International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 10 (2015) pp. 25027-25034

http://www.ripublication.com

Evaluating Vulnerability Tool To Analyze The Web Security Mechanism

Arunraj L, Vasanth S, Ramanathan M
Department of Computer Science and Engineering,
B S Abdur Rahman University
Chennai, India
arunraj@bsauniv.ac.in
Department of Computer Science and Engineering,
B S Abdur Rahman University
Chennai, India
vasanth2vasu@gmail.com
Department of Computer Science and Engineering,
B S Abdur Rahman University
Chennai, India
Mugunthan_1994@yahoo.com

Abstract

In this paper a tool is developed to evaluate mechanisms in web application security. Here existing vulnerabilities are found and are attacked automatically to assess the existing mechanism and the tool. The proposed vulnerabilities methods include attack in real time scenarios. In this paper, Evaluating Vulnerability tool (EVT) is developed to automate the evaluation process. This method used SQL injection, XSS and Denial of service (DoS) attacks to assess the effectiveness of the web application. The result shows that injecting vulnerabilities is not only an effective way to assess the web security but can also improve the weakness in it.

Keywords: Web security, Denial of service, vulnerability injection, SQL, XSS

Introduction

Web applications have increased a lot nowadays. They are used in most of the industries. A web application may be mails, commerce applications, blogs, etc. [1]. The omnipresence of the web application in economy and personal life can make it a target for hackers to exploit the system.

Vulnerability is a weakness which reduces information assurance in a system by the attacker. This system is based on the idea through which vulnerabilities are injected in a web application and attacking them automatically. To provide life to system large numbers of vulnerabilities are taken and attack them. This is done by the Evaluating vulnerability Tool (EVT). It point out the weakness in the system and shows how to improve it also. Both static and dynamic analysis of the web application is done to gather the required information. The tool focus on SQL injection, Cross Site Scripting (XSS) and Denial of Service (DoS) attacks.

Evaluating Vlunerability Tool

The method of this evaluating vulnerability tool is based on injecting realistic vulnerabilities to exploit the Intrusion Detection System (IDS). Once the evaluation is done it can be used to improve the weakness in the system. There are several ways to exploit the systems weakness using vulnerabilities. The most common once with high possibilities are SQL injection, Cross Site Scripting (XSS) and Denial of Service (DoS).

SQL Injection

In the SQL injection method tweaked scripts are given as inputs to expose the backend database [2]. This allows the hackers to have access to all the database related records. Once he gets all the required records from the database he could have access to confidential files. This attacks mostly occurs when the system process the query without proper validation.

Cross Site Scripting

The attacks target scripts which are embedded in a web page or application which are executed on the browser (client) and not the server side. The scripts mostly include JavaScript and Html [3]. The concept of XSS is to manipulate the client webpage in the needs of the hacker or malicious user. These manipulated scripts are executed when the page is loaded every time. The attacker changes some of the functions so that he can gain access to the webpage. Both the vulnerabilities occur, when the inputs are not validated by the system.

Denial of Service

The attacker targets the web pages so that the user cannot use any service from the webpage. The web application cannot easily make difference between ordinary traffic and an attack. Most of the web application servers can handle up to several hundred requests from a concurrent user. A single attacker can generate enough traffic to swamp the web application. The attacker targets the web server and keeps on sending request so that the web server cannot respond to the client's request.

TOOL SETUP

There are two external probes, the HTTP and SQL probe to monitor http and SQL data exchanged respectively. These probes send a copy so that it can be analyzed by the tool. As it is a web based attack it needs a web server for the http interaction. All

the client, database, server and tool are linked together so that the data can be analyzed. Figure 1 show a typical setup of the EVT tool with their respective probes.

