Session hijacking

Chapter 3

By: Ismaeel Ahmad&L& azhy Adnan

Supervisor and Editor: bilal H Ali

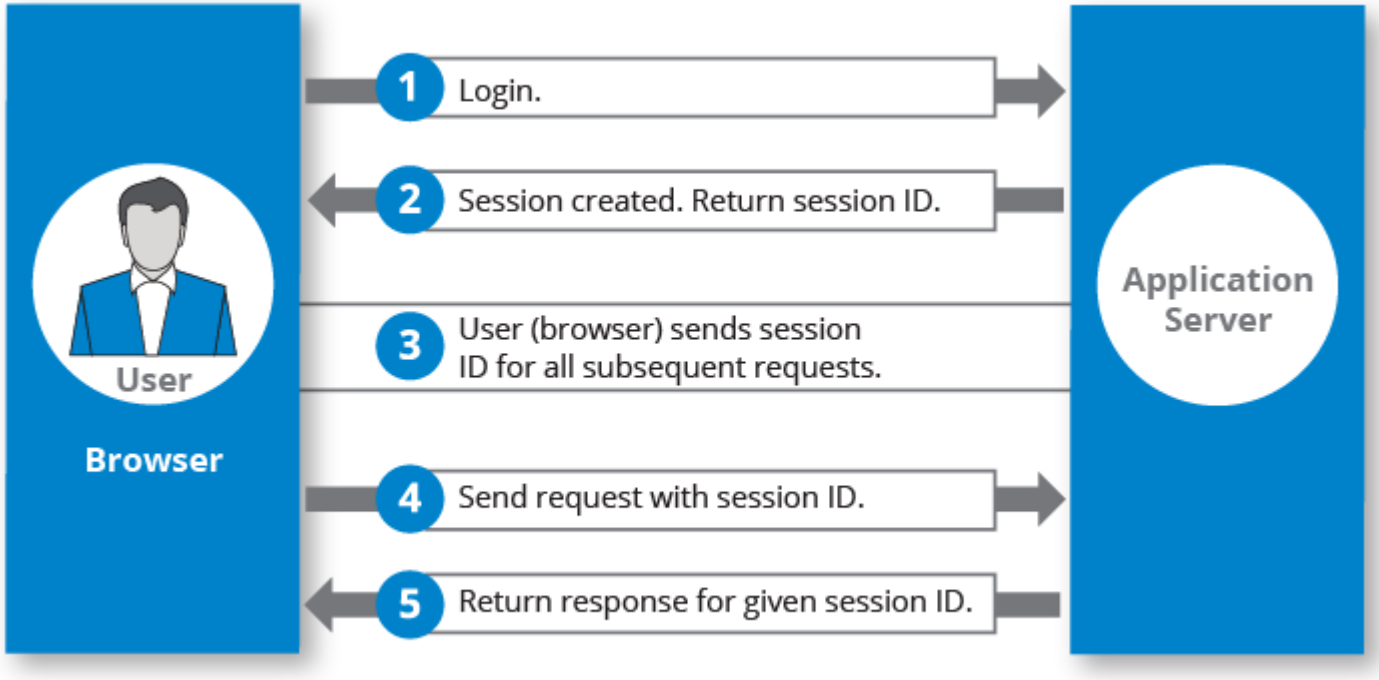
Outline

- What is Session ?
- What is session ID
- What is session hijacking ?
- Example of session hijacking
- How does session hijacking work ?
- Types of session hijacking
- Types of session hijacking methods
- Session hijacking tools
- How to prevent Session Hijacking
-

session

- In telecommunication, a session is a series of interactions between two communication end points that occur during the span of a single connection. Typically, one end point requests a connection with another specified end point and if that end point replies agreeing to the connection, the end points take turns exchanging commands and data ("talking to each other"). The session begins when the connection is established at both ends and terminates when the connection is ended.
- A web session is a series of contiguous actions by a visitor on an individual website within a given time frame. This could include your search engine searches, filling out a form to receive content, scrolling on a website page, adding items to a shopping cart, researching airfare, or which pages you viewed on a single website. Any interaction that you have with a single website is recorded as a web session to that website property.

session


User
Browser
1 Login.
2 Session created. Return session ID.
3 User (browser) sends session ID for all subsequent requests.
4 Send request with session ID.
5 Return response for given session ID.
Application Server

session

Why is a web session used? (Web session use case examples)

- To avoid storing massive amounts of information in-browser, developers use session IDs to store information server-side while enabling user privacy. Every time a user takes an action or makes a request on a web application, the application sends the session ID and cookie ID back to the server, along with a description of the action itself.

session

What is Session ID?

