ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 4, April 2018

Review On Enhanced Counter Measure for Web Site Vulnerabilities with Amplified Algorithm

Rashmi[1], Deepika Goyal[2]

MTech. Student, Department of Computer Science & Engineering, Advance Institute of Technology and Management, Palwal, Haryana, India[1]

Assistant Professor, Department of Computer Science & Engineering, Advance Institute of Technology and Management, Palwal, Haryana, India[2]

ABSTRACT: Web applications have picked up prominence throughout the years and have turned into a vital piece of our day by day lives connection. We utilize these applications all the time to interface with our loved ones, buy things on the web and access financial balances among others. In any case, these submission are not 100% protected, they are liable to a extensive cluster of vulnerabilities, for example, for example, SQL infusion, Cross site following , cross site reference fabrication and server side infusions among others. To find these shortcomings, web application scanners are utilized to report vulnerabilities found. The fundamental goal of this investigation is to play out a near investigation of open source powerlessness testing apparatuses, consider their calculation for these devices and propose an enhanced enlarged calculation. A reproduction to test and approve the expanded calculation was additionally created. This exploration centers around six of the open source web filtering apparatuses which, were tried against four online applications with known vulnerabilities to think about the devices abilities and highlights. Also, the calculation of these apparatuses was examined with a point of delivering an increased calculation that will be more precise in distinguishing web vulnerabilities.

KEYWORDS: Web Security, Cross Site Scripting, Sql Injection, Web Site Vulnerabilities.

I. INTRODUCTION

While the web foundation is created by exceptionally experienced specialists with defence imperfections and arrangements at the rear of their awareness, a portion of the network application are produced by beginner software engineers who have practically no learning of about network application protection. Hence, they create helpless network application that can be hacked uncovering the association's private data. Numerous associations utilizing online applications, encounter at least one types of security breaks. For example, programmers may access organization information, unapproved programs take client's login accreditations and send them to digital culprits, infections may likewise be utilized to execute unlawful exchanges and in addition other deceitful exercises. Programmers are additionally known to ruin organization's site and deny clients access to administrations. Though a few organizations timid far from publicizing such data to maintain a strategic distance from negative notoriety, the news discover their way to people in general area somehow. There is a need to recognize the security slip by in different associations and think of methods for limiting cybercrime

With headway in web advancements and move from customary work area application to electronic arrangements, the prominence of online applications has developed colossally. Today, the electronic applications are utilized as a part of security-basic situations, for example, medicinal, money related and military frameworks (Stuttard and Pinto, 2011). Despite the fact that the web framework is created by experienced developers with security worries in their psyche, a portion of the web applications are designed by less experienced counseling software engineers with practically no

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 4, April 2018

information about security. This uncovered the web application to different vulnerabilities and gives roads to digital crooks to increase unapproved access to private data. In one of the current examinations by the Ponemon Institute, they discovered that that 45% of ruptures surpass $500,000 in misfortunes. In the biggest of occurrences, numerous Fortune-recorded organizations have given investor direction that the misfortunes would shift from a couple of dollars to a large number of dollars.

With evolution of computers the life of people became more and more easily. They were able to keep their data on their devices, and started finding ways to make them accessible to others, for example say by using floppy, writable disks, which was followed by portable hard-disk, all these where expensive in their own way during their time. The data was very much private on personal devices like PC, laptops, mobile phones etc, therefore sharing data with others was considered to be expensive. As the world of computing got more advanced the ways for sharing data started becoming cheaper and cheaper. In recent years a new term has evolved called as Cloud which is provided by different provides, and which is nothing but facility or service of different resources or components like hardware, platform, storage's, software etc, and it is gaining importance because it frees the user from maintenance perspective on a investment of some money for the use of these services provided by cloud service providers. Now to provide such service to the client, naturally the provider's must have and rather can have access to resources which are used by the people/clients. Among the reasons these access are greatly required are for maintenance perspective. And definitely since billions of clients will be thinking about using such service, the infrastructure ought to be capable enough to support them, and these resources ought to be shared between billions of client's. Service availability, data synchronization between different devices, availability of data via any devices which includes browser facility make cloud more attractive. Now since the info gets shared or stored in providers area, the client gets worried about privacy of its data, although there are certain agreements and SLA which are agreed by cloud provider and client. Now although client have a platform to generally share the info, the expense of securing his/her data or in a nutshell making its data private gets costlier. The cloud term is of interest not just to the patient clients but to organizations as well. With organization as a consumer the concern of data security becomes multi-fold. Consider a typical example of small scale business that has different departments like HR, Finance, etc. We will focus on finance department since finance details of any business/company/organization are considered to be very sensitive and must be confidential.

