slight security trade-off is acceptable, transport mode’s efficiency can outweigh the risks.

### **When It’s Not Recommended:**

When communicating over public or untrusted networks, such as the internet, the risks associated with exposing the IP header are higher. In such cases, additional security measures (like using IPsec tunnel mode, which encrypts the entire IP packet) are recommended.

For communication involving highly sensitive data, where maximum security is paramount, ensuring that as much information as possible is encrypted (including the IP header) is advisable.

## **2. Tunnel Mode**

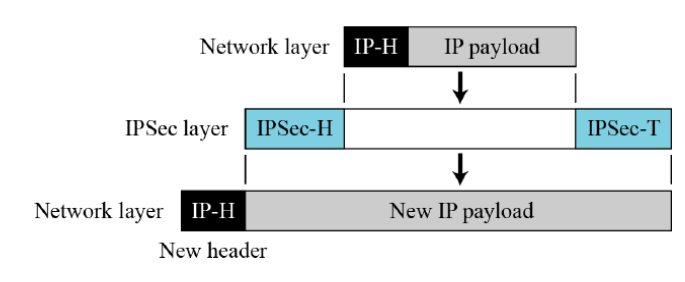
* Tunnel mode encrypts the entire IP packet (both the header and the payload) and then encapsulates it within a new IP packet with a new IP header.
* Commonly used for network-to-network communication, such as in VPNs (Virtual Private Networks).
* Ideal for scenarios where the communication needs to be secure and the IP addresses of the communicating devices need to be hidden from intermediate devices.

**How It Works:**

* The original IP packet is completely encapsulated within a new IP packet.
* Both the original IP header and payload are encrypted and authenticated.
* A new IP header is added, which is used for routing the packet to its destination.

**Example Scenario:**

* A company uses a VPN to connect two office networks securely over the internet. The routers at each office encrypt and decrypt the data, and the original IP addresses of the devices in each network are hidden from the internet.



## **IPSec Base Protocols**

### **Authentication Header (AH)**

**Purpose:**

* AH is used to provide data integrity, authentication, and optional anti-replay protection for IP packets. It does not provide confidentiality (encryption).

**How AH Works:**

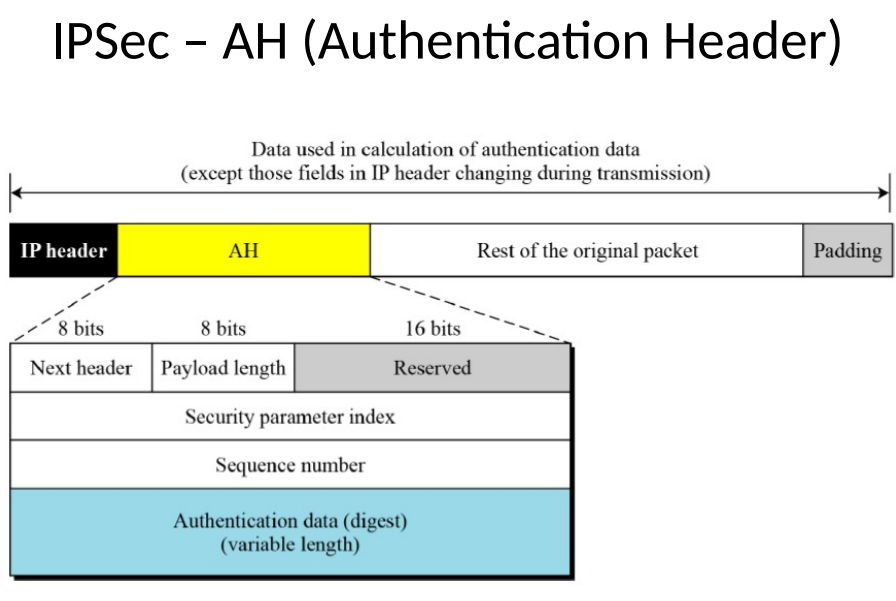
* **Extra Field:** Adds an extra header to the IP packet.
* **Integrity Check:** Ensures the data has not been altered during transmission.
* **Authentication:** Verifies the sender's identity.

**Key Components:**

* **Next Header (8 bits):** Indicates the type of protocol (e.g., TCP, UDP) in the payload.
* **Payload Length (8 bits):** Length of the AH header.
* **Reserved (16 bits):** Reserved for future use (set to zero).
* **Security Parameters Index (SPI, 32 bits):** Identifies the security association.
* **Sequence Number (32 bits):** Prevents replay attacks by ensuring each packet is unique.
* **Integrity Check Value (variable length):** A checksum to verify the packet's integrity.

**Usage:**

* **Authentication-Only Protocol:** Useful where encryption is restricted or not needed.
* **Faster Implementation:** Less computational overhead compared to encryption.



### **Encapsulating Security Payload (ESP)**

**Purpose:**

* ESP provides confidentiality (encryption), data integrity, authentication, and optional anti-replay protection.

**How ESP Works:**

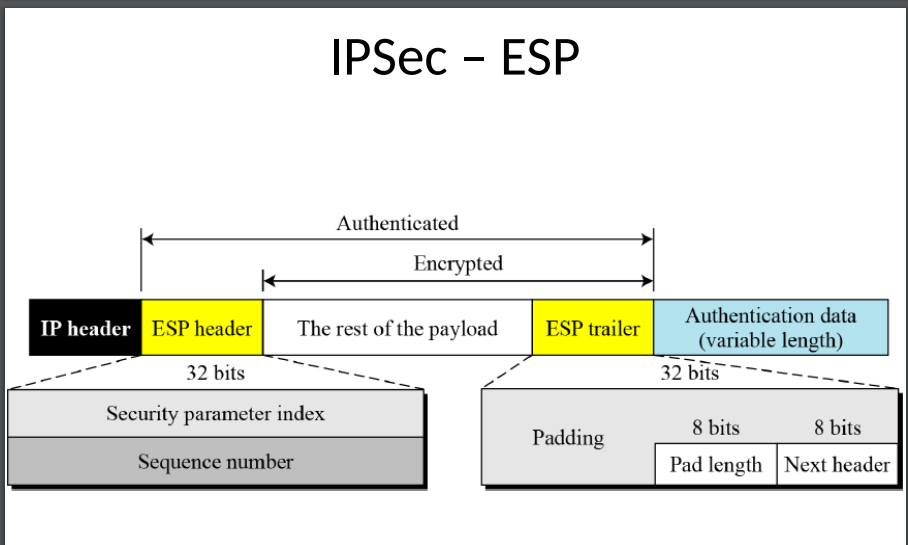
* **Encryption:** Encrypts the payload of the IP packet.
* **Encapsulation:** Adds ESP header and trailer around the encrypted data.
* **Authentication (optional):** Verifies the integrity and authenticity of the packet.

**Key Components:**

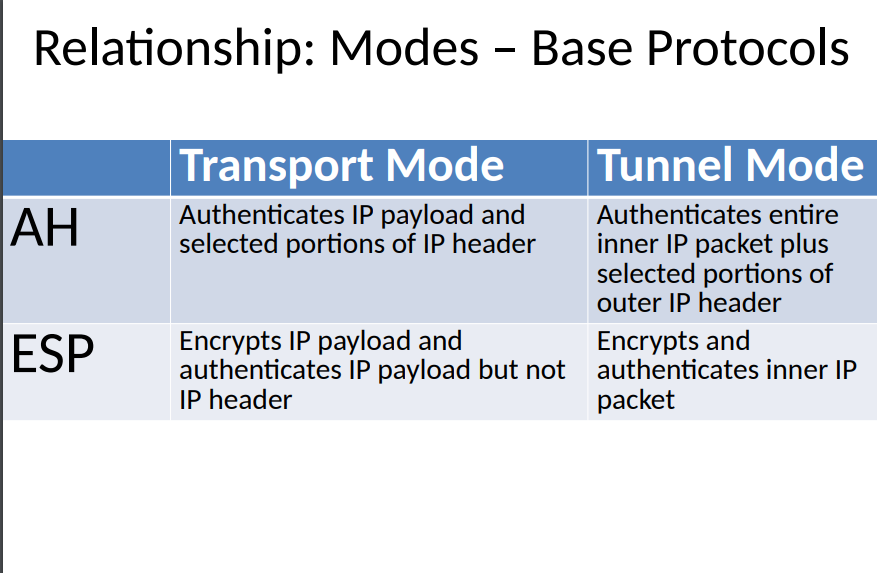
* **Security Parameters Index (SPI, 32 bits):** Identifies the security association.
* **Sequence Number (32 bits):** Prevents replay attacks by ensuring each packet is unique.
* **Payload Data (variable length):** The encrypted contents of the original IP packet.
* **Padding (0-255 octets):** Ensures the payload is aligned to the encryption algorithm's block size.
* **Pad Length (8 bits):** Indicates the size of the padding.
* **Next Header (8 bits):** Indicates the type of protocol (e.g., TCP, UDP) in the payload.
* **Integrity Check Value (variable length):** A checksum to verify the packet's integrity.

