International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878, Impact Factor (2014): 3.05

A Survey on Vulnerability and Attack Injection for Evaluation of Web Security Mechanism

Rani V. Bhor[1], Harmeet K. Khanuja[2]

Department of Computer Engineering, Marathwada Mitra Mandal's College of Engineering, Savitaribai Phule Pune University, India

Abstract: ***The use of web applications is increasing very widely in the field of global economy. Now a day's web application security becomes a critical issue. Web applications are vulnerable due to software defects. Developers used unchecked input fields at user interface. Attackers take advantage of it and exploit such vulnerability into the attack. Vulnerability is a weakness in the systems security that could be accidentally occur or intentionally violated and result in security failure. Hackers or attackers can exploit a vulnerable web application and cause serious damage. To protect information from such attacks many web security solutions presented by researchers, but these solutions are insufficient because new attacks and vulnerabilities are encountered every day. We need to review these methods and overcome limitations by developing new ones. This paper surveys the methods of vulnerability detection and prevention.***

Keywords: SQL injection, Web application, Web attack, Web security, Vulnerabilities, Exploits, Web application scanner

1. Introduction

Web applications have become a crucial part of commerce, entertainment and social interaction and they are rapidly replacing desktop applications. In the near future, they are expected to play critical roles in national infrastructures such as health-care, national security, and the power grid. Web applications have become one of the most important communication channels between various kinds of service providers and clients on the Internet. However, this also raises many security issues and exacerbates the demand for practical customer-friendly solutions. Although there are many approaches of vulnerability analysis, web applications require a more technology independent solution. Along with the increased importance of web applications, the negative impact of security flaws in such applications has grown as well. Vulnerabilities that may lead to the compromise of sensitive information are being reported continuously to network [6]. Costs of the resulting damages are increasing.

The main reasons for this phenomenon are time and financial constraints, limited programming skills, and lack of security awareness on part of the developers. Malicious users all around the world can exploit a vulnerable web application and cause serious damages. An attacker discovers new vulnerabilities and exploits every day. The security of web applications becomes a major concern and it is receiving more and more attention from governments, corporations, and the research community. Given the preponderant role of web applications in many organizations, one can realize the importance of finding ways to reduce the number of vulnerabilities. SQL Injection Attacks (SQLIAs) have emerged as one of the most serious threats to the security of database driven applications. Cross site scripting (XSS) is the type of attack in which hackers injects scripts like java script into clients trusted websites and stolen cookies or sessions and redirect to the malicious pages.

2. Literature Review

A literature survey has been carried out to motivate a critical analysis of the various state of the solution for the web application security. Malicious users all around the world can exploit a vulnerable web application and cause serious damages. The security of web applications becomes a major concern and it is receiving more and more attention from governments, corporations, and the research community. SQL Injection Attacks (SQLIAs) have emerged as one of the most serious threats to the security of database-driven applications. The Open Web Application Security Project (OWASP) identifies the most serious web application vulnerabilities, the top ten vulnerabilities in 2013 were:

1. Injection
2. Broken authentication and session management
3. Cross-site scripting
4. Insecure direct object reference
5. Security misconfiguration
6. Sensitive data exposure
7. Missing function level access control
8. Cross-site request forgery
9. Using components with known vulnerabilities: Heartbleed and shellshock in action
10. Unvalidated redirects and forwards

The Open Web Application Security Project (OWASP), an international organization of web developers, has placed SQLIAs as top most vulnerability among the top ten vulnerabilities of web applications.

2.1 Related Work

It is unfeasible to produce complex applications without defects, and even when this occurs, it is impossible to know it, prove it, and repeat it systematically [4]. Software developers cannot assure code scalability and sustainability with quality and security, even when security is defined from the ground up [5]. Many security problems are related to how bad different programming languages are in terms of tendency for mistake. String analysis and vulnerability detection is very important in the security due to this string analysis is widely studied. Data sanitization technique using reverse proxy is used to prevent SQLi and XSS attacks. This technique is used to sanitize the users input that may transform into database attack. Again a filter program is used which redirects the user input to the proxy server before it is sent to the application server. In the proxy server, data cleaning algorithm is trig-

gered using a sanitizing application [6]. Hybrid analysis framework is introduced by Monga et al [3] that blends static and dynamic approaches to detect vulnerabilities in web applications. The application code is translated into an intermediate form. The resulting static model is filtered to focus only on dangerous statements. This reduces model size where dynamic analysis will be conducted, mitigating the performance overhead of the dynamic taint analysis approach. This approach, as most taint analysis approaches (either static or dynamic), targets only injection-related vulnerabilities. Mohemed et al [4] introduce a new automated formal vulnerability analysis approach. This approach is based on formalized vulnerability definition schema. A part of this schema is the formal vulnerability signature. This signature specifies a set of invariants that confirm the existence of a given vulnerability in the target program. Commercial tools, like Acunetix Web Vulnerability Scanner, IBM Rational AppScan and HP Webinspect they can detect SQLi vulnerabilities, as well as several other vulnerabilities. There is also a wide range of open-source tools which can detect SQLI, such as Vega, W3af and Wapiti.

