SESSION HIJACKING AND ITS PROTECTION

Daya Shankar Singh
Asst. Professor
Sunbeam College for Women Varanasi-221109, Uttar Pradesh, India
E-mail:- vnsdaya@gmail.com

Abstract

There are various websites and protocols which have flaws, that's why it is vulnerable to hijack session of a user. A session may contain sensitive information so; you need to care for the session as a possible security hole. Since session is a key to indentify a valid user so it should be more secure. If someone is listening in or snooping on a network, it's possible that he can intercept a session ID and use it to look like he is someone else. In multiuser environment, it is also possible to access session data from the local file. One of the popular incident that exploits session hijacking attack vulnerability is Firesheep which is an extension of Firefox browser. It made it trivial to gain access to anyone's account while online using an open unsecure Wi-Fi connection. Many websites like Facebook, Flickr, Foursquare and many e-commercial sites are likely to be susceptible to it.

The main objective of this paper is to discuss mechanics of the act of session hijacking in TCP sessions i.e. hijacking at the network level and at Application levels i.e. hijacking HTTP sessions. The precaution and prevention will also be discussed in this paper.

Keywords:- *Session Hijacking, Session Fixation, TCP Session Hijacking, HTTP Session Hijacking.*

1. Introduction

1.1. What is a Session?

A lasting connection between a user (or user agent i.e. browser) and a server usually involving the exchange of many requests. It is Typically maintained by the server and Created on first request or after an authentication process.

User states are stored in table with an identification key known as session-id. Session-id also known as HTTP cookie is an alphanumeric key, returned by server to uniquely identify the user.

1.2.Why are they needed?

Sessions are generally needed to simplify authenticated communication between two or more parties. All the nodes are not required to authenticate for every activity or action, because of the session which exists.

1.3. What is a Session hijacking?

Session hijacking is a process of taking over an active HTTP, TCP, and UDP Session of user without their permission or knowledge by the hacker. Once it is implemented successfully, attackers assume the identity of the victim user and enjoying the same access to resources as the valid user.

Session hijack attacks are usually waged against users that are members of large networks containing a substantial number of open sessions via guided media or wireless media.

Protocols which are session oriented and having the longer length of communication session are more attractive to the attacker.

2.Types of Session Hijacking

2.1.Active Session Hijacking

Active session Hijacking involves hijacking

a already authenticated session.

Active Session Hijacking means that original user has logged in his account or profile and then attacker steal the cookies to hijack the active session and then disconnect the original user from the server.

Working Method:

In Active Session Hijacking, attackers use client side scripts to steal the original users cookies by involving social engineering tactics which includes emails, private messaging on forums and on other social networking websites.

Why we call it active session hijacking because attackers need to interact and need some actions to be performed by the victim to steal the session successfully.

2.2.Passive Session Hijacking

In passive session hijacking attackers does not hijack active session instead they capture the login credentials while the original user is trying to establish a new connection with the server, and attacker is sitting silently on the same network and recording the login credentials.

Working Method:

Passive Session Hijacking involves the use of network sniffing tools that captures data packet and exploit the vulnerability of ARP protocol by poisoning the network. Attackers analyze those captured data to retrieve login credentials of the user.

Why we call it passive session hijacking because attackers does not need to interact with the user and make him perform any specific actions.

3.Three-Way Handshaking

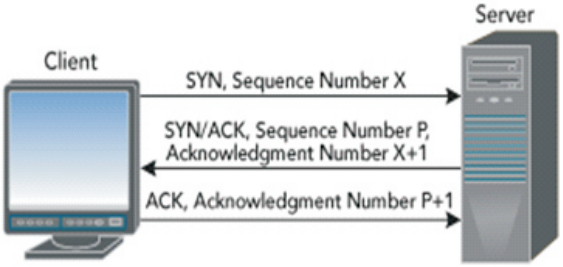
When two devices want to communicate with one another, they have to agree the technical parameters before communicate with one another.

In first step when a client wants to communicate with a server it builds a packet with the SYN (synchronization) bit set and an initial random sequence number, let X i.e. SYN, Sequence number X.

In second step when the server receives this packet, it responds to the client with a packet containing both the SYN (Synchronization) and ACK (Acknowledgement) bits set. The SYN initial sequence number is any random number let P, generated by the server and

ACK sequence number is X+1 i.e. SYN/ACK, Sequence Number P, Acknowledgement Number X+1.

In third step client sends acknowledgement packet to the server confirming its desire to communicate, with ACK bit set and acknowledge sequence number P+1 i.e. ACK, Acknowledgement Number P+1.



