

Sri Lanka Institute of Information Technology

**Session Hijacking**

**Individual Assignment**

IE2022 - Introduction to Cyber Security

Submitted by:

|  |  |
| --- | --- |
| Student Registration Number | Student Name |
| **IT20643904** | **Munasingha R.S.I** |

Date of submission - 06/10/2021

**Table of Contents**

[Abstract](#_Toc38443029) 3

[1. Introduction](#_Toc38443030) 4

[2. Evolution of the topic](#_Toc38443031) 6

* 1. Dangers Posed by Hijacking
  2. Why Is Session Hijacking So Effective?
  3. Process of Session Hijacking
  4. Packet Analysis of a Local Session Hijack
  5. Types of Session Hijacking

[3. Future developments in the area](#_Toc38443032) 23

[4. Conclusion](#_Toc38443033) 25

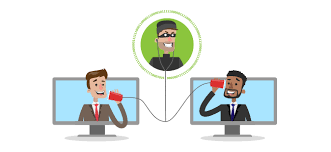
[5. References](#_Toc38443034) 26

# Abstract

In today's world, networks are vulnerable to various types of attacks. One of the most common types of attacks that can occur is “man-in-the-middle”. Session hijacking is the most common type of attack tried. It has a higher success rate than other attacks. Session hijacking is a type of attack that enables unauthorized access to a website or a user's session. One of the most common types of cyber attacks is session hijacking. It occurs when a website or network is attacked with a specially crafted session. It is a type of attack that can execute arbitrary code on a network. This paper aims to introduce the various security mechanisms that can prevent a session hijacking attack. This topic tackles various types of attacks that can severely affect the users' financial well-being. There are various ways that users can avoid getting attacked like this.

# 1 Introduction

Most important a topic in today's world is cyber security. various security threat lurk network at all times, waiting to exploit network vulnerabilities to harm users. Everything in today's world is associated with the web organization. online e-Commerce is most popular methods for customers to shop and conduct business online. Banks provide simple way to Manage online accounts & make online payments, making people's lives easier and faster. People rely on online transactions and payments without question. While conducting online transactions, sensitive information goes across the internet, and maintaining secrecy and integrity has become a major challenge. It is extremely difficult to guard each and every painful and sensitive information. We need to create the most mechanism that is secure manage all of this problem, which will give you us with assurance that all threats to users' confidential and sensitive information would be prevented. A number of the protection threats like Man-in-the-middle attack, sniffing, Denial-of-service (DOM), Phishing attack, ARP spoofing, SQL injections and much more which has malicious intention to get utilized of the vulnerability of the network and the internet application for the bad intentions. Aside from that, Session hijacking is really a technique used by many attackers around the world to attack the internet network. Session hijacking is really a technique used to steal confidential and sensitive information as it travels through the community. It's with the capacity of stealing information without the users' knowledge. This session hijacking attack can be carried out in lots of ways, the most common of which is the Man-in-the-middle attack. In this attack, the attacker places himself between two trustworthy connections and steals all of the important information being exchanged between your two systems. Using session hijacking, the attacker attacks in such a means that the victim does not have any indisputable fact that someone is stealing his/her information. Session Hijacking is capable of carrying away the attack without providing any warning or causing any noticeable changes to the info or information. Because the victim cannot imagine that someone has assaulted his/her system, an attacker can very quickly get all of the information that the attacker desired. This might be an advantage that is big the attacker because session hijacking provides all of the features to meet the attacker's intention without getting caught while doing the assault, giving them more motivation to do more attacks. As I investigate the session hijacking attack, it's necessary to realize the fundamentals of session hijacking and exactly how it works. As stated within the introduction, session hijacking is the entire process of stealing and compromising the user's confidential data by taking over an already established trusted and valid session between two system connections, additionally referred to as a man-in-the-middle attack. When a person logs in to a web application, it establishes a reliable session between the customer and the web server via three-way handshaking. Three-way handshaking is a procedure that enables you to create a trusted connection that is valid a session between a client system and a web server. When you've established a trusted and connection that is secure only the customer and server can communicate with each other and data. The attacker takes over a valid trusted connection and sends packets to a server as a genuine client, then receives packets from the server and sends them to the client as a genuine server in a session hijacking attack. The main advantage of a session hijacking attack is it has to do is keep listening to the network and hijack any valid session. that it does not require breaking any protection or security fire walls; all it needs to accomplish is keep listening into the network and hijack any session that is valid.



