

Prepared by Dr. Nermin Hamza

CYB740

Penetration test and Ethical hacking

Lecture 08



Session Hijacking

Lecture 08

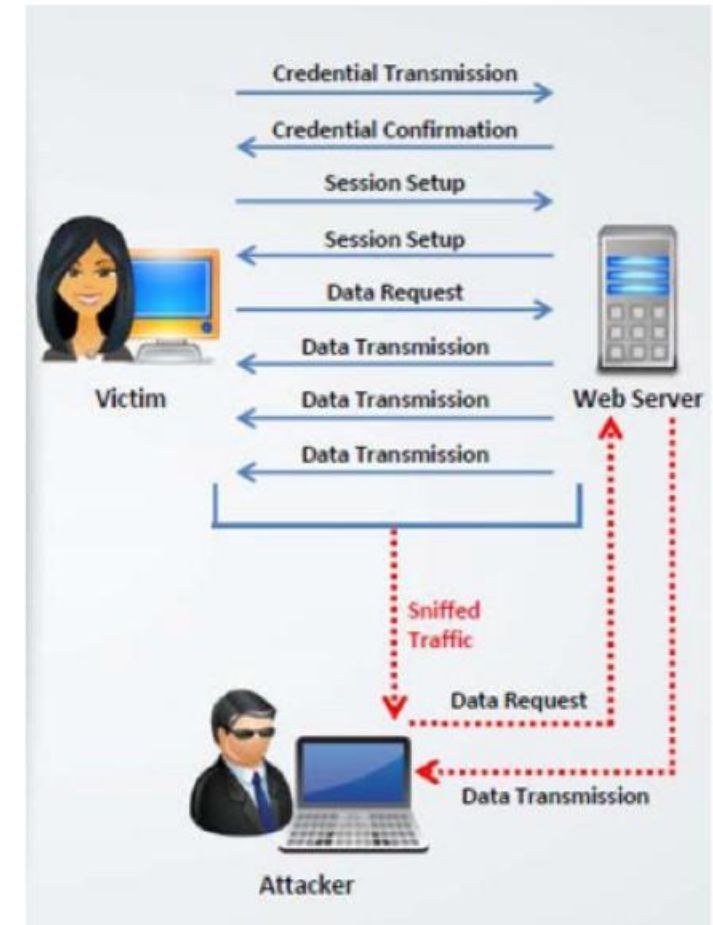
CYB740

Penetration test and Ethical hacking

Prepared by
Dr. Nermin Hamza

- What is Session Hijacking?
- Types of Session Hijacking
- Session Hijacking Process
- Why Session Hijacking is Successful?
- Spoofing vs. Hijacking
- What Do Attackers Gain from Session Hijacking?
- Application level session hijack
- Network level session hijack
- Countermeasures and Prevention

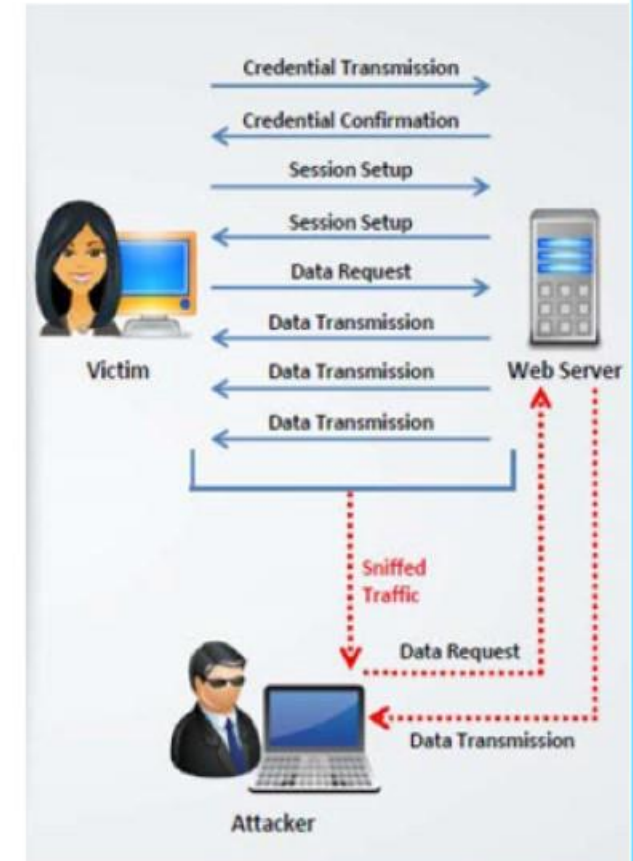
What is Session Hijacking?