The two machines have successfully established a session until one of the machines sends a RST (Reset) or FIN (Finish) packet to the other.

So, why this section is introduced here? Because sequence number play an important role for network session hijacking as given below section 4.

4.TCP Session Hijacking

TCP Session Hijacking is network level hijacking. TCP session hijacking actually deals with the successful prediction of the Initial sequence numbers that gets exchanged between two hosts during three way handshaking.

4.1.Methods of TCP Session Hijacking

Now before predicting an initial sequence number of a TCP three way handshaking, attackers need to be in between the client and server to successfully hijack the TCP connection, for which attacker can actually use these techniques.

- Source routing: attacker can actually use source routing in such a way that the data is being transferred between client and server through attacker. In that case attacker can see the data being exchanged and can see the Initial Sequence Number.

- Man in the middle attack: By Logically placing himself between the server and client, an attacker can similarly see the data connection going through him.

- Attacker can also use ICMP redirect to spoof himself as gateways so that data can be passed through him.

- Attacker can also hijack the TCP session by not predicting or looking for initial sequence number instead he can let the user establish a successful connection and then spoofing himself as a client by changing his MAC address with that of the original Host, attacker can send RST bit set to the server to reset the connection and starting a whole new connection from TCP three way handshaking and exchanging the new sequence numbers.

5.HTTP Session Hijacking

It is application level hijacking. If you think about some of the common websites you visit that require login credentials, those are great examples of session-oriented connections. You must be authenticated by the website with your username and password to formally set up the session, the website maintains some form of session tracking to ensure you are still logged in and are allowed to access resources (often done with a cookie), and when the session is ending the credentials are cleared and the

session ends. This is a very specific example of a session and even though we do not always realize it, sessions are occurring constantly and most communications rely on some form of session or state-based activity.

5.1.Methods of HTTP Session Hijacking

The principle behind most forms of session hijacking is that if you can intercept certain portions of the session establishment, you can use that data to impersonate one of the parties involved in the communication so that you may access session information.

HTTP Sessions Hijacking involves obtaining Session ID's for the sessions, which is the only unique identifier of the HTTP session. Session ID's can be found at three places:

1. In the URL received by the browser for the HTTP GET request.
2. With cookies which will be stored in clients computer.
3. Within the form fields.

Session ID can be obtain by sniffing, which is same as the Man in middle attack. Embedded session info in the URL is accessible by looking through the browser history or proxy server or firewall logs. A hijacker can sometimes reenter in the URL from the browser history and get access to a web application if it was poorly coded.

Another way is by Brute Forcing the Session ID's which involves trying a set of session id's based on some pattern. Brute forcing is a time consuming task but worked on some algorithm can produce results rather quickly.

Cookies are accessible on the client's local machine and also send and receive data as the client surfs to each page.

6.Defending Against Session Hijacking

6.1.Guidlines for the users and professionals

There are many different forms of session hijacking so the defenses for them can vary. Here are a few things you can do to better defend against session hijacking:

- Save Online Banking for Home - The chance of somebody intercepting your traffic on your home network is much less than on your work network. This isn't because your home computer is more secure, but the simple matter of fact is that if you only have one or two computers at home.
- Be aware - Smart attackers will not leave any evidence that they have been in one of your secure accounts but even the most seasoned hackers make mistakes. Being aware when you are logged into session-based services can help you determine if somebody else is walking in your shadow. Keep an eye out for things that seem out of place, and pay attention to “Last Logon Time” fields to ensure everything matches up.
- Secure your internal machines - Once again, attacks like these are most commonly executed from inside the network. If your network devices are secure then there is less of a chance of those compromised hosts being used to launch a session hijacking attack.
- Users should not login to websites with unsecure sessions on any network

that can be compromised.

. Use VPN when possible from any non secure network.

. Systems should have the latest security patches.

. Configure firewall to limit incoming/ outgoing traffic to applications (eg. DNS, email, WWW, FTP) that have a business need.

. Paying attention to https vs. non-https

. Properly signing out

. Not clicking on links but copying and pasting them.

. Minimize session expiration time as possible.

. Use SSL for all communications.

. Re-generating session-ids as given in the section 6.2.

6.2. Session Fixation

Session hijacking is when someone accesses either a client's cookie or session ID, and then attempts to use this data. Session fixation is attempting to set your own session ID. Session fixation and hijacking are easy to combat. We'll make use of the super global variables for the client's IP address and browser type to keep things secure.