Typically, the process of managing the state of a web-based client is through the use of session IDs. Session IDs are used by the application to uniquely identify a client browser, while background (server-side) processes are used to associate the session ID with a level of access. Organisations application developers have three methods available to them to both allocate and receive session ID information:

- Session ID information embedded in the URL, which is received by the application through HTTP GET requests when the client clicks on links embedded with a page.

Example: http://www.example.com/news.asp?article=27781;sessionid=IE60012219

- Session ID information stored within the fields of a form and submitted to the application. Typically the session ID information would be embedded within the form as a hidden field and submitted with the HTTP POST command.

Example: Embedded within the HTML of a page –

<FORM METHOD=POST ACTION=”/cgi-bin/news.pl”>

<INPUT TYPE=”hidden” NAME=”sessionid” VALUE=”IE60012219”>

<INPUT TYPE=”hidden” NAME=”allowed” VALUE=”true”>

<INPUT TYPE=”submit” NAME=”Read News Article”>

- Through the use of cookies. Example: Within the plain text of the HTTP server response –

  Set-Cookie: sessionID=”**IE60012219**”; path=”/”; domain=”www.example.com”; expires=”2003-06-01 00:00:00GMT”; version=0

Session ID

The two critical characteristics of a good session ID:

- Session ID Randomness

It is important that the session ID is unpredictable and the application utilises a strong method of generating random ID's. It is vital that a cryptographically strong algorithm is used to generate a unique session ID for an authenticated user. Ideally the session ID should be a random value.