What is Session Hijacking?

- Session hijacking refers to an attack where an attacker takes over a valid TCP communication session between two computers.
- Since most authentication only occurs at the start of a TCP session, this allows the attacker to gain access to a machine.

- Attackers can sniff all the traffic from the established TCP sessions and perform identity theft, information theft, fraud, etc.
- The attacker steals a valid session ID and use it to authenticate himself with the server.



Types of Session Hijacking

- **Active Attack:** In an active attack, an attacker finds an active session and takes over.
- **Passive Attack:** With a passive attack, an attacker hijacks a session but sits back and watches and records all the traffic that is being sent forth.

The essential difference between an active and passive hijacking is that while an active attack takes over an existing session, a passive hijack monitors an ongoing session.

Session Hijacking Process

- **Stealing:** The attacker uses different techniques to steal session IDs.
 - Some of the techniques used to steal session IDs:
 1. Using the HTTP referrer header.
 2. Sniffing the network traffic.
 3. Using the cross-site-scripting attacks.
 4. Sending Trojans on client machines.

Session Hijacking Process

- **Guessing:** The attacker tries to guess the session IDs by observing variable parts of the session IDs.
 - `http://www.hacksite.com/view/VW48266762824302`
 - `http://www.hacksite.com/view/VW48266762826502`
 - `http://www.hacksite.com/view/VW48266762828902`

- **Brute Forcing:** The attacker attempts different IDs until he succeeds.
 - Using **brute force attacks**, an attacker tries to guess a **session ID** until he finds the correct session ID.
- **Note:** Session ID brute forcing attack is known as session prediction attack if the predicted range of values for a session ID is very small.

Why Session Hijacking is Successful?

- No account lockout for invalid session IDs.
- Weak session ID generation algorithm or small session IDs.
- Insecure handling of session IDs.
- DNS poisoning, XSS, exploiting a bug in browser
- Indefinite session expiration time.
- Most computers using TCP/IP are vulnerable.

Spoofing vs. Hijacking

- **Spoofing Attack:**

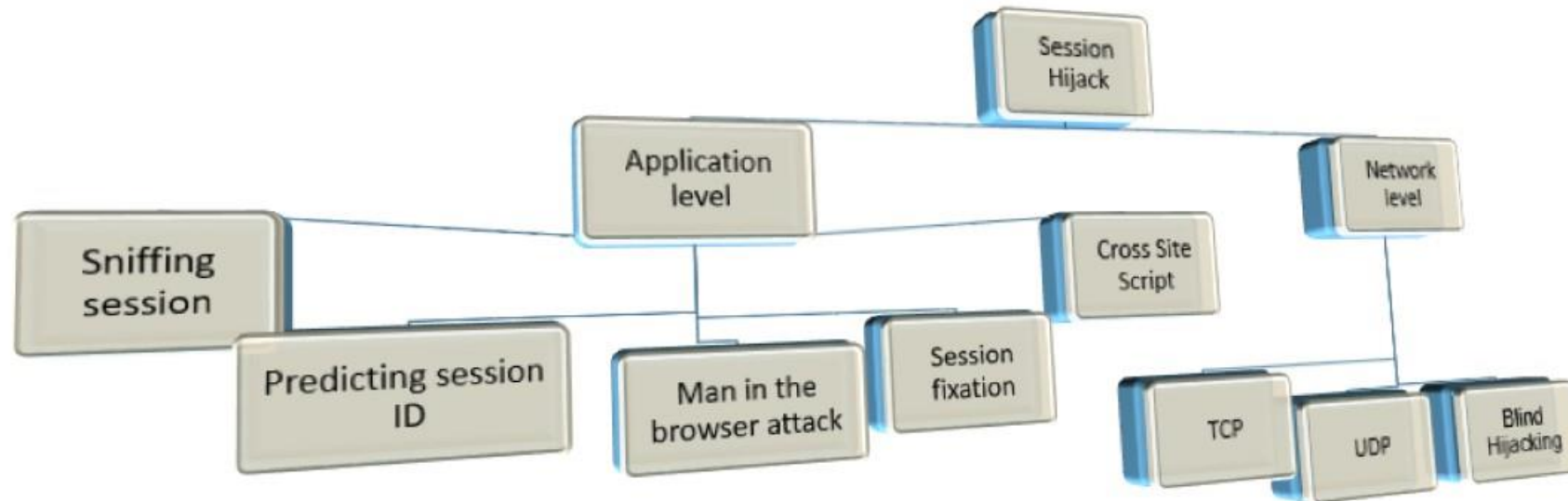
- Attack **pretends to be another user** or machine (victim) to gain access.
- Attacker does not take over an existing active session. Instead he initiates a new session using the victim's **stolen credentials**.

