Global Journals LaTeX JournalKaleidoscope™
Artificial Intelligence formulated this projection for compatibility purposes from the original article published at Global Journals. However, this technology is currently in beta. *Therefore, kindly ignore odd layouts, missed formulae, text, tables, or figures.*

# State of the Art Survey on Session Hijacking

Parves Kamal[1] and Parves Kamal[2]

[1] Saint Cloud State university

*Received: 7 December 2015 Accepted: 3 January 2016 Published: 15 January 2016*

## Abstract

With the advent of online banking more and more users are willing to make purchases online and doing so flourishes the online E-Business sector ever so more. Attackers are ever so vigilant and active now on web than ever to leverage the insecure web application and database that is out there on the internet to exploit. Today?s internet as we see are heavily integrated with sophisticated network whether it?s wired or wireless network. But the inherent compliancy to not integrating security while developing application leave it vulnerable to many attacks. One of the attack that has been prevalent now-a-days is: session hijacking.

***Index terms***— session-hijacking, CIA, spoof attack, CSS, SSL, captcha etc.

## 1 Introduction

here is various security threats that lurks around the internet. Especially in this age of Internet everything is connected to internet. Online E-Commerce heavily rely on online transaction for example bank provides users easy way of managing their account online. As the sensitive information passes around the internet the confidentiality, integrity and availability of such information become increasingly hard to protect. One needs to develop capable defensive mechanism to keep all the threats that poses threats to the CIA (Confidentiality, Integrity, and availability) of the information. Security threats like manin-the-middle attack, sniffing, Denial-of-service attack, ARP spoofing, session hijacking are some of the most prevalent attack performed daily by numerous attackers around the world on the internet.

A recent study performed by company Stake (Owned by Symantec) shown that 31% of e-commerce applications are vulnerable to session hijacking **??**Morana, Marco]. In the paper below I will go details on the session hijacking attack by giving the literature review of this attack. Also I will simulate the attack methodology to understand the mechanism better and finally will provide the general protection strategies for mitigating such attack.

## 2 II.

## 3 Literature Review

As we will be looking into the session hijacking let's get bit of background on what is session hijacking and how it works.

Session hijacking or Session Sidejacking both means taking over unauthorized already created trusted session in order to steal or compromise user's data. It's a well-known man-in-the-middle attack. A valid user who successfully logged into the webserver creates a session between him and the server. In session hijacking technique the attacker takes the control of the valid session from the user and replay packets to the server pretending to be the real user [Whitaker, A., & Newman, D. (2006)]. The advantage of such attack is that the attacker do not have to break into the defense of any firewalls, Intrusion detection system instead he/she can just listen to the network and take over any valid session.

One of the reason behind successful rake over such session is because of the way the server and the user authenticate themselves initially. In many cases only the server authenticate itself to the client in secure channel over HTTPS during the initial authentication phase and after the authentication the rest of the communication is done in clear plaintext. Session hijacking are of three types:

? Active session Hijacking ? Passive session Hijacking ? Hybrid Session Hijacking

# 4 a) Active session hijacking

In active session hijacking the attacker tries take over active session between the user and the server by either putting off the valid user from the connection and start making connection to the server masquerading as the valid user. The way attacker put off the valid user is by putting the active user out of the connection via Denial of service attack. Before making the valid user out of the valid active session he/she captures data that is sent back and forth between the user and the server by putting himself in between the connection between the connections and sniffing the data by packet capturing tool like Wireshark. In the figure below we see the three packets highlighted which is TCP three way handshake packet that are used to authenticate client to the server during the initial authentication session as shown below: In passive session hijacking the attacker captures all the packet between the user and the server and it send out valid packet to the user masquerading as server and same way sending packet to server masquerading as user. It's also referred as sessionreplay attack where the attacker basically replaying packets captured from the user and sending it to the server. The disadvantage of such attack is that the attack is valid until there is valid session still in continuation. If for some reason the server resets the connection or user logs off from the server the session will be terminated.

As shown in the figure above the attacker is replaying packet between user and the server and it modifies the packet as it goes from user to the server.

? Blind Spoofing attack ? Non-Blind spoofing attack

# 5 d) Blind Spoofing attack

In blind spoofing attack the attacker attacks the target machine without tempering with the connection. It simply captures all the packets between the client and the server and it tries to guess the TCP packet sequence number so that it can authenticate with the server. The problem with this type of attack is it's very hard to guess the TCP sequence number as it can be very random number which makes it harder to guess. Also its time consuming and the attacker might need to wait long time to get success with this type of the attack.

# 6 e) Non-Blind spoofing attack

In non-blind spoofing attack the attacker can actually monitor the traffic between the user and the target server. This way it's easy for the attacker to guess the next packet in case if it wants to guess the TCP sequence number of the next packet. It's hard to implement in today's network as the administrator now turns off the broadcast packet transmission around the network so unless the attacker can make the networking devices like switch and router to restart itself so it can capture the broadcast packet or by poising the CAM table of the switch it can place itself in the routing table and reroutes packet to itself for packet capturing.

In application level the attacker hijack the session as well as tries to create new session with newly constructed session ID's which can be stolen or guessed or crafted in a such way that it validates the attacker with the target machine to take over existing session or create new session [Sans.org,. (2015)].

The session ID's can be found in place like: [Ollman, Gunter] ? In the HTTP GET request that is made when clicking on the embedded link on the web page. ? When any HTTP post command issued typically with form that post data from client to the server. The session ID is hidden inside the form in the hidden field.

? Also the cookies are used to hold session ID's.

