Web Application security tool to identify the different Vulnerabilities using RUP model

Deven Gol[1] Nisha Shah[2]
Information Technology,
SVIT College, Gujarat Technology University,
Vasad, Gujarat 388306, India
Deven215@gmail.com

Priyank Bhojak[3]
Computer Engineering,
MBICT College, Gujarat Technology University,
Anand, Gujarat 388121, India
Priyankbhojak@gmail.com

Abstract -
Web applications become an important part for Interacting with the people of all kind now days. As the popularity of the web application increases like online transaction, net banking and many more, the role of web security has been increase as well. Web applications vulnerabilities let attackers to carry out malicious activities that range from gaining unauthorized access or stealing the sensitive data. Past research have shown a significant increase in the number of web application vulnerabilities which is still growing constantly. Vulnerability scanner is a tool used for verify whether web applications are vulnerable or not when they are subjected to improper input validation. Even though there are number of tools available for web application vulnerability still latest attacks (like attacks occur in HTML5) are hard to find. Web application security tool is designed to find out security holes in your web applications that an attacker can access to your whole system and data for malicious purpose. These tools used to find the multiple vulnerabilities including SQL injection, cross site scripting and many more etc. This paper demonstrates how easy it is for attackers to automatically discover and exploit web application-level vulnerabilities in a large number of web applications. This approach allows researcher/developer to develop an extensive good web application vulnerability scanner.

Keywords- *SQLiA, XSS, Broken Authentication, crawler, Web application vulnerability.*

I. INTRODUCTION

Web application is becoming so popular and significant part of our daily lives. As the increasing the use of web application, the web application security is becoming vital is so important for user's secret data. Because of the extensively used of web, any of the web application vulnerability will generally be observed and be broken by hackers. Hackers will get to access system through which he can easily access to the system which is used for malicious activity sooner or later.

In cyber security, the term vulnerability is applied to a security holes in a system that allows an attacker to break the integrity of that system [1]. Vulnerabilities may occur from improper software bugs, computer virus, XSS vulnerability, weak user passwords, and a most important SQLiA attack [11]. Basically testing is needed before deploying the web application.

The main purpose of any testing method can be summarized as explained below:

- Testing is carried out mainly to reveal the occurrence of errors or improper validation that is present during a program implementation.
- A superior testing method will have a privileged opportunity of discovering an error. A successful test case of tool should determine a latest error occurred or a regression error for better secure system.
- Many of the web application security vulnerabilities result from generic input validation and sanitization problems. Examples of such vulnerabilities are SQL injection, Cross-Site Scripting (XSS) and many more. Even though the most of the web application vulnerabilities are easy to recognize and to evade, many web developers are still not security-aware cause a huge number of vulnerable applications and web sites on the web.

Now it's also needed to understand about the method for testing a web application. Below listed two main approaches to testing software applications for the presence of bugs and vulnerabilities [8]:

- In white-box testing, the source code of the application is analyzed in an attempt to track down defective or vulnerable lines of code. This process is often integrated into the progress process by creating add-on tools for ordinary development environments.
- In black-box testing, the source code is not needed for examination so doesn't need to scan directly. As a substitute, particular input test cases are going to be generated and sent to the application. Then, the outcomes returned by the application are analyzed for unpredicted behavior that indicates errors or vulnerabilities.

So far away, white-box testing has not experienced widespread make use of finding protection flaws in web applications. An imperative reason is the incomplete detection capability of white-box analysis tools, in particular due to various programming environments and the complication of

applications that used in incorporate dataset, industry reason, and user boundary mechanism. Even many developers don't want to disclose about the source code. So it is quite difficult and not proper solution to provide white boxing technique. As a result black-box vulnerability scanners are used to find out security problems in web applications. These tools drive by launching attacks in opposition to an application and examining its reaction to these attacks. This is easy and proper solution rather than white box testing. The main contribution of this research paper is that we show how easy it is for hackers to automatically find out and utilize application-level vulnerabilities in a huge number of web applications. So detecting vulnerabilities is normally not an easy job, and not all of the frequent vulnerabilities can be effectively detected by automated existing vulnerability scanners. In addition to this paper will help to propose areas for Web Vulnerability Scanner tool improvement for security.