- **Hijacking:**

- Session hijacking is the process of taking over an **existing active session**.
- Attacker relies on the **legitimate user** to make a connection and authenticate.

What Do Attackers Gain from Session Hijacking?

- When cybercriminals have hijacked a session, they can do virtually anything that the legitimate user was authorized to do during the active session.
- The most severe examples include transferring money from the user's bank account, buying goods from web stores, accessing personally identifiable information (PII) for identity theft, and even stealing data from company systems.



Application-level session Hijacking

Application level :

Here the valid session token is stolen or predicted to take over the session.

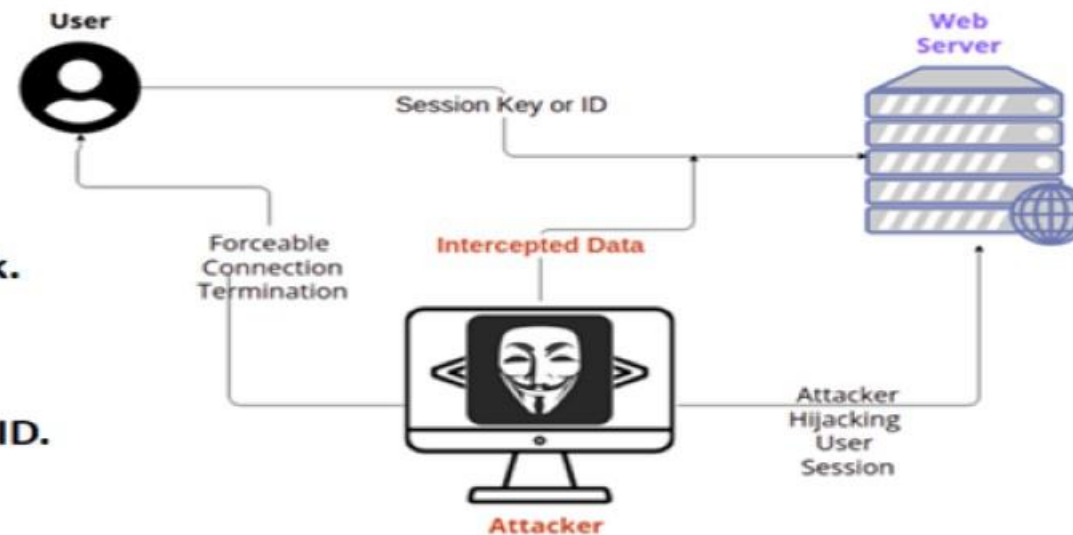
Session sniffing.

Man-in-the-browser attack.

Cross-site scripting.

Predictable sessions token ID.

Session fixation attacks.



Methods of session hijacking

- **Session sniffing:** This is one of the most basic techniques used.
- The attacker uses a sniffer, such as Wireshark, or a proxy, such as OWASP Zed, to capture network traffic containing the session ID between a website and a client.
- Attacker uses a sniffer to **capture a valid session token** or **session ID**.
- Once the attacker captures this value, he can use this valid token to gain unauthorized access.

Methods of session hijacking

- **Predictable sessions token ID.** Many web servers use a custom algorithm or predefined pattern to generate session IDs.
- The greater the predictability of a session token, the weaker it is and the easier it is to predict.
- If the attacker can capture several IDs and analyze the pattern, he may be able to predict a valid session ID.

- Attackers perform analysis of variable section of session IDs to determine the existence of a pattern.
- The analysis is performed manually or by using various cryptanalytic tools.
- Attackers collect a high number of simultaneous session IDs in order to gather samples in the same time window and keep the variable constant.