**Usage:**

* **Confidentiality and Integrity:** Provides both encryption and authentication for secure communication.
* **Flexible:** Can be used with or without authentication.



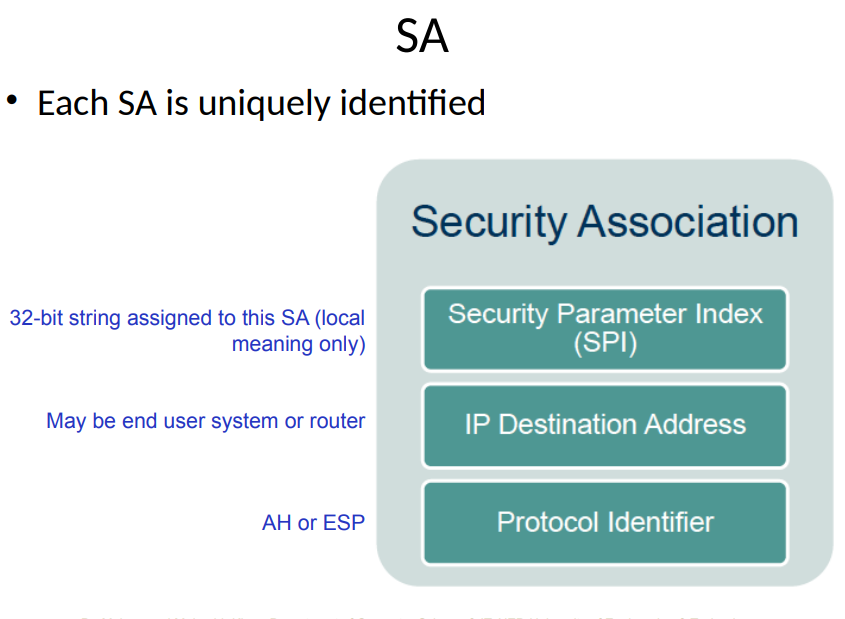
## 



## 

## **IPSec Connection Establishment**

1. **Phase 1: ISAKMP SA (**[**Internet Security Association and Key Management Protocol**](https://en.wikipedia.org/wiki/Internet_Security_Association_and_Key_Management_Protocol)**)**
   * **Purpose:** Establishes a secure and authenticated channel for negotiating further security associations.
   * **Steps:**
     + Mutual authentication between the two entities.
     + Establishment of a secure channel.
     + Generation of keys for Phase 2.
   * **Protocols Used:** Typically uses IKE (Internet Key Exchange) for this phase.
2. **Phase 2: SA (Security Association)**
   * Security associations (SAs) are the concept used by [IPSec](https://www.sciencedirect.com/topics/computer-science/internet-protocol) to manage all the parameters required to establish a [VPN](https://www.sciencedirect.com/topics/computer-science/virtual-private-networks) tunnel. In simple terms, SA is a set of parameters describing how communications are to be secured. SAs contain the following components: security keys and algorithms, mode of operation (transport or tunnel), key management method (IKE or manual key), and lifetime of the SA. IPSec stores all active SAs in a database called the security association database (SAD). The SAD contains all parameters needed for IPSec operation, including the keys currently in use. In order to have bidirectional communication, you must have at least two SAs, one for each direction of traffic flow.



****

## **Internet Key Exchange (IKE)**

**Purpose:**

* IKE is a protocol used to set up a secure and authenticated channel for negotiating security associations (SAs) in IPSec.

**How IKE Works:**

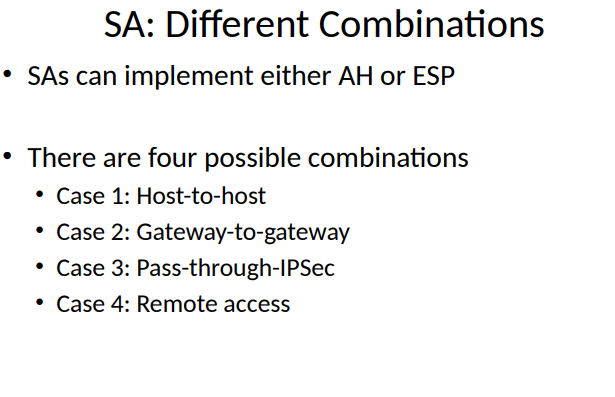
* **Generation and Distribution of Keys:** IKE generates and distributes secret keys.
* **Establishing SAs:** Creates SAs for secure communication between entities.

**Components of IKE:**

1. **ISAKMP (Internet Security Association and Key Management Protocol):**
   * **Framework:** Provides the framework for key exchange and SA negotiation.
   * **Functions:** Defines the procedures and packet formats for establishing, negotiating, modifying, and deleting SAs.
2. **Oakley:**
   * **Key Exchange Protocol:** Describes a series of key exchange techniques.
   * **Security:** Provides mechanisms for ensuring the security of key exchanges.
3. **SKEME:**
   * **Versatile Key Exchange:** Supports a variety of key exchange techniques.
   * **Efficiency:** Designed for lightweight and secure key exchange processes.

**IKE Phases:**

1. **Phase 1:**
   * **Purpose:** Establishes a secure, authenticated channel.
   * **Modes:** Main mode (more secure, more exchanges) and aggressive mode (fewer exchanges, less secure).
   * **Result:** Establishment of ISAKMP SA.
2. **Phase 2:**
   * **Purpose:** Negotiates the IPsec SAs.
   * **Result:** Creation of IPsec SAs for data transfer.

****

**Host-to-host**: Security is set up directly between two devices (like two computers) that are communicating.

**Gateway-to-gateway**: Security is set up between two network gateways (like routers), protecting the traffic between entire networks.

**Pass-through-IPSec**: This method allows encrypted traffic to pass through devices that don’t handle encryption themselves, usually for networks with existing security setups.

**Remote access**: Security is set up for individual users who connect from outside their main network, allowing them to securely access resources as if they were directly connected.

# **SSL / TLS :**

SSL (Secure Sockets Layer) and TLS (Transport Layer Security) are cryptographic protocols designed to provide secure communication over a computer network. TLS is the successor to SSL and is more secure. Here's a breakdown of how they work:

### **Purpose:**

* **SSL/TLS**: Ensure the privacy, integrity, and authenticity of data exchanged between clients (like web browsers) and servers.

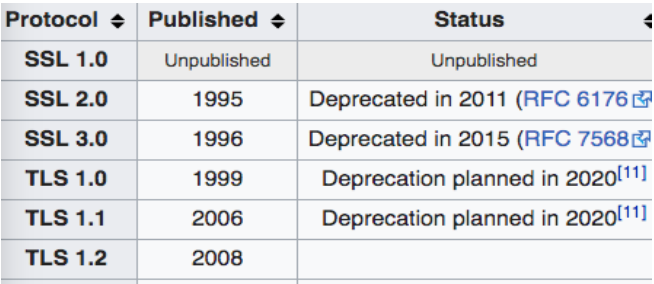
### **Key Concepts:**

1. **Encryption**: Protects data from eavesdroppers. Only the intended recipient can decrypt and read the data.
2. **Authentication**: Verifies the identity of the parties involved. This usually involves digital certificates issued by trusted Certificate Authorities (CAs).
3. **Integrity**: Ensures that data has not been tampered with during transmission. This is achieved through hashing.

### **SSL/TLS Differences and Improvements**

#### **Protocol Versions in Messages**

**Latest Version of TLS: 1.3 (2018)**

****

#### **Message Authentication**

* **SSL**: Uses **custom MAC** (Message Authentication Code) algorithms.
* **TLS**: Implements a **standardized MAC** called **HMAC** (Hash-based Message Authentication Code).
  + **HMAC Advantages**:
    - Proven in various implementations for its security and efficiency.
    - Operates with any hash function, not limited to MD5 or SHA, as SSL protocols specify.

