A review on prevention for Session Hijacking using One-Time Cookies

Neil Patel[1], Neel Patel[2], Manan Doshi[3], Yash Shah[4]

[1,2,3,4]*Computer Engineering Department, Shri Bhagubhai Mafatlal Polytechnic, Mumbai (India)*

ABSTRACT

In today's world, network security is in the limelight. Network security is an activity designed to protect the usability and integrity of network and the data flowing through it. Session hijacking attack in network security is one of the most effective attack attempted by adversaries. In this paper, we analyze the problems related to session hijacking and provide preventive measure to it. One-Time Cookie is one of the preventive measure for session hijacking. It is generated by binding the network layer and application layer through the reverse proxy server. This mechanism detects the change in browser due to which an adversary cannot get the illegal access. Since users are bind with machine and browser and with new disposable cookie for each request in the session.

Keywords* - *Session Hijacking, Web Security, One Time Cookie, Web Session Binding, Browser fingerprint

I. INTRODUCTION

Connecting people all over the world has become of utmost importance. Not only are they connected via GSM connections on mobile phones, but also through the internet. Internet is used for social networking sites, online transactions, online shopping, etc. As a result, it has led to increase in cyber- crime. There are various types of attacks a hacker or an attacker would perform on internet. The target of adversaries is to gain unauthorized control to cause interruptions, commit fraud, engage in blackmail or access private information.

Session management is a critical component of modern web applications, since it allows the server to track userspecific state, such as the authenticated account, across multiple requests. Unfortunately, many applications deploy session management over an insecure HTTP channel, making them vulnerable to eavesdropping, session hijacking or session fixation attacks. On the contrary, state-of-practice guidelines advocate the deployment of session management on a secure HTTPS channel, using the Http Only and Secure cookie attributes, effectively eliminating these wellknown session management attacks.

One of the dangerous attack in cybercrime is session hijacking. Session hijacking is also called as the man-in-themiddle attack. Session hijacking attacks are defined as taking over a TCP/IP communication session without their permission or knowledge. When implemented successfully, attackers assume the identity of the compromise user enjoying the same access to resources as a compromised user. Session hijacking can be implemented using the following fundamental scheme: Firstly, the user logins to the web application through valid authentication. These details are stored using cookies and delivered to the user, through an unsecured path.

This provides an apt way for attackers to sniff and get hold of these cookies. The attacker may then send erratic requests to the web application, thus, successfully hijacking the user's web application.

[1]Session hijacking attack is basically of two types:-Active session hijacking and Passive session hijacking. In an active session hijacking, an authenticated session is being hijacked. In

This method, the user already remains logged into the active session of his profile or account. The hackers try to steal the network cookies and thereby hijack the active session. The original user cannot further login into his/her profile and he is disconnected from the server. In this method of hijacking, the hacker does not attack any active session. They follow some different process to get the complete information of the login credentials of the user. When the user enters his login credentials on the system and tries to get access to his profile on the network, a hacker then steals his login credentials and hacks the user's account and profile information.

In this paper, we have analyzed a prevention technique for session hijacking. In this technique, we bind the network layer and application layer together through reverse proxy server. This reverse proxy server will generate session credentials such as session ID, browser ID and disposable one-time cookie (OTC). This mechanism detects the change in browser due to which an adversary cannot get the illegal access. Since users are bind with machine and browser and with new disposable cookie for each request in the session.

The flow of the paper is as follows. In section I we discuss about the literature survey. In section II we discuss about the threat model that have been assumed in the prototype. After threat model, we have discussed about the system overview and finally the working of session management.

II. HEADINGS

Web developers have periodically raised security concerns for session authentication cookies since their adoption. Several surveys [2] have demonstrated the various attacks possible with web authentication technique, including vulnerability to session hijacking attacks. As a result, security analysts have suggested ways to improve the robustness of cookies and session authentication. Park et al. [3] suggested cryptographic cookie mechanisms that provide better confidentiality and integrity to web session. In addition, Juels et al. [4] have proposed the use of cache cookie; which is a different form of persistent state in the browser. But the problem with these mechanisms is that they still rely on static cookie token to authenticate requests. Thus, security analysts have also explored the use of dynamic cookies to prevent arbitrary reuse.