**2 Evolution of the topic**

* 1. **Dangers Posed by Hijacking**

Hijacking is not difficult to initiate. The majority of computer that use “TCP/IP” are vulnerable to hijacking. You are able to do to guard yourself before you change to another protocol that is secure there is little. Many countermeasures is insufficient unless encoding may be used. The potential risks being big by hijacking include identity theft, fraudulence, information loss, and so on.

**The following elements are vulnerable to Hijacking:**

* **One-time Passwords (S/key, Smart cards, Challenge Response)**

Connection hijacking is a risk a with all one-time password schemes. after the service or user has authenticated it self, her or his connection can be taken over. Based on the website www.webopedia.com "S/ Key is a one time, Challenge response Password scheme that is used to authenticate data access. the goal of S/key is to eradicate the requirement to send same password over a Network every time a password is needed for access. "

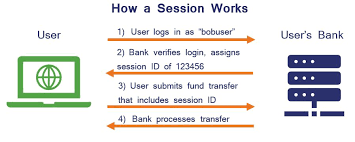
* **kerberos**

Because encryption isn’t enabled by default, Security is just a concern that is major it is similar to the password scheme that is one-time, which can be effortlessly hacked.

* **Source Address Filtering Router**

If your network's security is based on filtering packets from Unknown Sources, it is at risk of network address spofing attacks. an unknown host could insert it self into the middle of an connection that is existing.

* **Proxies with Source Address Control**
* Many Proxies Control Access to commands that are certain on the source address of the requestor. the source address is easily Vulnerable to Active or Passive sniffer s.
* No simple actions have actually yet been found that can secure a network from sniffing that is active or passive. You will be better willing to make smart safety decisions for the network if you are aware of this threat.

****

* 1. **Why Is Session Hijacking So Effective?**

**The following elements contribute to the success of session hijacking:**

* **Session ID Generation Algorithm weakness:** At the moment, most webs generate session IDs using "linear algorithms" that are based on effortlessly predictable variables like time or internet protocol address. The attacker can effortlessly reduce the search Space required generate a session id that is valid by studying the sequential pattern and creating many requests.
* **Indefinite Session Expiration Time:** Indefinite session expiration times give an attacker an unlimited amount of time to guess a session id that is valid. the 'Remember Me' option on websites is an example of this. If an attacker obtains a user's cookie file, he or she can use static Session ids to gain access to the user's account. if Attacker is able to gain access to proxy server that caches or logs session IDs, he or she can perform session hijacking.
* **Clear Text Transmission:** The session ID can potentially be sniffed across a set system if SSL just isn't made use of even though the session ID cookie is sent to and through the browser. The SSL wouldn't normally protect the provided information in cases like this. In the event that session IDs retain the logon that is real in the sequence and are usually grabbed, an attacker's work becomes even easier**.**
* **Small session ids:** Even when a good cryptographic procedure is utilized, an Active Session id can be easily determined if string is short.
* **Insecure handling:** By tricking the browser into visiting another web site, the attacker can retrieve the stored session id data. attacker can take advantage of then the information ahead of the session expires. this can be Accomplished in a variety of Ways, including domain name system poisoning, cross site scripting exploitation, and exploiting browser bugs, among others.
* **Invalid Session IDs Have No Account Lockout:** The attacker can decide to try any number of times with different session IDs embedded in an authentic URL if the web pages have no sort of account lockout. An attacker are able to keep undertaking before the session that is real is discovered. The process is known as brute pushing the session IDs. the web server will not display any warning messages or complaints during the session ID brute force attack . As a result, an assailant can figure out the Original Session id.

**Success of session hijacking is dependent on all of the factors listed above.**

* 1. **Process of Session hijacking**

It is simpler to enter the system as a authentic user than it is to enter that directly. hijacking works by locating an existing Session and hijacking it after a authentic user have gained access and been authenticated. The attacker can remain connected to your session for hours after it was hijacked. this gives attacker plenty of time to install Back-doors or gain additional Access to the system. The fact that attacker impersonates a legitimate user is one of the most significant that session hijacking is difficult to detect. As a result, all traffic directed to user's Internet Protocol address is routed through attacker's systam.