# 7 f) Obtaining Session ID's g) Sniffing

One of the way the hijacker can steal session ID'S are by sniffing out the network traffic just like taking over TCP session. This way the attacker monitors traffic to see if there is any unencrypted packets traversing and by finding so it can redirect the traffic through a host that it can monitor. Unencrypted traffic often has session ID inside and attacker can easily get the session ID and use it to take over already established session or create new session.

# 8 Fig. 3 : Passive Session Hijacking c) Hybrid Session Hijacking

In hybrid session hijacking the attacker uses both passive and active mode to complete the attack.

The attacker monitors the traffic pattern between the user and the server and wait for the right session to take over.

This type of session hijacking relies on spoofing and it can be further categorized to two types:

There are number of ways anattacker can steal session ID'S. Some of the ways are described below:

# 9 h) Brute Forcing

Another way the attacker can get the session ID is either guessing the session ID's or by attempting different session ID until it gets the right one. It can be automatic attack where attacker sets up certain pattern and it looks through all the patterns until it finishes. This type of attack is particularly successful if the session

ID number generation is not Random number and there is high chances the attacker will guess the session ID correct.

# 10 i) Misdirected Trust

Another form of attack where what attacker does is HTML injection or CSS (Cross Site Scripting) attack to misdirect valid traffic to the attacker. This way it can steal the session ID as the data is sent back from server to the host. This sort of attack relies heavily on the vulnerability of the web application on which this attack is performed since the success of the HTML injection and the CSS attack depends on the defensive mechanism of the web application it is attacking to. j) Tools Used For Session Hijacking Some of the tools used to steal session Hijacking are:? Hunt ? T-Sight ? Juggernaut ? TTY Watcher ? Hamster and Ferret ? Wireshark ? Ethereal III.

# 11 Attack Methodology

Session attack methodology can be shown in following steps as shown below in the figure

# 12 Fig. 4 : Session Hijacking Steps

We will be showing a session hijacking in a simulated environment in Virtual Environment where the set up will be as follows:? Victim Machine (Windows 7 VM) ? Attacking Machine (Kali Linux VM) ? Sniffed Router/Switch

The Tool we will be using for carrying out the attack are as follows:? Kali Linux ? Ettercap ? Hamster And Ferret

The kali Linux tool will be used as attacking machine to sniff out the traffic from victim machine which is windows 7 VM and Router. Our simulated Attack looks like following below: We will be stealing HTTPS connection from The VICTIM to get the USER Login and Password he/she put in.

For our demonstration purposes the IP network configuration is as follows:

The

# 13 b) Setting up Attacker Machine

We need to at first set up Attacker machine Kali Linux the Men in the middle between the router and the victim machine Windows 7.

We at first check our connectivity from Attacker machine to the Victim machine by pinging our victim machine as shown below: Now in order to crack HTTPS connection we need to have SSL strip in the attacker machine. So we type in the following command in our attacker machine and Press Enter after each command above: SSLstrip Download Code: cdcurl http://www.thoughtcrime.org/software/sslstrip/ssl strip-0.9.tar.gz > sslstrip-0.9.tar.gz tarxzf sslstrip-0.9.tar.gz cd sslstrip-0.9

Now we need to forward the Traffic generated in HTTP by forwarding the IP traffic by NAT forwarding in our Attacker Machine.

We do that by uncommenting the net.ipv4. ip_forward=1line inside the/etc/sysctl. conf file.

We do that by following command cp /etc/sysctl.conf /etc/sysctl.conf.bak vi /etc/sysctl.conf We find thenet.ipv4.ip_forward=1 line and uncomment it. Then we save the file CONTROL+X and save it. Now we need to set up IP tables Rule in the command prompt of the attacker machine as follows: iptables -t nat -A PREROUTING -p tcp –destination-port 80 -j REDIRECT –to-port 8080 iptables -t nat-L We see from following figure the output of the iptables we configured above Fig. **??**: IP forwarding Now we need to set up SSLtrip to act as sniffing between victim and the attacker machine to strip any HTTP connections from the victim machine.

On the attacker machine we type in the following command to install the ssstrip Cd sslstrip-0.9 python sslstrip.py -p -l 8080

We need to keep the windows open as it will generates traffic as the victim machine browse to any webpages with its browser: Fig. **??** : sslstripsetup Now the sslstrip will generate its traffic captured from the victim's machine and save it to its logfile. So we need to monitor its logfile in order to capture information.

On the attacker machine we type in following command to open the log file and keep it open to monitor capture traffic from the victim's machine as shown below: cd cd sslstrip-0.9 tail -f sslstrip.log The victim did not see the login page of his online banking been strip down from HTTPS to HTTP as shown above.

# 14 Fig.12 : Login ID and Password Stealing by HTTP Session Hijacking

Now the attacker is inside the session as long as the victim's will be and do any further attack as he/she might find it useful.

# 15 IV.

# 16 Survey Analysis

A survey was done about the awareness of the Session hijacking. Between researchers, common users and the administrator. As expected the common users have very less knowledge about the session hijacking followed by the Administrator. Surprisingly the administrator though they knew about the session hijacking had very little knowledge on how to prevent it. For successful mitigation of session hijacking one needs to have awareness as well as secure operation policies implemented in the organizations. The graph below shows the session hijacking awareness between common user, administrator and the researchers.

# 17 Counter Measure to Session Hijacking

There are number of ways session hijacking can be prevented. The countermeasure against session hijacking discussed below provided are based on recommended session hijacking techniques [CEHv8. Ethical Hacking and Counter Measures].We will be dividing the session hijacking in two layer of OSI layer as:? Network layer ? Application layer VI.