#### **Key Material Generation**

* **SSL**: Utilizes **RSA** or **Diffie-Hellman** for key exchange and key material generation.
* **TLS**: Uses **HMAC** and its **pseudorandom function** (**PRF**) **output** to generate key material.
  + This results in more robust and flexible key material generation compared to SSL.

#### **Certificate Verify**

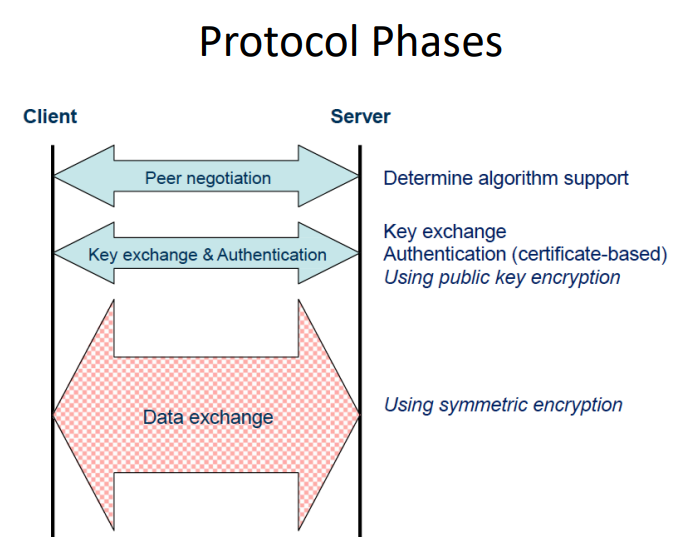
* **SSL**: CertificateVerify message requires a complex series of messages and procedures.
* **TLS**: Simplifies this process by including all verified information in the previously exchanged handshake messages.

#### **Alert Protocol Message Types**

* **Modifications in TLS**: TLS includes some modifications to alert protocol message types for better clarity and functionality compared to SSL.

### **Protocol Phases of SSL/TLS**

This diagram represents the different phases of the SSL/TLS protocol, showing how a secure connection is established and maintained between a client and a server.



#### **1. Peer Negotiation**

**Purpose**: Determine the algorithms that both the client and server support and will use for the connection.

* **Client**: Sends a list of supported encryption algorithms, hashing algorithms, and other protocol options.
* **Server**: Selects the strongest algorithms from the list provided by the client and informs the client of its choice.

**Key Actions**:

* **Algorithm Support**: Both parties agree on the encryption methods (cipher suites) they will use, which include key exchange algorithms, encryption algorithms, and hashing algorithms.

#### **2. Key Exchange and Authentication**

**Purpose**: Exchange cryptographic keys and authenticate each party to establish a secure connection.

* **Client**: Generates a random value (pre-master secret), encrypts it with the server's public key (obtained from the server's certificate), and sends it to the server.
* **Server**: Decrypts the pre-master secret using its private key.

**Key Actions**:

* **Key Exchange**: Securely exchange keys using public key encryption methods like RSA or Diffie-Hellman.
* **Authentication**: Verify the identity of the server (and optionally the client) using digital certificates. This step ensures that the server is who it claims to be, preventing man-in-the-middle attacks.

#### **3. Data Exchange**

**Purpose**: Securely transmit data between the client and server using symmetric encryption.

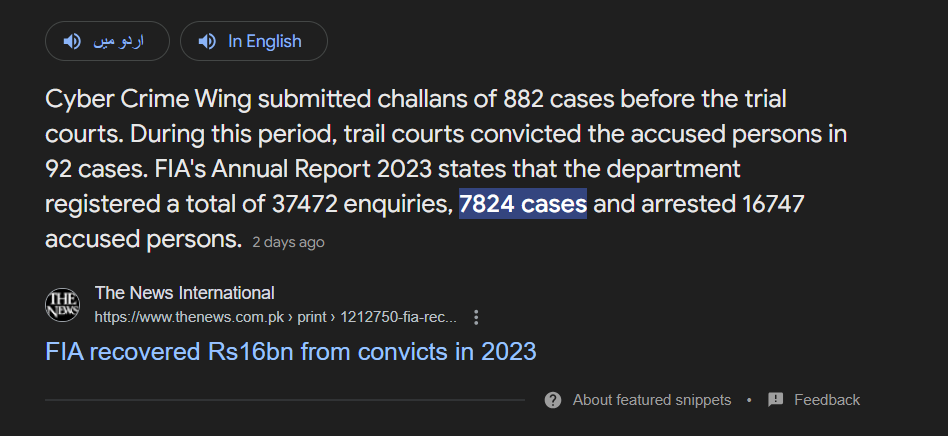
* **Client and Server**: Both parties now generate session keys using the pre-master secret and the agreed-upon algorithms from the peer negotiation phase.
* **Symmetric Encryption**: Use these session keys to encrypt and decrypt the data sent between them.

**Key Actions**:

* **Symmetric Encryption**: Provides efficient and secure data transmission because symmetric encryption is faster than asymmetric encryption.

# **Cyber Security Crimes in Pakistan**

### **Federal Investigation Agency (FIA) Statistics**



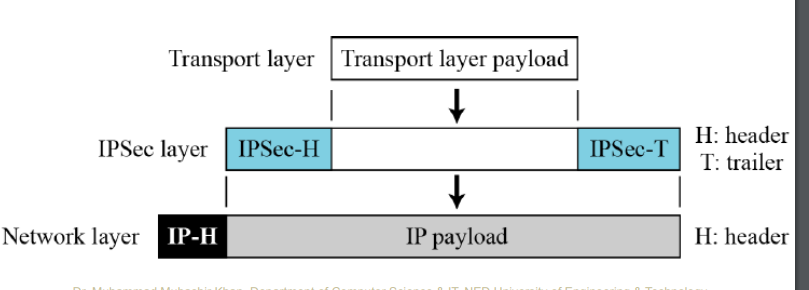
The **Pakistan Electronic Crimes Act (PECA)** of 2016 addresses various forms of cybercrime, including malware attacks. The key sections relevant to malware and similar attacks are:

1. **Section 4**: **Cyber Terrorism** – This section covers the use of electronic means to intimidate, coerce, or harass individuals or entities, including the use of malware to cause widespread harm or disruption.
2. **Section 6**: **Unauthorized Access to Information Systems** – This section deals with unauthorized access to computer systems and data, which can include using malware to gain access to or control over systems.
3. **Section 7**: **Unauthorized Access to Critical Information Infrastructure** – This specifically addresses unauthorized access to critical infrastructure, which can involve deploying malware to disrupt essential services.
4. **Section 8**: **Data Theft** – This section covers the unauthorized copying, transmission, or distribution of data, including data theft facilitated by malware.
5. **Section 9**: **Electronic Fraud** – This section addresses fraud committed using electronic means, including fraud perpetrated through malware.
6. **Section 10**: **Cyberstalking** – This includes using electronic means to stalk or harass individuals, which can involve malware for surveillance or harassment purposes.

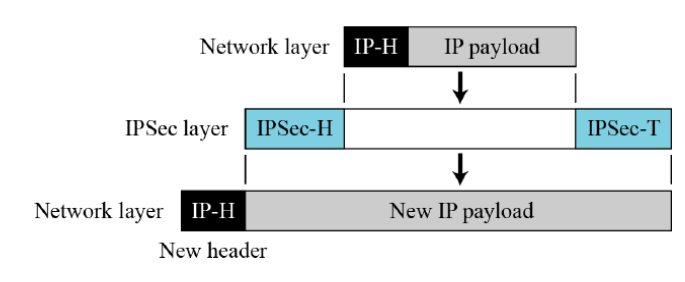
These sections collectively address various aspects of malware-related activities, from unauthorized access and data theft to broader cybercrimes.

