Prevent Session Hijacking Using Cookie Authentication

G.Suresh Kumar
Student, CSE Department
SRKR Engineering College
Bhimavaram

Sri. V.V Siva Rama Raju
Asst.Professor, CSE Department
SRKR Engineering College
Bhimavaram

***Abstract*— Session Hijacking is an attacking mechanism that allows an attacker to impersonate as a legitimate user to the server, thereby giving the attacker the same access rights as the legitimate client. It is one of the attacking methodologies of renowned security attacks such as Man-In-The-Middle. This project is set to discuss on some of the concepts of Sessioin Hijackin in Man-In-The-Middle attack and procedures to prevent the session hijacking with the help of forms authentication mechanism.**

***Keywords*— Session, Hijacking, Man-In-The-Middle, Attack, Authentication, Security, Attacker, Victim, Client, Server, Cookie.**

I. INTRODUCTION

The Internet is a place of abundant data and with the abundant information, there comes greater security risks as well. With several security threats that lurk round the internet, providing security becomes harder than anyone could ever imagine. Today, E-Commerce sector, largely rely on online transactions. . For example, Shopping websites provide users an easy way in order to manage their account online. Sensitive information such as usernames and passwords passes around the internet. Maintaining the confidentiality, integrity of such information become harder than anything. Security threats like man in-the-middle attack, Denial-of-service attack (DOS), Distributed Denial-of-service attack (DDOS), ARP spoofing, session hijacking, sniffing are some of the most dangerous attacks that were being performed daily by numerous attackers around the world. Among those security threats, some attacks such as man-in-the middle and session hijacking are considered as more dangerous because of their unique attacking procedures. According to a recent study conducted by a security giant, who is owned by Symantec, shown that 31% of e-commerce applications are most vulnerable to the session hijacking procedure [Morana, Marco]. These attacks mostly operates on attacker being impersonated as a victim and that's the reason why this attack is considered as the most dangerous of all and when the intrusion happens the server will not be able to know the legitimacy in the connection and attacker might possess a greater chance to access the sensitive information.

II. LITERATURE STUDY

Before diving into the details, we necessitate to know some collective terms and attacking methodologies in order to realize the project promptly. Session hijacking or Session Sidejacking both comes under the attacking methodologies of Man-In-The-Middle that means to strike over the authorized user and impersonate as the legitimate one to the server to steal or compromise user's data. In order to do it, the attacker should consider these measures. The attacker takes the control of the valid session information or session id is from the user and sends them to the server pretending to be the real user [Whitaker, A., &Newman, D. (2006)]. This type of attack is more advantageous as the attacker does not have to care about the defense measures of the victim such as firewalls, Intrusion detection system, etc. Instead of breaking, all these security holes in his way the attacker can just simply sit and listen to the ongoing network traffic and tends to capture the valid sessions.

These types of attacks over a network are called as Session hijacking. On that point are primarily three types of session hijacking methodologies known to be in existence, which includes.

- Active Session Hijacking
- Passive Session Hijacking
- Hybrid Session Hijacking

a) Active Session Hijacking

In this type of attacking methodology, the attacker tries to take over an active session, which is present between the legitimate user and the server by either disabling the valid or legitimate user from the connection and starts establishing connection to the server. Therefore,, he attempts to masquerade himself (Attacker) as the valid user. There are various ways by which the attacker can possibly make out this; one of the proven and efficient ways to disable the legitimate user is by attacking him via Denial of service (DOS Attack). Before disabling the valid user out of the validity from the server, active session the attacker captures the data that is sent to and fro between the user and the server. In this way, he (attacker) puts himself in between the connection that was established between the victim and the server. He sniffs the data by packet capturing tool. Wireshark was widely considered as one among those tools that helps to sniff the data that is going back and forth between the client and the server. Using a sniffer one can manage to sniff the traffic in the network and also manage to capture the data packets as well. These data packets are the principal origin of communication objects in between the client and the server. The individual who can hijack them has a possibility to obtain the sensitive data that is being sent. You can see the attacking procedure in the figure [1] given down below.