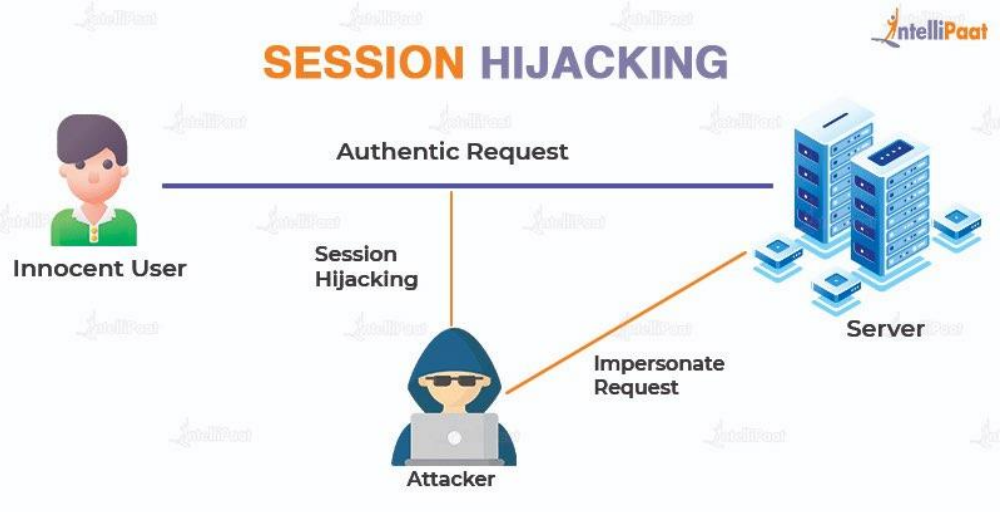
- Session ID Length

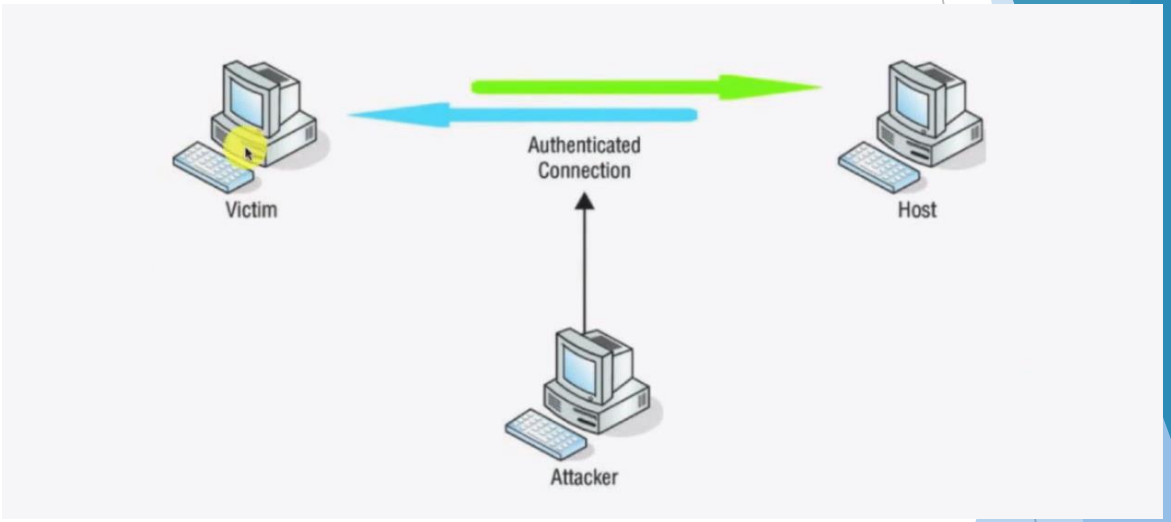
It is important that the session ID be of a sufficient length to make it infeasible that a brute force method could be used to successfully derive a valid ID within a usable timeframe. Given current processor and bandwidth limitations, session ID's consisting of over 50 random characters in length are recommended – but make them longer if the opportunity exists.

What is session hijacking?

- Session hijacking (also known as Cookie hijacking or Cookie side-jacking) is one of the most sophisticated man-in-the-middle attacks which gives the attacker access to the victim’s web sessions. It also refers to the attacker’s ability to take control over a portion of the user’s session. This process would provide them access to sensitive data such as personal and financial data (PII and PCI) that might be protected using a passkey or passphrase.

session


IntelliPaat
SESSION HIJACKING
Authentic Request
Innocent User
Session
Hijacking
Server
Impersonate
Request
Attacker


Victim
Authenticated
Connection
Host
Attacker

Example of Session Hijacking

- A **session attack** takes advantage of data leaks in the **compression ratio of TLS requests**. This then gives them access to users' login cookies which can be used to hijack the users session. One such incident occurred in September, 2012, when an organization of session hijackers called CRIME breached an organization's website.
- CRIME ended up hijacking the session by **decrypting HTTPS cookies** set by the website and authenticated themselves as users by brute force, siphoning a considerable amount of data.

Session Hijacking Phases

1. Sniffing
2. Monitoring
3. Session failure
4. Prediction of session ID
5. Forcing the target to go offline
6. Injection code

Session Hijacking Phases

1. Sniffing: The first step is to find an active session between the user and the server, and try to stand between them. Hackers use sniffing tools like Wireshark to find session information and capture traffic
2. Monitoring: The session is then monitored for vulnerabilities and protocols that can be exploited. Hackers also look for the valid authentication packets flowing between the user and the server.
3. Session failure(Retrieval)

Hackers then use the available data and information to find the valid session ID. They try to predict the sequence number that can enable access to the session. It is a crucial step because if the incorrect sequence number is used, the server may reset the session or terminate the attempt.

Session Hijacking Phases (process)

4. Prediction of session ID(Stealing session ID)

Man in the middle attack, cross-site scripting, brute force attacks, etc. is used to steal the session IDs.

5. Forcing the target to go offline: After predicting the session ID, the hackers launch a DoS attack to force the user to go offline. It is important for the attacker to ensure that the user is offline because if a session is accessed by two parties, it can cause an ACK storm.

6. Hijacking

It is the final stage where the hackers take over the session between the user and the server. Here, they will also spoof the IP address to appear legitimate to the server.

Session Hijacking Types

1. **Active Session Hijacking** : An Active Session Hijacking occurs when the attacker takes control over the active session. The actual user of the network becomes in offline mode, and the attacker acts as the authorized user. They can also take control over the communication between the client and the server. To cause an interrupt in the communication between client and server, the attackers send massive traffic to attack a valid session and cause a denial of service attack(**DoS**).

2. **Passive Session Hijacking** : In Passive Session Hijacking, instead of controlling the overall session of a network of targeted user, the attacker monitors the communication between a user and a server. The main motive of the hacker is to listen to all the data and record it for the future use. Basically, it steals the exchanged information and use for irrelevant activity. This is also a kind of **man-in-middle** attack (as the attacker is in between the client and the server exchanging information.