SQL Injection: Sql-Injection commonly means all those attacks to an application, usually Web, in which the program queries an SQL database using variables passed by the user without having previously checked them. As is clear from this first, minimal, explanation, this is not a problem directly attributable to PHP but more commonly applicable to all Web languages, whatever the SQL database used; MySQL is obviously not an exception. While it is extremely simple to avoid these kinds of problems, an incredibly high number of commercial and non-commercial applications are subject to this vulnerability due, as usual, to excessive trust in user input. Trust that, in this case, can truly be fatal and lead to unauthorized access, destruction of corporate databases and so on. One of the most typical examples of Sql-Injection is that of modifying a SELECT query to access data that would normally not be visible to the user or gain access to an authentication system with administrator privileges. Everything is extremely simple. In order to access this application by passing it off for another user, it will be sufficient to enter the login of these and the following password: "123 'OR' '='" (123 is inserted by pure example, there could be any value). In this way the SQL query executed by the program will be: "SELECT * FROM users WHERE username = 'mario' AND password = '123' OR " = "". By changing the password in this way, the execution of our script will no longer generate a MySQL error for incorrect syntax, but will authenticate the user correctly. The addition of OR " = " (the closing apex is added by the PHP query and, in the intentions, had to close the password field) in fact makes the control of the password superfluous. It is in fact asked to MySQL that the password is 123 or, alternatively, that " = ". This second clause is clearly always true and therefore MySQL will find at least one result. We will analyze in the following paragraphs how to avoid the risk of SQL Injection, it is worth stressing now that the two main limits that a hacker will encounter in trying to exploit this type of attack are: the lack of knowledge of the database structure (table names and fields) and, in this specific case, the identification of the administrator's username. In the first case the problem is hardly surmountable since PHP, in case of SQL error, in a standard configuration does not display the query that gave the error, thus not exposing the hacker valuable information on tables and fields. This "protection" is still missing for Open Source applications. In this case, in fact, anyone can download the application and see in a few seconds how the database

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 4, April 2018

tables are structured. In the second case it could be quite easy to get around the obstacle by trying, with another SQL Injection, to view the entire user database or alternatively looking for information on the website itself. Precisely for this reason it is highly recommended never give too trivial names to the fields and tables of a database and above try to use a little 'imagination for the username of the administrator trying to avoid terms like admin or root.

Preventing SQL Injection is a relatively simple task and can be done in various ways depending on your preferences. Let's now analyze, one by one, all the possibilities to protect your web application from SQL Injection. It is advisable to always adopt at least one of the following advice if you do not want to have unpleasant surprises once your product is in the hands of customers. A first solution is provided directly by the php.ini configuration file. By setting true the magic_quotes_gpc entry PHP will take care to add escape characters in front of all the contents in cookies, Post and Get. This solution, however, is not optimal as it may cause incompatibility with many third-party web applications and also do not protect from "malicious" inputs not coming from cookies, Post or Get and in particularly complex applications it is not impossible that, studying the application with particular effort, a hacker can find other ways to put his SQL Injection somewhere. A second solution is to use addslashes () or similar functions like mysql_escape_string () on all the variables inserted in an SQL query. In this way you will always be sure that in all past values the quotes will be converted into \ 'making them so inoffensive. It should however be noted that this precaution, as the previous one is really safe ONLY if, in SQL queries, the apex is used even when numerical values are manipulated. In all queries it is advisable to always use syntax of the type id = '1'. Only in this case will you really be protected from SQL Injection, otherwise a hacker could still add an OR = 1 = after the desired id without our addslashes () can in any way help us. The problem is not limited to the use of the quotes only, but it is presented in the same way as a string delimiter in the query "(quotes) Fortunately, mysql_escape_string () escapes all the characters that can be used as delimiters of string solving many problems. A last alternative, the most complex but also the safest, is to check each variable inserted in a query with regular expression or other specific functions. If for example we know a priori that a variable must contain an integer we could use a solution proposed under the scheme.

Cross Site Scripting : Common translations include "cross-site scripting attacks" and "cross-site scripting attacks". Generally referred to as XSS attacks, the description of the domestic web security expert Chen Xiaozhong is a kind of "you out of the bag, I'm sorry, he Really cool!" Attack! Different from the general attack pattern directly against web applications, the object of XSS attack is not the website itself, but the computer and user browsing the website, which is somewhat similar to the attack mode of “dashing across the hill”; because many websites do not check the input value well.,allowing attackers to send out malicious code and outputting the problematic HTML data through the website. The final processing and execution of this problematic HTML data is the browser on the user's computer. The mode of fighting cattle across the hill allows users to execute problematic code through a trusted website without knowing it, in order to achieve the purpose of the attack. What about the website itself? Because it is only responsible for outputting the malicious instruction code sent by the attacker into HTML data, the website itself will not generate an error message, and therefore, the manager often cannot timely notice the problem of his own website! Recently, many domestic corporate websites have been revealed to have XSS problems, including even domestic large-scale security companies, and the vulnerabilities may exist for many years, whether it has been used for further attacks, it is difficult to prove! However, according to the records and comments on the website of the Grand Cannon, we can also find that most companies are not precise enough to fix XSS loopholes, and there is no systematic repair mode (Note 4). This also highlights the fact that many domestic business units, even security companies, still lack the correct concepts and understanding of XSS weaknesses.