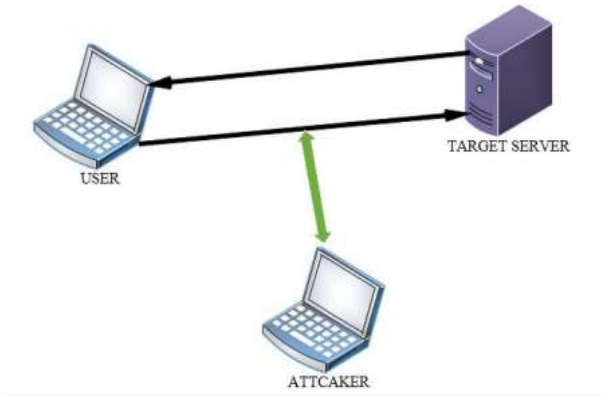

Fig.1: Active Session Hijacking

b) Passive Session Hijacking

Passive session hijacking works on the same prospectus as the active session hijacking. The only difference is that, passive session hijacking the attacker captures all the packets between the user and the server and sends out a valid packet to the user by masquerading himself (Attacker) as server and same way sending packet to server impersonating as a legitimate user. In this, the attacker work is only to monitor the ongoing traffic in between the server and the client. Unlike active session hijacking, passive session hijacking brings more security to the attacker. However, in this, he has to take care about the impersonation tactics very seriously or else there is a probability that he might become tripped up in the center. This procedure also suffers from a major disadvantage. The disadvantage is that the attack is valid until there is valid session still in continuation between the server and the user. If the server resets the connection for some grounds or if the user logs off from the server abruptly, then the session will be ended and the attacker might lose a chance to approach successfully. You can see the attacking methodology in the figure [2]. You can observe that attacker goes on to act as a legitimate client to the server and in the same way act as a server to the victim

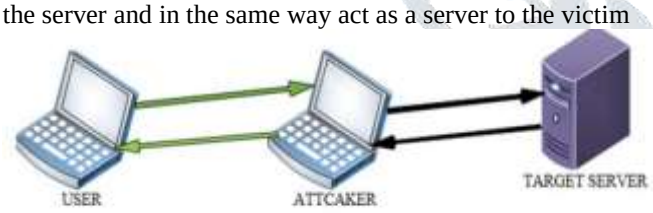

Fig.2: Passive Session Hijacking

You can keep the difference clearly here as the attacker here stays idle and just manages to capture data by getting the client staying in the connection, unlike the active session hijacking methodology, where the hijacker disables the client altogether.

c) Hybrid Session Hijacking

Hybrid session hijacking is a combined form of both passive and active modes. The attacker monitors the traffic pattern between the legitimate user and the server and waits for the right chance to take session over and apply it for his own use. In this, the attacker has to wait until he sees certain vulnerability in the ongoing network traffic and when it comes, he certainly calls for the chance and launches the attack along the network. Spoofing is considered as one of the main attacking prospective of this hybrid session hijacking. The spoofing attacks were further categorized into two types.

- Blind Spoofing
- Non-Blind spoofing

d) Blind Spoofing attack

In blind spoofing attacking methodology, the attacker attacks the victim machine without tampering any things with the ongoing connection. The attacker simply captures all the packets between the legitimate client and the server and then the attacker tries to guess the TCP packet sequence number. By guessing the sequence the attackers find a way to authenticate himself with the server through that number. There is a disadvantage with this attacking measure as it's very hard to guess the TCP sequence number due to a fact that the transmitted number can be any random number which makes it very hard to guess. It's time consuming and due to this the attacker might need to wait a long time in order to get success.

e) Non-Blind spoofing attack

In non-blind spoofing attack, the attacker can actually monitor the ongoing traffic, which is established between the legitimate user and the server. This helps the attacker to guess the next packet in case if it wants to guess the TCP sequence number of the preceding packet. It's hard to implement this type of attacking the methodology in today's network as the administrator now possess a mechanism to disable the broadcast packet transmission around the network so unless the attacker can make the networking devices like switch and router to restart on its own. There are also some places where the session ID's are known to be in existence. The places includes.

- In the HTTP GET request, where we get to visit a webpage by clicking the link embedded in a web page.
- During any HTTP post command being issued, Session Id will be hidden inside the form
- The third one is the cookies, which usually have the session id's.

You can certainly find the session id in any of these following methods. These are some of the session hijacking methods by which an attacker can gain access to the system, but what's sniffing.

f) Sniffing

Sniffing is a way which the hijacker can steal session ID'S. The attacker can do this by sniffing out the network traffic and this process is actually just like taking over a TCP session which we discussed earlier. Through this way the attacker can be able to monitor the traffic to see if there are any unencrypted packets present in between the server and the victim. Unencrypted traffic often has session ID's inside and if the attacker gains the session ID, he can utilize it to take over established session or he can also create a new session as well. Software such as Wireshark, Cain and Abel, Pcap are some of the software by which one can easily sniff the network.

