Kalpa Publications in Computing
Volume 2, 2017, Pages 177–187
ICRISET2017. International Conference on Research and Innovations in Science, Engineering &Technology. Selected Papers in Computing

Automated Web Application Vulnerability Detection With Peneteration Testing

Priyank Bhojak[1] ,Vatsal Shah[2] ,Kanu Patel[3] ,Deven Gol[4]
[123]I.T. Department,BVM
Vallabh Vidhyanagar,India
[4]C.E. Department,
SOEC, Ahmedabad,India
priyankbhojak@gmail.com,vatsal.shah@bvmengineering.ac.in,
kanu.patel@bvmengineering.ac.in, deven215@gmail.com

Abstract

The rate of web application threats is growing more and more now in days. The most of software bugs are result from inappropriate input validation. It should lead to attack of confidential information, breaking of knowledge integrity. We develop a scanner for detecting SQ injection and XSS type software-bugs which is based on hidden web crawl and make open source scanner with the aim of hidden web crawling which may be require authentication. In this research paper we presents a new technique to find vulnerability which include advantages of black-box analysis of different web pages. And at the end we shows evaluation table which mention comparison of our scanner with two other web scanner tool. So finally this paper additionally shows how easy it is to scan web application bugs with dynamic analysis and retrieve hidden web pages from web applications.
Keywords— SQL Injection, Web application vulnerability, Penetration Testing, Hidden webpages crawler, Web Scanner tool

R. Buyya, R. Ranjan, S. Dutta Roy, M. Raval, M. Zaveri, H. Patel, A. Ganatra, D.G. Thakore, T.A. Desai, Z.H. Shah, N.M. Patel, M.E. Shimpi, R.B. Gandhi, J.M. Rathod, B.C. Goradiya, M.S. Holia and D.K. Patel (eds.), ICRISET2017 (Kalpa Publications in Computing, vol. 2), pp. 177–187

1 Introduction

Web based application is turning out to be increasingly prevalent and critical piece of our lives. As the imperative part of web based application, Security of web application is getting to basic [1]. In Security, the word vulnerabilitiy is connected to a shortcoming in a framework that allow an attacker to damage the respectability of that system. Recent scenario in web applications the vast majority of seen vulnerabilities are Cross Site Scripting (XSS), Structured Query Language (SQL) Injection, feeble passwords, Broken Authentication utilizing SQL Injection, Auto-complete enabled on information fields, Directory Listing, Invalidated Redirects and so forth. These vulnerabilities have been notable for quite a long time. XSS and SQL infusion assaults are basic which can be run through simple scripts. Identifying vulnerabilities is for the most part not a simple errand, and not the majority of the basic vulnerabilities can be effectively identified via computerized scanners.

Most of the software bugs in web application are result from invalid input sanatization.[9]. these vulnerabilities may be SQL injection and Cross-Site Scripting .Although the dominant part of web vulnerabilities are straight forward and to maintain a strategic distance from, numerous web designers are, shockingly, not security-mindful. Subsequently, there exist an expansive number of venerable applications as well as websites on World Wide Web. Most important ways to deal with testing programming applications for the web applications are static (white-box) and dynamic (black-box).

White-box test is a method of testing software in which the internal working is known to the tester. White box testing validates the internal code and therefore often focuses primarily on improving security and making the flow of inputs/outputs more capable and optimized. In white box testing, the tester is often testing for internal security holes. Now, If we talk about black box testing , than special code or script are generated and send to web application. After that result or response is gathered and find if any errors or bugs are there or not. So black box testing is work externally, hence no source code is needed for this testing.

At the end, we demonstrate that how concealed web crawler work into the identification framework or scanner instrument to sidestep the verification of web application and recognize web application helplessness existed in them by reenacting web assaulting and investigating the information of reaction. In second area we depict run of the mill web assaults. In area 3 we initially proposed the portrayal of web application vulnerabilitiy (SQL Injection and XSS) and location component. In segment 4 and 5 portray shrouded web slithering system and Analysis XSS and SQL infusion Attack separately. Area 6 indicates after effect of scanner with examination with various scanners. At long last we finish up this paper and examine our future works.

2 Different Webattacks

2.1 SQL-injection attack

SQL-injection assaults depend on infusing strings into database questions that change their planned use. This can happen if different website doesn't use proper client input [6]. There are numerous assortments of SQL. A question ordinarily brings about a solitary result set that contains the inquiry comes about. Aside from information recovery and upgrades, SQL proclamations can likewise change the structure of databases utilizing Data Definition Language explanations.

