International Journal of Engineering Research and Development
e-ISSN: 2278-067X, **p**-ISSN: 2278-800X, www.ijerd.com
Volume 6, Issue 9 (April 2013), PP. 43-47

Browser Authentication and Inline Code Analyzer with Common Centralized Browser Security Violation Report Management System for Mitigating Web Attacks

D.Nandhini[1], Kalpana.G[2], Abhilash.R[3]
[1]M.Tech,Information Security and Cyber Forensics, Dr.M.G.R. Educational and Research Institute University
[2] Asst.Professor, Dr.M.G.R. Educational and Research University,
[3]M.Tech,Information Security and Cyber Forensics, Dr.M.G.R. Educational and Research Institute University

Abstract:- Delivering malware through web pages has become a common factor nowadays. This is due to poor security on web browser.web browser do not have any control on scripts loaded on it. Attackers may include malwares through scripts which makes users to redirect to malicious web pages. Thus attackers can exploit many attacks due to insecure browsing such as cross-site scripting, drive-by-downloads (malicious code gets downloaded and executed without user's intent). Scripts executed on web browser may steal cookies presented on the browser. By stealing cookies, the attacker may obtain victim's confidential information such as account, pin number and other passwords. Browser also lacks protection mechanism such as authenticating and authorizing the user on client systems. Due to this poor authorization on web browser, server suffers from DDOS attacks. This paper includes a browser component and provides authentication on a browser in order to protect user against attacks on both browser and server systems.

Keywords:- Browser DOM extension, Report generator, Centralized browser authenticator.

I. INTRODUCTION

Initially browser-based attacks are performed through untrusted websites. But nowadays, attackers gain access to trusted websites to launch exploits due to web vulnerabilities. The improper coding of web page makes the attacker to get chance to attack victim through browser. The attacker inserts their malicious code on the script which when loaded on browser may result in browser based attacks such as cross-site scripting(XSS),click-jacking, cookie poisoning attack ,browser event-handling attack, drive-by-download attack[2,3] . Improper validation of user input causes server to prone to many attacks such as Sql injection, Remote file inclusion (RLI), Cross-site scripting (XSS)

Several techniques have been proposed for browser-based attacks such as Iceshield[10], Webpawet, Client honeypot, Nozzler , Prophiler[11]. But all these techniques does not provide real time protection and fails to analyze code for dynamic web pages. Cookies prevention techniques do not provide security for unencrypted cookies.

With this proposed browser DOM[4] component and browser authentication[5,6] we are able to provide real time protection against various scripting and DDOS attacks.

II. SECURITY THREATS IN BROWSER

Most important security threats on browser includes Activex controls, plug-in, java, cookies, authentication and scripts. All these components on browser has variety of vulnerabilities which may allow attackers to launch different type of attacks. Threat due to authentication: Due to lack of browser authentication, any user on a browser can easily carry out many identity attacks and malicious downloads and executables attacks as

A. Brute-Force Login attack

Brute-Force Login attack carried out to find the password of a user account using different guesses based on the available information about the user. Client resubmits HTTP Request with credentials included when the client does not get authorized on server.

```
GET /members/docs/file.pdf HTTP/1.1
Host: target
Authorization:Basic b3dhc3A6cGFzc3dvcmQ=
```

B. Drive-by-downloads

When a user visits a compromised web page, malicious codes such as Trojan horse may get downloaded and gets executed on user computer at the back end without user's knowledge Threat due to java scripts: Weak javascript methods such as concat(),substring(),unescape() and other similar string functions are used to insert malicious payload which results in cross site-scripting attack. Weak scripts also possess chance to insert malicious code behind iframes on web page may result in click-jacking attack

C. Cross-site scripting(XSS)

In this type of attack, malicious code are injected into scripts of trusted web pages that gets executed on client browser which results in loss of identity and other browser information about victim.

Sample code for xss:

The following PHP code is vulnerable to XSS attack via the 'name' parameter.

```
<?php
$name = $_REQUEST ['name'];
?>
<html><body>Hello, <?php echo $name; ?>!</body></html>
```

D. Click-jacking attack

It is also known as UI-Redressing attack. In this the user is intended to click a link which makes the user to redirect to a website which they are not intended to link or to a malicious web page.

Target site is inserted behind iframe as:

```
<script>
window.defineSetter("location" , function(){});
</script>
<iframe src="http://target site"></iframe>
```

E. Cookie poisoning

It means changing of contents of cookies in order to steal the user identity or redirecting the user to malicious web page

Cookies of users get stored on cookie.txt file.

```
<!--?php
   $cookie = $HTTP_GET_VARS["cookie"];
   $steal = fopen("cookie.txt", "a");
   fwrite($steal, $cookie ."\n");
   fclose($steal);
?-->
```

Threats due to browser components: Browser DOM components works dynamically on web pages which allows attacker to import their malicious DOM in the browser which results in browser event-handling attack.