Vega is GUI-based, cross-platform tool written in Java, which can be extended using its Java script API. W3af is a free open-source web application scanner designed and implemented for finding and exploiting SQLI and web application vulnerabilities. Some of the existing Web application scanners are based on predefined rules and known defects recorded in vulnerability databases [8]. They use vulnerability databases, such as OSVDB (Open Source Vulnerability Database), to scan for possible existence of directories and files that malicious users usually try to find and treat as an entry point. Most popular scanners like AppScan and ZAP Proxy provide rule-based SQL injection detection capabilities, through which they can construct a number of of attacking exploits. All these tools try to identify points in a web application that can be used to inject malicious code.They perform attacks that target these points and monitor how the application responses to the generated attacks. Zoran Djuric developed a Black-box testing tool for detecting SQL injection vulnerabilities. The black-box approach is based on simulation of SQLI attacks against web applications. Thus, the scope of analysis is limited to HTTP responses and HTML pages received from the application server [8].

Fonseca et al. developed a methodology to automatically inject realistic attacks in web applications. This methodology consists of analyzing the web application and generating a set of potential vulnerabilities. Each vulnerability is then injected and various attacks are mounted over each one. The success of each attack is automatically assessed and reported [11]. List of possible types of vulnerabilities affecting web applications is huge. According to Open Web Application Security Project SQLi and XSS are at top of that list. A SQLi attack modify the input fields of the webpage so it can alter the query send to the back-end database. A XSS attack injecting scripting code or HTML code in vulnerable web pages. When user visit a trusted website attacker convince user to click on the URL that contains malicious code. This can result in stealing of browser cookies and other sensitive user data. Most common vulnerabilities found in the web application is missing function call extended (MFCE). This MFCE fault type represents vulnerabilities caused by an input variable that should have been properly sanitized by a specific function, which the developer forgot to include in the code. Most of the SQLi vulnerabilities come from exploitation of numeric field. Marco Vieira et al. analyze 715 vulnerabilities and 121 exploits of 17 web applications using field data on past security fixes. They analyze the web application written in a weak type language and some written in strong type languages. According to their results applications written with strong typed languages have smaller number of vulnerabilities and exploits. Weak typed are the preferred targets for the development of exploits.

2.2 Comparative Study

Table 1: Comparative study of vulnerability and attack injection

Sr. No.	*Author*	*Title*	*Method*
1	J. Fonseca, M. Vieira, and H. Madeira	"Evaluation of web security mechanisms using vulnerability and attack injection", IEEE transaction on Dependable and secure computing, 2014.	Vulnerability and attack injector tool which inject vulnerabilities can be exploits automatically after words.
2	Abdul Razzaq, Khalid Latif, H.Farooq ahmad, Ali Hur, Zahid anwar and Peter Charles Bloodsworth	"Semantic security against web application attacks", ACM, 2013	Ontology models, semantic approach: It provides an effective security mechanism against web application attacks by capturing the context of a web application and its protocol.
3	J. Fonseca,N. Seixas, M. Vieira, and H. Madeira	"Analysis of field data on web security vulnerabilities", IEEE transaction on Dependable and secure computing, 2014	This paper presented a field study on SQL injection and XSS. It analyzes source code of security patches of widely used web applications.
4	Zoron Djuric	"Black Box testing tool for detecting SQL injection vulnerabilities", IEEE, 2013.	SQL injection vulnerability Detection tool: Black-box vulnerability scanner for detecting SQLI vulnerabilities.
5	N.Neves, J.Antunes, M.correia, P.verissimo and R. Neves	"Using attack injection to discover new vulnerabilities", IEEE conference on Dependable systems and networks, 2006	Attack injector tool (AJECT) to support the discovery of vulnerabilities in network servers.
6	J.Fonseca, M. Vieira, H. Madeira	" Testing and comparing web vulnerability scanning tools for SQLi and XSS attack", IEEE conference on Dependable computing, 2007	Software fault injection technique: identifying all points where bugs can be injected and then injecting the bug and can be used to test and compare the performance of the scanners.
7	Mohemed Al-	"Supporting auto-	Formal vulnerabili-

	morsy, John Grundy and Amani S. Ibrahim	mated vulnerability analysis using formalized vulnerability signatures ", ACM, 2014	ty signature described using OCL. Program analysis of the target system to locate signature matches is performed using formal signature.

3. Conclusion

In this literature survey we have studied a different methods used to detect and prevent web attacks. We have concluded that major web application vulnerabilities are generating from the source code defects. The attacker can invoke the application with a malicious input that is part of an SQL command that the application executes. This permits the attacker to damage or get unauthorized access to data stored in a database. We have a more research scope in this area to overcome current security issues and improve efficiency of security mechanisms.