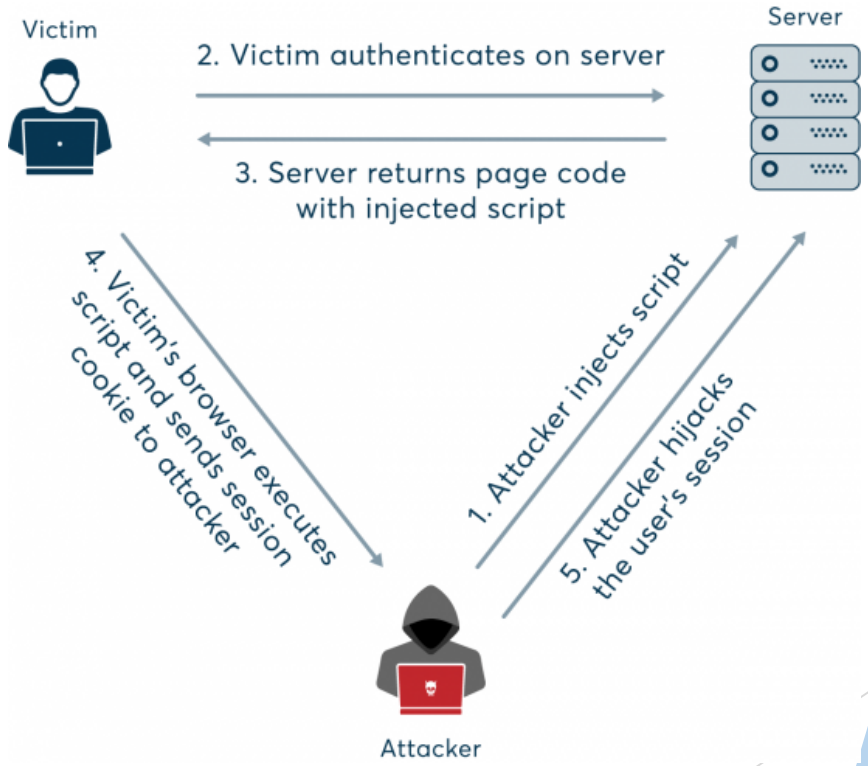
3. **Hybrid Hijacking** : The combination of Active Session Hijacking and Passive Session Hijacking is referred to as Hybrid Hijacking. In this the attackers monitors the communication channel (the network traffic), whenever they find the issue, they take over the control on the web session and fulfill their malicious tasks.

Session Hijacking Types

1. Cross-Site Scripting
2. Session-Side-Jacking
3. Session Fixation
4. Brute Force
5. Man in the Browser
6. Man in the Middle
7. IP Spoofing

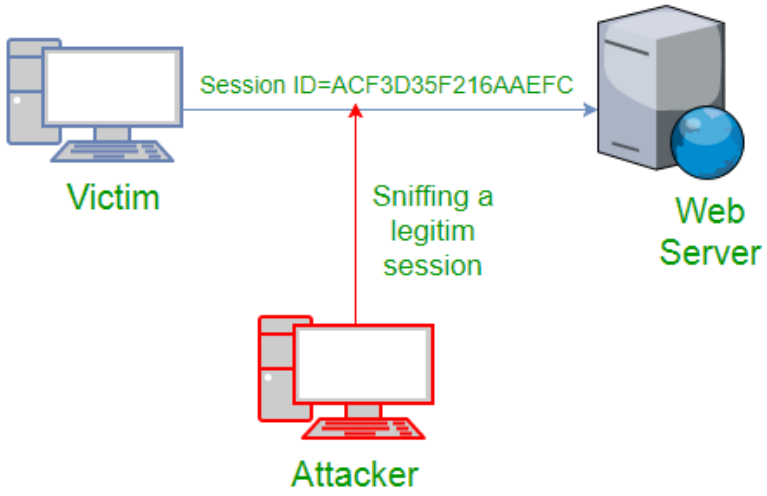
Cross-Site Scripting (XSS)(Active)

The hijacker finds weak spots in the target server and takes advantage by inserting scripts into the web page. This page then loads this code, thinking everything looks legitimate on the client side. Once this cod



Session Side-Jacking (Active)

This, also known as Session Sniffing, is a more active type of attack. But for this type, the hijacker needs to have access to the user's network traffic. So, to achieve that, the hijacker or attacker uses packet sniffing techniques like Kismet or Wireshark to monitor and steal session cookies after searching the user's session.