- **How to Predict a Session Token**
- **Captures:** Attacker captures several session IDs and analyzes the pattern.
 - <http://www.juggyboy.com/view/JBEX25022014152820>
 - <http://www.juggyboy.com/view/JBEX25022014153020>
 - <http://www.juggyboy.com/view/JBEX25022014160020>
 - <http://www.juggyboy.com/view/JBEX25022014164020>
- **Predicts:** At 16:25:55 on Feb-25, 2014, the attacker can successfully predict the session ID to be <http://www.juggyboy.com/view/JBEX25022014162555>
 - JBEX: Constant
 - 25022014: Date
 - 162555: Time

Methods of session hijacking

- **Session fixation attacks.** This technique steals a valid session ID that has yet to be authenticated.
- Then, the attacker tries to trick the user into authenticating with this ID.
- Once authenticated, the attacker now has access to the victim's computer. Session fixation explores a limitation in the way the web application manages a session ID.

Session Fixation Attack

- Attacker exploits the vulnerability of a server which allows a user to use fixed session ID.
- Attacker provides a valid session ID to a victim and lures him to authenticate himself using that session ID.

- Several techniques to **execute Session Fixation** attack are:
 - Session token in the **URL argument**
 - Session token in a **hidden form field**
 - Session ID in a **cookie**

Methods of session hijacking

- **Man-in-the-browser attack.** . Once the victim is tricked into installing malware onto the system, the malware waits for the victim to visit a targeted site.
- The man-in-the-browser malware can invisibly modify transaction information and it can also create additional transactions without the user knowing. Because the requests are initiated from the victim's computer, it is very difficult for the web service to detect that the requests are fake.

- Man-in-the-browser attack **uses a Trojan Horse** to intercept the calls between the browser and its security mechanisms or libraries.
- It works with an already installed Trojan horse and acts between the **browser and its security mechanisms**.
- Its main objective is to cause financial deceptions by manipulating transactions of **Internet Banking systems**.

Steps to Perform Man-in-the-Browser Attack

1. The Trojan first infects the **computer's software** (OS or application).
2. The Trojan installs malicious code (extension files) and saves it into the **browser configuration**.
3. After the user restarts the browser, the **malicious code** in the form of extension files is loaded.
4. The **extension files** register a handler for every visit to the webpage.
5. When the page is loaded, the extension uses the **URL** and matches it with a **list of known sites** targeted for attack.
6. The user logs in **securely** to the website.
7. It registers a **button event handler** when a specific page load is detected for a specific pattern and compares it with its targeted list.

8. When the user clicks on the button, the extension extracts all the data from all form fields and modifies the values.
9. The browser sends the **form** and **modified values** to the server.
10. The server receives the **modified values** but cannot distinguish between the original and the modified values.
11. After the server performs the transaction, a **receipt is generated**.
12. Now, the browser receives the receipt for the **modified transaction**.
13. The browser displays the receipt with the **original details**.
14. The user thinks that the **original transaction** was received by the server without any interceptions.

Methods of session hijacking

- **Cross-site scripting.** Cybercriminals exploit server or application vulnerabilities to inject client-side scripts into web pages. This causes the browser to execute arbitrary code when it loads a compromised page. If Http Only isn't set in session cookies, cybercriminals can gain access to the session key through injected scripts, giving them the information, they need for session hijacking.

- **Cross-Site Scripting (XSS)**: XSS enables attackers to **inject malicious client side scripts** into the web pages viewed by other users.
- If an attacker sends a **crafted link** to the victim with the **malicious JavaScript**, when the victim clicks on the link, the JavaScript will run and complete the instructions made by the attacker.

