Design and Implementation of VAPT Tool for Cyber Security Analysis using Response Analysis
Prashant S. Shinde¹, Shrikant B. Ardhapurkar²
PG Scholar¹, Assistant Professor²
Department of Computer Technology
Yeshwantrao Chavan College of Engineering, Nagpur, Maharashtra, India
prashant.ss21@gmail.com¹, shrikant.999@gmail.com²

Abstract:
In last two decades, use of the internet and online resources has been increased. It has also brought security concerns to new levels. As compromising with security issues don't seem to be reasonable, organizations facing very significant challenges in securing their web applications from rising cyber threats. Vulnerabilities are system flaws or weaknesses that may lead to security breach. It is necessary to ascertain the vulnerabilities and install security patches. As, these vulnerabilities can be used by attackers to launch cyber-attacks and have access to confidential information. Therefore finding Vulnerabilities and install security patches to their web application has become prime issues to each internet facing organization. Vulnerability Assessment and Penetration Testing (VAPT) techniques help them for auditing, patching their web application's security. In this paper, we have implemented an automated vulnerability assessment and penetration testing (VAPT) Tool for finding vulnerabilities from the web application on the basis of response coming from the web server. Apart from this, it exploits the XSS vulnerable links. This tool can be used for internal security auditing of web applications.

Keywords: Cross-Site Scripting (XSS), Cyber Security, Sql Injection (SQLi), Vulnerability Assessment and Penetration Testing (VAPT)

I. INTRODUCTION
In today’s era, we are dependent on the internet for various kind of needs. Web applications are used for the variety of tasks such as social networking, e-commerce as well as banking related critical tasks. But these web applications may contain vulnerabilities which may cause serious damage confidentiality of our valuable data.

Organizations facing challenges to secure their data from such threats. VAPT techniques help them in detecting such vulnerabilities and install security patches. As these vulnerabilities can be used by attackers to launch different kinds of attacks. Lack of awareness and improper defensive measures gives rise to malicious activities. So finding out such security loopholes is a major concern for every web facing organization. Early detection and remediation of vulnerabilities is achieved by using VAPT techniques.

In this paper, we have developed a VAPT tool which identifies vulnerabilities from a web application by analyzing the response coming from the web server. Vulnerabilities found in the webpage will be reported which can further be used to exploit XSS vulnerable links and to grab information from the targets.

1. Web Application Vulnerabilities:
The Vulnerabilities are nothing but system flaws, bugs, misconfiguration that make it vulnerable to the attacks. There are types of vulnerabilities that can be found in any web applications namely Cross-site Scripting (XSS), SQL Injection (SQLi), directory traversal, Failure to Restrict URL Access, file inclusion etc.

Cross-site scripting (XSS) is a kind of security vulnerability found in the web applications in which the attacker can inject client-side scripts into web pages which are viewed by other users [19]. XSS attacks are of two types i.e. Persistent XSS and Non-persistence XSS. The Persistent or stored XSS is a kind of attack in which attacker submits malicious code and server saves that code within the database, so a victim is able to retrieve the stored data from the web application.

Whereas Non-persistent XSS attack happens when user input is instantly received in the form of error message, search result, or any other response from the web application. This error message may contain some or all of the input provided by the user as a part of the request which can be permanently stored

SQL injection happens when a user injects an SQL commands via input of a web page in an SQL statement and this injected SQL command make alterations in SQL statement and compromises the security of a web application [1].

Likewise, Directory Traversal is an input manipulation attack vulnerability which uses directory traversal sequences to access or manipulate arbitrary files and resources on the web server [17]. This kind of vulnerability occurs due to insufficient filtering/validation of browser inputs from users.

File inclusion vulnerability permits an attacker to include a file, typically through a script on the web server. This vulnerability happens because of the use of un-validated user-supplied input, this can cause code execution on the web server and client-side, Denial of service (DoS), information theft/ manipulation.
Moreover, Failure to restrict url Access helps attackers to bypass the web-site’s security by accessing files directly rather than following links. This enables an attacker to access information source files directly rather than using the web application [17].

2. Overview of VAPT:

Vulnerability Assessment and Penetration Testing in combination are more useful for application evaluation with detailed threats view in an application for vulnerability mitigation [9]. VAPT tests are done on a periodic basis for assurance about web application's security and its operations. Financial losses, preserving Corporate Image and rationalizing Information Security investments are prevented with the help of VAPT. For identification and prevention of security risks, proactive approach towards VAPT testing is beneficial to stay protected from unauthorized access, data corruption or financial loss.

VAPT methodology is conducted in two major components. In first part existing vulnerabilities are analyzed and discovered and in second part detected set of vulnerabilities are exploited for evaluation of its severity and impact over target system. Vulnerability assessment is a passive approach whereas penetration testing is an active approach where security professionals simulate attacks, to test the target website and its tolerance power against attacks.

Vulnerability assessment is a technique in which systematic and proactive approach for discovering vulnerability is being followed. It is used for discovery of known and unknown bugs in the system.

Penetration testing performs attacks for assessment of security posture of a system or network. It is a proactive approach towards security assessment so the VAPT tester pretends to be an attacker and tries to exploit the identified set of vulnerabilities [9].

Section II of the paper describes the related work regarding Vulnerability Assessment and Penetration Testing methodologies, further, Section III discuss Implementation of proposed VAPT tool. Finally, Section IV deals with the conclusion and future scope for improvement and future enhancement in the Tool.