A web application may have software bug if anyone can attack with sql query string. This kind of activity mostly done thorough inserts malicious code in different user input. Now we consider one following example which authenticating users from database.

SELECT id, MyLastLogin from Client WHERE Userid = 'mynamea' AND userpassword = 'mypassworda '

This kind of sql query mostly use for check user authentication. So attacker mostly targets this type of sql query. If we consider query, it retrieve id and mylastlogin data for userid "mynamea" as well as userpassword "mypassworda" from table client. Now suppose client enter userid and userpassword in to input field , query look like following.

Sql-Query = "SELECT ID , LastLogin FROM client WHERE Userid = '" + userName + "' AND userpassword = '" + password + "'"

In web application code if developer does not use input filter or input validation than attacker can inject some code which might be alter its meaning in case of executing sql query.for example anyone insert username and password like following.

Userid: ' OR 1=1 --
Userpassword :

Using the provided form data, the vulnerable web application constructs a dynamic SQL query for authenticating the user as shown in

SELECT ID, myLastLogin FROM client WHERE Userid = ''OR 1=1 -- AND userpassword = '

In sql-query SELECT ID, LastLogin FROM Users WHERE User = ''OR 1=1 -- AND Password = ' Shows '--' character mean that its comment in SQL. So after '--' is mostly mistreated by sql database. And if sql engine check "OR 1=1" than it means it's always true for every field in table . So whenever database engine executing this type of query it returns all user data, means its valid login for that userid.

2.2 Cross-Site-Scripting web attack

Cross Site Scripting which is also called XSS, allow attacker to inject client-side script in input fields of web application. The most widely recognized one establish in web application now in day is call reflected XSS. The web structure on the site neglects to perform input acceptance, and at whatever point a hunt question is entered that doesn't gave back any outcomes, the client is shown a message that likewise contains the unfiltered seek string. Anyone can insert script like Java-Script, VB-Script, Active-X, HTML into an website to try to find access to receptive information of dynamic websites which are vulnerable.

For example if any client enter string "<B>My Name </B>", the Bold markers (like, <b>) which may not filtered, So whenever client search with this type of script , result come from server that "no matches found for **My Name**" (here string show in bold letter). This means here XSS software bug is available in web application or web site.
There are two type of cross site scripting, one is Stored XSS and second is reflected XSS.

3 Web-Vulnerability Detection

Software bug scanner or web scanner mostly develop in four section. At the first section web pages are gathered from the web application with the use of web crawler. After that dynamic analysis or penetration testing will be done with the use of some predefine code attack. And third section check the output of web attack which is done by penetration test and examine output to make decision that bug is present or not. And at the in fourth section scanner make report of different vulnerability present on that web application.

The web crawler interfaces with web-applications, and accumulates data for scanning. Discovery part develops web demand with some predefined assaulting code. Web spider or web crawler sits tight for the reaction and breaks down it, once the predefined catchphrases can be distinguished in the reacted information, the weakness is recognized.

3.1 Detecting Vulnerabilities

The mainly productive method for discover Software bugs is manually check the code in web-sites. This method is require more time, need professional ability to overlooked errors. These methodologies can be isolated into wide classifications: discovery testing and white-box testing. Static analysis test comprises of looking at the code with-out of execute it.
And web scanner mostly scan automatically without source code.which follow black box testing concept. So special code or script are generated and send to web application. After that result or response is gathered and find if any errors or bugs are there or not. So black box testing is work externally means from an outside perspective, hence no source code is needed for this testing. to the examination of system execution.

3.2 Attack Component

In this section, Scanner scans each and every field of web forms. Means web pages which are contain web forms (input fields) is scan because it is entry point of web forms and may contain software bug.

Now Scanner extract method which is use for submit the form. It may be get or post. And also find the parameters or attributes which are used in form fields. After than scanner may attack on different fields with appropriate values using get or post method. And server reply back with some response or information through HTTP protocol.