F. Browser event handling attack

In this attack, malware is installed through browser extension. Such as when user uses search box to find out any word in a web page, the results of search box will gets redirected to the word on a attacker's web page.

Fake search bar on browser:

III. SECURITY THREATS IN SERVER

Server side threats includes Brute Force Attack, Open Relay, Botnet, DoS, Cross-site Scripting, SQL Injection, Malware, Unpatched Software, Careless Users. With this available threats,server are prone to various types of attacks which includes Remote File Inclusion(RFI) .

Threat due to poor authentication: When user input is not validated properly, server may prone to various types of attacks such as

A. Brute-Force Login attack

Brute-Force Login attact is carried out to find the password of a user account using different guesses based on the available information about the user.

B. Distributed Denial-Of-Service (DDOS):

Requests from multiple compromised systems(virus/unauthenticated clients) to a web server causing it to be flood with requests from attackers making it to shut down and also server cannot process the requests coming from legitimate users due to traffic.

C. Sql injection:

Vulnerability on web server makes the user to access server database through sql commands supplied by user. This is due to improper validation and sanitization of user inputs.

```
' union
SELECT table_name,null,null,null,null,null,null,null,
null,null,null FROM INFORMATION_SCHEMA.TABLES –
```

IV. EXISTING SOLUTIONS ON BROWSER

A. **Search engine alerts:** Search engine sometimes alert the users when there is presence of malicious contents, but it is difficult for search engine to keep track of it.

B. **Anti-virus solutions:** Many anti-virus programs like kaspersky, AVG alerts the user any malicious links are present while browsing.

C. **No script:** In this method, either part or full scripts of web page are blocked. But this is not a good solution to prevent user against attacks.

D. **URL filtering:** Filters out the malicious web page, but maintaining the database for blacklisted web pages may be too difficult.

V. EXISTING SOLUTIONS ON SERVER

A. **Validating user inputs:** By properly validating user input, server side scripting attacks can be prevented. Since then also attacker uses legitimate user passwords which are stolen from cookies.

B. **Using authentication and Encryption techniques:** Allowing legitimate user by authenticating and encrypting the database contents on web server can prevent server from attacks.

C. **Implementing a secure network:** A secure network infrastructure with DMZ on firewalls can prevent server from outsiders to access network.

D. **Administering the server:** Maintaining logs of requests, user accessing server files and logs of file contents can be maintained

VI. PROPOSED SYSTEM

The proposed system enhances the security on client (browser) and server side by incorporating various security measures. In the proposed system, it provides secure accessing of website by authenticating the user at the browser itself using DOM component. It will prevent malicious logins and thereby DDOS attacks on server. Various targeted attacks and identity theft makes hole in web security. Existing solutions such as anti-virus, bonnets, IDS/IPS becomes weak for evolving attacks. This new system will authenticate each user on a browser and thereby protects entire web communication against cybercriminals

A. Browser Components:

1) ***Browser authentication component:*** Browser authentication refers to authenticating and authorizing user on a browser. When any user uses browsers to surf the web content, they must be first registered with centralized server. The centralized server stores usernames, passwords of browser's users irrespective of browser type. When user registration takes place on browser, additional authentication such as phone verification is done similar to mail server authentication service. Once the user registration on server gets over, he may login to access the web content whose authentication will be verified at the centralized server.

1. a) Possible of Attacks

- Attacker can carry out following attacks when tries to attempt to login with available information of a genuine user
- Brute-Force Login Attack,
- Dictionary Attack.

1. b) Solution include

Centralized server records the number of attempts of each user.

- When an attacker exceeds the limit of attempting, particular account will be blacklisted at the server.
- The centralized server will thereby monitor web activity of particular account.

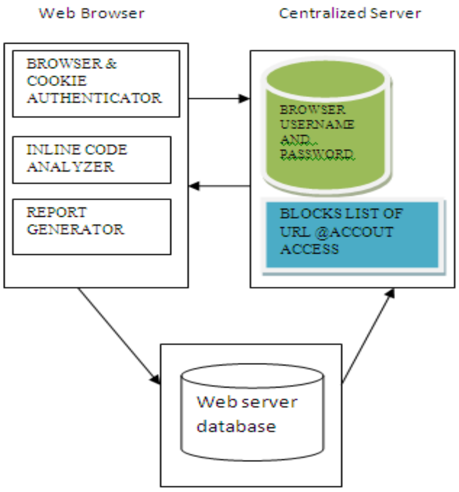
Web Browser
Centralized Server
BROWSER & COOKIE AUTHENTICATOR
INLINE CODE ANALYZER
REPORT GENERATOR
BROWSER USERNAME AND PASSWORD
BLOCKS LIST OF URL @ACCOUT ACCESS
Web server database

Fig 1: Browser components and Centralized Security Violation Report Management System

2) Inline code analyzer component:

Inline code analyzer [2] performs analyzing of scripts on web pages for the presence of any malicious links or parameters passed from malicious site. Javascript methods such as concat(),substring() and similar string functions becomes exploit for attackers to inject and call parameters from malicious site and passed to user on webpage. The proposed analyzer component checks out all vulnerable javascript methods and hook them which means it checks out parameters of those methods which are called. This inline code analyzer works for de-obfuscated code.