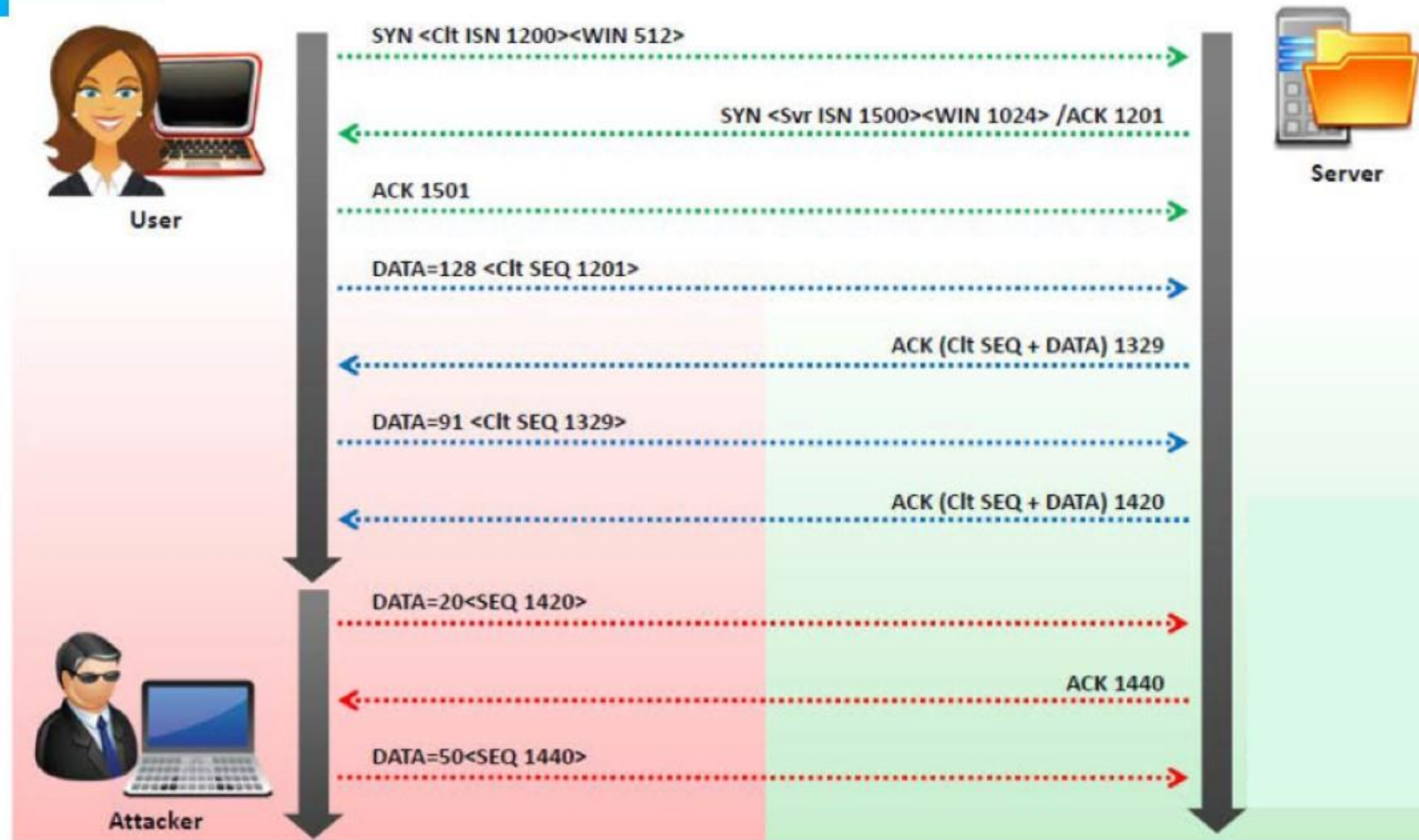
Network-Level session Hijacking

Network-level Session Hijacking

- The network-level hijacking relies on hijacking **transport** and **Internet protocols** used by web applications in the application layer.
- By attacking the network-level sessions, the attacker gathers some **critical information** which is used to **attack the application level**.
- Network-level hijacking includes:
 - Blind Hijacking
 - UDP Hijacking
 - TCP/IP Hijacking

TCP/IP Hijacking

- TCP/IP hijacking is a hacking technique that uses **spoofed packets** to take over a connection between a victim and a target machine.
- The victim's connection hangs and the attacker is then able to **communicate with the host's machine** as if the attacker is the victim.
- To launch a TCP/IP hijacking attack, the **attacker must be on the same network as the victim**.
- The target and the victim machines can be anywhere.