Tool Architecture

Figure 2 show the internal components of the EVT [4]. The internal components include the vulnerability injector and attack inject followed by the respective probes. There are 4 stages in the architecture

1. Preparation stage
2. Vulnerability injection stage
3. Attack Load generation stage
4. Attack stage

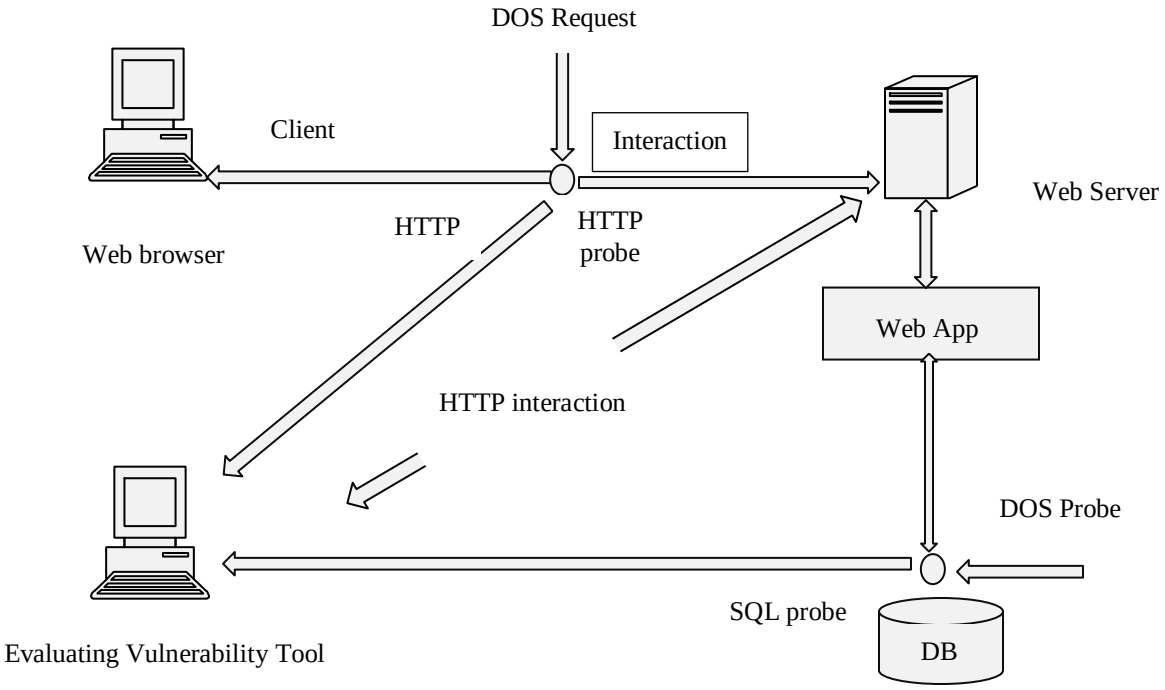

Figure 1: EVT in Setup

Preparation Stage

This phase is where all the functionalities are tested using the web application. The http, SQL and DoS communications caught and processed for later using the two probes. The interaction takes place from only the client point of view so that the vulnerability can be analyzed. The output will be the correlation of the input values and the queries which will be sent back to the backend database for SQL, the web browser for XSS and the web server and the client for DoS respectively.

Vulnerability Injection Stage

This stage is where the vulnerabilities are injected into the web application. For this, the information and relevant information are analyzed to execute the attack on the

web application [5]. The stage starts on analyzing the web applications source code followed by the location of the vulnerability where it can be injected.

Attack Load Generation Stage
Once the source code of the web application including the vulnerability injection are collected the malicious Attack Load the will be used to attack each vulnerability. The malicious activity data is called the Attack Load which is needed to attack the vulnerability given. This data depends on the interaction patters from the first phase by changing the input values of the vulnerable variables.

Attack Stage
The final stage is where the web application is interacted again. This interaction consists of attack payloads to exploit the vulnerabilities so it's called malicious interaction. This attack changes the SQL query that is sent to the database server through web application for SQL attack, html data that is sent back to the user in the case of XSS attacks and number of request are send from a client to server to stop the service of the web server.

