

TCP Session Hijacking Implementation by Stealing Cookies

D.Madhavi¹

¹Assistant Professor

¹Department of Information Technology

¹VR Siddhartha Engineering College, Vijayawada, India

Abstract— Session Hijacking is a form of Man-in-the-middle attacks which are increasingly creating unbelievable impact on all of web based sensitive information transactions. They are hard to detect but easy to do. It refers to the exploitation of valid computer session to gain authorized access to the information or computer system services. This paper demonstrates how the session hijacking done at network layer and application layer by stealing cookies. It is necessary and important to understand how this threat is working and design networks and applications that will be less vulnerable to it. This paper will be discussing about what are the protocols and tools are required to do it and its enactment. It is easy to avoid this attack only if the designer knows how the attacker can do.

Key words: session hijacking, session hijacking tools, man-in-the-middle attack, cookies, Wireshark

I. INTRODUCTION

Session Hijacking often referred to as an impersonation attack. It is a commonly used attack. A session is any interactive information interchange or conversation between two or more communicating devices or between a computer and user. It is a semi- permanent setup. It is established at a certain point in time and then torn down at some other later point of time. Each communication session may have more than one message in each direction between two communicating nodes in a computer network.

When user logs into any website, a session with session ID is created on the webserver for that user. Basically, sessions contain the entire user's information. For any page request, the user name and password are not needed. Each user has a unique identifier called as "session ID" or "Session Identifier" to authenticate particular user to the session. This session ID is passed between the web server and the user's computer system at every message request, reply and vice versa. A session ID is used to manage the session which is called "session token".

Session Hijacking is also called as "Session Side jacking", is a form of Man in the Middle attack in which a malicious attacker has access to the transport layer and can eavesdrop on communications. When communications are not protected attackers can steal unique session token and impersonate the victim on the target. This allows the permission to the attacker to access the account and data. The state of the session is maintained using cookies. Those Cookies are also used to identify the user. Hacker can simply sniff/capture/ stole cookies on the browser and impersonate the victim and carry out the activities on behalf of victim. Here the hacker has full access to the victim's account without knowing his user's details such as user name or password.

Session Hijacking is widely used to hack the website accounts such as E-mails: Gmail, yahoo, Facebook, twitter, rediff, and online banking transactions widely used for E-commerce, Point of Sale [4] et all. In those

applications session ID is stored in the form of cookies in the client's browser used to identify the user uniquely. To hijack some one's session, we need to steal the session information of that corresponding user.

There are three different types of session hijacking attacks [1] [5]: passive, active and hybrid session hijacking.

A. Active Session Hijacking:

Attacker takes control over the session by masquerade as a genuine user. This way of attacking is also known as Daniel of Service. Here, attacker puts the user into Offline mode and attacker is in stealth mode and monitors the packets over the network.

B. Passive Session Hijacking:

Attacker starts off with passive mode and listens all the data and captures them for future attacks. Disadvantage is Attacker might not succeed on the user to impersonating the server until user is alive, it will not if user logs off from the server.

C. Hybrid Session Hijacking:

In this the attacker implements both passive and active mode to successfully complete the attack. Here, attacker monitors the traffic and uses the session to impersonate. An Example of this attack is public unprotected wireless network (Here, attacker access multiple sessions and he has to wait for the right session and hijack the session from user). It is of two categories.

- (a) Blind spoofing-attack
- (b) Non-blind spoofing-attack.

Session Hijacking can be done at two levels.

- (a) Network Level
- (b) Application Level

Network level hijacking is a TCP and UDP session Hijacking and Application level is a HTTP session Hijacking. In application layer, a session ID is obtained through HTTP session hijack. This paper only considers the TCP and HTTP session hijacking. To defend a network with session hijacking, security measures have to be implemented at Application level and Network level.

Session Hijacking sometimes also known as "Cookie Hijacking". There are three types of security services authentication, integrity, and confidentiality are provided by securing the Cookies [1] [8] [9]. Therefore it is important to secure Cookies to avoid TCP session hijack attacks. Authentication verifies the cookies owner, Integrity protects against unauthorized modification of cookies, and confidentiality protects against cookies values being revealed to an unauthorized entity.