Given below PHP code demonstrates encoding the information with an md5 function call to prevent these potential security holes.

```
<?php
session_start();
$user_check =
md5($_SERVER['HTTP_USER_AGENT'] .
$_SERVER['REMOTE_ADDR']);
if (empty($_SESSION['user_data'])) {
session_regenerate_id();
echo ("New session, saving user_check.");
$_SESSION['user_data'] = user_check;
}
if (strcmp($_SESSION['user_data'],
$user_check) !== 0) {
session_regenerate_id();
echo ("Warning, you must reenter your
session.");
$_SESSION = array();
$_SESSION['user_data'] = $user_check;
}
else {
echo ("Connection verified!");
}
?>
```

When a browser first requests the page in a session is started. In that session, we stored the encoded combination of the IP address and browser type. That way, when the user returns to this page, we can compare the value stored in the session versus a fresh computation of the IP address and browser type. If the two don't match, we potentially have a hijacker, so we pick a new ID and clear out any saved data for that session. That way, the hijacker cannot retrieve any of the private information stored in the session. This doesn't cause a problem for legitimate users, because they aren't going to change

browser or IP addresses in the middle of a session with your web site.

6.3.Shared Hosting Concerns

If you don't have your own dedicated server or are on a server that has multiple users, it can be very dangerous to use the default PHP settings to store your user's session data in a temporary directory. Normally, all users have access to that temporary directory, so they can easily pilfer private data from the session, including the session ID.

To make your session data more secure, you can set the session.save_path configuration parameter with the ini_set function to change the path where sessions are stored, as given below. Make sure that these are stored below the web root directory.

```
<?php
ini_set('session.save_path',
'/home/user/sessions/');
 session_start();
?>
```

Be sure that whichever folder you choose is created and has the correct permissions for the PHP interpreter to write the session data. Typically, this means the file must be writable by the permission group www-data. This folder shouldn't be readable or writable by general users at large.

7.Conclusion

Session hijacking is a serious threat to Networks and Web applications on web as most of the systems are vulnerable to it. Although above explanation and countermeasures will give insight to the defender to protect his /her network and web application, but it will also raise the security bar and will force the hijackers to apply more complex attacks to compromise the system. Networks should be tested and monitored continuously in order to make them impenetrable by the intruders. Session-hijacking attacks against wireless networks is easier than the wired networks because of the open broadcast nature of the wireless networks. Another important outcome of the experiments is the importance of encryption. Encryption and key management are the most important components of a secure network. If the encryption is strong and key management is well-organized, the attackers cannot reach their goals even if they exploit some other vulnerabilities.

References

1. Mrs. Mridu Sahu and Rainey C. Lal "Controlling IP spoofing through packet filtering" Int.J.Computer Techology & Applications,Vol 3 (1),155-159 ISSN:2229-6093.
2. Rupinder Gill, and Smith, Jason and Clark, Andrew "Experiences in Passively Detecting Session Hijacking Attacks in IEEE 802.11 Networks" in: Proceedings of 4th Australasian Information Security Workshop (Network Security), 16-19 January 2006, Hobart, Tasmania.
3. Learning PHP and MySQL By Michele Davis, Jon Phillips Chapter-14
4. Understanding Man-in-the-Middle Attacks – ARP Cache Poisoning (Part 1) http://www.windowsecurity.com/arti

cles/Understanding-Man-in-the-Middle-Attacks-ARP-Part1.html

5. Understanding Man-In-The-Middle Attacks – Part2: DNS Spoofing http://www.windowsecurity.com/articles/Understanding-Man-in-the-Middle-Attacks-ARP-Part2.html
6. Understanding Man-In-The-Middle Attacks - Part 4: SSL Hijacking http://www.windowsecurity.com/articles/Understanding-Man-in-the-Middle-Attacks-ARP-Part4.html
7. Internet Crime Complaint Centre link: www.ic3.gov
8. Hassell, J. (2006). The Top 5 Ways to Prevent IP Spoofing. Retrieved from the web July 15,2006. Http://masc2279.no-ip.org/gadgets-toys/internet/the-top-five-ways-to-prevent-ipspoofing/
9. Using Microsoft Windows IPSec to Help Secure an Internal Corporate Network Server. Retrieved from the web July 20, 2006,http://www.microsoft.com/downloads/details.aspx?familyid=a774012a-ac25-4a1d-8851- b 7 a 0 9 e 3 f 1 d c 9 & displaylang=en
10. http://www.microsoft.com/technet/technetmag/issues/2005/01/SessionHijacking/default.aspx