# **Diagrams:**

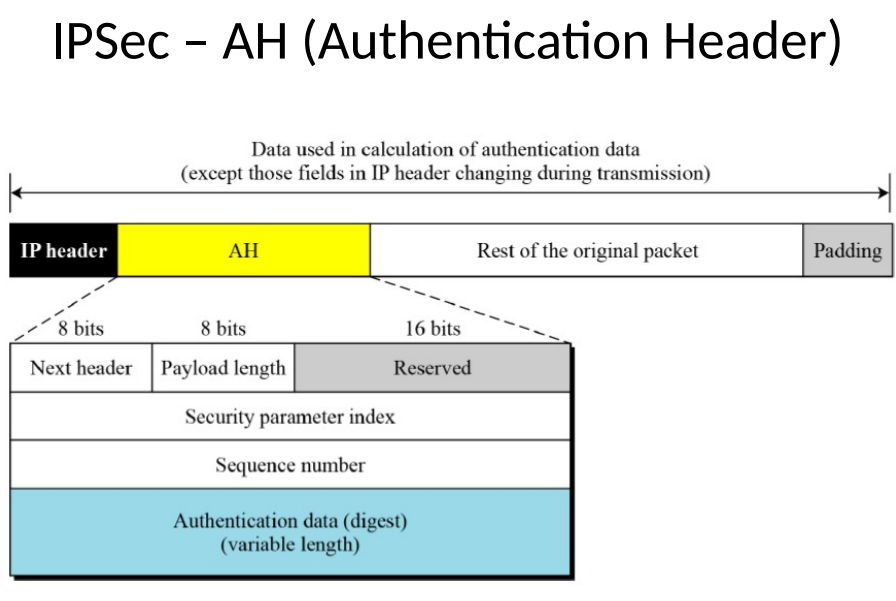
## **Transport mode:**



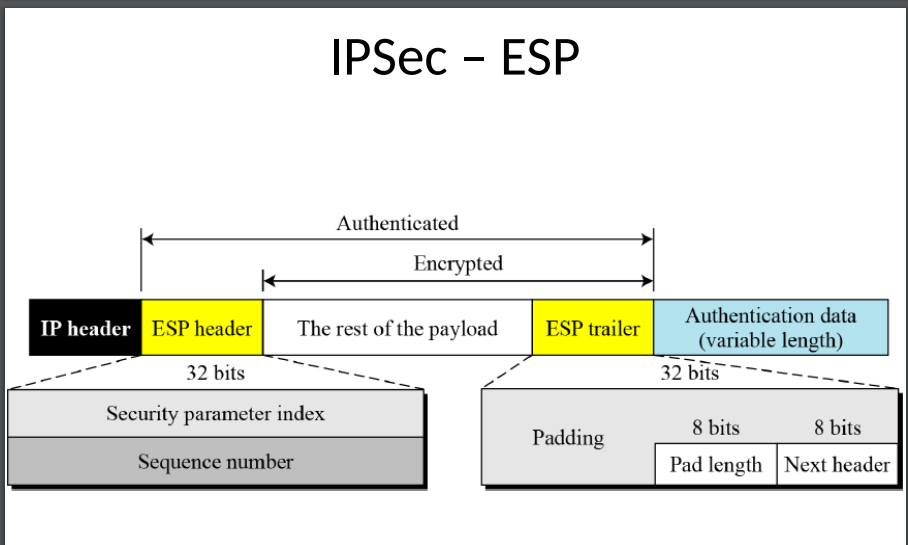
## **Tunnel Mode:**



## **AH:**



## **ESP:**



# **Attacks:**

## **What is ARP?**

ARP stands for Address Resolution Protocol. It’s a way for computers on a local network to find each other. When a computer wants to communicate with another computer on the same network, it needs to know the other computer’s hardware (MAC) address. ARP helps by matching the IP address to the MAC address.

### **How ARP Works:**

1. **Computer A wants to talk to Computer B**.
2. **Computer A sends an ARP request**: "Who has this IP address?"
3. **Computer B responds**: "I have that IP address, and here’s my MAC address."
4. **Computer A stores this info** in its ARP cache for future use.

### **What is ARP Cache Poisoning?**

ARP cache poisoning (also called ARP spoofing) is a trick that a malicious person can use to mess with the network communication.

### **How ARP Cache Poisoning Works:**

1. **Malicious Computer sends fake ARP messages** to other computers on the network.
2. **These fake messages** say something like, "Hey, I am Computer B" with a false MAC address.
3. **Other computers update their ARP caches** with the fake information.
4. **Now, when these computers want to talk to Computer B**, they mistakenly send the data to the Malicious Computer instead.

## **What is ICMP?**

ICMP (Internet Control Message Protocol) is a network protocol used by devices to send error messages and operational information, like indicating that a service is not available or that a route should be used to reach a destination.

### **What is an ICMP Redirect?**

An ICMP redirect message is a specific type of ICMP message used by routers to inform hosts (computers or devices) that there is a better route available for a particular destination. This is intended to optimize network routing.

### **How an ICMP Redirect Attack Works:**

1. **Normal Behavior**:
   * A host sends data to a router to be forwarded to the destination.
   * If the router knows of a better route, it sends an ICMP redirect message to the host.
   * The host updates its routing table to use the new route for future data to that destination.
2. **Attack Scenario**:
   * An attacker sends a fake ICMP redirect message to a host on the network.
   * This fake message tells the host that the best route to a particular destination is through a router controlled by the attacker.
   * The host updates its routing table with the attacker’s suggested route.
   * Future traffic from the host to that destination is sent to the attacker’s router instead of the legitimate one.