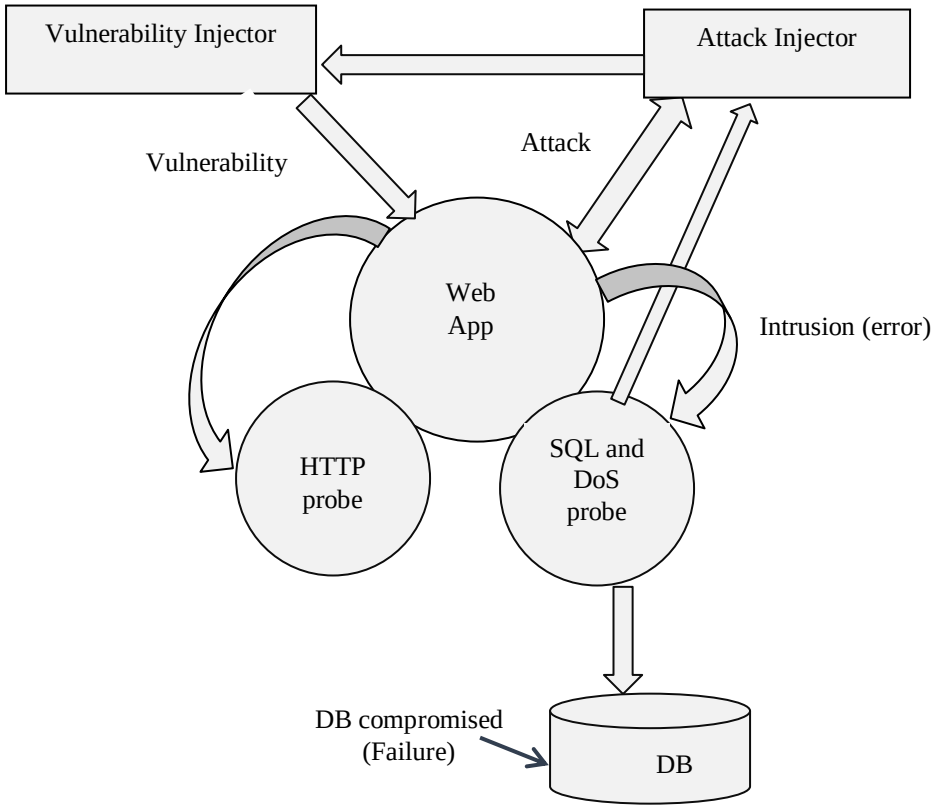

Figure 2: Internal Components of EVT

Attack Injection Scenario

This consist of two stages

1. Inline
2. Offline

Inline

Security assurance mechanism under evaluation is also being executed by EVT. EVT is used to evaluate security assurance mechanism and tools, like IDS for databases, reverse proxies, web application firewall and web application IDS [6]. For example, when assessing IDS for database, IDS should be placed after SQL probe, such that IDS is between database and SQL probe. During the attack stage, when IDS inspect the SQL query that is sent to the database, the attacker injector tool also monitors the output of IDS to detect whether the IDS has detected the attack or not. The entire process is without human interaction (automatic). The output of EVT contains logs of the IDS. The attacks which were not detected by the IDS, the security head could gather the data of the weakness in the IDS and gather information so that he could improve it.

Offline

A set of realistic vulnerabilities are provided for later use once the EVT is executed. This offline scenario is where EVT injects vulnerabilities into the web application to check if it is exploited or not [7]. Once the attack is over it can be used to train and evaluate security members who are doing penetration testing, code review, estimating number of vulnerabilities that are still present in the code, testing static code analyzer, evaluate web application vulnerability scanner and so on. It can also be used in honey pots which are prepared to collect data about the hackers. It gathers details about how hackers operate, which asset they want to attack and how they use the other vulnerabilities to attack the whole system. It also can be applied to assess the quality of the test case developed for a given web application. In this setup the attack can be injected once or multiple depending upon a vulnerability scanner or a training to a group of security team.