A.THREAT MODEL

In this model, the attacker's aim is to establish the current session credentials of the legal user. It uses active and passive methods to sniff network traffic and capture the tokens. The attacker can modify the attack by removing the workstation computer from the session and using their identity for further use. In this technique, they have considered two ways by which attacker can modify the attacking methods. Firstly, the adversaries will try to inject the session credential by accessing the victim`s machine. Second, the adversaries will try to inject the session credential from remote machine. In victim machine, there is absence of plugins due to which session credentials are exploited. In remote machine, transmission is done over unsecured protocol. As a result, it the session.credentials are established.

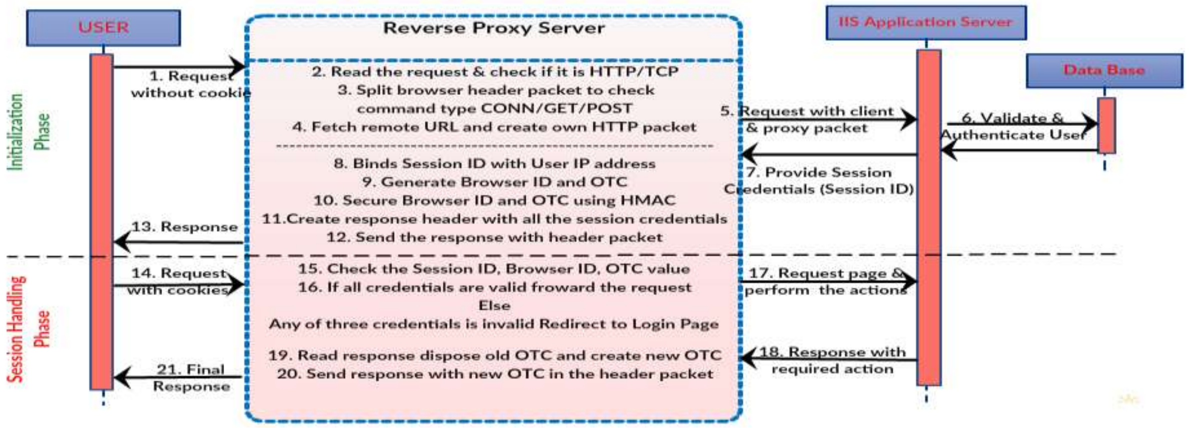
USER
Reverse Proxy Server
IIS Application Server
Data Base
Initialization Phase
Session Handling Phase
1. Request without cookie
2. Read the request & check if it is HTTP/TCP
3. Split browser header packet to check command type CONN/GET/POST
4. Fetch remote URL and create own HTTP packet
5. Request with client & proxy packet
6. Validate & Authenticate User
7. Provide Session Credentials (Session ID)
8. Binds Session ID with User IP address
9. Generate Browser ID and OTC
10. Secure Browser ID and OTC using HMAC
11.Create response header with all the session credentials
12. Send the response with header packet
13. Response
14. Request with cookies
15. Check the Session ID, Browser ID, OTC value
16. If all credentials are valid froward the request
Else
Any of three credentials is invalid Redirect to Login Page
17. Request page & perform the actions
18. Response with required action
19. Read response dispose old OTC and create new OTC
20. Send response with new OTC in the header packet
21. Final Response

Fig 1. One Time Cookie method: flow diagram

B. SYSTEM OVERVIEW

In this technique, the network layer and application layer are bind together in order to prevent the session credentials from hijacking. The technique uses a reverse proxy server that will generate session credentials. The session credentials include session ID, browser ID and unique disposable one-time cookie (OTC). . The session ID is generated by binding the application level and network level together to detect change of machine for current session. OTC and browser ID is again bind with session ID so that it can detect change of cookie or browser for current session. For each new session request it will generate new disposable cookies.