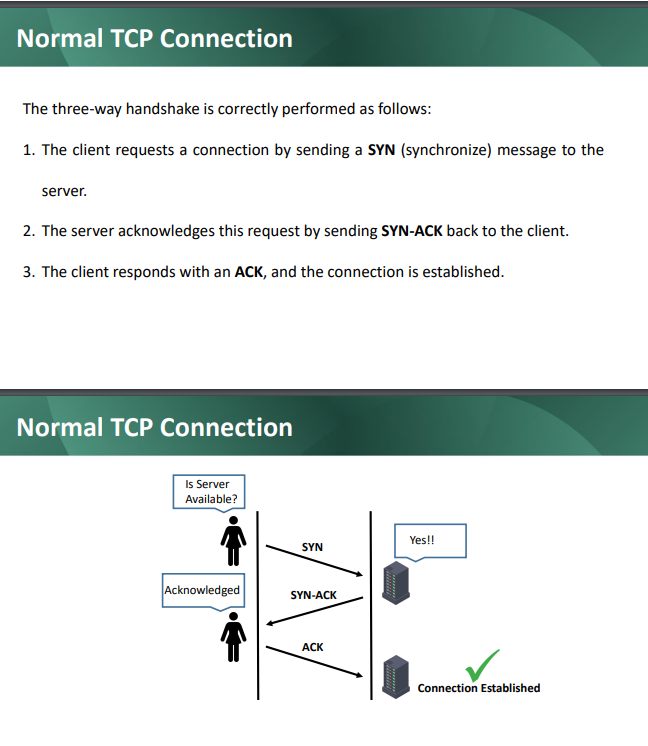
## **Smurf Attack**

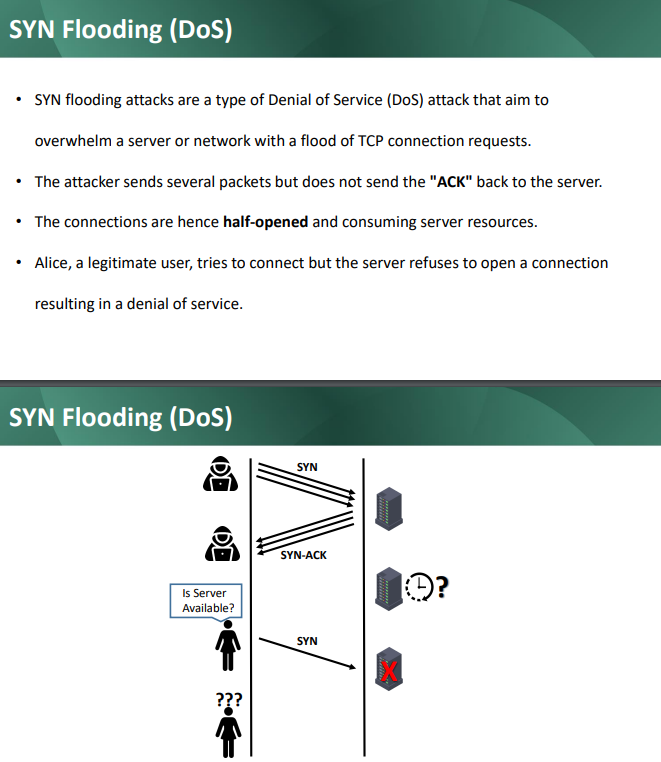
### **How a Smurf Attack Works:**

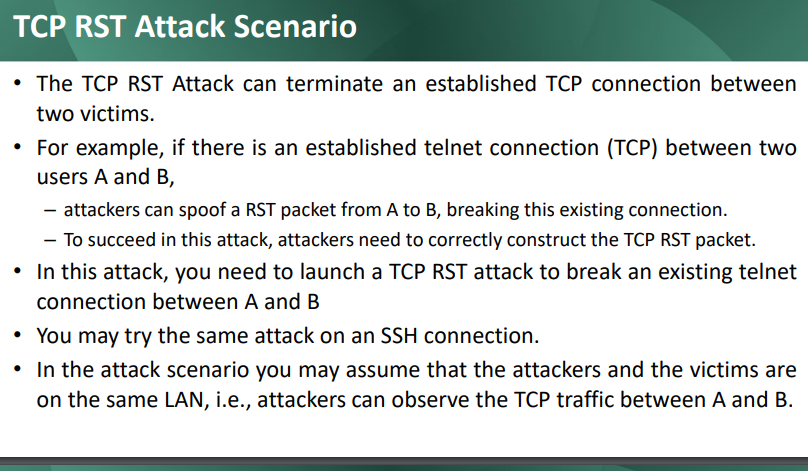
1. **Spoofing the Source IP**:
   * The attacker spoofs (fakes) the IP address of the target system, making it appear as if the ICMP echo request (ping) messages are coming from the target’s IP address.
2. **Broadcasting the Ping**:
   * The attacker sends the ICMP echo request to an IP broadcast address, which means that all devices on the network will receive the ping request.
3. **Amplification**:
   * Every device on the network responds to the ping request, sending an ICMP echo reply back to the spoofed IP address (the target system).
4. **Overwhelming the Target**:
   * The target system is overwhelmed with a flood of ICMP echo replies from all the devices on the network, consuming its bandwidth and processing resources, potentially causing it to crash or become unresponsive.

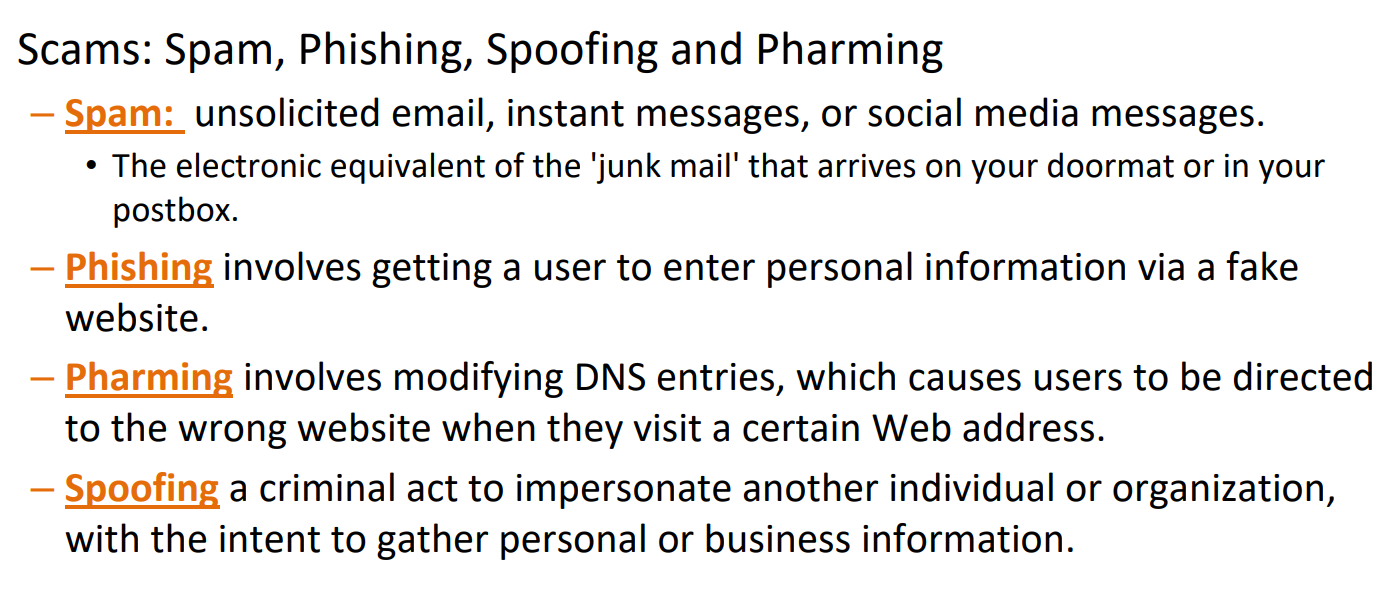
## **Ping of death**

Ping of death: is a type of attack on a computer system that involves sending a malformed or otherwise malicious ping to a computer. When the target computer reassembles the malformed packet, a buffer overflow can occur, causing a system crash and potentially allowing the injection of malicious code.