The 3-Way Handshake

- If the attacker can anticipate the **next sequence** and **ACK number** that the User will send, he/she will spoof User's address and start a communication with the server.
1. Bob initiates a connection with the server and sends a packet to the server with the **SYN flag set**
 2. The server receives this packet and sends back a packet with the **SYN + ACK flag** and an **ISN (Initial Sequence Number)** for the server
 3. Bob sets the **ACK flag** acknowledging the receipt of the packet and increments the sequence number by 1
 4. Now, the two machines successfully **established a session**

- According to the diagram, the next expected sequence number would be 1420. If you can transmit that packet sequence number before the user does, you can desynchronize the connection between the user and the server.
- After establishing the connection between the attacker and the server, though the user sends the data with the correct sequence number, the server drops the data considering it as a resent packet.
 - Note: Before the user could send the next data packet, attacker predicts the next sequence number and sends the data to the server. This leads to establishment of connection between attacker and the server.
 - To conduct a session hijacking attack, the attacker performs three activities:
 - Tracks a session
 - Desynchronizes the session
 - Injects attacker's commands in between

UDP Hijacking

- A network-level session hijacking where the attacker sends **forged server reply** to a victim's UDP request before the intended server replies to it.
- The attacker uses **man-in-the-middle** attack to intercept server's response to the client and sends its own forged reply.

Note : UDP does not use packet sequencing and synchronizing.

Blind Hijacking

- The attacker can inject the **malicious data or commands** into the intercepted communications in the TCP session even if the source-routing is disabled.
- The attacker can send the data or commands but has no **access to see the response**.

Countermeasures

Countermeasures Session Hijacking Detection Methods

- Manual Method
 - Using Packet Sniffing Software
 - Normal Telnet Session
 - Forcing an ARP Entry
- Automatic Method
 - Intrusion Detection Systems (IDS)
 - Intrusion Prevention Systems (IPS)

Protecting against Session Hijacking

- Use **Secure Shell (SSH)** to create a secure communication channel.
- Pass the **authentication cookies** over HTTPS connection.
- Implement the **log-out functionality** for user to end the session.
- Generate the **session ID** after successful login and accept sessions IDs generated by server only.
- Ensure data in transit is **encrypted** and implement **defense-in-depth** mechanism.
- Use **string** or **long random number** as a session key.
- Use different **user name** and **passwords** for different accounts.

- Implement **timeout()** to destroy the session when expired.
- Do not transport session ID in **query string**.
- Ensure **client-side** and **server-side** protection software are in active state and up to date.
- Use strong **authentication** (like Kerberos) or peer-to-peer VPN's.
- Configure the appropriate **internal** and **external spoof rules** on gateways.

Methods to Prevent Session Hijacking: To be Followed by Web Developers

- Create session IDs with lengthy strings or random number so that it is difficult for an attacker to guess a valid session IDs.
- Regenerate the session ID after a successful login to prevent session fixation attack.
- Encrypt the data and session key that is transferred between the user and the web servers.
- Expire the session as soon as the user logs out.
- Reduce the life span of a session or a cookie.

Methods to Prevent Session Hijacking: To be Followed by Web Users

- Do not click on the links that are received through mails.
- Use Firewalls to prevent the malicious content from entering the network.
- Use firewall and browser settings to restrict cookies.
- Make sure that the website is certified by the certifying authorities.
- Make sure you clear history, offline content, and cookies from your browser after every confidential and sensitive transaction.
- Prefer https, a secure transmission, rather than http when transmitting sensitive and confidential data.
- Logout from the browser by clicking on logout button instead of closing the browser.

Approaches Vulnerable to Session Hijacking and their Preventative Solutions

Issue	Solution	Notes
Telnet, rlogin	OpenSSH or ssh (Secure Shell)	It sends encrypted data and makes it difficult for attacker to send the correctly encrypted data if session is hijacked
FTP	sFTP	It reduces the chances of successful hijacking
HTTP	SSL (Secure Socket Layer)	It reduces the chances of successful hijacking
IP	IPSec	It prevents hijacking by securing IP communications

Approaches Vulnerable to Session Hijacking and their Preventative Solutions

Any Remote Connection	VPN	Implementing encrypted VPN such as PPTP, L2PT, IPSec, etc. for remote connection prevents session hijacking
SMB (Server Message Block)	SMB signing	It improves the security of the SMB protocol and reduces the chances of session hijacking
Hub Network	Switch Network	It mitigates the risk of ARP spoofing and other session hijacking attacks

Thank You