# 18 Network Layer a) Use of SSL at all time

Use SSL connection whenever it's possible. SSL (Secure Socket layer) Provide end to end encryption which make it really hard for attacker to look into any data passing over this encrypted SSL channels uses public key and symmetric key which are of 128/256 bits. Since it provides the integrity as well as the confidentiality sniffing and loss of information is protected while using SSL connection.

# 19 b) Use SSH for Remote Connection:

Often the remote connection to network devices or web server is required for the administrator for remote administration. SSH can protect the network as it guards against the IP spoofing as well as the data is encrypted. An attacker if has access to the target network can force the connected SSH user out of the connection but he/she cannot replay the packet as the data will be encrypted **??**Webopedia].

# 20 c) HTTPS Connection Only

It is very important to use HTTPS connection while login to your webserver, or any E-commerce site like Online banking, shopping sites as it encrypts the data with SSL as mentioned earlier to encrypt the authentication data back and forth. Attacker even if is successful to capture data will not be able to make any sense out of the data.

# 21 d) Implementing IPSec Protocol in Network Layer

IPSec protocol ensures the secure exchange of the IP packet and it provides two protection service. In transport mode it encrypts the data of the packet while in tunnel mode it encrypts the data as well as the header of the packet making the attacker hard to guess where the packet is going and coming from.

# 22 e) IDS/IPS Implementation

Implementing IDS/IPS along with firewall with proper rules can detect IP spoofing, packet sniffing which is the key to the session hijacking at the network layer. For example the rule can be set up as ignoring source routed packets or even blocking the sourcerouting completely. ARP poisoning as shown above in the simulated attack can be prevented by implementing static ARP table or by monitoring ARP table with tool like "arpwatch". Other techniques like ICMP redirection disabling can make it even harder for attacker to perform the MITM (Men in the Middle Attack).

# 23 Global Journal of Computer Science and Technology

Volume XVI Issue I Version I 10 Year 2016

# 24 ( ) f) Application Layer

Application layer deals with attacks on Web as our attack involved in URL session ID hijacking we will see below the countermeasure that can prevent such attacks.

# 25 g) Strong Session ID

Session ID is key to authenticate, create, reestablish connection with server. Session ID key must be strong nor predictable and it needs to be truly random. The session ID management system both in the client side and the server side needs to implement strong session management system. Following are some of the steps that can be taken to generate strong Session IDs

? Making the Session ID Random -As mentioned earlier the more random the session ID is more it's harder for attacker to guess or brute force the session ID. For making robust random session ID one can put the session number generation to a statistical analysis test.

? Making The Cookie or the session ID longer -The longer the session ID is harder it will be to brute force against. It will be very difficult to brute forcing against session ID of 50 characters in given time.

? Use Server generated Session IDs -Often the client side use its own session ID's which is less vulnerable to session hijacking. ? Forced Log Out -There should be a mechanism to log out user and prompt for re-authentication for new connection that way the attacker cannot use the same session ID to take control of the session. So every new connection there should be new authentication and log out of the current authenticated user.

? Generate ID after the authentication -Often before the authentication is performed the session ID is generated and shared that way the session ID is exposed to the attacker and they can carry out session fixation attack. So for security reason the session ID should be generated after the authentication is done.

? Token Regeneration-Once in a while if the session token is regenerated it becomes hard for the hacker to remain in valid session as after certain time the session token becomes useless. Webserver can be implemented in a way to regenerate session tokens giving the attacker less time to be on a session [Martin Eizner, and Roy McNamara "A Guide to Building Secure Web Applications].

? Time-Out-Time out should be implemented after certain period of inactive time period so that the attacker cannot exploit any idle session.

? Proper Input Validation Checking -Proper form input validation checking needs to be implemented from the server side. Often the Cross site scripting, HTML injection vulnerability allows the attacker to take over the web application and thus exploiting the session.

? Detecting Session ID Brute Forcing attacks -OWASP suggest using booby traps session tokens to detect any brute forcing on session ID token.

[Search Software Quality. ( **??**015)]. It's a token which is attached to the actual session token to detect any brute force on tokens. User should be aware of why using encrypted connection always, when to use proxy, VPN connection or to have strong password set up for their online account etc. All these will add up to the better safe environment against session hijacking.

# 26 VII. Observations & Recommen Dations

In this paper the simulated attack on CITY bank session hijacking was analyzed from literature and practical point of view and also the countermeasure to such attack was explored in the end. The actual attack though did not yield in catastrophic effects but the researcher was startled to see how attacker was able to easily get into the victim's session just by modifying Cookie or session ID changes. Such attack can further exploits vulnerable system inside the bank's infrastructure which can enable the further severe exploitation to be successful. The hacker can get the users data and email ID. Nonetheless it's been projected that the user data loss will prosper further scamming and fishing attacks. The general recommendation to prevent such further attack encrypted and longer session ID with time out and effective IDS/IPS with Brute forcing detection mechanism to deter any attacker in carrying out such attacks in future.

# 27 VIII.

# 28 Conclusion

In this short survey paper we tried to have look at the session hijacking attack and its implementation with demo Attack. The attack carried out by the attacker though was not known in terms of details that much but the security expert stated it was due to session hijacking attack. Session hijacking has been on the rise on recent past mainly due to the users/developers/administrators lack of awareness and poor session management of some of the web application and servers on the internet. By putting the effective countermeasure mentioned in the countermeasure section of this paper one cannot fully prevent such attacks but can at least make attacker to come harder and use some other tricks rather than the usual attack performed in this paper. Also it's recommended to test the defensive mechanism that are in place and also monitor to deter, prevent and counter attack on such attacks if ever take place. [1] [2] [3]

[1] © 2016 Global Journals Inc. (US) 1

[2] © 2016 Global Journals Inc. (US)

[3] Year 2016 ( )