II. ARCHITECTURAL VIEW

Most of the communications are protected by providing credentials at session setup. Hijacking a TCP Session [7] is one of the categories of attack. Attacker neither intercepts nor injects data into existing communications between two

hosts. Instead, the attacker creates a new session (that means impersonates) or uses old ones done at both supplication and network level. The architecture of session Hijacking represents the steps of Session Hijacking [1] [3] [9].

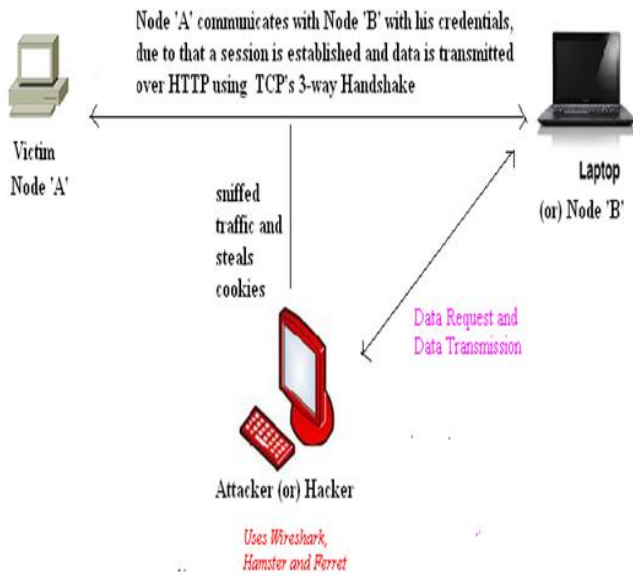


Fig. 1: Session Hijacking Architecture

TCP Session Hijacking is done in five steps are

- (1) Locating a target that needs to be attacked
- (2) Finding an active session that wants an attacker to be attacked
- (3) Observe the capture packet and analyze the stolen cookies and predict the sequence number
- (4) Make the user i.e. the victim system to be offline
- (5) Attacker takes over the session and maintains the connection

To perform the Session Hijacking: Wireshark, Hamster and Ferret are used in this paper as discussed in the Tools section.

III. PROTOCOLS AND TOOLS

A. Protocols:

1) HTTP:

HTTP (Hyper Text Transfer Protocol) [2] is a stateless protocol. Sessions are created and managed using HTTP protocol. User has to establish a TCP connection on server over a port number 80. Cookies are generated for each and every session while communication between two nodes in a network. Since when user logs into any website, a session with session ID is created and maintained on the webserver for that user. Which is the target for stealing sessions and this is the last stage of session hijacking.

2) TCP:

TCP abbreviation for Transmission Control Protocol is a reliable end-to-end connection oriented protocol in a TCP/IP network. To establish a session for a client to the server, a client must follow a structured system for session management; this system is called "Three Way Handshake". Formally TCP id defined in RFC 793 and extensions are given in RFC 1323. General TCP session establishment process is shown in the figure 2.

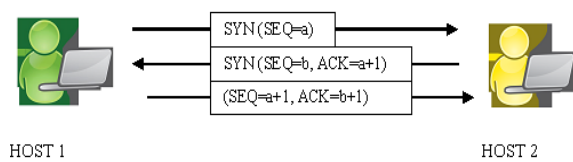


Fig. 2: TCP Three-Way- Handshake

Hijacking a TCP Session [7] is done successfully with sequence numbers used and transfer along with the TCP communication process.

B. Tools:

Some of the tools used in session hijacking are Hunt [11], T-Sight, Juggernaut, TTY Watcher, Firesheep, IP Watcher, Paros HTTP Hijacker, Hamster and Ferret, Wireshark and Ethereal and so many to hijack TCP based sessions. In this paper, the session hijacking software packages such as Wireshark, Hamster and Ferret are used to enactment of session hijacking is as shown below section. Tools used for session hijacking on attacker's side are presented in the table1.