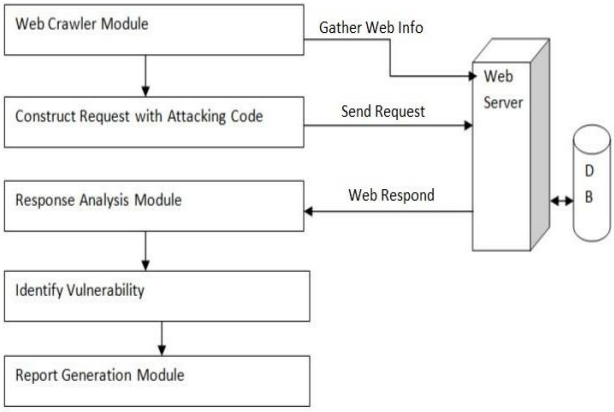


Figure 1. How Automated vulnerability scanner work

As per above figure1, first web crawler or web spider module connect with website and collect the web pages of that website. Now Scanner build web-request and along with send code for attack. Response analysis module or section get the response and if scanner find some specified keywords, software bug or error is recognized. After that report will produce based on category and risk level-wise.

3.3 Hidden web crawling strategy

Presently in days numerous web administrations oblige clients to enter the right confirmation or intuitive data, so they can furnish clients with the right comparing web administration or site page. What's more, there will be numerous intelligent components or structures in the site pages reacted after the verification, through which websites can collaborate with clients & give essential web administrations and data. Nonetheless, if there exist possible security issues in the cooperation amongst users and the web sites, it will bring about intense results to clients. Subsequently, if the data of these intuitive units in the shrouded web amid the way toward slithering, it can enormously enhance generally speaking of the security weakness location.[2]

By getting to approval, web-spider can be validated by the targeted framework and increase site content behind the login page, and proof the approval of data after reply of confirmation, for example, we can say session & cookie or local storage, etc. If chance that there are different links or from-fields in the reply pages, every one of them might be as the beginning stage of traversal so the pages will be gathered recursively, and got the more secret web pages and get better attack component.

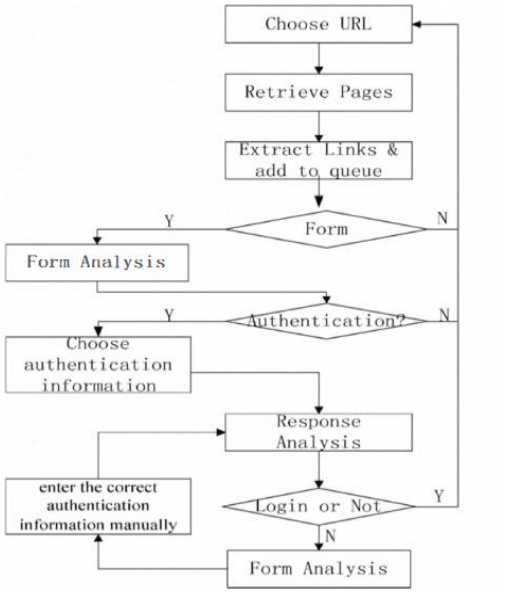


Figure 2. Hidden web-spider technique

3.4 Analysis SQL-injection & Cross Site Scripting Attack

Third but the most important section which take reply from web application and identify keywords so scanner can decide that bug or vulnrability is present or not. First make a web request and sends it to the targeted website with some predifine code for blackbox test. And receive response page which is check some specify keywords which may contain bugs. [17]. The following code can be use for easy and simple XSS bug injection.

< Ṣcript > alert ('Hacker Inside XSS ');</ Ṣcript >

Any developer is use some input validation or filter marks than code will be more secure and attack is not possible on web application.

To find reflected XSS from web application we addbelow string to one array, add a corresponding string (to look for in response), with the same index, in the other array.

$payloads=
array('<webvulscan>','javascript:alert(webvulscan)');

The corresponding string that we look for it in response after submitting above payload.

```
$harmfulResponses =
array('<webvulscan>','src="javascript:alert(webvulscan)"');
```

So now first we check that requested URL pass through this function or not. we have to check requested URL to have contained any parameter or not as shown below:

```
$log->lwrite("$urlToCheck does contain parameters");
                $scheme = $parsedUrl['scheme'];
                $host = $parsedUrl['host'];
                $path = $parsedUrl['path'];
                $query = $parsedUrl['query'];
```

If requested URL contains parameter than for each parameter we have to check it with current payload. If any error found in URL then after we have check into response page body, and there if harmful response found then reflected XSS present in those URL.

