International Journal of Pure and Applied Mathematics
Volume 116 No. 9 2017, 75-79
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
Special Issue

ijpam.eu

A STUDY ON WEB APPLICATION SECURITY STATE

[1]S. Shelgin, [2]R. Kavitha
[1,2]Assistant.Professor
[1,2]Department of Computer Science and Engineering,
Bharath University, Chennai-73, Tamil Nadu, India.
[1]shelgin.cse@bharathuniv.ac.in, [2]kavitha.cse@bharathuniv.ac.in

Abstract: Web Application are very important, Ubiquitous Distributed Systems whose current security relies primary on server-side mechanisms, In order to support end-to-end security users with client access to server through a set of web page, Those pages contains script code to be executed dynamically within the client web browser, and web client must be enhanced. This introduce motation event transform: an easy to use client site mechanism. HTML, CSS, JAVA script, C#, etc. coding are use for the page designing web page designing, and the Protocols are used "HTTP, SMTP, POP3, MIME, IMAP", Most web application aim to enforce simple, intuitive security policies, such as, web best email, disallows the untrusted email message, even so, and Application_ Specific security policies, and whose Implementation required only Straightforward changes to existing web browsers. We give numerous example of attractive. Web applications was subjected to a plethora of successful attacks, such as, Cross-Site Scripting, session riding, cookie theft, Browser hijacking, and the recent self-propagating worms in social networking sites and web based email.

Keyword: web application, web security flows, Vulnerabilities.

1. Introduction

The latest technology of internet & network increase the interest of user to accessing the computer environments. These page contain script code to be executed dynamically within the web browser. Most web application goal to enforce simple, initiative security policies, such as web-based email, disallowing any scripts emails which is untrusted. The custom approach consider the location of hardware these assessments were primary of definite point in time and primarily action of complaining-based reviews. In the network-based technology, the primary concern is on the network and the also, an assessment of network technology is forced on the management and implementation of real time controls to meat to provide continuous monitoring and to meat business needs[1-3]. Security is not only one-time even it is not sufficient to secure your code just once. A secure environment act must deal with all stages of a program's lifecycle. Secure web applications are only possible when a used a secure SDLC. By designing we secure the program[5], during the development.

2. Review from literature

Not any language can prevent insecure code, although there are language features which can be aid or hider a security conscious developer. The insecure software is already undermining our financial, defence, healthcare, energy, and other critical infrastructure. Like our digital infrastructure gets increasingly interconnected and complex the difficulty of achieving application security increases exponentially we can't afford longer to tolerate relatively simple
security problems like those presented below. The vulnerabilities explained in this paper are[4].

- SQL Injection.
- Cross Site Security (XSS).
- User Name Enumeration.
- Format String Vulnerabilities.
- Remote Code Execution.

3. Vulnerabilities

As the name suggests, this vulnerability an attacker allows to run arbitrary, system level code on the vulnerable server and recover any desired information contained therein. Improper coding errors read to this vulnerability during the penetration testing assignment it is difficult to discover this vulnerability, at time. But such problems are often revealed while doing a source code review[6]. However, when testing web application is important to remember that the explanation of this vulnerability can level to total system compromise with the same rights as the web server itself.

Rating :- Highly Critical.

Here we will look at such type of critical valnebility:
Exploiting register_globals in the PHP: Register_globals is a PHP setting that controls the availability of "superglobal" variables in a PHP script[7] (like data posted from a user's form, data from cookies or URL-encoded data). In the earlier releases of the PHP, register_globals was set to "on" by default, which made a developer's life easier - but this lead to was widely exploited and less secure coding[8-10].