**Attacker can obtain access to a session in a variety of ways.**

**Tracking connection:** found utilizing a network sniffer, or identifies the right individual by scanning with a tool like Nmap to get the target having an simple TCP sequence prediction. This is accomplished to make sure that the sequence that is correct acknowledgement numbers are grabbed, as TCP checks packets utilizing sequence and/or acknowledgement numbers. The assailant utilizes these figures to create his / her packets.

**Desynchronizing connection**: A state this is certainly desynchronized each time a connection between target and host is set up; or in a well balanced condition with no data transmission; or even the host's series number does not equal client's acknowledgement number.

Sequence number or acknowledgement number [SEQ/ACK] of server must be changed to desynchronize connection between the target and host. this is accomplished by sending NULL data to server, allowing server's SEQ/ACK numbers to advance while target machine is incapable of registering such an increment. Example; prior to desynchronization, attacker observes session without interfering in any way.

From then on, the attacker directs a amount that is huge of "Null Data" to the server. This information is only utilized to change the ACK number from the server and has now hardly any other impact. both the target and server are now Desynchronized.

The other method is to deliver a reset flag to server in order to disconnect connection on server side. this should ideally happen during the connection's initial setup. The attacker's goal is to break server connection and establish a new one with a various sequence number.

the attacker monitors host for SYN/ACK packets from the server. When the attacker notices the packet, he immediately sends a RST packet to the server as well as a SYN packet with the same features, such as a port number, but a different sequence number. When the server receives the RST packet, it terminates the connection with the target and opens a new one based on the SYN packet, but with a different sequence number on the same port. To get acknowledgement, the server sends a SYN/ACK packet to the target after establishing a new connection. This is noticed (but not intercepted) by the attacker, who sends an ACK packet back to the server. The server has finally reached a condition of stability. When the target gets the initial SYN/ACK packet from the server, the primary purpose is to keep the target conversant and move to the established state. Both the server and the target are now asynchronous but stable.

This may also be done using a FIN flag, however it may lead the server to react with an ACK, disclosing the attack via an ACK storm. The reason for this is a fault in the approach utilized to hijack a TCP connection. When the host gets an unsatisfactory packet, it sends the expected sequence number in response. This undesirable packet causes an acknowledgement packet to be generated, resulting in a never-ending cycle for each data packet. As both the server and the target attempt to verify the right sequence, the disparity in SEQ/ACK number creates unnecessary network traffic. These packets are not retransmitted if they are lost since they do not contain data. Because TCP is based on IP, a single packet loss stops the unwanted interaction between server and target.

the desynchronizing stage is included in hijack sequence to keep target host unaware of attack. without desynchronization, attacker can inject data to server and even maintain her or his identity by faking IP address. however, they must also deal with server's response being transmitted to target host.

**Using Attacker's packet to inject:** Now that attacker has disrupted connection between server and target, He or She has the option of injecting data into network or actively participating as the man in the middle, sending data from target to server and vice versa, injecting and reading data as desired.

2.4 **Analysis of Packets from a Local Session Hijack**

Session hijacking attacks are high-level attack vectors that affect many systems. Many systems that are connected in a LAN or on the Internet use TCP communication protocol for transmitting data. For connection establishment between two systems and for the successful transmission of data, the two systems should establish a three-way handshake. session hijacking is the process of gaining control of a session by utilizing this three-way handshake method.

**The attacker engages in three activities to carry out a session hijacking attack:**

* Tracks a session
* Desynchronizes the session
* Injects attacker's commands

Monitoring or tracking a session may be accomplished simply by sniffing the traffic. The next phase in session hijacking is desynchronization. This is straightforward if the client's next sequence number is known. If you know the sequence number, you can use it to hijack the session before the client. The sequence numbers can be determined in two methods. One way is to sniff the traffic, identify the ACK packet, and then determine the next sequence number based on the ACK packet. The data might also be sent with assumed sequence numbers, which is another alternative. The second technique is less trustworthy. You may simply detect the sequence number if you have network access and can sniff the TCP connection. This type of session hijacking is known as "local session hijacking."

The attacker delivers the data with the predicted sequence number before the user does. The server and the attacker are now in sync. As a result, a connection between the attacker and the server is formed. When a connection is established between the attacker and the server, the server discards the data as a resend packet, even if the user provides data with the right sequence number. Because the user is uninformed of the attacker's actions, he or she may resend the data packet if an ACK for his or her TCP packet is not received. The server, on the other hand, drops the packet yet again. As a result, an attacker launches a local session hijacking attack.