II. RELATED WORK

The usage of Personal Computers, handheld devices and propelled cells has staggeringly extended throughout late years, as substantiated by Stuttard and Pinto (2011) web applications have been created to perform in every practical sense every profitable limit you could realize on the web. These fuse Online Shopping, Social Networking, Gambling and Online club, Banking, Web look for, Auctions, Webmail, and Interactive website pages among others. In a report dispersed by Whitehat "86% of all destinations attempted by Whitehat Sentinel had no short of what one bona fide lack of protection, and usually, essentially more than one – 56% to be correct. (Whitehat, 2015)

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 4, April 2018

As showed by Shema (2011), various affiliations rely on modified web applications to realize business shapes. These may fuse absolute applications, or include modules, for instance, on the web, login pages shopping crates, and diverse sorts of dynamic substance. A segment of these item applications in your framework could be made in-house. Likewise, some may be legacy locales with no alloted proprietorship or support. Physically looking at these for stipulations and arranging their criticalness for remediation can be a staggering task without dealing with attempts and using robotized gadgets to improve accuracy and profitability.

Agents are continually responding to requests from both inside and outside the affiliation's corporate framework using gadgets, for instance, tablets, mobile phones or PCs. While this has tremendous focal points, the negative drawback is the way that software engineers may misuse accessibility to increment unapproved access to basic association information. Subsequently, it is essential for any association to ensure that they guarantee their web applications and reduce the probability of a security break to their electronic structure. Testing the weakness of web applications with modernized penetration testing instruments conveys by and large lively results. Starting at now, there are various such gadgets, both business and open source.

III. PROPOSED METHODOLOGY

The calculation was be tried by making an interpretation of it in to a reproduction created utilizing Microsoft .NET and Python advancement stage. The re-enactment will be keep running against the few web applications and the outcomes gathered about discovery precision, the time taken to check a given application and in addition the unwavering quality and consistency. After the testing procedure, the consequences of the recreation were contrasted and alternate opens source web scanners too. The Amplified calculation was composed with a point of enhancing shortcomings that were found with existing calculations. A discovery approach will be received with a point of enhancing application scanners. The device used to test and approve the proposed Amplified calculation will exhibit the enhanced capacity of the device. Amid the advancement of this calculation, separate and vanquish approach was received. This implies the code is built to slither every one of the Webpages in a web application and sweep for the different vulnerabilities autonomously.The below mentioned scheme will work with 64 bit based XOR and Shift Substitution encryption to form and establish the secure communication among the transaction components and resources.

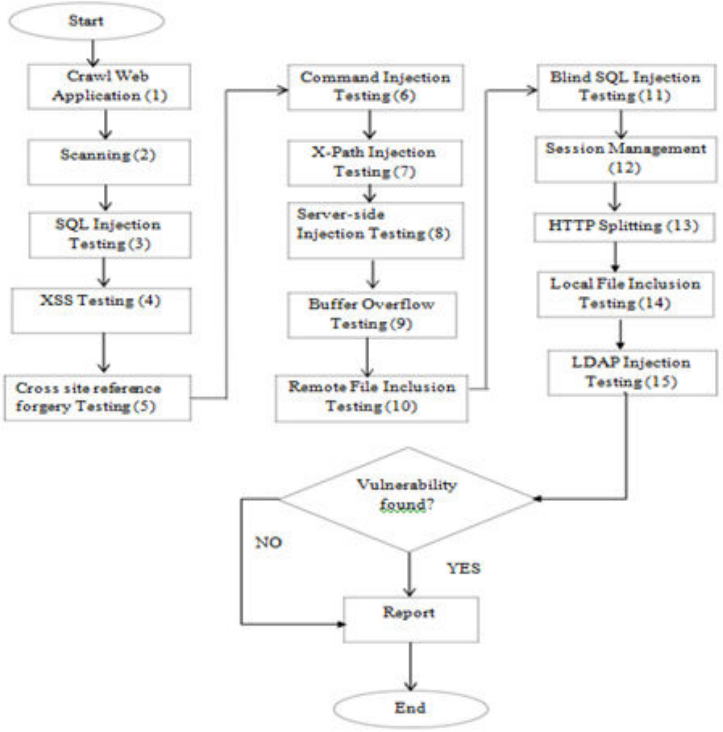

Figure 2: Architecture and Flow Diagram for proposed Amplified Algorithm Scheme.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 4, April 2018