When register_globals is set to "on" in php.ini, it can allow a user to initialize several previously not initialized variables remotely. Many a times an uninitialized parameter is used to protect from unwanted files from an attacker, and this could lead to the execution of arbitrary files from local/remote locations. For example require ($page . ".php");
Here if $page parameter is not initialized and register_globals is set to "on," then server will be vulnerable to remote code executed by including any arbitrary file in $page parameter[11]. Now let's look at the exploit code:
Countermeasures

3.1 SQL Injection

SQL injection is a very recent approach however it's still common among attackers. This technique allows Associate in Nursing assailant to retrieve crucial data from an online server's info [18]. Depending on the application's security measures, the impact of this attack can vary from basic data revelation to remote code execution and total system compromise [3,5].

Rating: Moderate to Highly crucial

Here is an example of vulnerable code in that the user-supplied input is directly utilized in a SQL query[19]:

• <form action="sql.php" method="POST" /> <p>Name: <input type="text" name="name" />
 <input type="submit" value="Add Comment" /></p> </form> <?php $query = "SELECT * FROM users WHERE username = '"; $result = mysql_query($query); ?>

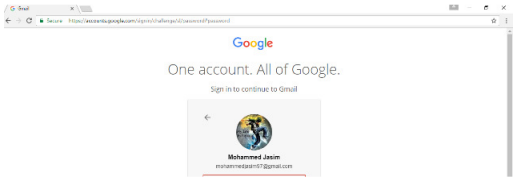
The script will work unremarkably once the username does not contain any malicious characters. In other words, when submitting a non-malicious username (steve) the question becomes:

$query = "SELECT * FROM users WHERE username = 'steve'";

However, a malicious SQL injection query can result in the subsequent attempt:

$query = "SELECT * FROM users WHERE username = '' or '1=1'";

As the "or" condition is often true, the mysql_query function returns records from the info. A similar example, using AND and a SQL command to generate a selected error message, is shown in the URL below in Figure one.



It is obvious that these error messages help Associate in Nursing assailant to induce a hold of the knowledge that they're probing for (such because the

1. Maximum recent PHP versions have register_globals set into off by default; however few users are change the default setting for applications which require it. This can be set to "on" or "off" either in a php.ini file or in a .ht access file. The variable must be properly initialized if register is set to "on." Administrators who are question application developers who insist on using register_globals[13].
2. before processing it an absolute must to sanitize all user input. As far as possible, ignore using shell commands. However, they need it, ensure that only filtered data is used to construct the string to be make sure to escape the output and executed[14].

info name, table name, usernames, password hashes etc). Thus displaying custom-made error messages could be an honest workaround for this drawback, however, there is another attack technique referred to as Blind SQL Injection where the assailant remains able to perform a SQL injection even once the applying doesn't reveal any info server error message containing helpful data for the assailant[20-21].

3.2 Countermeasures

1.Avoid connecting to the info as a superuser or as the database owner. Always use custom-made info users with the clean minimum needed privileges needed to perform the assigned task.
2.If the PHP magic_quotes_gpc function is on, then all the POST, GET, COOKIE data is free mechanically.
3.PHP has two functions for MySQL that sanitize user input: addslashes (an older approach) and mysql_real_escape_string (the counseled method). This function comes from PHP >= four.3.0, so you ought to check initial if this operate exists which you are running the most recent version of PHP four or five. MySQL_real_escape_string prepends backslashes to the following characters: \x00, \n, \r, \, ', "and \x1a.

3.3 Format String Vulnerabilities

This vulnerability results from the use of unfiltered user input because the format string parameter in certain Perl or C functions that perform format, such as C's printf().A malicious user may use the a way and sophisticated format tokens, among others, to print data from the stack or probably alternative locations in memory [4,8]. One may additionally write discretional knowledge to discretional locations mistreatment the woman format token, which commands printf() and similar functions to write back the quantity of bytes formatted. This is assuming that the corresponding argument exists and is of sort int *.Format string vulnerability attacks fall into three general categories: denial of service, reading and writing [14,15].