Tool Name	Operating System	Type	Purpose
Wireshark/ Ethereal	Windows/ Linux	Open Source	Packet Capturing Tool
Ferret and Hamster	Windows/ Linux	Open Source	Hijacking Tool {Side Jacking}
Back Track	Linux Platform	Open Source	Hijacking

Table 1: Tools Used For Session Hijacking

1) Wireshark:

Wireshark is an open source software project released under the GNU General Public License. In 2008, Wireshark is released with first Wireshark [6] Developer and User Conference, called SharkFest. Another name for Wireshark is Ethereal. It is most popular security test tool [6] or software package used for trouble shooting, capturing and analyzing packets that are flows in a network.

Wireshark is not an intrusion detection system. If some strange things on network may happen then Wireshark does not warn but it helps to figure out what is really going on in a network. It will not manipulate things whereas; it will measure things on the network.

2) Hamster And Ferret:

Hamster and Ferret [12] [13] are used to steal cookies of TCP session. In this paper, Session Hijacking attack is done on a victim machine by impersonates the user and access the searched webpage from our attacking machine.

Protection against Session Hijacking can be done by using encryption, use a secure protocol, limit the incoming connections, minimize remote access, educate the employees, be cognizant to pay attention to last logon time and place and secure internal network devices.

IV. EXPERIMENTAL RESULTS

Consider a victim machine that is searching the data on Google search page by sending a request to the Google server. Aim is to hijack this session and display the victim's browser on attacker's web browser. Assume both Victim and attacker is using the Internet Explorer (IE) browser.

An overview of Wireshark main window is for capturing packets in a network to hijack a TCP session is shown in Fig. 3.

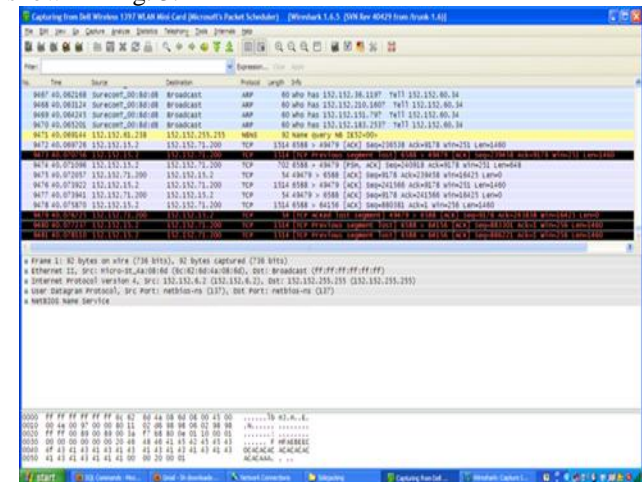


Fig. 3: Wireshark Main Window

Selecting an interface in a Wireshark to capture a particular system in a network is going to be captured is as shown in the Fig. 4.

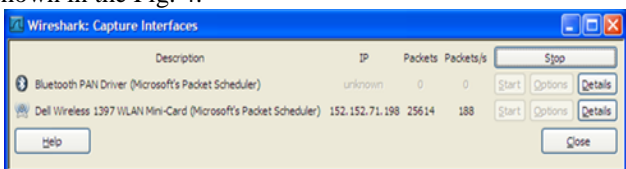


Fig. 4: Selecting an interface in Wireshark

After selecting the interface to capture and analyses the packets through a particular node and find the sequence number of an attacker host is shown in the Fig. 5.

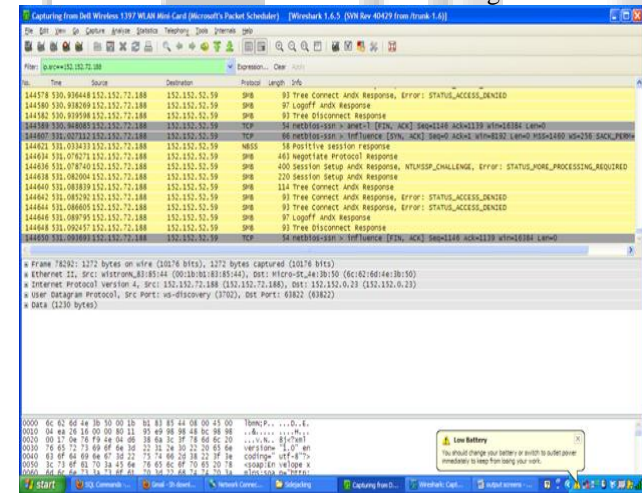


Fig. 5: Capturing the traffic of the victim user's browser

Capture the traffic of the victim user browsing of some query on Google search page. After capturing, the file named in this example is packet_info2.pcap as shown in Fig. 6.