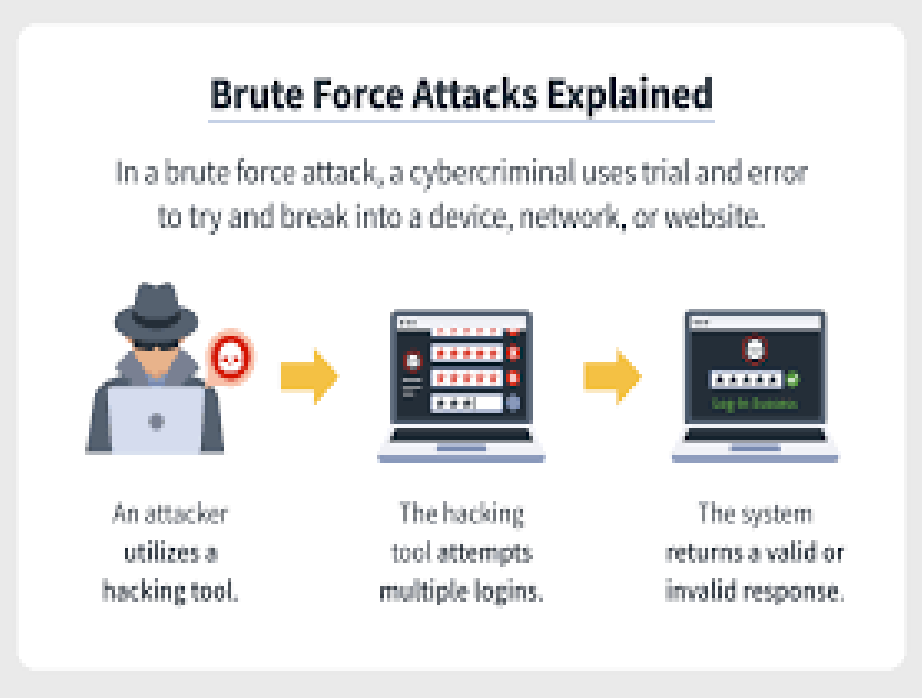


Session Fixation(Active)

In this type of attack, attackers create a session ID, and the user uses this session ID after being tricked. The session ID can be set via URLs or forms through emails, leading to the attacker's website. Once the user logs in, the hijacker gains access to the user's data.