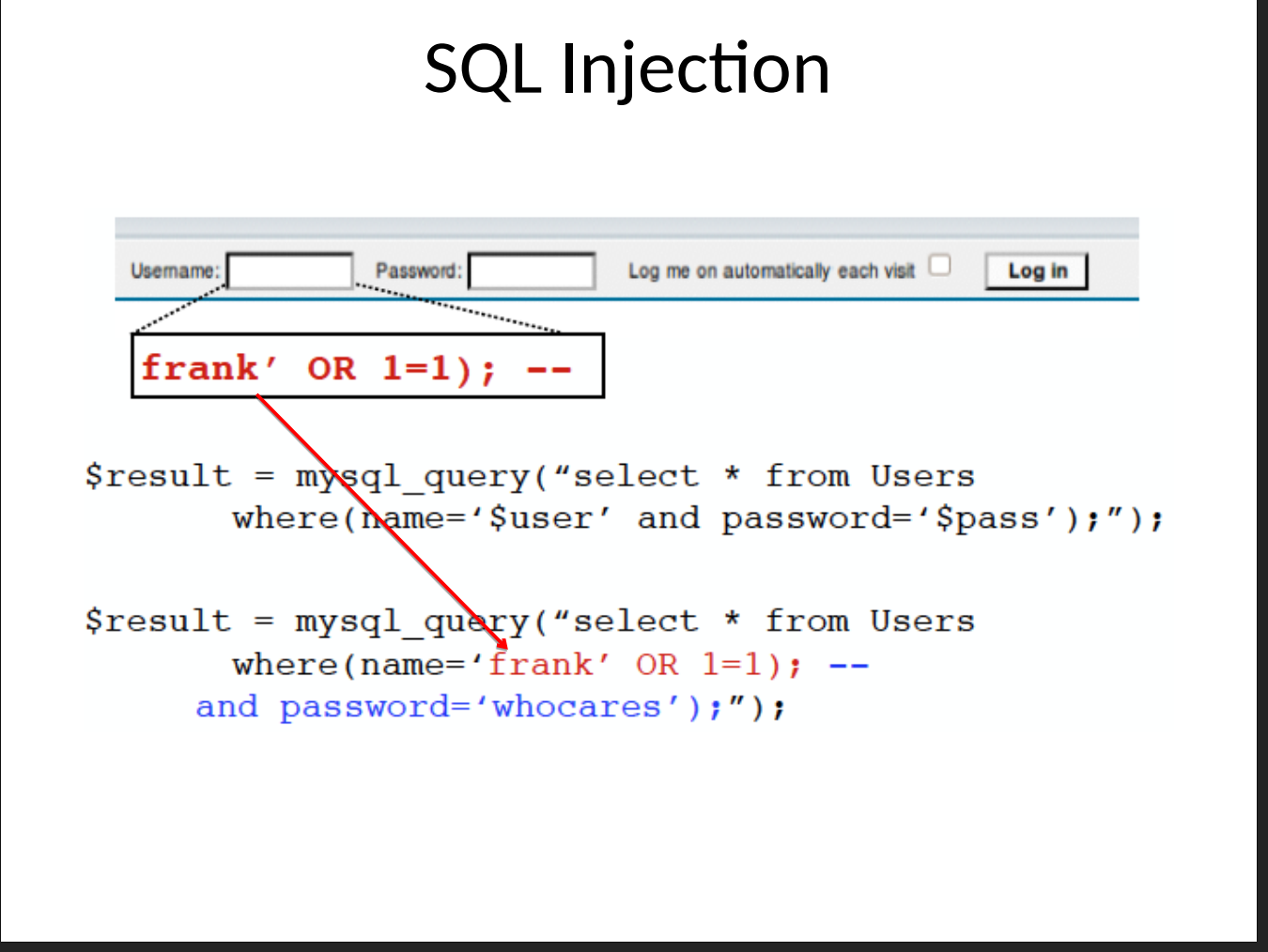


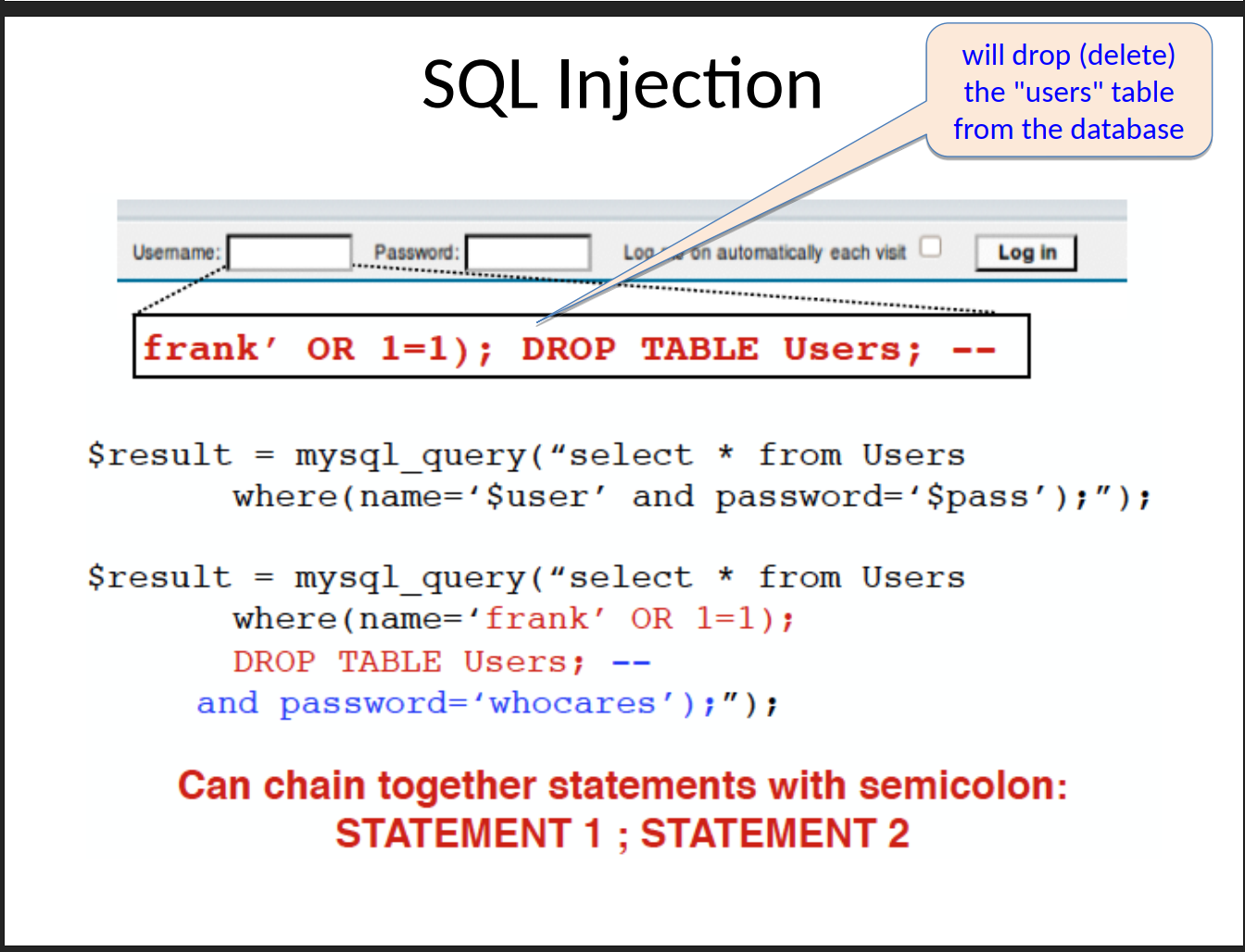




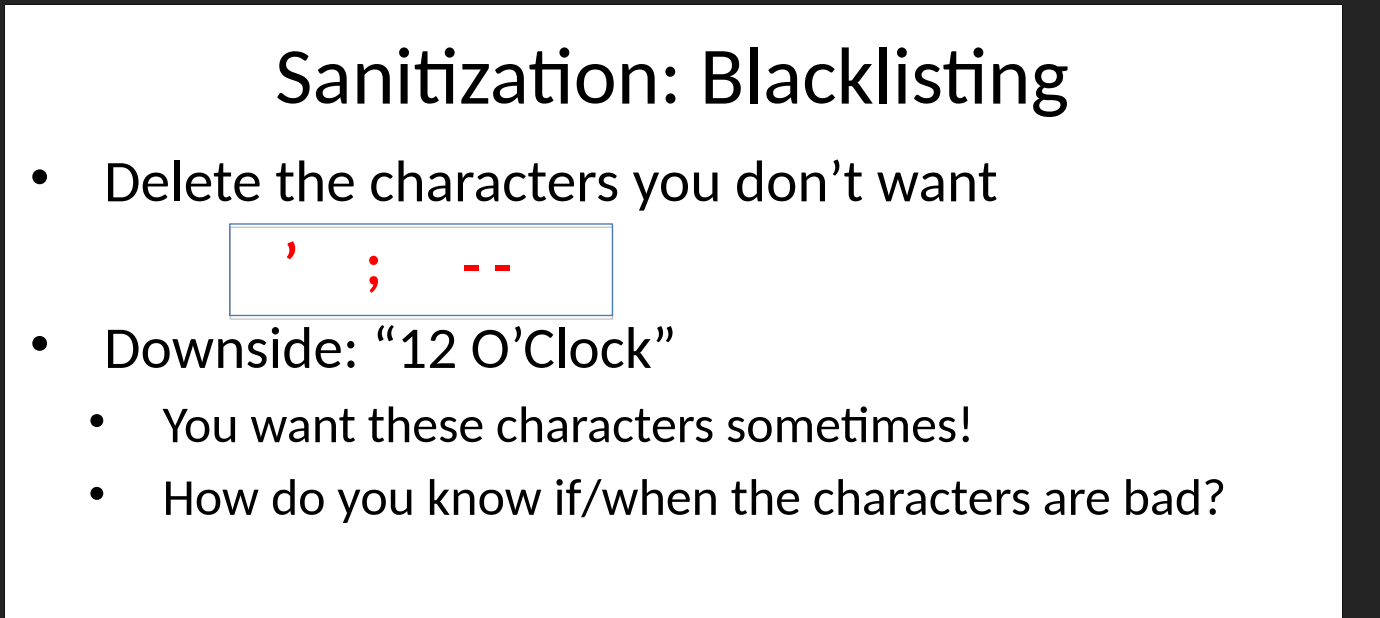


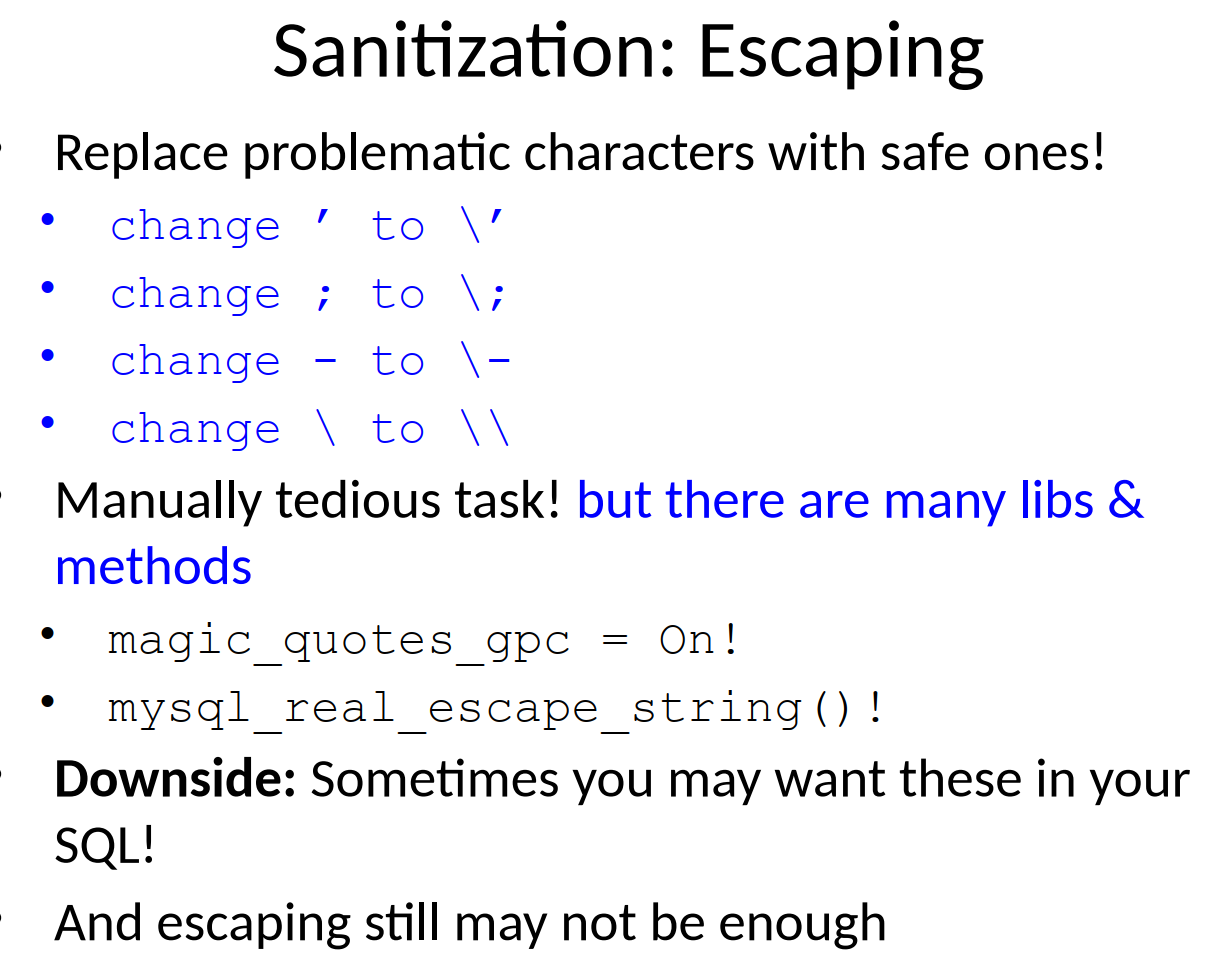