**2.5 Types of Session Hijacking**

There are fundamentally three variety of session hijacking attack which does the work that is same various ways. We shall look at all three types of session hijacking in detail, which will provide us with a more in-depth understanding for the session assault that is hijacking its techniques.

**There are three forms of hijacking**

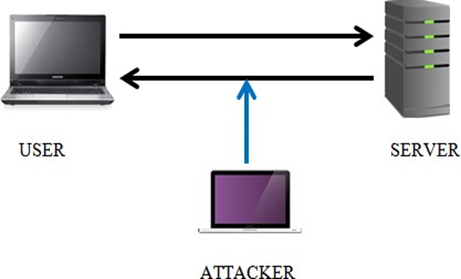
• Active session hijacking

• Passive session hijacking

• Hybrid session hijacking

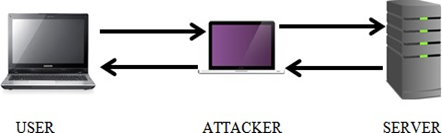
**Active Session Hijacking**

active session hijacking is a technique in which an attacker attacks an existing session between a user and a server. Using a DOS, the attacker attacks an active session, knocking out the valid user and putting himself in the place of the valid user (DOS). Before launching a DOS attack, the attacker sniffs connection and captures data packets between user and server using a packet capturing tool such as “wireshark”. Denial of service occurs when an attacker floods target with traffic. The attacker sends a large number of requests to target network, making it unavailable to the server. Result is, target system is unable to use the services sent by server, and the target machine may shut down or crash in order to handle the traffic. the server waits for a while before sending another request to the user machine for connectivity, at which point the attacker masquerades and accepts as a valid user and sends an acknowledgment to server, and the attacker establishes a connection with the server in place of a valid user.

****

**Passive Session Hijacking**

The attacker inserts himself between the server and the valid user, sending valid packets to the user while masquerading as a server, receiving packets from the user and sending them to the server while masquerading as a valid user. In this Passive Session Hijacking technique, the attacker may transmit all data via the attacker system and even make modifications to data packets that neither the user nor the server can notice. In this manner, the attacker can collect all of the information required for his/her nefarious activity. The attacker, on the other hand, has one disadvantage. They can only access the data and information between the user and server while the session is active; if the user logs out, attacker is unable to access the data and the session is permanently ended.



**Hybrid Session hijacking**

Hybrid Session Hijacking is a combination of Active and Passive Session Hijacking. To achieve his/her goal, the attackers use both types of session hijacking techniques, which can be active and passive. it is classified into two types:

**Blind Spoofing Attack**

Blind Spoofing is a form of attack where the attacker attacks prospective system without making any noticeable changes to the server to victim device connection. An attacker merely captures the network that is entire between the user and the server in purchase to find out the TCP sequence number to be able to gain authentication with the server and full control throughout the session. But there is however a problem that is big this attack: it is extremely difficult to acquire out or guess the TCP sequence number through captured packets because TCP sequence number is a random number and every time it creates a fresh and random TCP sequence number, which makes it very hard to discover the correct series number, which takes a long time and requires the attacker to keep capturing packets. Finding the sequence that is correct might take a long time, and the attacker must have patience so that you can achieve these kinds of attacks.

**Non-Blind spoofing attack**

Non-Blind spoofing attack in which attacker is on the network that is same victim and it is for a passing subnet as the victim, enabling the attacker to monitor the traffic between the victim while the server. Because the attacker can understand packets traveling through the exact same network, it is easy for the attacker to monitor traffic from the system that is same. Attackers keep an eye on the connection and decide to try to guess the TCP sequence quantity of the packets that are next purchase to utilize the TCP sequence number to gain authentication over the connection. Attackers determine the series that is right and re-establish a connection utilizing the server using that number. Nonetheless, the issue that is main this attack is that today's routers do not allow packets to be broadcast in the system; instead, they keep it switched off to be able to protect the packets. The attacker resets the connection in order to position himself between routers and capture the first broadcast packet in order to avoid this problem.