References

[1] Mohemed Almorsy, John Grundy and Amani S. Ibrahim, "Supporting Automated Vulnerability Analysis using Formalized Vulnerability Signatures," ACM, 2012.
[2] M. Dowd, J. McDonald, and J. Schuh, "The Art of Software Security Assessment: Identifying and Preventing Software Vulnerabilities ", Addison Wesley Professional, 2006.
[3] J. Fonseca,N. Seixas, M. Vieira, and H. Madeira, "Analysis of Field Data on Web Security Vulnerabilities", IEEE Transaction on dependable and secure computing, vol. 11, no. 2, march/april 2014.
[4] P. Giorgini, F. Massacci, J. Mylopoulos, and N. Zannone, "Modeling Security Requirements through Ownership, Permission and Delegation", Proc. IEEE Intl Conf. Requirements Eng., 2005.
[5] L. Hatton, "The Chimera of Software Quality", IEEE Software, vol. 40, no. 8, pp. 104-103, Aug. 2007.
[6] Kirti randhe, "Security Engine for prevention of SQL Injection and CSS Attacks using Data Sanitization Technique", IJIRCCE, 2015.
[7] M. Monga, R. Paleari, and E. Passerini, "A hybrid Analysis Framework for Detecting Web Application Vulnerabilities", ICSE Workshop S/W Engineering for Secure Systems, 2009.
[8] Zoron Djuric, "Black Box TestingTtool for Detecting SQL Injection Vulnerabilities", IEEE, 2013.
[9] J. Duraes and H. Madeira, "Emulation of Software Faults: A Field Data Study and a Practical Approach", Trans. Software Eng., vol. 32, 2006.
[10] J. Fonseca, M. Vieira, and H. Madeira, "Vulnerability & Attack Injection for Web Applications", Proc. Intl Conf. Dependable Systems and Networks, pp. 93-102, 2009.
[11] Jose Fonseca, Marco Vieira, and Henrique Madeira, "Evaluation of Web Security Mechanisms Using Vulnerability and Attack Injection", IEEE transaction on Dependable and secure computing, Vol. 11, No. 5, 2014.
[12] M. Howard, D. LeBlanc, and J. Viega, "19 Deadly Sins of Software Security: Programming Flaws and How to Fix Them", McGraw-Hill, 2005.
[13] IBM Global Technology Services, "IBM Internet Security Systems X-Force1 2010 Trend & Risk Report", technical report, IBM Corp., 2011.
[14] N. Jovanovic, C. Kruegel, and E. Kirda, "Precise Alias Analysis for Static Detection of Web Application Vulnerabilities", Proc. IEEE Symp. Security and Privacy, pp. 27-36, 2006.
[15] C. Le Gues et al., "A Systematic Study of Automated Program Repair: Fixing 55 Out Of 105 Bugs for $8 Each", Proc. Intl Conf. Software Eng., pp. 3-13, 2012.
[16] B. Livshits and S. Lam, "Finding Security Vulnerabilities in Java Applications with Static Analysis", Proc. USENIX Security Symp. , pp. 18-18, 2005.
[17] F. Long, "Software Vulnerabilities in Java", Cert. technical note, Software Eng. Inst., Carnegie Mellon Univ., 2005.
[18] R. Mays, C. Jones, G. Holloway, and D. Strudinsky, "Experiences with Defect Prevention", IBM Systems J., vol. 29, pp. 4-32, 1990.
[19] OSVDB, "Open Sourced Vulnerability Database", http://osvdb.org, May 2013.
[20] N. Tomatis, R. Brega, G. Rivera, and R. Siegwart, "May You Have a Strong (Typed) Foundation Why Strong Typed Programming Languages Do Matter", Proc. IEEE Intl Conf. Robotics and Automation, 2004.
[21] J. Walden, M. Doyle, G. Welch, and M. Whelan, "Security of Open Source Web Applications", Proc. Intl Symp. Empirical Software Eng. and Measurement, 2009.
[22] Abdul Razzaq, Khalid Latif,H.Farooq ahmad, Ali Hur, Zahid anwar and Peter Charles Bloodsworth, "Semantic security against web applicationattacks", ACM, 2013.
[23] N.Neves, J.Antunes, M.correia, P.verissimo and R. Neves, "Using attack injection to discover new vulnerabilities", IEEE conference on Dependable systems and networks, 2006.
[24] Abdul Razzaq, Khalid Latif, H.Farooq ahmad, Ali Hur, Zahid anwar and Peter Charles Bloodsworth, "Semantic Security Against Web Application Attacks", ACM, 2013.
[25] C Anley, "Advanced SQL Injection in SQL Server Applications" White Paper Next Generation Security Software Ltd., 2002. http://www.nextgenss.com/papers/advanced sql injection.pdf
[26] M. Howard and D Le Blane, "Writing Secure Code", Microsoft Press, Redmond, Washington, second edition, 2003.
[27] S.McDoland, "SQL Injection. Modes of Attack, defence and why it matters", White paper, GovernmentSecuity.org, April 2002