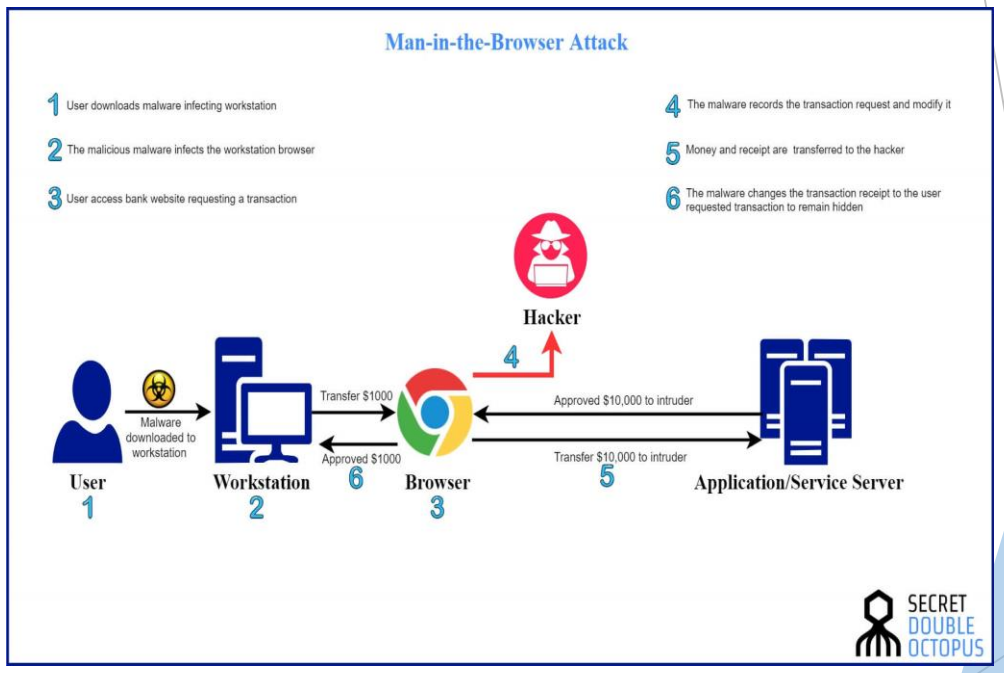
Brute Force

This works mainly if the website or the target user uses predictable session Ids, where the attacker has to guess them and perform the attack. Another scenario is if the hijacker gains access to a list of session IDs from a website with weak security measures.



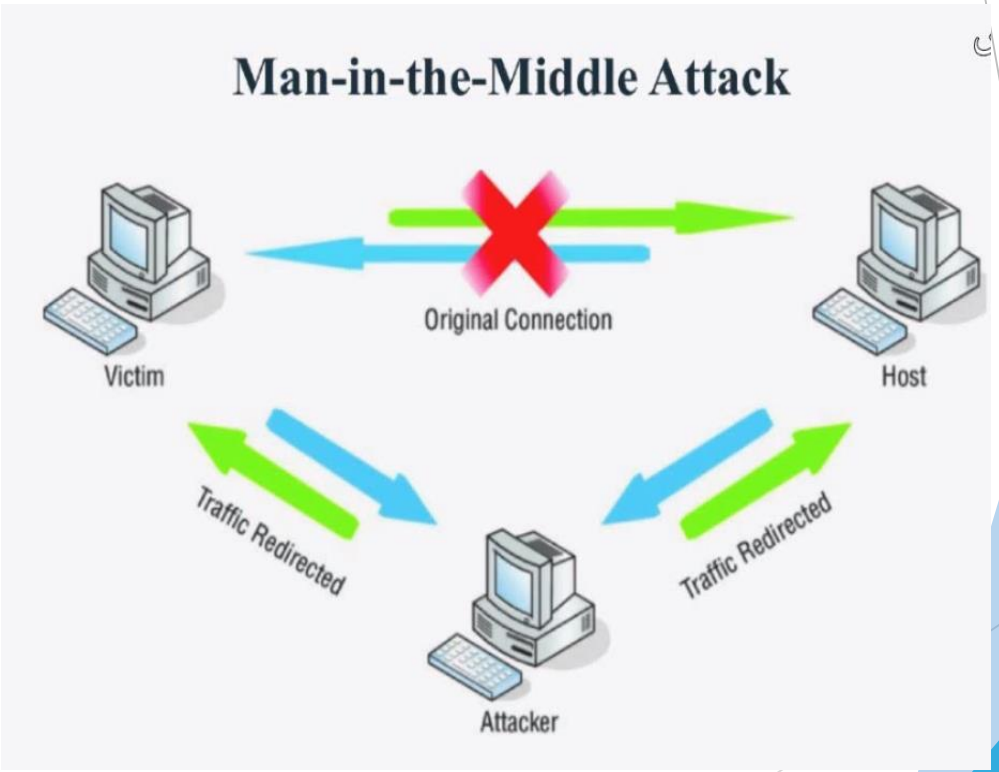
Man in the Browser(Hybrid)

This is also known as Man in the Middle Attacks or Malware. Here, the attacker infects the user's computer with malware and viruses, allowing them to hijack a session. It is very tough to detect any issues with the web application or the site's security in this type of attack.



Man in the Middle Attacks(passive)