However, the issue that is foremost this attack is that today's routers don't allow packets to be broadcast within the network; instead, they keep it turned off in purchase to guard the packets. The attacker resets the connection to be able to place himself between routers and capture the first broadcast packet to avoid this problem.

The session ID is a number that is one-of-a-kind by the host to a specific individual during a trip to a website. The session ID is assigned within the cookies form field or, on occasion, in the URL field. The session ID can be generated in a variety of methods. Some internet servers create session IDs by incrementing figures which can be static however this method is not very suitable or safe; If the attacker is actively monitoring the packets provided for the server through the victim machine, the attacker can easily imagine the session ID.

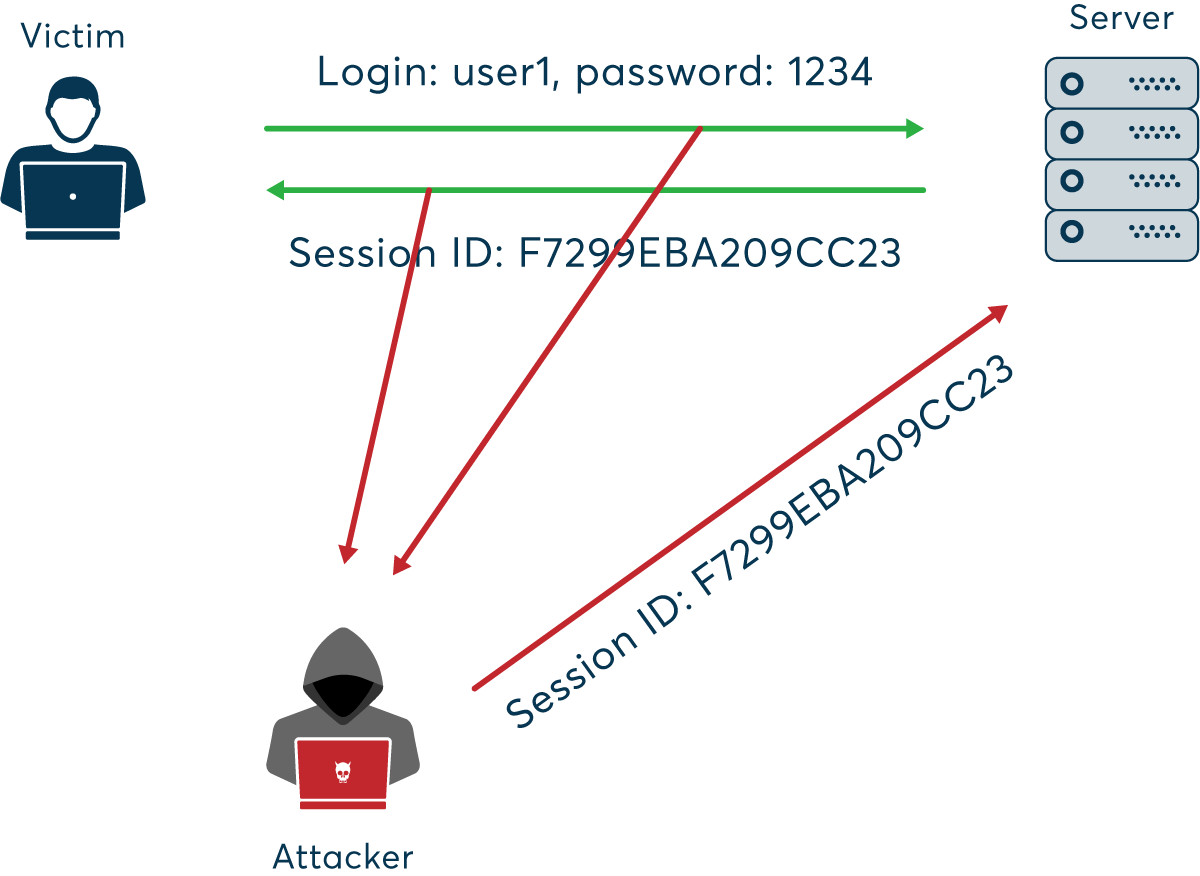
This method of producing session ID is not suggested because the attacker can easily imagine the incrementing of fixed numbers. Utilizing an algorithm to create a session ID is the way that is best to make session IDs more complex and secure than incrementing fixed figures. Using algorithms to create a session ID involves various methods being complex make session IDs more secure and hard to guess and predict by any attacker. Making use of algorithms to generate a session ID involves various methods that are complex make session IDs more secure and difficult to guess and predict by any attacker.