II. UNDERSTANDING WEB ATTACKS

A. *SQL injection attack*

SQLiA is attack against a database-driven web application, in which attacks are performed through injecting invalid input strings into the database of the system that modify it for their deliberate use[11]. This can be possible if a web application does not provide appropriately filtering the user input. SQL injection can be possible through inserting new keywords, different combination of operators and statements into the original SQL statement. If the attack is successfully performed, the attacker will directly pass an SQL attack code inside the back-end database of the vulnerable application for execution. For some of the conditions he may interrelate with the entire file system with full privilege and even perform system calls as a legitimate user.

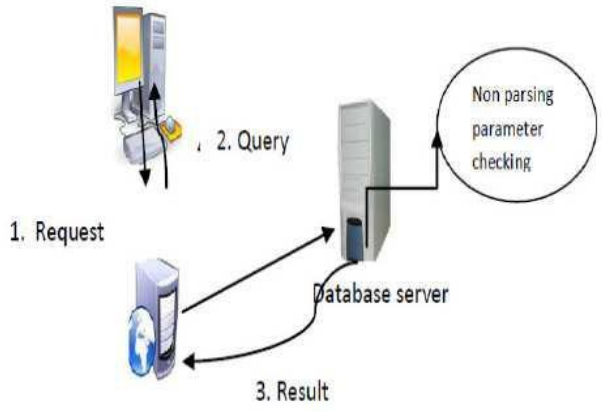

Fig. 1 scenario of SQLIA attack

A web application is vulnerable to an SQL injection attack if and only if an attacker is able to run the SQL statements inside an existing database of the application. It can be normally possible through injecting malicious input into user fields of SQL statements that are used to compile the SQL query.

As listed below example [07] of a web application that uses a query for authenticating its users.

SELECT IDno, LLogin FROM User_table WHERE UserName = 'ejfgh' AND Password = 'xgykz ';

This query retrieves the IDno and LLogin fields of UserName "ejfgh" with password "xgykz" from table User_table . These types of queries are frequently used for assessment of the user login identification and therefore these are the major targets for an attacker. In given example, a login page prompts the legitimate user to enter his username and password in to the given form. When the form is filled with all the details and submitted, its fields are used to assemble an SQL query [07] that authenticates the legitimate user if it verifies by the server.

SQLQuery = "SELECT IDno,LLogin
FROM User_table
WHERE UserName = ' " + userName + " ' AND Password = '
" + password + " ' ";

If the login application does not provided with accurate input validation of the form fields, the attacker can easily inject malicious strings into the query that will alter its semantics and attacks launched successfully. Let's take an example [07], think about an attacker entering user identification such as

UserName: 1' OR '1'='1 --
Password :

Using the provided login form data, the vulnerable web application constructs a dynamic SQL query for authenticating the legitimate user as shown in below [07]:

SELECT IDno,LLogin FROM User_table WHERE UserName = ' 1' OR '1'='1 -- AND Password = ';

The "--" command used to point out a comment in Tra nsact-SQL which is mostly undetected afterword data. Hence, all subsequent the "--" is unobserved by the SQL databa se engine. Because of the starting quote in the input string, the user name string is ended, while the "1' OR '1'='1" adds a phrase to the query which evaluates to true condition for the every row in the table. When executing this query by the system, the database returns all user rows of the system, which applications often interpret as a valid login or legitimate user login.

B. *Cross-Site Scripting (XSS) attack*

XSS allows attackers to inject client-side script in a web page for malicious purpose. Static websites are less risky as compare to dynamic. The attacker injects script, such as JSP, VBScript, ActiveX, HTML tag, or flash in a web application to attempt to get access to sensitive information such as Dynamic websites created in php or others are vulnerable. XSS attacks are normally easy to perform, but complicated to avoid and can cause important damage [2].

There are mainly two types [2] of XSS attacks are famous as known: *reflected* and *stored* XSS attacks.
Reflected XSS attack is the type of XSS which is mostly found in vulnerable application. Unfortunately, the search form on the web site fails to perform input validation that means simply bypass the authentication, and each time a search query is entered that does not acknowledged with any results, the user is displayed with a message that also contains the unfiltered search string.