12 


Figure 1: Fig. 1 :Fig. 2 :E

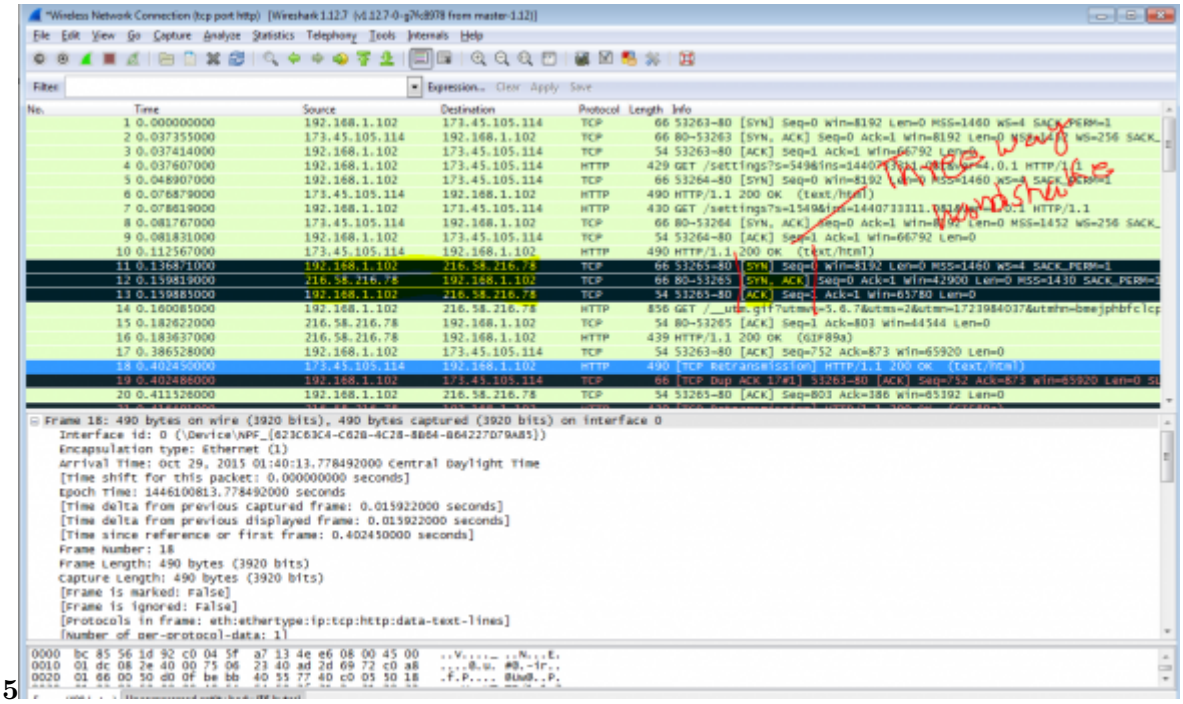
5 

Figure 2: Fig. 5 :