**Stealing session ID’s**

Numerous methods have for stealing session IDs, several of which become given below.

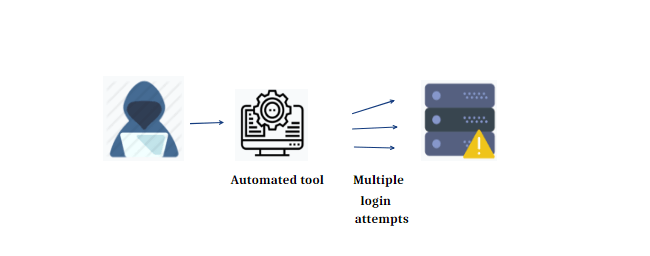
1. **Sniffing**

One sort of attack is sniffing, in which the attacker hijacks the network and grabs the session ID. The attacker continues to monitor the victim's network traffic, looking for any unencrypted packets. If the attacker discovers any unencrypted packets, it checks to see if they include a session ID. If the attacker receives the session ID from the packet, it takes over the previously established session between the victim and the server, and the attacker gains access to all of the target machine's information.



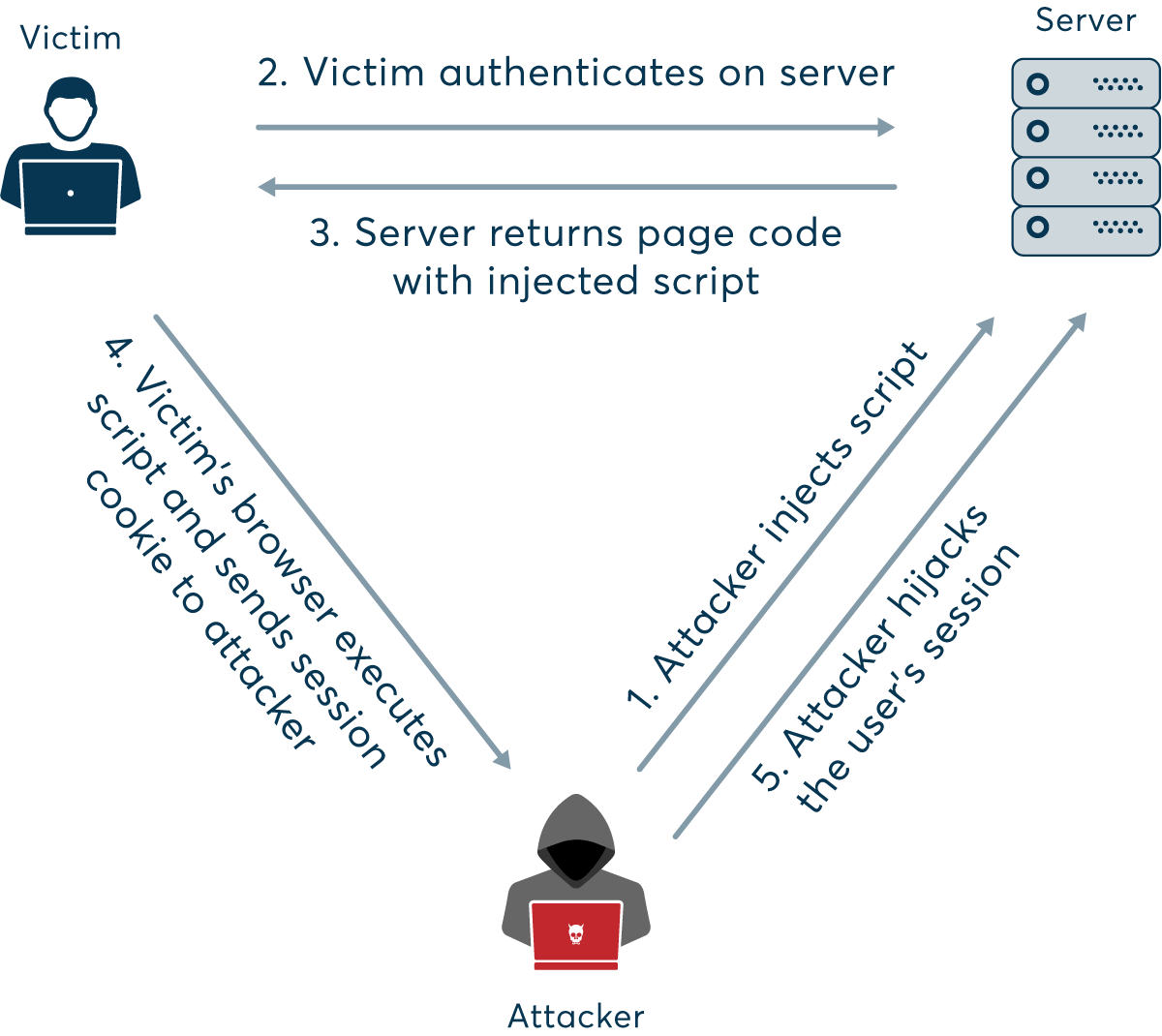
1. **Brute Force**

Brute Force is a attack that is well-known can crack any character, number, symbol, or unique character combined username, password. This Brute Force attacker might take a time that is long but he guarantees to finish the task. The attacker can specify the combination of characters, symbols and number, as well as how many words in this Brute force attack. Assume the victim has eight digits of password and the attacker has set the length of the password to eight, so this brute force checks all the combinational words set by the attacker or over to the digit that is 8th. An attacker must have a whole large amount of patience and time for you complete this sort of attack. This force that is brute in the target network checks all associated with combinations set by an individual to crack the session ID and provides appropriate session ID to attacker, whom then takes over the valid already created session.



1. **Cross Site Scripting**

Cross-site scripting is another means of stealing the session ID (XSS). This XSS, also known as a client-side code injection attack, enables the execution of a malicious script into any website or online application. This attack simply exploits website weaknesses by turning any user input into unencrypted or encoded data and transmitting it in plain text to an attacker. However, the attacker has a substantial disadvantage in that this approach is only effective on vulnerable targets; many websites have been patched to address these security issues.



**tools for session hijacking**

Session Hijacking can be completed using a variety of tools.

* Burp Suite
* Ettercap
* BetterCAP
* Hamster
* ferret
* T-Sight
* Hunt
* Juggernaut
* sslstrip
* JHijack