1. REVERSE PROXY SERVER:

It is a type of proxy server that typically sits behind the firewall in a private networks and directs client requests to the appropriate backend server. In this system we use this server to combine browser's session ID and network IP address with a proxy server. When a user requests, the proxy server will send the request to the user only if the user is validated by the system. When a request with the same application session ID of genuine user is used with a different machine, the system will know that the session is stolen. Thus it will remove the session cookie from the request, and the application session is invalidated.

2. ONE-TIME COOKIE:

One-Time Cookie are generated by the reverse proxy server for each request of the user. It provides more security than the alternative authentication cookies that does not require volatile state in web browser. OTC is generated by reverse proxy server for each request of the user and is disposed once it is verified by server.

Thus, the legitimate user is verified without storing any volatile data.

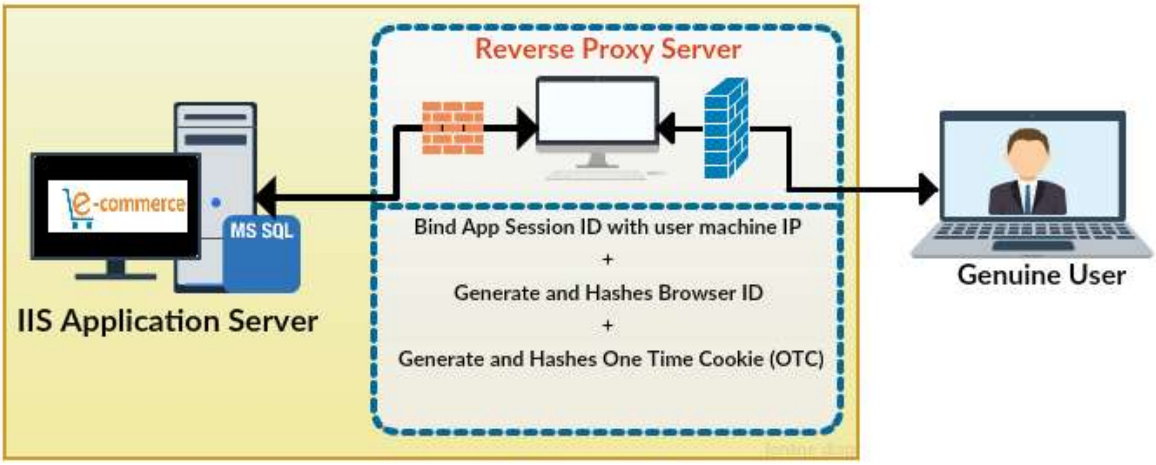
Reverse Proxy Server
e-commerce
MS SQL
IIS Application Server
Bind App Session ID with user machine IP
+
Generate and Hashes Browser ID
+
Generate and Hashes One Time Cookie (OTC)
Genuine User

Fig 2. Modus Operandi of Reverse Proxy Server to Manage Session Authentication

III. INDENTATIONS AND EQUATIONS

A. **Disadvantages**

i. Dependent on the Reverse proxy server

This method is dependent on the initial configuration and then continuous monitoring of the reverse proxy server's proper functioning for maintaining security integrity side of this method.

ii. Only 1 session/user this affects the web services that offer the facility of session creation from web applications and websites together.

B. **Advantages**

i. Insider session hijacking prevention

This method requires the adversaries to connect with the same reverse proxy server the victim is connected to, in which case they would not be able to hijack the session. This kind of network layout, with one proxy/reverse proxy acting as middle ware between end-machines, is common in organization work-groups and which makes this method appropriate for insider session hijacking prevention.

ii. Prevent social engineering attack

This method provides protection against malicious activities accomplished through psychological manipulation of people into performing actions or divulging confidential information ex. phishing attacks. Preventing the man in the middle attack ensures that no adversary can leverage people into false activities.

iii. Prevent browser theft

This method prevents multiple session logins through a single browser with the help of browser IDs

iv. Prevent CSRF attack

The three tired protection using session ID, browser ID and OTCs offering security at various levels of interaction. This ensures no false requests can be forged by attackers to achieve session riding.

v. Prevent XSS attack

Dynamic OTCs generated for each request-response prevents the session hijackers from using browser plug-ins to inject malicious scripts into vulnerable web applications by pretending to be the legitimate user.

vi. Support highly distributed environment

This method is suitable for scalable applications as with the increased number of users the security of the system is not compromised. Preventing multiple sessions from different machines or even a single host makes it possible to add any number of hosts to the network.