g) Cookies, Session & Authentication

Cookies are small files which were being stored in the user's computer by the server in order to authenticate him. A session is a collective group of information on the server that binds with the associated cookie information. They usually reside on the server. Server denote the sessions with the help of ID's which are called as session ID's. Authentication is the action of proving something or some person to be true. It's an

act of verifying a person by the server. The server uses cookies in order to affirm the authenticity of the user. These three are active components of identifying the authenticity of the user.

h) Secure Socket Layer (SSL)

Secure Socket Layer is one of kind security layer which was present in between the client and the server in order to obtain a secured communication in between them. SSL contains encryption certificates and keys through which the communication line gets encrypted so that no one can be able to look for the information that is persisting to and fro between the host and the client. You can observe the way of SSL connection with the browser in the form of HTTPS in the url tab in the browser. Although SSL possesses a spacious range of advantages, it also got its faults as well. The demerits include higher cost and complex architecture. In order to protect a website from the intruder one should buy the SSL certificate and install.

i) Firebug Tool

Firebug is an open source extension which was being developed to work along with Firefox. It is practiced to do live debugging, monitoring of any website's HTML, DOM, JavaScript, XHR and CSS books on the site. Initially, it was developed solely for the design of web security testing, but later on it was being utilized for the process of tampering, unsecured logins, hijacking the session and so along. Firebug tends to make changes immediately and also gives constant feedback to the user who applies it.

III. Existing System

Many sites in the internet suffer from this type of session hijacking attacks because of a procedure which makes them vulnerable to these types of attempts. First and foremost, we have to observe that everything passes away over the HTTP protocol was not safe and due to this, HTTPS, which is an innovative form of HTTP with Secured Socket Layered protection was put in. To protect a website from Session Hijacking attacks, we must force SSL at all times which bring maximum security. But, passing our information only through HTTPS possesses many hurdles. The price in order to obtain force SSL at all time is very high and small web sites can't be able to afford it. Not just pricing, forcing SSL at all times increase the degree of overhead due to SSL architecture dependency on good server hardware, software, number of site visitors, and session duration as well. Hence, they come up with a solution for this problem, The solution that they came up is to use SSL only for the login page and by doing this they make sure that the user's credentials are transmitted safely to the web server, so that no intruder can gain access to those credentials from the outside. Later on the credential transmission, the web server authenticates the user by the means of an authentication cookie. This cookie is to help acknowledge the genuineness of the user and also act as a way for all his requests as well. But, there is a major drawback in this system which make the system vulnerable to the attack. The downside of this approach is that the authentication cookie is transferred to and fro from the server insecurely over http protocol, which in turn leads an attacker to gain access to the server by gaining access to them first.

IV. Attacking Methodology

In lodge to possess a well-defined discernment of this procedure, let's work through the attacking methodology by which the attacker gains access to the system using this vulnerability. You can observe the step by step procedure in the figure[3] down below.

Fig.3: Attacking Methodology

1. In step 1, the credential transmission (username and passwords) occurs in between the client and the server
2. In step 2, confirmation of the credentials that were communicated in the step 1 passes.
3. In step 3, A HTTP data request will be proceeded by the user to the host.
4. In step 4, HTTP response will be transmitted by the host.
5. Meanwhile, In step 5 the intruder capture all data, transmitted over http rest that are being performed in both step 3 and step 4. In this step he mainly concentrates on capturing the cookies that were being transmitted between the client and the server.
6. In step 6, Using those captured cookies he makes a request to the server imposing himself as the legitimate customer.
7. In step 7, web severs authenticates the attacker by considering him as authentic as he used the cookies which are utilized by the host in order to authenticate the actual user.

And then, that's how an attacker gains access to the server by imposing himself as the actual user. As for the real user, the attacker can block him with DOS attacks which entrance completely disable him to access the server.

V. Proposed System

In order to overcome this problem, I propose a solution. The suggested resolution is to use two cookies instead of one cookie for authentication purpose during the first HTTP response phase. The new cookie was called as secure cookie. Along with it, we can use asp.net form authentication to power it up. This process also looks same as like the one that was illustrated above, but the major change here in this procedure is that a new cookie called Secure Cookie was introduced along with the ordinary Authentication Cookie. In lodge to possess a

clear understanding about this project, a small website was acquired by asp.net which has the accompanying web pages.

- Login.aspx
- Logoff. aspx
- DefaultPage.aspx
- My Account.aspx

You can see the default page of the website in the figure[4] given below. To illustrate it clearly, there are two versions of this website. Secured one and the Unsecured one. In these both web pages, My Account is the web page that keeps back the sensitive date and we have to protect it against the attack. Merely, as there is no presence of a secured cookie, unsecured version of the site suffers from making access to that My Profile page, while Secured version will not afford access to it.