**5.6 Session Hijacking Counter Measures**

Session hijacking can be avoided in a variety of ways. However, these methods are contingent on the users' willingness to take security seriously. It is said that attackers take advantage of a user's lack of knowledge about security to steal important data, and that attackers will sometimes deceive a user and steal information. From the Open System Interconnection mode, we can divide the countermeasures against session hijacking in to as network and application layer.

**Network layer**

1. ***Secure socket layer***

Constantly use a “SSL” to encrypt data from starting to end. as a result, whatever data passes through this network secure socket layer is encrypted, making it difficult for an attacker to see the exact data that passes through the network. Even if an attacker obtains the data, extracting the real data from the packets is extremely difficult. SSL channels use symmetric key of “256 bits” and public key of 28 bits, making the encryption method more complex, strong and secure.

1. ***Use secure shell***

Secure shell, also known as “SSH”, is a network protocol that allows users to access remotely located system or computer in a secure manner. this secure shell provides a strong method of authentication as well as stronger method of encryption between two systems in insecure network, assisting the user in avoiding session hijacking attacks.

1. ***Use of HTTPS***

when login to any website or web server, or when do online banking, online shopping, or E-commerce, it is critical to use the (Hyper Text Transfer Protocol Secure) connection. A user should be aware that when doing any online work, the URL should always be in Hyper Text Transfer Protocol Secure format because this secures connection and indicates that it is secure link for online work. If link does not begin with https, it is insecure, and data will be transmitted in plain text.

**Application layer**

1. ***Complex and strong session ID***

Each application supplies the users with a session identifier, additionally known as a Session ID or a token, which will be assigned to the session when the session is used and intended to notify the users and application server to be able to trace the progress of the consumer together with authentication state of the users into the web application. This session Id will be legitimate before the session expires, at which point it will probably be invalid.

1. ***Random session ID***

Using a program that is random generation method causes it to be much more problematic for an assailant to imagine the session ID. Long session ID: If the session ID is long enough, it provides safety this is against brute force attacks.

1. ***Session ID generated by server***

Always utilize a session that is server-generated, which makes the session ID more complex and strong because computers use an algorithm to build the session id, and it's also very hard for an attacker to crack.