Rating:Moderate to Highly crucial

Here is the piece of code in miniserv.pl which was the cause of vulnerability in Webmin:

if ($use_syslog&& !$validated)) { syslog("crit", ($nonexist ? "Non-existent" : $expired ? "Expired" : "Invalid"). " login as $authuser from $acpthost"); }
In this example, the user supplied information is among the format specification of the syslog decision.
The vectors for a simple DoS (Denial of Service) of the online server ar to use the woman and %0(large number)d within the username parameter, with the former causing a write protection fault among Perl – and resulting in script abortion. The latter causes a large quantity of memory to be allotted within the perl method[24].

3.4 Countermeasure

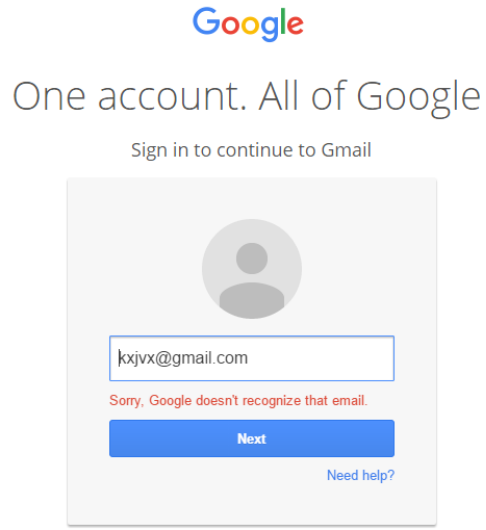
Edit the source code so the input is correctly verified.
Rating: Less to Moderately Critical
Here is a sample piece of code which is susceptible to XSS attack:
<form action="search.php" method="GET" /> Welcome!! Enter your name: <input type="text" name="name_1" />
<input type="submit" value="Go" />
</form><?php echo " Your Name
"; echo ($_GET[name_1]); ?>
In this example, the value passed to the variable 'name_1' isn't sanitised before reechoing it back to the user. This can be exploited to execute any absolute script.
Here is some example exploit code:
http://victim_site/clean.php?name_1=<script>code</script> or
http://victim_site/clean.php?name_1=<script>alert(document.cookie);</script>

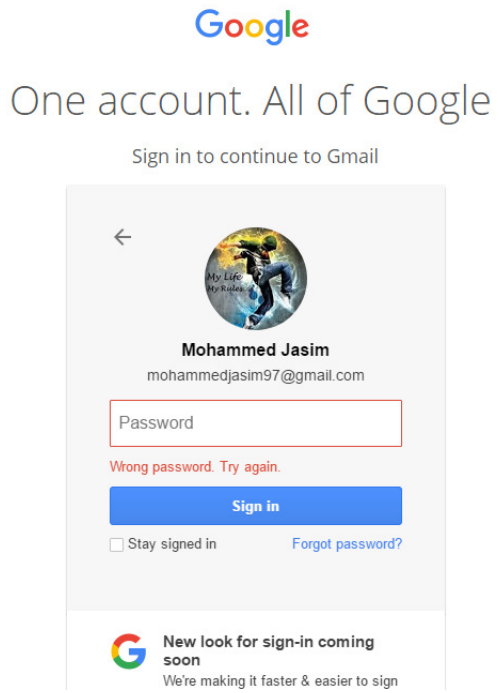
3.5 Countermeasures

The above code will be altered within the following manner to avoid XSS attacks:
• <?php $html= htmlentities($_GET['name_1'],ENT_QUOTES, 'UTF-8'); echo "<p>YourName
"; echo ($html); ?>

3.6 Username enumeration

Username enumeration is a sort of attack where the backend validation script tells the aggressor if the provided username is correct or not[10,11,17]. Exploiting this vulnerability facilitates the attacker to experiment with totally {different|completely different} usernames and confirm valid ones with the help of those different error messages.
Rating: Less Critical
Here is an example of login screen:





Below the response given when a valid username is guessed correctly:
Username enumeration can facilitate Associate in Nursing aggressor WHO makes an attempt to use some trivial usernames with simply guessable passwords, such as test/test, admin/admin, guest/guest, and so on. These accounts are usually created by developers for testing functions, and many times the accounts area unit ne'er disabled or the developer forgets to vary the positive identification.
During pen testing assignments, the authors of this article have found such accounts don't seem to be only common and have simply guessable passwords, but at times they additionally contain sensitive data like valid mastercard numbers, passport numbers, and so

on. Needless to say, these could be crucial details for social engineering attacks.