IV. CONCLUSION AND FUTURE WORK

Decision regarding the Amplified algorithm : The projected Amplified calculation is broad in the implementation of its discovery component against network or web-applications vulnerabilities. The projected Amplified calculation intelligence more vulnerabilities and demonstrate the capable way whereas revealing found vulnerabilities. Nevertheless while the projected Amplified calculation did not examine 100% of the current vulnerabilities. There is have to build the calculation creeping part keeping in mind the end goal to guarantee that it executed "profound" slithering. Furthermore the outcomes exhibited demonstrates that the proposed calculation should be upgraded to do the filtering in a brief timeframe. More research is expected to think of a modern calculation that has the ability to distinguish greater defenselessness.

REFERENCES

1. Alssir, F. T., & Ahmed, M. (2012). Web Security Testing Approaches: Comparison Framework. In Proceedings of the 2011 2nd International Congress on Computer Applications and Computational Science (pp. 163-169). Springer Berlin Heidelberg.
2. Antunes & Vieira (2012). Defending against web application vulnerabilities. Computer, (2), 66-72.
3. Bau, J., Bursztein, E., Gupta, D., & Mitchell, J. (2010). State of the art: Automated black-box web application vulnerability testing. In Security and Privacy (SP), 2010 IEEE Symposium on (pp. 332-345). IEEE.
4. Chen, S. (2014). wavsep. Available: http://sectooladdict.blogspot.com/2014/02/wavsep-web-application-scanner.html. [Accessed 09 July 2015.]
5. Dessiatnikoff, A., Akrout, R., Alata, E., Kaaniche, M., & Nicomette, V. (2011). A clustering approach for web vulnerabilities detection. InDependable Computing (PRDC), 2011 IEEE 17th Pacific Rim International Symposium on (pp. 194-203). IEEE.
6. Dougherty, C. (2012).Practical Identification of SQL Injection Vulnerabilities. 2012. US-CERT-United States Computer Emergency Readiness Team. Citado na, 34. . [Accessed: 08th June 2015]
7. Doupe, A., Cova, M., & Vigna, G. (2010). Why Johnny can't pentest: An analysis of black-box web vulnerability scanners. In Detection of Intrusions and Malware, and Vulnerability Assessment (pp. 111-131). Springer Berlin Heidelberg. [Accessed: 10th June 2015]
8. Fonseca, J., Vieira, M., & Madeira, H. (2014). Evaluation of Web Security Mechanisms using Vulnerability & Attack Injection. Dependable and Secure Computing, IEEE Transactions on, 11(5), 440-453.
9. Granville, K . (2015).Nine Recent Cyber-attacks against Big Businesses. New York Times [online] Available from http://www.nytimes.com/interactive/2015/02/05/technology/recent-cyberattacks.html?_r=1. [Accessed 08 July 2015.]
10. Howard, M., LeBlanc, D., & Viega, J. (2010). 24 deadly sins of software security [electronic book]: Programming flaws and how to fix them. New York: McGraw-Hill.
11. Jovanovic, N., Kruegel, C., & Pixy, E. K. (2010). A Static Analysis Tool for Detecting Web Application Vulnerabilities (Short Paper). In Proceedings of the 2006 IEEE symposium on Security and Privacy, Washington, DC, IEEE Computer Society (pp. 258-263).
12. Kalman., G. (2014). Ten Most Common Web Security Vulnerabilities.[online] Available from: http://www.toptal.com/security/10-most-common-web-security-vulnerabilities [Accessed 08 July 2015.]
13. Kals, S., Kirda, E., Kruegel, C., & Jovanovic, N. (2014). A web vulnerability scanner. In Proceedings of the 15th international conference on World Wide Web (pp. 247-256). ACM.
14. Khoury, N., Zavarsky, P., Lindskog, D., & Ruhl, R. (2011). Testing and assessing web vulnerability scanners for persistent SQL injection attacks. In Proceedings of the First International Workshop on Security and Privacy Preserving in e-Societies (pp. 12-18). ACM.
15. McQuade, K. (2014). Open Source Web Vulnerability Scanners: The Cost Effective Choice?. In Proceedings of the Conference for Information Systems Applied Research ISSN (Vol. 2167, p. 1508). [Accessed: 18th June 2015]