1.) Unsecured Version Demonstration - Let's consider how this work in Unsecured version first

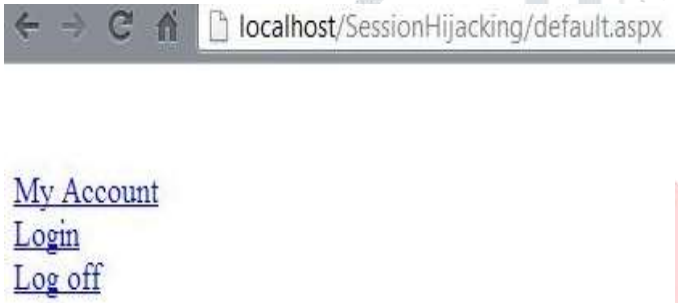

Fig.4: Default Web Page

Step-1: Lets login into the unsecured version of the website using credentials. You can see the demonstration of the login procedure in the figure [5].

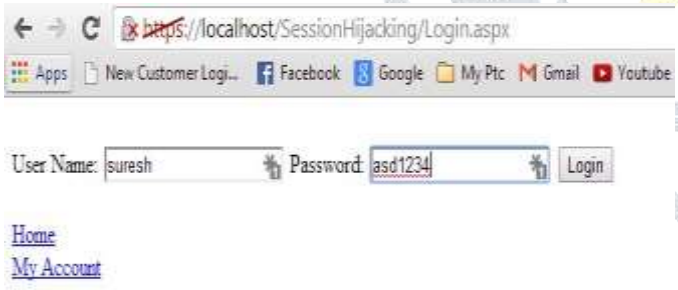

Fig.5: Default Web Page

Step-2: As there is no presence of the secured cookie and the authentication cookie is not being transmitted via SSL, the attacker can gain admittance to the protected page by simply capturing the authentication cookie and gain admittance to the protected page with the aid of browser extensions such as Firebug. You can distinctly see that My Account which we suffer to protect is being accessed with the authentication cookie being fed to the firebug tool.

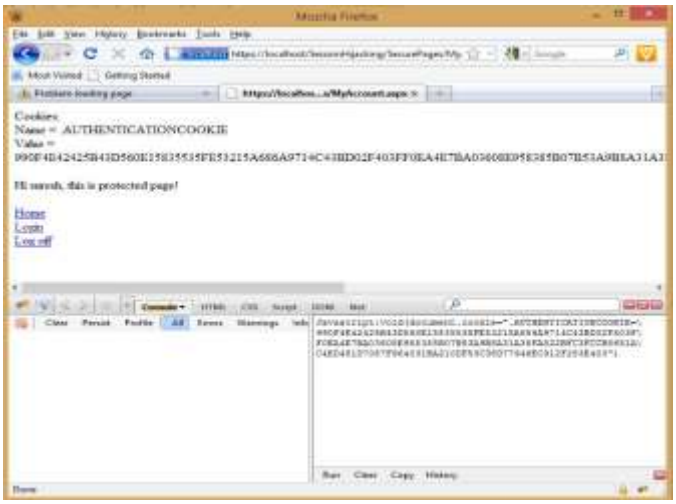

Fig.6: Firebug & Protected Page Access

2.) Secured Version Demonstration – Let's see how this work with the secured version of the website. We can switch between the secured and unsecured version of the website by changing the key value FORCE_SECURE to true which is present in the website code. Later modifying the version of the web site, we can see the Secure Cookie in action. You can observe this in the figure [7] given below.

Fig.7: Secured Version Demonstration

In the secured version of the website, Secured Cookie was transmitted via SSL and gets stored in the server and the Authentication Cookie was used for the request and reply purpose as usual. But since the Secured Cookie was transmitted to the server via SSL, the attacker gain access to Authentication Cookie only. When he tries to use this via Firebug to gain access to My Account page like he used it in the unsecured version, he automatically gets redirected to the login page once again. You can see the demonstration in the figure [8] given below. Instead of expressing the prompt "Hi, This Is Protected Page" it will simply redirect him towards the login page.

