Technique to protect cookies using dual Protection Method

Prof. Swapnil Waghela
Department of Computer Science & Engineering, Acropolis Technical Campus, Indore, India

Prof. Nitisha Tiwari
Department of Computer Science & Engineering, SIRT, Sage University, Indore, India

Abstract-- Cross site scripting (XSS) is security vulnerability in World Wide Web. Information sharing between computers is increasing day by day. Web security is most important topic to be discussed. Web applications often use cookies to maintain authentication between the user and web application where XSS attack is a popular attack to steal cookies between client and server. By using XSS technique, attacker insert malicious script onto the application’s output. Some Web Application Vulnerabilities like Cross Site Scripting (XSS or CSS) and SQL Injections are happened because of poor input validation.

In this paper we are proposing a new and light weight technique for “Cookies Alteration”. This technique aims to repay the cookie useless for the attacker. Our technique is to implement a proxy that will recreate the cookie that is sent back and forth between the browser (Client) and server (web application). If attacker steals cookies then these cookies will be worthless for him. Our aim is not only to protect cookies but render the user’s data secure.

Keywords-- Cookies, Web Proxy, XSS Attack

I. INTRODUCTION

The Cookies are a mechanism to provide state full communication over HTTP\(^1\). Cookies are mainly used to store the session Id’s or personal information in today’s web applications. Cookies are sent by the web application as a part of the response message using Set-Cookie header. The browser stores cookie in its database, and includes the cookies with every subsequent request to the web application. This is shown in the diagram below.

Types of Cookies:
In general cookies can be classified into two types, this depends on a attribute of cookie that is their life time named as “expires” is represented by positive, or negative value \(^2\)\(^3\).

Types are:
- Session cookies are temporarily used; they are discarded when the browser is closed. They expire attribute is set with negative value.
- Persistent Cookie can be kept longer until they expire; they are stored on a disk and survive across a computer restart.

II. BACKGROUND WORK

One simple way is to disable cookies, but it may cause the web server denying to work without the cookies. Other techniques to protect cookies from the attackers \(^4\).
IP Mapping:
The web server maps IP addresses of users with cookies and denies any access that comes from invalid IP. This helps to shift the problem but it does not work where the users access the Internet through web proxy.

Http Only attribute:
Http Only attribute is a Microsoft extension, it can also be included in the cookies before being sent to the browser. With the Http Only attribute, the browser deny scripting language to access those cookies. The Http Only attribute is originally not a part of HTTP; the browsers that are not aware of this attribute will ignore it and consequently remain vulnerable.

Secure cookies:
Secure cookies mean that the clients and the web servers only send the cookies via the SSL connections.

No solution mentioned above guarantee that the cookie will safe from XSS attack. We propose a new approach which aims not to protect the cookies but instead render the cookie unusable for the attackers.

III. PROBLEM DEFINITION

we are facing is to save cookies from XSS attack, we have two type of attacks that can be used to steal cookies from system, the potentiality of these attacks can be judged as even the firewall cannot stop the scripts from being executed and once they are executed there is no chance to save cookies, the cookies will be sent to attackers system whose address is embedded in the script. More over we still have an option, if we close the browser it will not execute, the greater problem is without our permission browser will execute those script and even after our cookies are sent to the attacker’s computer we won’t know anything about it.

XSS Attack

Cross Site Scripting (also known as XSS or CSS) is generally believed to be one of the most common application layer hacking techniques. In an XSS attack, a Web application is sent with a script that activates when it is read by an unsuspecting user’s browser or by an application that has not protected itself against cross-site scripting. Because dynamic Web sites rely on user input, a malicious user can input malicious script into the page by hiding it within legitimate requests. Once XSS has been launched, the attacker can change user settings, hijack accounts, poison cookies with malicious code, expose SSL connections, access restricted sites and even launch false advertisements.

- Non persistent (or reflexive XSS) means that malicious code is not persistently stored in a vulnerable server, but it immediately echoed by the vulnerable server back to victim. This can be understood by an example: suppose a person is accessing www.bank.com in order to do an online transaction, at the same time the victim might also be accessing www.attacksite.com, and be persuaded into clicking the below link

  <a href="http://www.bank.com/" "document.cookie;"""
  Document.location=""http://www.attacksite.com/stealcookie.php?"
  +document.cookie;">"

  Click here to win a million Dollars</a>

The script will get executed and the cookies of www.bank.com will be sent to www.attacksite.com. The owner of attack site can use cookies to impersonate www.bank.com with respect to the person.

- Persistent (or stored) XSS means that the malicious code is persistently store in a server’s storage, and may later be embedded in an HTML page and send to the victim. This can also be explained by an example. Suppose a Script is posted on an online message board of www.bank.com

  Click Here To See A New Promotion
  
  <SCRIPT>
  Document.image[0]=
  </SCRIPT>

The victim who reads a message will receive the malicious script as a part of message. The victim’s browser will then execute the malicious script which will later send the cookies of www.bank.com to www.attacker.com

  The main problem with this is we cannot stop script to run once it has enter with html script, browser will run the script and the coding of script will send our cookies to the attacker.

IV. SOLUTION DOMAIN

We propose a new technique for cookie protection named “Preventing XSS attack by cookie Modifying USING A PROXY SERVER”, which we will implement as a part of web proxy. With this technique in place, the web proxy will automatically modify the value if the name attribute in the cookie with a randomly generated unique id, before sending the cookies to the browser, so the browser will keep the randomized value in its database instead of the original and the original value along with the randomized value as key will be saved in a hash table. The returned cookies from browser will be rewritten back to the original name/value sent by server and then forwarded to the server. Now as the browsers data base does not store the original value even if XSS attacker steals the cookies from the browser’s database. The cookies cannot be used later to impersonate the users.

V. TERMINOLOGY USED

Web Proxy:
The proxy as an intermediary or a middleman that fulfills transactions on behalf of clients, many organizations allow the users only to access the internet via web proxy. It is an important control point for web surfing which is commonly built in with various security capabilities. The web proxy can be a separate device or a part of fire wall. It must sit between client and server, act as both the client to web server and as a server to the client i.e. the browser. All web connections from the client are intercepted at the web proxy, and then the web proxy will initiate new web connections to the web servers on behalf of the clients.
VI. CONCLUSION

The proposed mechanism for Protecting Cookies from cross site scripting attack has guarantee that the cookie will be safe from XSS attack. Cookies will be safe from Persistent XSS attack and Non-Persistent XSS attack.

REFERENCES


43. D Gourley, B Totty M Sayer, S Reddy and A Aggrawal, HTTP the definitive Guide 1st elderly media US 2002


45. Suman Saha, Dept. of Computer Science and Engineering, Hanyang University, Ansan, South Korea, Consideration Points: Detecting Cross-Site Scripting, IJCSIS Vol. 4 2009


48. Conference on Web Application Development (Web Apps), 2010