**SQL Injection**

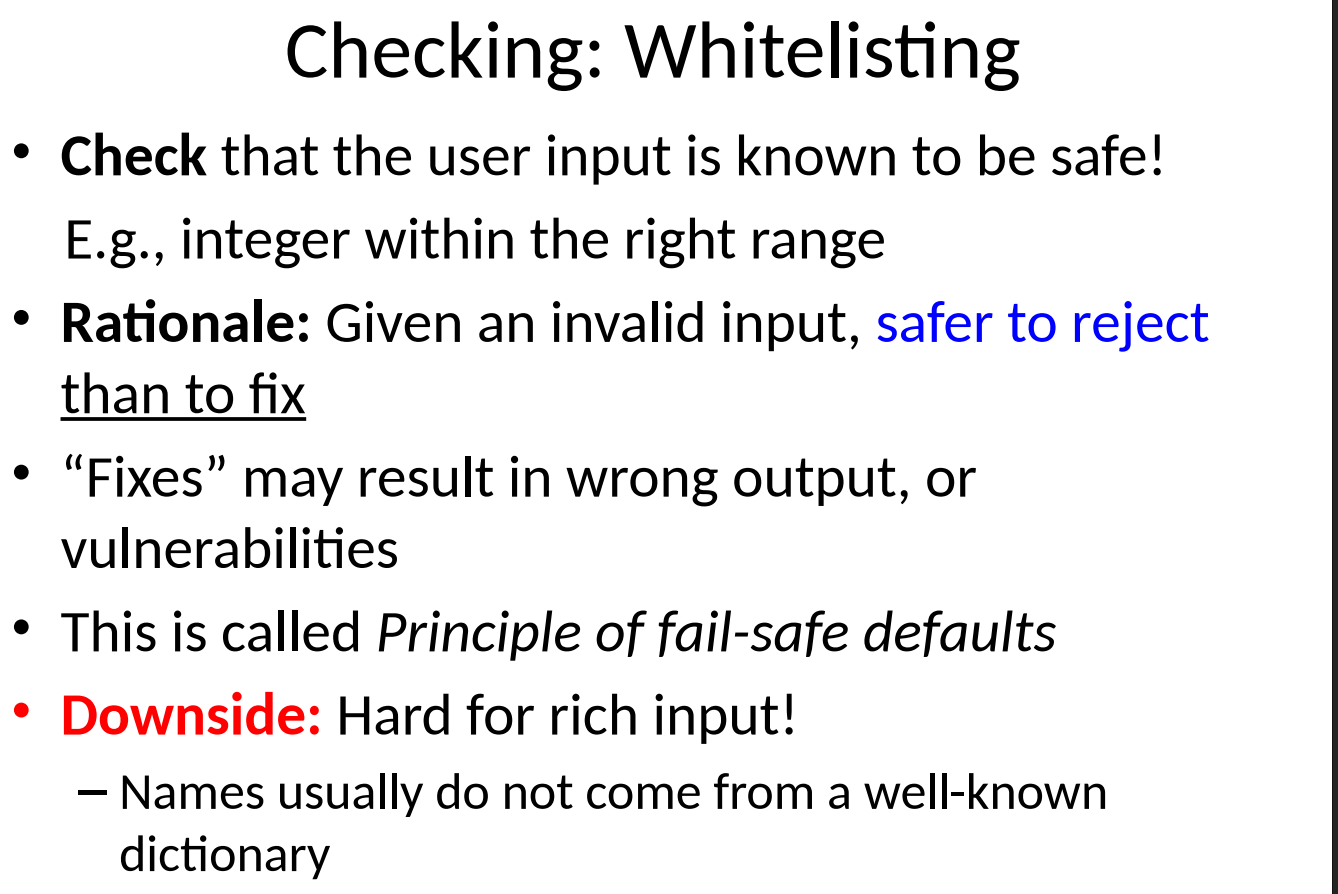
****

****

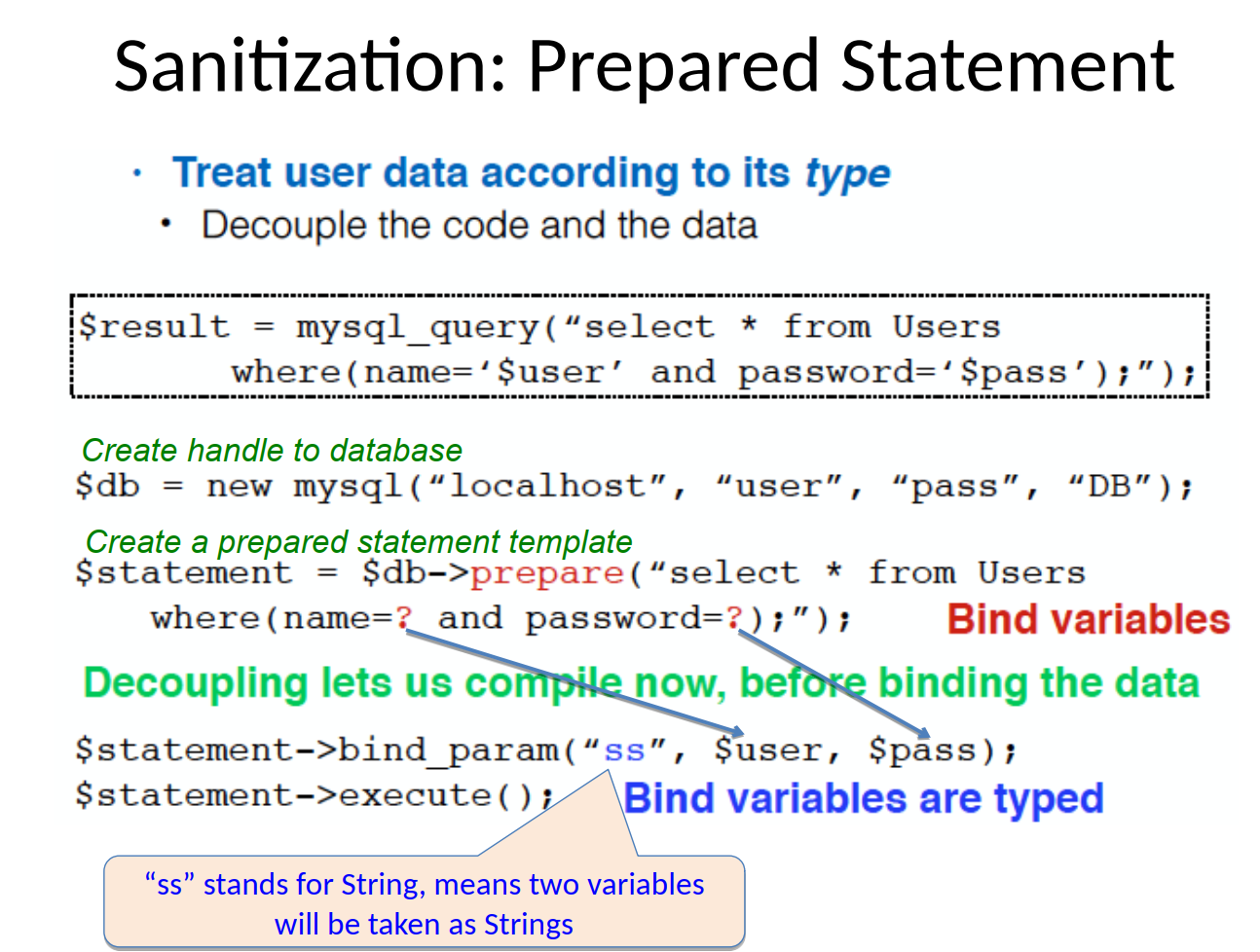
**Countermeasure for SQL Injection**

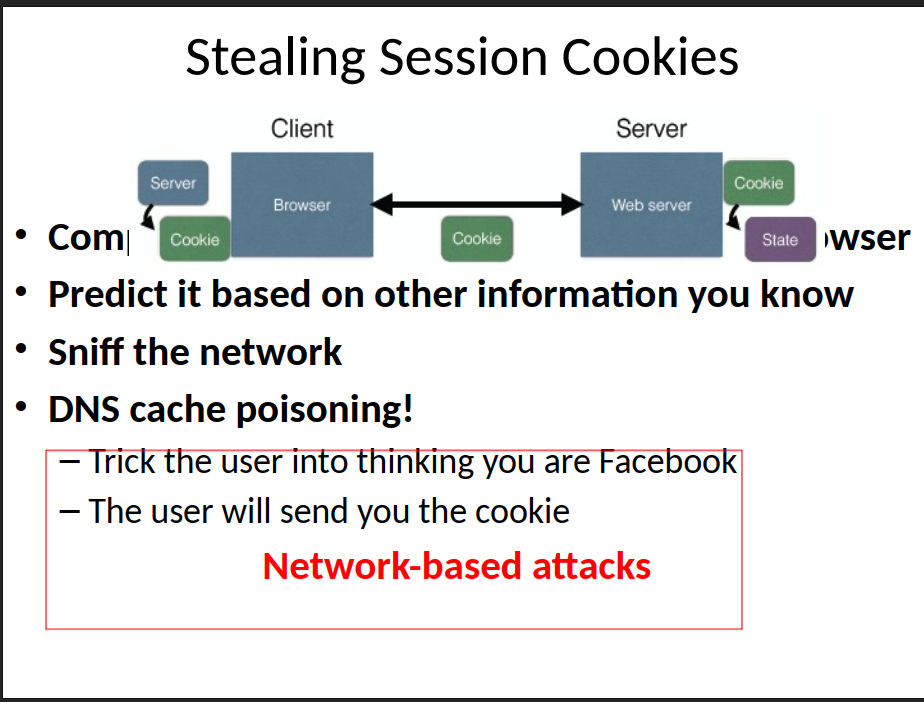