Attack Scenario

An attack is considered successful only if has the ability to check for the errors. To avoid attacks developers reduce the number of errors that will be displayed to the users so that it cannot be attacks with the vulnerability. Some experiments are conducted to demonstrate the purpose of EVT. Vulnerabilities are injected in the web application to verify the attack performance and quality of the attack that is injected [8]. The experiment uses Linux and apache web server and has MySQL database for the backend. Table 1 shows the results of target web application using attack injection.

Table 1: Result on Attack Injection on Web Application

Web Apps	Files Attacked	Code lines	Vuln. Injected	Attacks	Attacks successful	Vuln. Attacked suucessfully
Kiwirik	kiwi_editpge.p	900	3	81	31	2
	kiwi_index.ph	645	1	4	3	1
	kiwi_login.php	302	2	18	0	0
	Total	1847	6	103	34(35%)	3(55%)
phpA	search.php	1400	3	40	40	3
	login.php	220	1	19	19	1
	viewform.php	690	1	5	5	1
	viewtable.php	1206	4	83	83	5
	Total	3516	9	147	147(100%)	10(100%)
Refs	edit_page.php	306	25	520	55	25
	edit_title.php	165	3	193	49	5

Figure 3 shows the graph that is generated for the attack injection on the three web applications. A comparison can be made through the number of files attacked, number of codes in the web application, vulnerabilities that are injected and the successful attacks that are made through the web application.

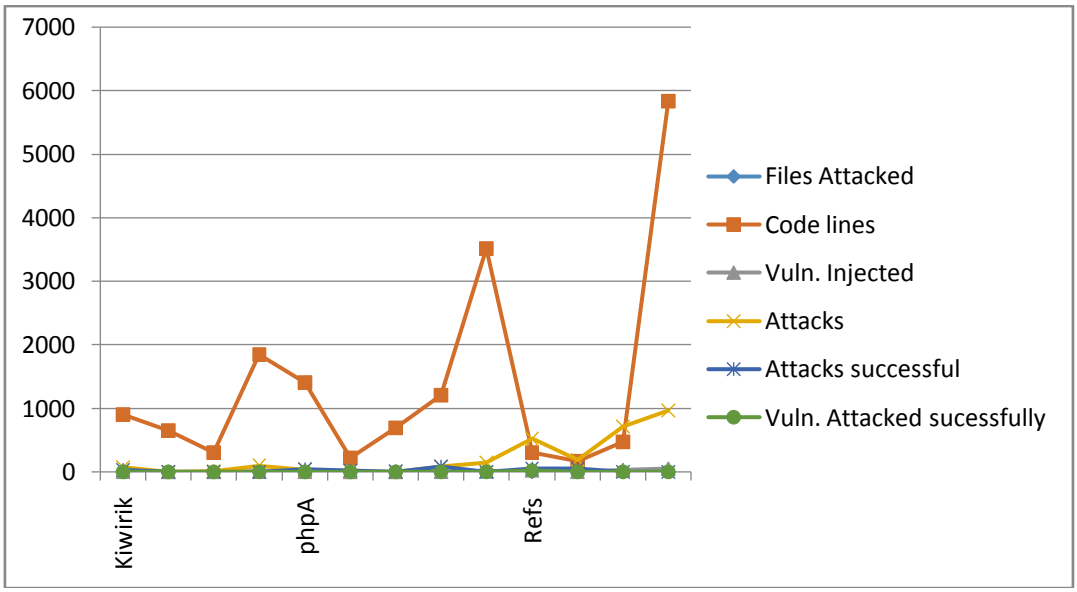

Figure 3: A Graph on successful attack

Table 2 shows the results that are evaluated using IDS. To evaluate the performance of the IDS we should know its ability to detect various attacks. This is done by analyzing the ratio between successful attacks and false positive or alarms that are show by the IDS [9]. Both the metrics are used to analyze the characteristics of the IDS [10]. With the ratio a message can be sent to developers so that they can know if they want to make improvements to the IDS.