Inline code analysis is done based on the following observations:

- Any injection in HTML elements
- Suspicious redirects: Scripts can modify <meta> and *location* elements to redirect the page to some malicious page
- If an HTML element allows embedding a souce URL, check if the source filename matches the element. Ex: <img src = "http://www.example.com/a.js"> <img> is an anamoly
- Cloaking: Hiding a malicious webpage within invisible iframe on the page.
- Scripts with <object> tag.
- Any link with Domains extension like .cn, .ru, .co.cc, etc

3) Cookie authenticator:

Cookies stores files of user information on web browser. Cookie stealing are achieved by attackers through scripts .So to prevent cookies from stealing and poisoning, access to cookies is done using cookie authentication on browser[8,9]. Only the authenticated user on browser will be allowed to access cookies.

4) Report generator:

This reporting component on browser checks out for Brute-Force Login attack on browser and if there is multiple numbers of failed authentications occurs, it reports to centralized server.

B. Server Side Authentication:

When any server receives multiple requests from any user, it checks with centralized server to find out whether it is from legitimate user. Many worms or illegitimate users can carry out DDOS attack. So, if a server receives multiple requests from any illegal ip address, it just blocks those requests and also updates centralized server with those malicious or illegal ip addresses and websites.

VII. ADVANTAGES OF PROPOSED SYSTEM

1. This proposed system will prevent illegal users to carry out attacks at the browser end itself.
2. Prevents server from DDOS attacks.
3. Maintains users of browser through which we can identify the suspect if any attack occurs.
4. Centralized server maintains report of unauthorized users.
5. Cookie authenticator prevents cookie-based attacks.

VIII. CONCLUSION

Browser authentication is one of the important factors in preventing web page attacks on browser side. So based on the proposed system, authenticating and authorizing each user on browser we can identify the attacker and also prevent the browser end itself. Browser and cookie authenticator denies access to unauthorized user. Inline code analyzer works as a guard to browser against web page attacks. This proposed system also prevents web server from DDOS attacks. This proposed report generator prevents unauthorized access by reporting to centralized server.

REFERENCES

[1]. Basic Browser authentication - http://watirwebdriver.com/basic-browser-authentication/

[2]. “SurfGuard JavaScript instrumentation-based defense against Drive-by downloads” - SachinV. Chiplunkar, N.N. –ieee xplore

[3]. BrowserGuard: A Behavior-Based Solution to Drive-by-Download Attacks- Hsu, Fu-Hau Tso, Chang-Kuo; Yeh, Yi-Chun; Wang, Wei-Jen; Chen, Li-Han –ieee xplore

[4]. Modeling the HTML DOM and browser API in static analysis of JavaScript web applications. Simon Holm Jensen, Magnus Madsen, Anders Møller

[5]. Stronger Password Authentication Using BrowserExtensions Blake Ross lake@cs.stanford.edu,Collin, Nick Miyake, Dan Boneh, John CMitchell - http://crypto.stanford.edu/PwdHash/pwdhash.pdf

[6]. Strong User Authentication on the Web-David Chou Microsoft Corporation-August2008 http://msdn.microsoft.com/en-us/library/cc838351.aspx

[7]. “Cookies, Authentication, and Advanced Requests” - http://lwp.interglacial.com/ch11_01.htm

[8]. Managing Search for Controlled-Access Content: “Cookie-Based Authentication Scenarios” Google Search Appliance software version6.8 Posted October 2010 - https://www.developers.google.com/search-appliance/documentation/68/secure-search/secure_search_cookieauthscenarios

[9]. IceShield: “Detection and Mitigation of Malicious Websites with a Frozen DOM” Mario Heiderich, Tilman Frosch, and Thorsten Holz- http://www.nds.rub.de/media/emma/veroeffentlichungen/2011/06/21/iceshield-raid11.pdf

[10]. Canali, Marco Cova, Giovanni Vigna, Christopher Kruegel “Prophiler: A fast filter for the large-scale detection of malicious web pages” In: 20th International World Wide Web Conference (WWW 2011)

[11]. C., Livshits, B., Zorn, B., Seifert, C.: “Zozzle: Fast and Precise In Browser JavaScript Malware Detection”