To check the SQL injection we add one function testForSQLi($urlToCheck, $urlOfSite, $testId) that checks the URL for forms and attempts to submit SQL strings into each value of the forms. In this function we add some regular expression to one array for the checking into response URL code, that get from when requested URL give response, those array given below:

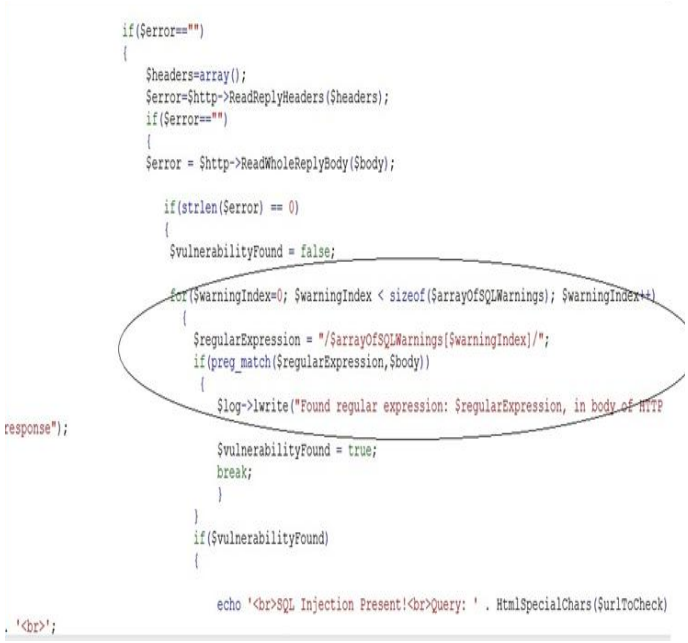
```
$arrayOfPayloads = array( " ' " , ' " ' , ' ; ' , ' ) ' , ' ( ' , ' . ' , ' -- ' ) ;
```

After this array was submitted we add one function to check the error present into response code or not. In that function we take some known SQL error present now a day in market. If admin want to add some new one than also add it into this function that given below

```
arrayOfSQLWarnings = array(
    "supplied argument is not a valid MySQL",
    "mysql_fetch_array\\(\\)",
    "on MySQL result index",
    "You have an error in your SQL syntax;",
    "You have an error in your SQL syntax near",
    "MySQL server version for the right syntax to use",
    "\\[MySQL\\]\\[ODBC",
    "Column count doesn't match",
    "the used select statements have different number of columns",
    "Table '[^']+' doesn't exist",
    "DB Error: unknown error",
    ":[\s]*mysql",
    "mysql_fetch",
    "System\\.Data\\.OleDb\\.OleDbException",
    "\\[SQL Server\\]",
    "\\[Microsoft\\]\\[ODBC SQL Server Driver\\]",
    "\\[SQLServer JDBC Driver\\]",
    "\\[SqlException",
    "System.Data.SqlClient.SqlException",
    "Unclosed quotation mark after the character string",
    "'80040e14'",
    "mssql_query\\(\\)",
    "odbc_exec\\(\\)",
    "Microsoft OLE DB Provider for ODBC Drivers",
    "Microsoft OLE DB Provider for SQL Server",
    "Incorrect syntax near",
    "Syntax error in string in query expression",
    "ADODB\\.Field \\(0x800A0BCD\\)<br>",
```

Figure 3.Examples of some error messages

Presently after that blunder capabilities we check does the URL went into this capacity hold constraint and submit payloads as those parameters on the off chance that it does. Into reaction body if discover any parameter as blunder inclined then its check for which customary expression are there. Through that character it discovers which SQL weakness is available into structure as checking appeared into code:



```
if($error=="")
{
    $headers=array();
    $error=$http->ReadReplyHeaders($headers);
    if($error=="")
    {
    $error = $http->ReadWholeReplyBody($body);

      if(strlen($error) == 0)
      {
       $vulnerabilityFound = false;

      for($warningIndex=0; $warningIndex < sizeof($arrayOfSQLWarnings); $warningIndex++)
        {
          $regularExpression = "/$arrayOfSQLWarnings[$warningIndex]/";
          if(preg_match($regularExpression,$body))
           {
             $log->lwrite("Found regular expression: $regularExpression, in body of HTTP
response");
             $vulnerabilityFound = true;
             break;
             }
        }
        if($vulnerabilityFound)
        {

             echo '<br>SQL Injection Present!<br>Query: ' . HtmlSpecialChars($urlToCheck)
. '<br>';
```

Figure 4. check for SQL injection present or not

4 Experiment results

Testing the web vulnerability scanner consists of several tests. Test created by us on some existing web applications: WebGoat, dvwa, Mutillidae, bwapp. Specified all tests were performed with the web vulnerability scanner to scan for all vulnerabilities.To compare our open source scanner, we performed tests on some web applications more accurately, We checked the output of certain combination of inputs, which gives desirable result, or not. We scan the some web application to get the result for different web vulnerability. Here in following table we conclude and show result based on different website tested using three scanners.