Table 2: Ratio of IDS Evaluation

Web Apps	Files Attacked	Vuln. Injected	Total attacks	Successful attacks	Attacks detected by IDS	IDS false positives
Kiwirik	kiwi_editpge.php	3	81	31	31	40
	kiwi_index.php	1	4	3	3	1
	kiwi_login.php	2	18	0	0	25
	Total	6	103	34	34(35%)	66(90%)
phpA	search.php	3	40	40	40	0
	login.php	1	19	19	19	0
	viewform.php	1	5	5	5	0
	viewtable.php	4	83	83	83	0
	Total	9	147	147	147(100%)	0(0%)
Refs	edit_page.php	25	520	55	56	300
	edit_title.php	3	193	49	38	35
	Total	28	713	104	94(92%)	335(60%)
	Grand total	**49**	**963**	**285**	**275(98%)**	**401(68%)**

Conclusion

A review on the evaluation vulnerability tool and its performance on web security mechanism in analyzed. This proposed attack shows effective ways to evaluative web security mechanisms like IDS providing during a breach in the system. By injecting vulnerabilities and attacking using EVT weakness in IDS is found. These results were important in improving the IDS. EVT is used to evaluate commercial vulnerabilities like SQL injection, XSS and DOS. The scanners were used to evaluate most of the vulnerabilities in the system. This clearly shows that there is development in the IDS so that the whole system can be secure.

References

[1] D. Avresky, J. Arlat, J, C. Laprie and y. Crouzet, “Fault Injection for Formal Testing of Fault Tolerance”, IEEE Trans. Reliabilty vol. 3, no. 3, pp. 443-455, 1996.

[2] J. Fonseca and M. Vieira, “Mapping Software Faults with web Security Vulnerabilities”, Proc. IEEE Int’l Conf. Dependable Systems and Networks, June 2008.

[3] J. Carreira, H. Madeira and J. G. Silva, “Xception: Software Fault Injection and Monitoring in Processor Functional Unit”, IEEE Trans. Software Eng., vol 24, no. 2, Feb 1998.R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[4] J. Fonseca, M. Vieira and H. Madeira, "Testing and Comparing Web Vulnerability Scanning Tools for SQLi and XSS Attacks", Proc. IEEE Pacific Rim Dependable Computing, Dec 2007.

[5] J. Fonseca, M. Vieira and H. Madeira, "The Web Attacker Perspective-A Field Study", Proc. IEEE Int'l Symp. Software Reliability Eng., Nov 2010.

[6] G. Buehrer, B. Weide and P. Sivilotti, "Using Parse Tree Validation to Prevent SQLi Attacks", Proc Int'l Workshop Software Eng. And Middleware, 2005.

[7] Y. M. huang, S. K. Huang, T. P. Lin and C. H. Tsai, "Web Apllication Security Assessment by Fault Injection and Behaviour Monitoring", Proc. Int'l World Wide Web, pp. 148-159, 2003.

[8] J. Fonseca, M. Vieira and H. Madeira, " Detecting Malicious SQL", Proc. Conf. Trust & Privacy in Digital Business, Sept 2007.

[9] Liang Hu, Xiaoming Bi, " Research of DDoS attack mechanism and Its Defence Frame", Computer Reseach and Developmet (ICCRD), 3rd International Conference, pp.440-442, March 2011.

[10] A. Yaar, A. Perrig and D. Song, "PI: A Path Identification Mechanism to Defend Againt DDOS Attacks", in proceedings of the IEEE symposium on Security Privacy, pp. 93-109, May 2003.