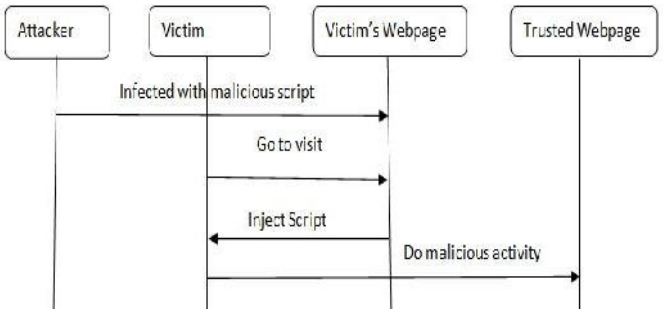

Fig 2: Sequence view of XSS attack

The second type of XSS attack is called stored XSS attack as its name suggests, the dissimilarity compared to the reflected attack is that the malicious script is not instantly reflected back to the victim or legitimate user by the server, but that will be stored inside the vulnerable application for later retrieval.
The attack string can be URL encoded so that the content is unreadable for the average Internet user. An attack is successful if a victim visits an URL containing the XSS attack. This can be achieved by e.g. E-mail from a sender in which the user has trust.

III. VULNERABILITY DETECTION

As we are going to use RUP based Framework it correlate the four Phases of the system are listed below [01]:

Inception : Requirements capture and analysis
Elaboration : System and class-level design
Construction : Implementation and testing
Transition : Deployment

The Web vulnerability scanner proposed in this research consists of four components which are described as per the RUP based model:

First Module of our Framework is using crawler that is Crawling component which is used for collecting a set of target web sites for the system.
Second Module of the system is web application Attack component which launches the configured attacks like SQLiA, XSS etc. against these targets collected web sites by crawler.
Third Component is important component as used for Analysis which observes the results returned by the web applications to verify whether an attack was successful.
And the last and forth Module of this system is used to generate a report of the scanning process.

The main function of web crawler is interrelating with the web applications, and to collect the information for better execution such as web pages forms etc. for detection engine of the system.

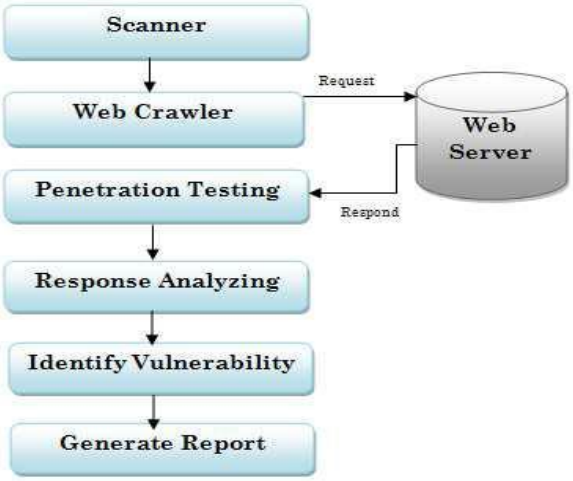

Fig. 3: Proposed System

Detection engine is used to provide the web request through some specified attacking code or script.
Detection engine waits for the response getting from the web server that is also analyzed and if it will detect the specified vulnerable script in the responded data, the vulnerability is recognized successfully.

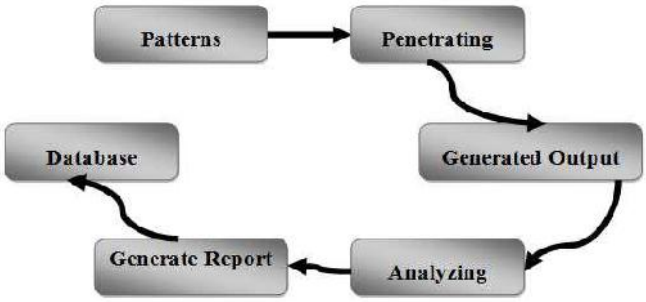

Fig.4: Flow of the System

1. **Web crawler**
   A Web crawler is a internet bot that is used for browses the World Wide Web in a methodical, robotic approach which can copy all the pages they visit for afterward processing explore them much more quickly [7].
   It will gather all the pages, scripts from the Web for better performance afterwards.
   So in this Proposed Research we are going to build a RUP based Framework vulnerability tool which is

used a web crawler that is crawls the web and gather information about the website and gather all link into the website.