****

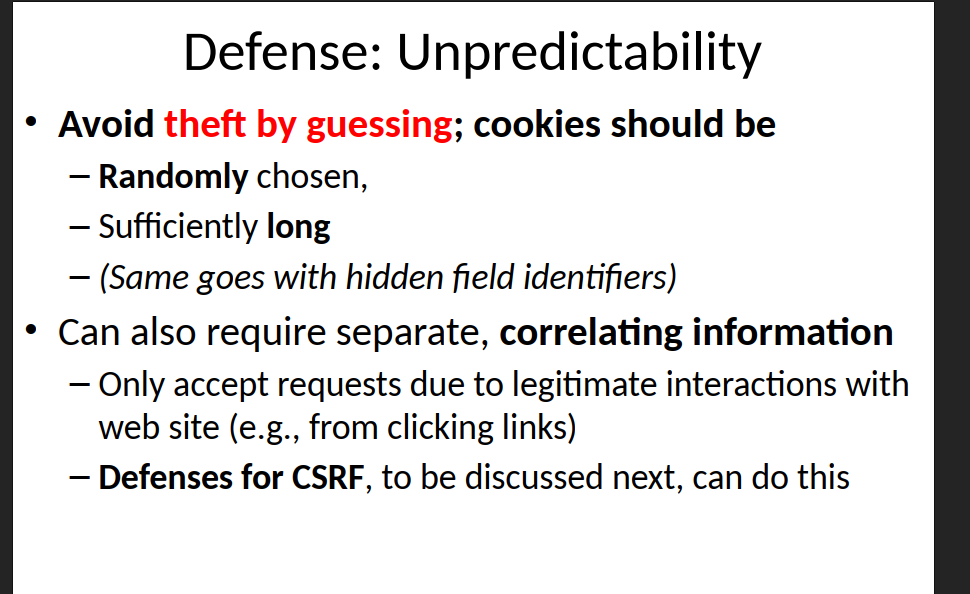
****

****

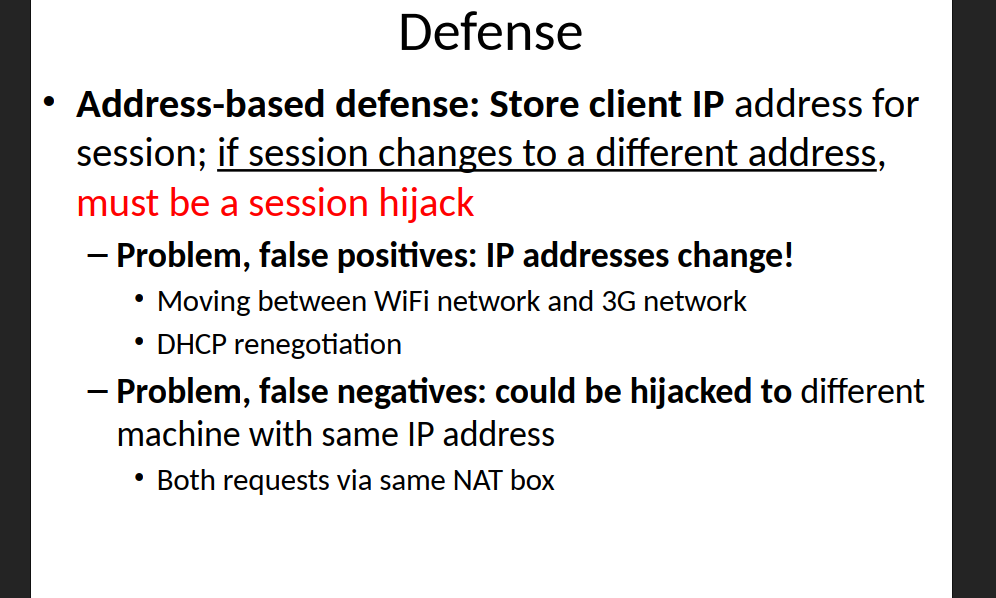
**Preferred Method**

****

****

****

****

****