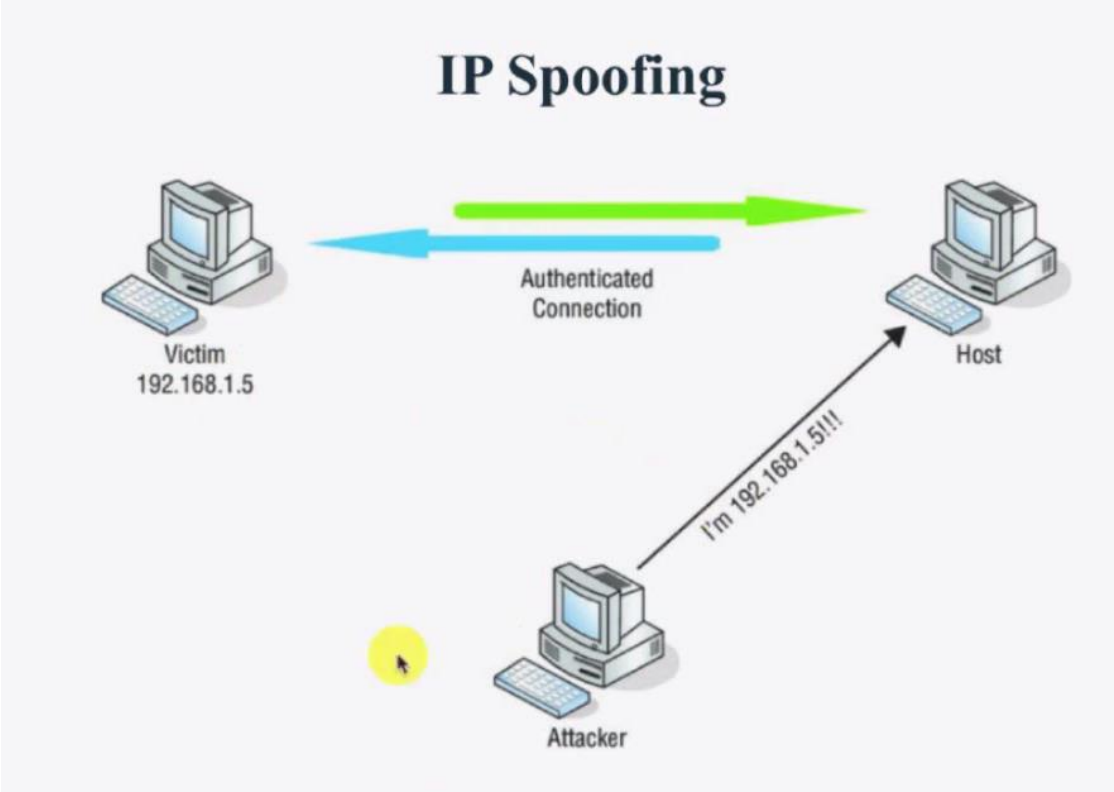
This is also known as Man in the Middle Attacks or Malware. Here, the attacker infects the user's computer with malware and viruses, allowing them to hijack a session. It is very tough to detect any issues with the web application or the site's security in this type of attack.



IP Spoofing

This is also known as Man in the Middle

IP spoofing, or IP address spoofing, refers to the creation of Internet Protocol (IP) packets with a false source IP address to impersonate another computer system. IP spoofing allows cybercriminals to carry out malicious actions, often without detection. This might include stealing your data, infecting your device with malware, or crashing your server.



Session Hijacking Tools

Zaproxy

The OWASP Zed Attack Proxy (ZAP) is an integrated penetration testing tool for finding vulnerabilities in web applications.

Burp Suite

Burp suite allows the attacker to inspect and modify traffic between the browser and the target application.

It analyzes all kinds of content, with automatic colorizing of request and response syntax.

JHijack

A Java hijacking tool for web application session security assessment.

A simple Java Fuzzier mainly used for numeric session hijacking and parameter enumeration.

How to Prevent Session Hijacking

- In order to protect yourself from being hijacked while in a session, you need to strengthen the mechanisms in web applications. This can be done through communication and session management. Here are a few ways you can reduce the risk of session hijacking:

1. **HTTPS:** The use of HTTPS ensures that there is SSL/TLS encryption throughout the session traffic. Attackers will be unable to intercept the plaintext session ID, even if the victim's traffic was monitored. It is advised to use HSTS (HTTP Strict Transport Security) to guarantee complete encryption.
2. **Session Key:** It is advised to regenerate session keys after their initial authentication. This renders the session ID extracted by attackers useless as the ID changes immediately after authentication.

How to Prevent Session Hijacking

3. **Public Hotspot:** Avoid using public WiFi to protect the integrity of your sessions and opt for secure wireless networks.
4. **VPN:** Use a Virtual Private Network (VPN) to stay safe from session hijackers. A VPN masks your IP and keeps your session protected by creating a “private tunnel” through which all your online activities will be encrypted.

How to Prevent Session Hijacking

5. Lock Session ID
6. Use a good algorithm to generate the session ID
7. Clear text
8. Use a strong session ID