It will need to save the captured file into the Hamster directory. When that file is in place, I use to process the file with Ferret as shown in Fig. 7. Ferret will process the file and create a hamster.txt file that may be used by Hamster for the actual hijacking of a TCP session. Start the Hamster which executes the available processed data on Ferret. Hamster itself runs as a proxy that provides an interface for browsing and it can be used as a stolen

session cookies. Hamster can easily started without command line options as shown in Fig. 8.

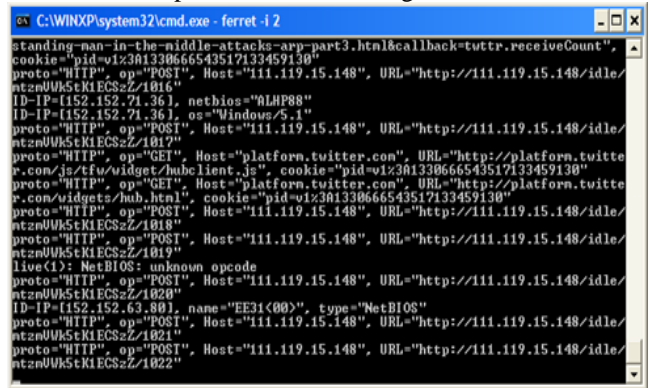


Fig. 7: Processing the captured file with Ferret

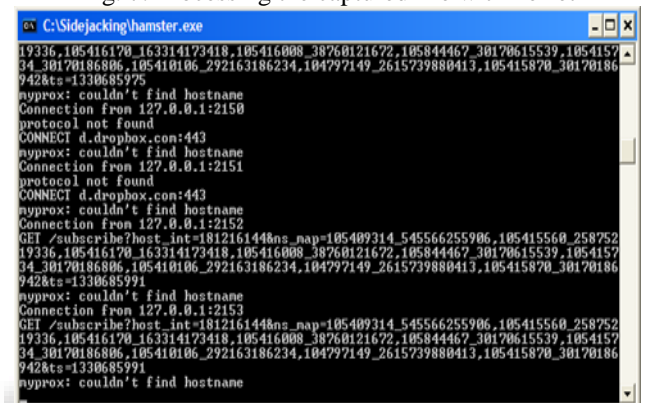


Fig. 8: Starting Hamster

After capturing all the packet information and processing the Ferret console is closed automatically as shown in Fig. 9.

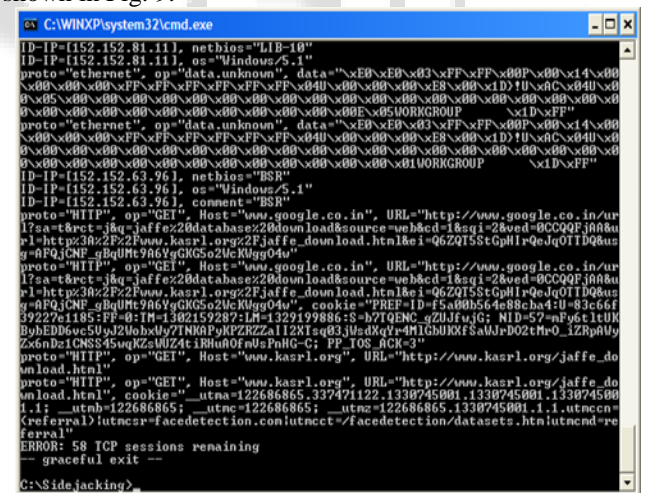


Fig. 9: After processing cookies, exiting Ferret

Now proxy settings have been applied to IE Browser window. We can access the Hamster console in your browser by browsing http://hamster as shown in Fig. 10. Hamster will use the file created by Ferret to produce the list of IP addresses of intercepted sessions and displays those IP addresses in the right panel of the browser and left panel of the browser is populated with the sessions available for Hijacking as shown in Fig. 10. But our file only contains a single IP address of the victim machine, so select to choose for clicking a single session that is going to be hijacked.