		Scanner 1	Scanner 2	Our Scanner
Site tested	***Vulnerabilities***	***Number of Vulnerability Detected***		
Site 1	Cross Site Scripting	5	7	8
	SQL Injection	8	7	7
	Directory Listing	11	9	12
	Broken Authentication using SQL Injection	1	1	1
	Autocomplete enabled on input fields Enabled	0	0	0
Site 2	Cross Site Scripting	6	5	8
	SQL Injection	3	5	6
	Directory Listing	12	9	13
	Broken Authentication using SQL Injection	0	0	0
	Autocomplete enabled on input fields Enabled	1	0	2

TABLE I. Experiment Results

5 Conclusions and future works

The primary contribution of this web scanner is to show that it is so simple to get and search software bug or vulnrability in different web sites. We exhibited another arrangement of consequently distinguishing vulnerabilities in certain web applications, for example, Broken Access Control, Insecure Configuration Management, SQL infusion, XSS. The assaults were produced utilizing some data about the objective web application supplied by its test administrators and the test system consequently sent the assaults to the web application to check whether the assaults were effective or not. Our Scanner has some usefulness like Open source Web application, Modular outline, Automation.

Later on, our scanner will incorporate enhancing identifying increasingly web security vulnerabilities. We will accomplish more concentrate on complex-structure, secure shrouded pages, codes building and the reaction dissecting.

6 Acknowledgements

We are grateful to IT Department, BVM ,V.V.Nagar,Anand for their support and for given essential direction concerning project development . We would like to express our deep and sincere gratitude to Mr. Kaushal Bhavsar for providing guidance.

References

[1].Dafydd Stuttard , Marcus Pinto "The Web application Hacker's Handbook Finding an Exploiting Security Flaws " second edition ©2011

[2]. 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,p.- 14-18.

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

[4]. Nuno Antunes and Marco Vieira , "Defending against Web Application Vulnerabilities", University of Coimbra, Portugal, 0018-9162/12/$31.00 © 2012 IEEE, vol.-2,p.- 66-72.

[5].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

[6].Vebjørn Moen, Andr´e N. Klingsheim, Kent Inge Fagerland Simonsen, and Kjell Jørgen Hole "Vulnerabilities In E-governments" ,University of Bergen fmoen,klings,kentis,kjellhg@ii.uib.no

[7]. 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

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

[9].David Shelly, Randy Marchany, Joseph Tront "Closing the Gap: Analyzing the Limitations of Web Application Vulnerability Scanners" Virginia Polytechnic Institute and State University

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

[11]. Stefan Kals, Engin Kirda, Christopher Kruegel, and Nenad " SecuBat: A Web Vulnerability Scanner" Stefan Kals, Engin Kirda, Christopher Kruegel, and Nenad Jovanovic Secure Systems Lab, Technical University of Vienna

[12]. Elizabeth Fong Romain Gaucher Vadim Okun Paul E. Black " Building a Test Suite for Web Application Scanners " National Institute of Standards and Technology Gaithersburg, MD

[13]. Top 10 web security threats part-1 URL:http://www.emateecontent. org/security/top-10-web-security-threats-part-1/

[14]. Top 10 web security threats part-2 URL: http://www.emateecontent. org/security/top-10-web-security-threats-part-2/

[15]. Hai Zhou LING "towards the automation of vulnerability detection in source code" Master of Computer Science, Concordia University, Montréal, Québec, Canada

[16]. Security vulnerabilities in modern web browser architecture MIPRO, 2010 Proceedings of the 33rd International Convention Date of Conference: 24-28 May 2010 Author(s): Silic, Marin;Krolo, Jakov ; Delac, Goran Faculty of Electrical Engineering and Computing, University of Zagreb, Unska 3, 10000, Croatia

[17]. Testing and Comparing Web Vulnerability Scanning Tools for SQL Injection and XSS Attacks Dependable Computing, 2007.PRDC 2007. 13th Pacific Rim International Symposium on Date ofConference: 17-19 Dec. 2007 Author(s): Fonseca, J. CISUC - Polytechnic Inst. of Guardia, Guardia Vieira, M. ; Madeira, H.

[18]. Using web security scanners to detect vulnerabilities in web services Dependable Systems& Networks, 2009. DSN' 09. IEEE/IFIP International Conference on Date of Conference: June 29 2009-July 2 2009 Author(s): Vieira, M.;Antunes, N. ; Madeira, H. Dept. of Inf. Eng., Univ. of Coimbra, Coimbra, Portugal