3. Conclusions

Web applications reach out to a bigger, less-trusted user base than legacy client-server applications, and yet they area unit additional at risk of attacks. Many firms area unit beginning to take initiatives to forestall these varieties of break-ins. Code reviews, extensive penetration testing, and intrusion detection systems area unit simply a few ways in which firms are battling a growing downside. Unfortunately, most of the solutions available these days area unit mistreatment negative security logic (working with a list of attacks and making an attempt to forestall against them). Negative security logic solutions can forestall illustrious, generalized attacks, but area unit ineffective against the kind of targeted, malicious hacker activity outlined in this paper. In this paper,we have incontestable 5 common net application vulnerabilities, their countermeasures and their criticality. If there is a standardized message among each of those attacks, the key to mitigate these vulnerabilities is to sanitize user's input before processing it.

References

[1]OWASP Top 10 Web Application Vulnerabilities. http:// www.applicure.com/blog/owasp-top-10-2010
[2]E. Chien. Malicious Yahooligans. http://www. symantec.com/avcenter/reference/malicious. yahooligans.pdf, 2006.
[3]S. Di Paola. Wisec security. http://www.wisec.it/ sectou. php? id=44c7949f6de03, 2006.
[4]Ú. Erlingsson and F. B. Schneider. IRM enforcement of Java stack inspection. In Proc.IEEE Security and Privacy, 2000.
[5]O. Hallaraker and G. Vigna. Detecting malicious JavaScript code in Mozilla. In Proc. IEEE Conf. on Engineering of Complex Computer Systems, 2005.
[6]B.Hoffman.Ajaxsecurity. http://www.spidynamics. com/assets/documents/AJAXdangers.pdf, 2006.
[7]T. Jim, N. Swamy, and M. Hicks. Defeating script injection attacks with browser-enforced embedded policies. In WWW, 2007.
[8]Udayakumar R., Kaliyamurthie K.P., Khanaa, Thooyamani K.P., Data mining a boon: Predictive system for university topper women in academia, World Applied Sciences Journal, v-29, i-14, pp-86-90, 2014.
[9]Kaliyamurthie K.P., Parameswari D., Udayakumar R., QOS aware privacy preserving location monitoring in wireless sensor network, Indian Journal of Science and Technology, v-6, i-SUPPL5, pp-4648-4652, 2013.
[10] Brintha Rajakumari S., Nalini C., An efficient cost model for data storage with horizontal layout in the cloud, Indian Journal of Science and Technology, v-7, i-, pp-45-46, 2014.
[11] Brintha Rajakumari S., Nalini C., An efficient data mining dataset preparation using aggregation in relational database, Indian Journal of Science and Technology, v-7, i-, pp-44-46, 2014.
[12] Khanna V., Mohanta K., Saravanan T., Recovery of link quality degradation in wireless mesh networks, Indian Journal of Science and Technology, v-6, i-SUPPL.6, pp-4837-4843, 2013.
[13] Khanaa V., Thooyamani K.P., Udayakumar R., A secure and efficient authentication system for distributed wireless sensor network, World Applied Sciences Journal, v-29, i-14, pp-304-308, 2014.
[14] Udayakumar R., Khanaa V., Saravanan T., Saritha G., Retinal image analysis using curvelet transform and multistructure elements morphology by reconstruction, Middle - East Journal of Scientific Research, v-16, i-12, pp-1781-1785, 2013.
[15] Khanaa V., Mohanta K., Saravanan. T., Performance analysis of FTTH using GEPON in direct and external modulation, Indian Journal of Science and Technology, v-6, i-SUPPL.6, pp-4848-4852, 2013.