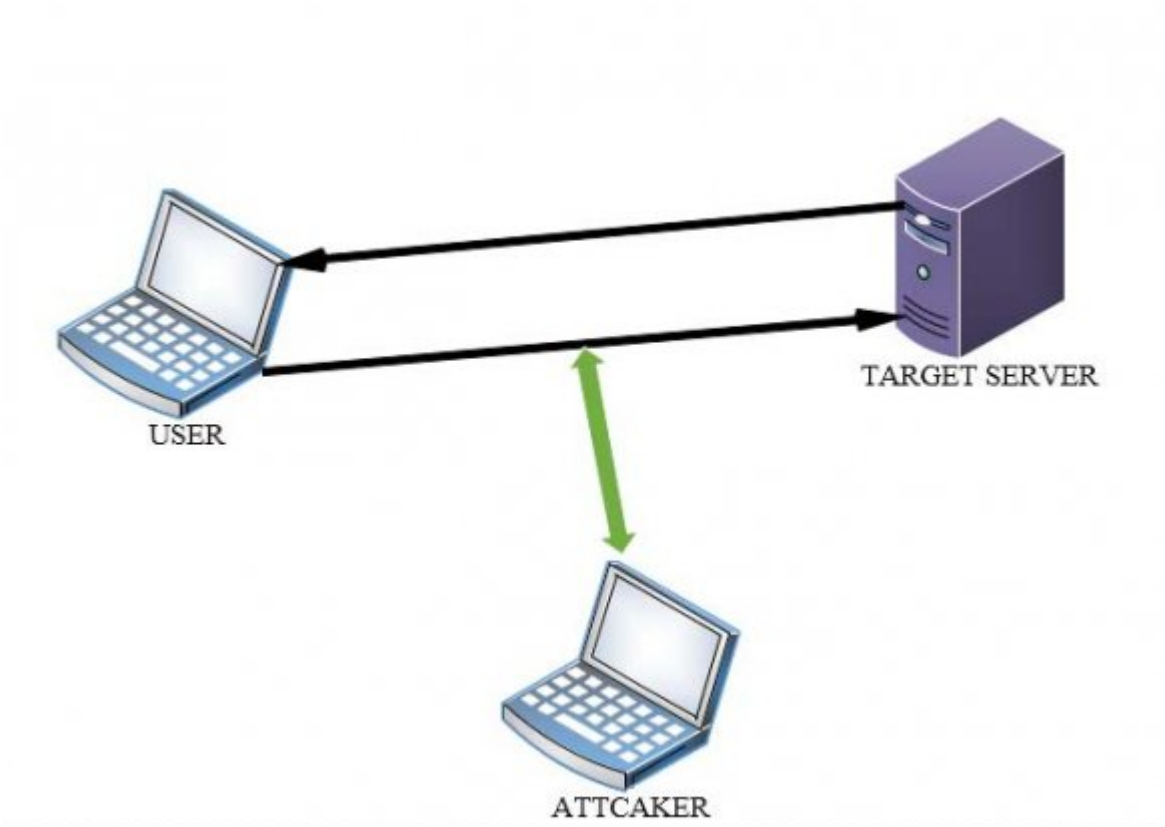
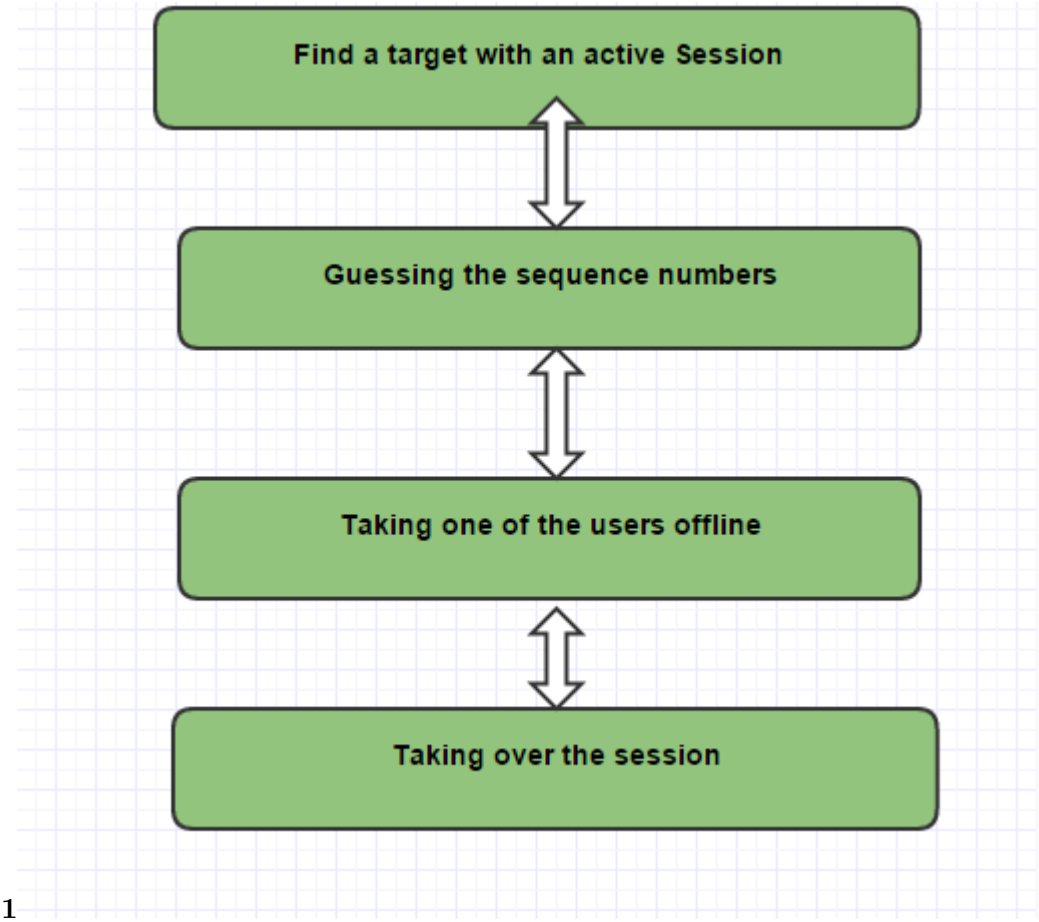



Figure 3:

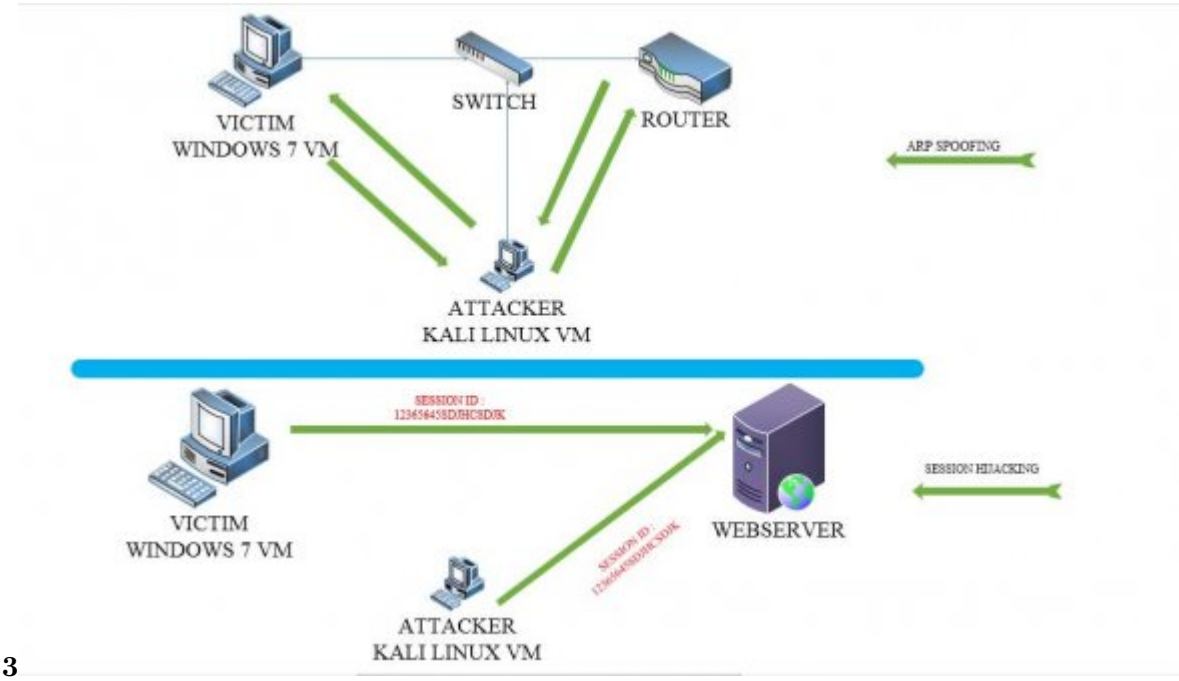

**6**

Figure 4: Fig. 6 :