* Encrypted session ID: best way for protecting the session ID is to encrypt it. encrypted session IDs travel through the network in an encrypted form, rendering them unidentifiable.
* Automatic log out: there should be an Implementation or Mechanism in place to force a log out at a specific time and have server request that the client re-establish connection with new session ID. It assists the session and session ID in avoiding session hijacking and maintaining the user's authentication.

# Future developments in the area

The researchers are working on creating a smart system that would recognize and prevent session hijacking attempts. Several research-based algorithms and methodologies have been presented and successfully applied.

* Ensuring a secure cookie generation
* Implementing the CIA technology
* Using an encrypted connection for transmission
* Locking a session to the users who belong to it

**Ensuring a secure cookie generation**

Cookie is one that contains all of the user data that is sent to the server in order to authenticate a specific user. Some data stored in cookie are as follows:

* Password
* Username
* Session ID
* creation date - timestamp
* Expiration time

**implementing a CIA technology:**

an affordable robust solution was proposed, which is to change the cookiegenerated for very single transmission between the client and server that uses CIA technology. This is accomplished by running a JavaScript on the server and making the cookie available only to the client's web browser; and also this prevents cross site scripting. protocol used to secure the wireless network was later created as a standard plug in for web browsers such as Firefox, Chrome, and others.

**Using an encrypted connection:**

The HyperText Transfer protocol Securityprotocol was developed to secure all network based transmissions. HTTPS is implemented using a secure socket layer (SSL), also known as Transport Layer Security. The only disadvantage of this technology is that it is not suitable for every website due to the costs involved in implementing and maintaining such a service.

**Locking a Session to its corresponding user:**

Locking a session to a certain user is a very successful approach for avoiding session hijacking since it assures that even if the session is hijacked, the attacker's information is rendered worthless. The author of a session locking research article recommends using a unique fragment identifier, a unique HMAC algorithm, and a shared secret key between the client and the server. The first SSL login token is kept and reused in the browser's fragment identifier, resulting in a rapid and low-cost approach for safeguarding the session and avoiding eavesdropping.

Advancements mentioned above are research-based; certain tools have been developed by numerous infrastructure behemoths such as Cisco, Juniper, and others.

Some of the tools are as follows:

* + Juniper network VPN devices
  + CISCO Intrusion detection tool
  + State of the art – Cisco switches
  + Firewalls

# Conclusion

# This article goes into detail about the session hijacking attack and shows how detrimental it is to network security. Many individuals are still ignorant of these sorts of assaults, and network security specialists are dismissive of them. Some online applications and web servers still have inadequate session management due to a lack of awareness about session hijacking attempts. I covered numerous defenses to the session hijacking assault in this work, which does not prevent the attack but makes it more difficult for the attacker to succeed. Many adjustments to the web app and server are still required to permanently halt the session hijacking king of attack. This paper focuses primarily on the countermeasure of session hijacking.references

* “Session Hijacking Module 11”- CEHv8. Ethical Hacking and Counter Measures.
* [*"Prevent Session Hijacking by Binding the Session to the Cryptographic Network Credentials"*](https://www.springer.com/computer/security+and+cryptology/book/978-3-642-41487-9)*. Burgers, Willem; Roel Verdult; Marko van Eekelen (2013).  Proceedings of the 18th Nordic Conference on Secure IT Systems (NordSec 2013).*
* ["CookiExt: Patching the browser against session hijacking attacks"](https://dx.doi.org/10.3233/jcs-150529).-  Bugliesi, Michele; Calzavara, Stefano; Focardi, Riccardo; Khan, Wilayat (2015-09-16*Journal of Computer Security*.
* Burgers, W.: Session proxy, a prevention method for session hijacking in blackboard. bachelor thesis, Institute for Computing and Information Sciences, Radboud University Nijmegen, The Netherlands. Bachelors Thesis (July 2012)
* https://en.wikipedia.org/wiki/Session\_hijacking
* <https://owasp.org/www-community/attacks/Session_hijacking_attack>
* https://cwatch.comodo.com/blog/website-security/what-is-session-hijacking/
* <https://diarium.usal.es/pmgallardo/2020/10/26/list-of-session-hijacking-tools/>
* https://www.netsparker.com/blog/web-security/session-hijacking/