2. Attack Component

In the second phase of system by means that the crawling phase has completed then this component scans each page gathered by crawler for the existence of web forms. The motivation is that the fields of web forms represent our entrance points to web applications.

Penetration testing: In this procedure we are going to penetrate the elements which are gathered by crawler via web crawling the URL. Penetration is done by the script, which contains different patterns of the attacks for penetration. Penetration testing is going to be implemented as black box testing.

The web application vulnerability identification step is a very essential phase in penetration testing [8].

This will allows the user to resolve the weaknesses of the target system and the point where to launch the attacks. Penetration testing approach is based on simulation of web attacks against web applications. In fact black box penetration testing component contains the four-step process explained below:

1) The first step of this phase is to identify all pages being part of the web application. This task can be fulfilled automatically using web crawlers.

The second step is to extract Data Entry Holes from the pages visited and the result set of DEHs should be analyzed.

The third step is simulation of attacks each constraint in all DEHs is fuzzed with malicious scripts or patterns and used within an HTTP request sent to web application.

2) The forth and last step provide every acknowledged HTTP reply is scanned for indication of present web application vulnerability.

3. Analysis module

In this module after an attack has been launched, the analysis module has to parse and understand the server response.

An analysis module makes use of attack-specific response criteria and keywords to estimate a confidence value to make a decision [4] if the attack was successful or any false positives are possible.

4. Generate Report

This is the last module in that we generate a report of vulnerability list detected by penetrating testing in the web page. And report will generate by the category-wise of attacks in pie chart form and tabular form.

IV. Experimental Setup and Results

The basic environment is developed on windows 8 machine. We use the PHP language for our tool. As a front end tool we use Dreamweaver 8.As a back end tool MySQL and Wamp server.

Figure 5 shows the result of crawler phase. It will fetch URL of selected request.

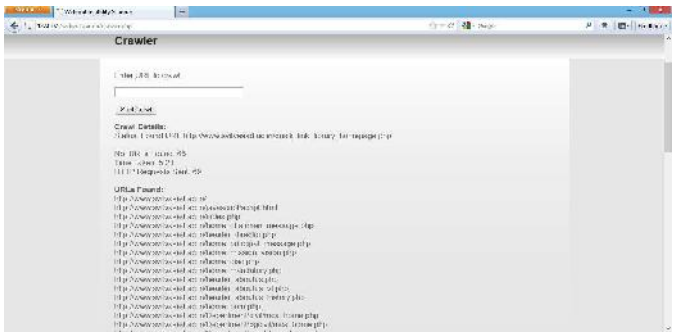

Fig 5: Fetching URL of SVIT using Crawler

After fetching the URL Module of the system Attack component which launches the configured attacks against these targets collected web sites by crawler. This is forwarded to Analysis component that will observe the results returned by the web applications to verify whether an attack was successful. Figure 6 depicts the progress of Discovery phase in our web application tool.

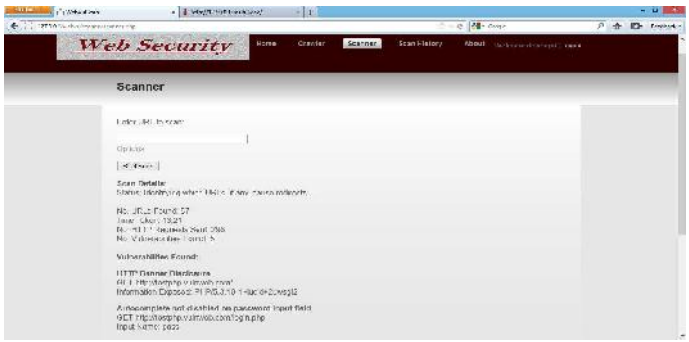

Fig 6: After Analysis of scan test

V. Conclusion & Future Work

The basic idea about this research is to explain how we can do automatically discover and exploit web application- level vulnerabilities in a large number of web applications. Many of the web application security vulnerabilities result from generic input validation problems. A tool to identify the Web application vulnerabilities based on the RUP (Rational Unified Process) Model based on web crawling is proposed in this research. In the future, our research will includes improving on detecting web security vulnerabilities. In this research we proposed a better approach to build SQL injection and XSS vulnerability detection in proper manner and more works on studying the complex-form on analyzing, the attacking codes constructing and the response analyzing. To this end, we implement a generic web vulnerability scanner that analyzes web sites for exploitable SQL and XSS vulnerabilities specially.