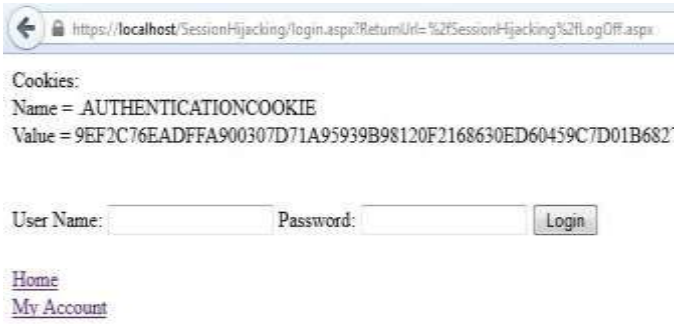

Fig.8: Redirect To Login Page

That's how we can block the attacking procedure using secured cookie. There are also some other background coding mechanisms that work in the background to make this procedure possible. Plans such as EncryptionHelper which is applied to encrypt cookies and Cookie Helper, to handle the functioning of cookies comes handy while providing protection to the site. As for the attacker, he gets to know the presence of the second cookie only when he creates an account of his own on the website and examining the working strategy of the website clearly.

VI. Survey Analysis

Agreeing to a study conducted by a renowned security organization, the quotient of the awareness of this particular approach was measured and the result went on to be really surprising. Agreeing to that view that took common people, computer administrators and researchers as the targets of their sight, the awareness quotient went from 0 to 10% only among the common people and while administrators went to somewhere close to 10 to 40 percent only, which shows that administrators also receive the least sum of knowledge on how session hijacking works. You can witness the study results in the image [9] down below. In order to obtain the successful mitigation of session hijacking attacks and methodologies, one needs to have greater awareness and also must possess a willingness to operate the tight secure operation policies implemented in the organizations. Not only that, we all know session hijacking was meant to be an attack for smaller websites whose security features are not up to the mark. But, many session hijacking attacks were recently being conducted on bigger websites as well such as Siemens and Gitlabs. Hence, Session Hijacking is considered as the nightmare above all the security attacks.

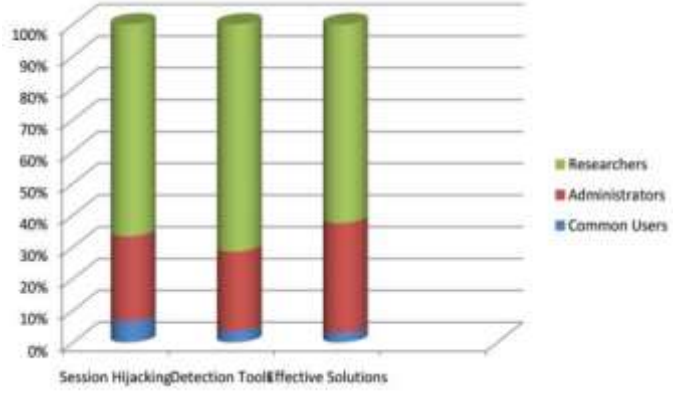

Fig.9: Session Hijacking Awareness [Louis, J. (2011)]

VII. Conclusions

Session hijacking is considered a serious threat to every network and web applications in the world wide web. In this paper, we tried to explain various aspects of session hijacking, and how one can gain access to the system by imposing himself as the legitimate client. In the universe of the internet where we can find out the existence of rapid development clearly, securing everything might just be a myth. Although, we strive hard to fix ourselves, there will unquestionably be a loophole in every method or system that we humans create and these loopholes can be chased down by the other individuals who possess the same knowledge just like the creators who created those systems. As long as creators and creations exist, destroyers also tend to go up as well. Thus, our goal is to attempt to transcend them in that race because without the destroyers there are no challenges, and without challenges there will be no advances.

References

[1] State of the Art Survey on Session Hijacking Written By Parves Kamal. Available Global Journal of Computer Science and Technology: ENetwork, Web & Security Volume 16 Issue 1 Version 1.0 Year 2016

[2] Morana, Marco. "Make It and Break It: PreventingSession Hijacking And Cookie Manipulation."Secure Enterprise. Online – http://nwc.securitypipeline.com/howto/5370124.

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

[4] Ollman, Gunter. "Web Session Management: Best Practices in Managing HTTP Based Client Sessions." Technical Info: Making Sense of Security. Availabilty Online In the website as following http://www.technicalinfo.net/papers/WebBasedSessionManagement.html.

[5] Webopedia: Online Computer Dictionary for Computer and Internet Terms and Definitions. http://www.webopedia.com/

[6] Cergis Blog Topic On Session Hijacking And Security Avaible Online: blog.cergis.com

[7] Louis, J. (2011). Detection of session hijacking. University Of Bedfordshire. Survey Analysis Graph.

[8] Sans.org,. (2015). Available Online https://www.sans.org/reading-room/whitepapers/ecommerce/overview-session-hijacking-networkapplication-levels-1565/