**91011**

Figure 5: Fig. 9 :Fig. 10 :Fig. 11 :




**13**

Figure 6: Fig. 13 :

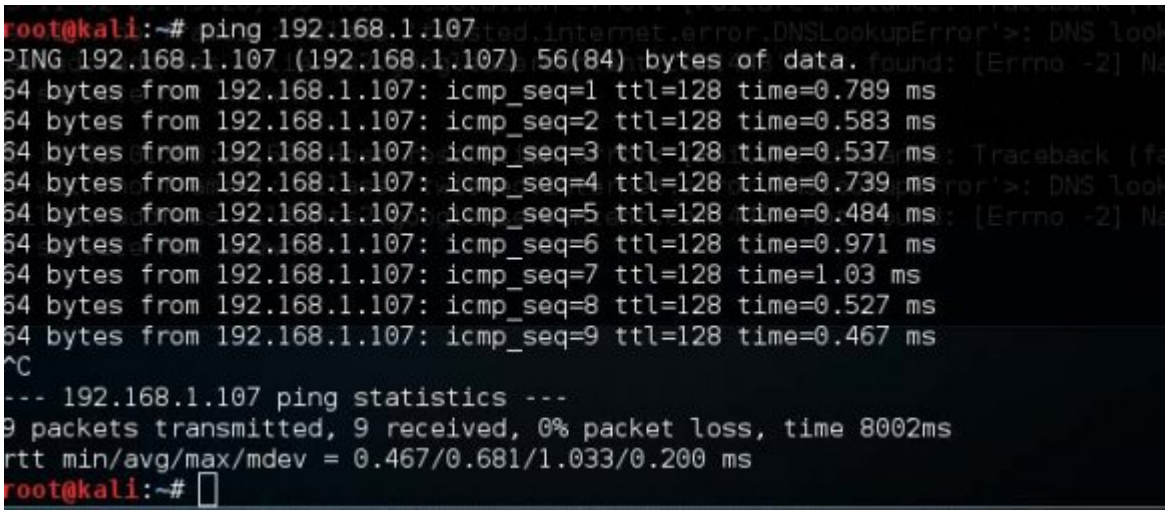



Figure 7:

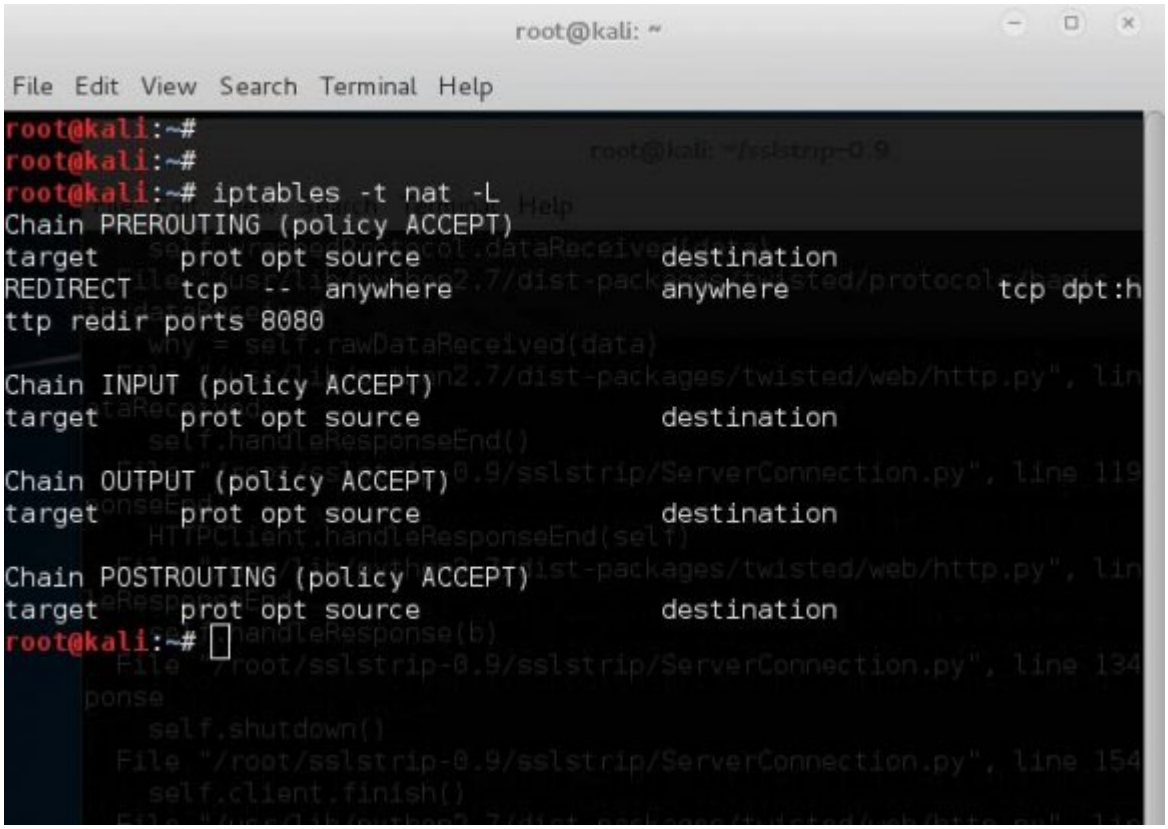



Figure 8:

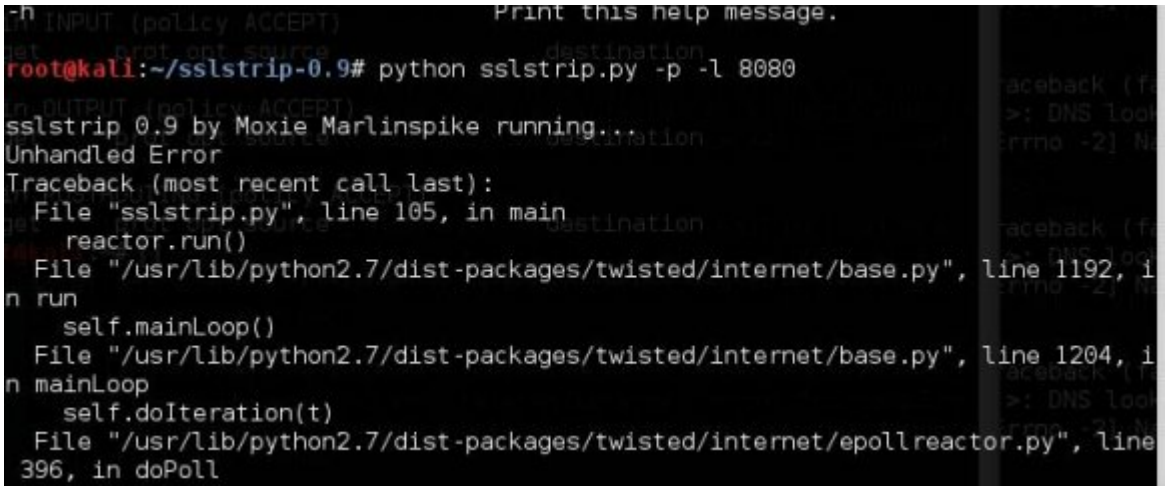



Figure 9:

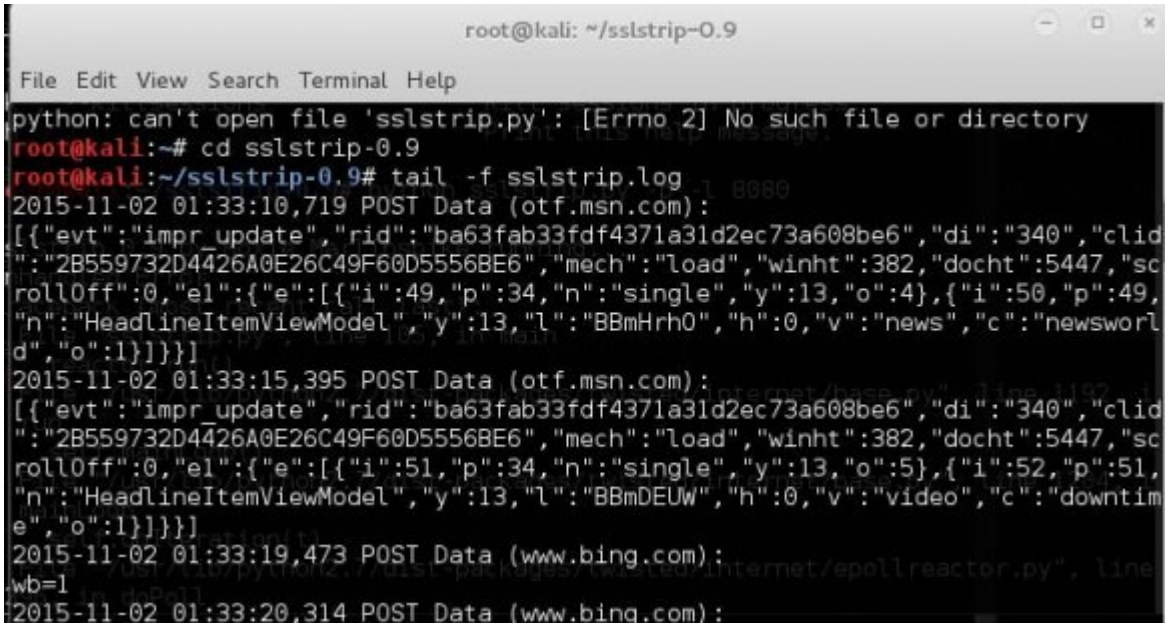



Figure 10:

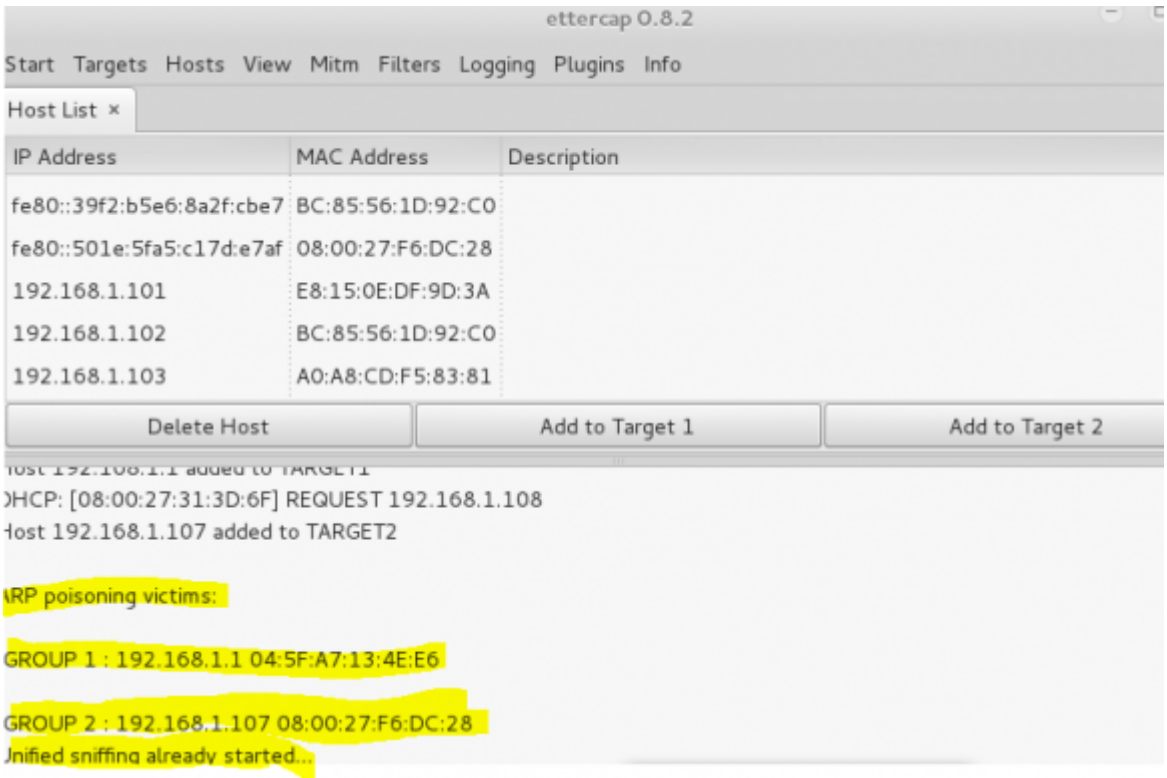

ettercap 0.8.2
Start Targets Hosts View Mitm Filters Logging Plugins Info
Host List ×
IP Address
MAC Address
Description
fe80::39f2:b5e6:8a2f:cbe7 BC:85:56:1D:92:C0
fe80::501e:5fa5:c17d:e7af 08:00:27:F6:DC:28
192.168.1.101 E8:15:0E:DF:9D:3A
192.168.1.102 BC:85:56:1D:92:C0
192.168.1.103 A0:A8:CD:F5:83:81
Delete Host
Add to Target 1
Add to Target 2
)HCP: [08:00:27:31:3D:6F] REQUEST 192.168.1.108
Host 192.168.1.107 added to TARGET2
ARP poisoning victims:
GROUP 1 : 192.168.1.1 04:5F:A7:13:4E:E6
GROUP 2 : 192.168.1.107 08:00:27:F6:DC:28
Unified sniffing already started...

Figure 11:

[Sans and Org (2015)] , Sans , Org . https://www.sans.org/reading-room/whitepapers/ecommerce/overview-session-hijacking-network-application-levels-1565 2015. 30 October 2015.

[Louis (2011)] *# References Références Referencias 6. CEHv8. Ethical Hacking and Counter Measures*, J Louis . Accessed: 10-. https://www.wiziq.com/tutorial/714466-CEHv8-Module-11-SessionHijacking 2011. Oct-2014. University Of Bedfordshire (Session Hijacking Module 11)

[Curphey et al. ()] *A Guide to Building Secure Web Applications*, Mark Curphey , David Endler , William Hau , Steve Taylor , Tim Smith , Alex Russell , Gene Mckenna , Richard Parke , Kevin Mclaughlin , Nigel Tranter , Amit Klien , Dennis Groves , Izhar By-Gad , Sverre Huseby , Martin Eizner , Martin Eizner , Roy Mcnamara . 11 Sept. 2002. 20 Dec. 2004. (The Open Web Application Security Project)

[Do you need to encrypt session data Security.stackexchange.com. Retrieved (2015)] 'Do you need to encrypt session data'. http://security.stackexchange.com/questions/18880/do-you-need-to-encrypt-session-data *Security.stackexchange.com. Retrieved*, 2015. 1 November 2015.

[Zarei ()] *IMPROVE CAPTCHA'S SECURITY USING GAUSSIAN BLUR FILTER*, Ariyan Zarei . http://arxiv.org/ftp/arxiv/papers/1410/1410.4441.pdf 2014. (Accessed: 15-Dec-2014)

[Morana (2004)] *Make It and Break It: Preventing Session Hijacking And Cookie Manipulation*, Marco Morana . http://nwc.securitypipeline.com/howto/53701241 23 Nov. 2004. 20 Dec. 2004. (Secure Enterprise)

[Webopedia ()] *Online Computer Dictionary for Computer and Internet Terms and Definitions*, Webopedia . 2004.

[OWASP Guide to Building Secure Web Applications and Web Services Session Management (2015)] 'OWASP Guide to Building Secure Web Applications and Web Services'. http://searchsoftwarequality.techtarget *Session Management*, 1156684/OA SP-Guide-to-Building-Secure-Web-Applications-nd-Web-Services-Chapter 2015. 1 November 2015. Search Software Quality (11-Session-Managem ent)

[Whitaker and Newman ()] *Penetration testing and network defense*, A Whitaker , D Newman . 2006. Indianapolis, IN: Cisco Press.

[Ollman (2004)] *Web Session Management: Best Practices in Managing HTTP Based Client Sessions*, Gunter Ollman . http://www.technicalinfo.net/papers/WebBasedSessionManagement.html 20 Dec. 2004. (Technical Info: Making Sense of Security)