IV. CONCLUSION

To ensure a high-quality product, diagrams and lettering MUST be either computer-drafted or drawn using India ink. Figure captions appear below the figure, are flush left, and are in lower case letters. When referring to a figure in the body of the text, the abbreviation "Fig." is used. Figures should be numbered in the order they appear in the text. Table captions appear centered above the table in upper and lower case letters. When referring to a table in the text, no abbreviation is used and "Table" is capitalized.(Font size 10, Times New Roman) We have discussed about the current and updated general techniques to prevent session stealing that binds the session with network IP with the combination of Browser dabs (fingerprinting) and One Time Cookie (OTC). A secured communication channel is used for authenticating the user on the server side reverse proxy. Using this technique, the web application has effectively prevented session stealing attacks. This method is not dependent on secure protocol to protect the network credentials; also long-living cookies can also be used with our application. In this technique, OTC and browser ID, a secure alternative to authentication cookies. OTC is not only resistant to session hijacking, but also maintains the simplicity and improves performance of the cookies. Moreover, browser ID will offer another security layer to web applications by reducing the threats associated with cookies. This technique has significantly improved the security of web sessions with minimum impact on scalability and performance. This technique can be improved by just incorporating HTTPS protocol with the session credentials to handle the network connection.

REFERENCES

[1] Jerry Louis, “Detection of Session Hijacking,” in University of Bedforshire, January 2011.

[2] C. Visaggio, “Session Management Vulnerabilities in Today’s Web,” in IEEE Security and Privacy, 8:48–56, 2010.

[3] J. S. Park and R. Sandhu, “Secure Cookies on the Web,” in IEEE Internet Computing, 4:36–44, 2000.

[4] A. Juels, M. Jakobsson, and T. Jagatic, “Cache cookies for browser authentication (Extended Abstract),” in IEEE Symposium on Security and Privacy, 2006.

[5] A. Liu, J. Kovacs, and M. Gouda, “A secure cookie protocol,” in Proceedings of the International Conference on Computer Communications and Networks (ICCCN), 2005.

[6] C. Blundo, S. Cimato, and R. D. Prisco, “A Lightweight Approach to Authenticated Web Caching,” in Proceedings of the The Symposium on Applications and the Internet, 2005.

[7] Nick Nikiforakis, Wannes Meert, Yves Younan, Martin Johns, and Wouter Joosen, “SessionShield: Lightweight protection against session hijacking,” in 3rd International Symposium Engineering Secure Software and Systems (ESSoS 2011), volume 6542 of Lecture Notes in Computer Science, pages 87–100. SpringerVerlag, 2011.

[8] Rolf Oppliger, Ralf Hauser, and David Basin, “SSL/TLS session-aware user authentication – or how to effectively thwart the man-in-the-middle,” in Computer Communications , 29(12):2238–2246, August 2006.

[9] Willem Burgers, “Prevent Session Hijacking by Binding the Session to the Cryptographic Network Credentials,” in Institute for Computing and Information Sciences, Radboud University Nijmegen, The Netherlands. 2013.

[10] ItaloDacosta, “One-Time Cookies: Preventing Session Hijacking Attacks with Stateless Authentication Tokens,” in Converging Infrastructure Security (CISEC) Laboratory Georgia Institute of Technology.

[11] Unger, T.; Mulazzani, M.; Fruhwirt, D.; Huber, M.; Schrittwieser, S.; Weippl, E., "SHPF: Enhancing HTTP(S) Session Security with Browser Fingerprinting," in Availability, Reliability and Security (ARES), 2013 Eighth International Conference on , vol., no., pp.255-261, 2-6 Sept. 2013.

[12] Nattakant Utakrit. Review of browser extensions, a man-in-the-browser phishing techniques targeting bank customers. 2009.

[13] Chirag R Desai, Dr. Narendra M Shekokar, “Prevention of session hijacking attack using enhanced binding technique”