ACKNOWLEDGMENT

I would like to express my deep and sincere gratitude to Mr. Priyank Bhojak my advisor for this research.

References

[1] Priya. R. L1 , Lifna. C. S ,"Rational Unified Treatment for Web Application Vulnerability Assessment " ©2014 IEEE.

[2] Guowei Dong1, YanZhang2,Xin Wang1,Peng Wang2, Liangkun Liu2, "Detecting Cross Site Scripting Vulnerabilities Introduced by HTML5", Renmin University of China, China ©2014 JCSSE.

[3] Yuan-Hsin Tung, Chen-Chiu Lin, Hwai-Ling Shan Telecommunication Lab.,Chunghwa Telecom Co., Ltd., Taiwan,ROC , "Test as a Service: A framework for Web security TaaS service in cloud environment ", Beijing University of Posts and Telecommunications, Beijing, China. 978-1-4244-6769-3/10/$26.00 ©2014 IEEE,p.- 14-18.

[4] 12Yuan-Hsin Tung,23Shian-Shyong Tseng , 1Jen-Feng Shihl, 1Hwai-Ling Shanl 1Telecommunication Lab., Chunghwa Telecom Co., Ltd., Taiwan,ROC , "A Cost-Effective Approach to evaluating Security Vulnerability Scanner ", Beijing University of Posts and Telecommunications, Beijing, China. © IEICE ,2013.

[5] Xin Gopal R. Chaudhari, Prof. Madhav V. Vaidya Department of Information Technology,SGGS IE & T, Nanded, Maharashtra, "A Survey on Security and Vulnerabilities of Web Application ", ©2014 IJCSIT.

[6] Rajesh M. Lomte1 , Prof. S. A. Bhura2 Computer Science & Engineering Department, BNCOE, India , "A Survey on Security and Vulnerabilities of Web Application IOSR-JCE, 2013

[7] Xin Wang, Luhua Wang, Gengyu Wei, Dongmei Zhang and Yixian Yang, "Hidden Web Crawling For Sql Injection Detection ", Beijing University of Posts and Telecommunications, Beijing, China. 978-1-4244-6769-3/10/$26.00 ©2010 IEEE.

[8] Andrey Petukhov and Dmitry Kozlov, "Detecting Security Vulnerabilities in Web Applications Using Dynamic Analysis with Penetration Testing", Dept. of Computer Science, Moscow State University

[9] [illegible]

[10] Katkar Anjali S and Kulkarni Raj B, "Web Vulnerability Detection and Security Mechanism", International Journal of Soft Computing and Engineering (IJSCE),ISSN:2231-2307, Volume-2, Issue-4, p.-237-241

[11] https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

[12] V. Suhina, S. Groš and Z. Kalafatić, " Detecting vulnerabilities in Web applications by clustering Web pages" , Faculty of Electrical Engineering and Computing,University of Zagreb , Croatia.

[13] Acunetix Ltd. Acunetix Web Vulnerability Scanner. http://www.acunetix.com/, 2005.

[14] Jeremiah Grossman WhiteHat Security founder & CTO "Website Vulnerabilities Revealed " WhiteHat Security.

[15] J.Dhanamma, and T. Rohini, "The Unified Approach for Organizational Network Vulnerability Assessment", IJSEA, Vol 4, No.5, September 2013.

[16] A. Riancho, "w3af User Guide"–Document Version 2.1, August, 2012.

[17] I. Jacobson, G. Booch, and J. Rumbaugh, "Rational Unified Process – Best Practices for Software Development Teams", Rational Software Corp.,White Paper , TP026B, Rev 11/01.

[18] P. Kruchten, "The Rational Unified Process 3rd Edition: An Introduction". Reading, MA: Addison-Wesley Longman, Inc., 2004.

[19] W. Royce, "Software Project Management: A Unified Framework". Reading, MA: Addison-Wesley Longman, Inc., 1998.