II. RELATED WORK

In 2009, Adam Kiezun, Philip J. Juo, et al. [6] proposed an automatic technique for creating inputs/attack vectors that expose SQLI and XSS vulnerabilities from applications. For performing exploitation their technique first produces sample inputs then make symbolical tracking of taints through execution, and mutation of inputs. The proposed tool creates attack vectors, and has some false positives. These attack vectors works without making any modification in the code. It is a kind of white box testing tool which requires source code of application. This tool generates a set of concrete inputs then executes the program under test with each input, and dynamically observes data flows.


This tool makes use of black box testing mechanism for analysis of potential vulnerabilities present in the web applications. It is made up of two major components as Spider and Scanner. The spider crawls the website and finds out injections points whereas scanner performs injection test and response analysis. And for verification they used National Vulnerability Database (NVD).

In 2013, Michelle E Ruse and Samik Basu [12] proposed a two-phase technique for detection of XSS vulnerability and prevention of XSS attack which relies on translation of web applications. First phase translates the web application code into a language where recently developed concolic testing tools were available and second phase instrument the application code by including monitors based on I/O dependencies captured from first phase. Exploitation of vulnerabilities is checked by monitors at runtime. Due to this prototype implementation, XSS vulnerabilities are identified and are exploited.

In 2014, Sugandh Shah, B. M. Mhetre [3] proposed an automated VAPT Testing Tool named NetNirikshak 1.0 for Services assessment and Security analysis Posture. It is useful for finding out the vulnerabilities based on the running Services and target system applications. It detects SQL Injection vulnerabilities and reports the Identified vulnerable links on the Target. Addition to this, the tool exploits the identified SQLI vulnerable links and steals confidential data from Target. The Email is sent for the generated report and all the traces of Scan are removed for ensuring the Confidentiality of the VAPT Report. It uses passive approach for detection of service vulnerabilities with the help of National Vulnerability Database (NVD) and active approach for detection of application vulnerabilities by performing Blind SQL injection, Error-Based SQL injection.

In 2014, Geogiana Buja, Dr. Kamarularifin Bin, Abd Jalil, et al. [9] proposed a detection model for detecting and recognizing SQL Injection vulnerability based on the defined and identified criteria and generate a report concerning the vulnerability level of the web application. This model relies on Boyer Moore string matching algorithm in which every string or input files are scanned for the defined attributes of the SQL Injection Pattern of attack.

In 2014, Rocha, T.S., Souto, et al. [13] developed a tool ETSS Detector, which works automatically for web applications analysis to find XSS vulnerabilities. It is generic and modular vulnerability scanner works on the principle of automatically analyzing web applications with the help of information contained on web applications to detection of vulnerabilities. ETSS Detector also identifies and analyses all data entry points of the application and generates the code injection tests. ETSS Detector is constructed on techniques that ensure the proper filling of form fields with valid data permitting the pages to be successfully submitted.

In 2013, Michelle E Ruse and Samik Basu [12] proposed a two-phase technique for detection of XSS vulnerability and prevention of XSS attack which relies on translation of web applications. First phase translates the web application code into a language where recently developed concolic testing tools were available and second phase instrument the application code by including monitors based on I/O dependencies captured from first phase. Exploitation of vulnerabilities is checked by monitors at runtime. Due to this prototype implementation, XSS vulnerabilities are identified and are exploited.

In 2014, Sugandh Shah, B. M. Mhetre [3] proposed an automated VAPT Testing Tool named NetNirikshak 1.0 for Services assessment and Security analysis Posture. It is useful for finding out the vulnerabilities based on the running Services and target system applications. It detects SQL Injection vulnerabilities and reports the Identified vulnerable links on the Target. Addition to this, the tool exploits the identified SQLI vulnerable links and steals confidential data from Target. The Email is sent for the generated report and all the traces of Scan are removed for ensuring the Confidentiality of the VAPT Report. It uses passive approach for detection of service vulnerabilities with the help of National Vulnerability Database (NVD) and active approach for detection of application vulnerabilities by performing Blind SQL injection, Error-Based SQL injection.

In 2014, Geogiana Buja, Dr. Kamarularifin Bin, Abd Jalil, et al. [9] proposed a detection model for detecting and recognizing SQL Injection vulnerability based on the defined and identified criteria and generate a report concerning the vulnerability level of the web application. This model relies on Boyer Moore string matching algorithm in which every string or input files are scanned for the defined attributes of the SQL Injection Pattern of attack.

III. DESIGN AND IMPLEMENTATION OF PROPOSED TOOL

As illustrated in Figure 1. The proposed model contains five major parts such as, Target Discovery, Scanning, Vulnerability Detection, Exploitation and Report Generation.
The working of proposed tool is described in following sequence of steps.

Steps:
1. Login to the website via credentials
2. Fetch the complete list of files from the website
3. Scan each of files to get HTML content
4. For each HTML content, get the GET and POST from page, and create a list with request type and variable names
5. Use each of the GET and POST request with each attack vulnerability tests and check for the vulnerabilities
6. Exploit the vulnerable pages with XSS inputs
7. Generate a report for vulnerabilities present, total scan time, variables and pages discovered.

1. Target Discovery

This module aims to discover the target i.e. to discover all files present in the target. Target discovery is carried out by taking administrator credentials of the website and URL as input from the user. With the help of these parameters, we created ftpRequest. Collect the list of pages/files present using method ListDirectoryDetails(); collects the response and create a list of webpages.

2. Scanning

This module discovers HTML content and to find the GET/POST Request methods and variables present in the webpage. first it will find forms, then search for methods like GET, POST etc. and finally it will find variables present in webpage. It will use string matching to find out input tags and variables. All the variables discovered are stored in a collection.

3. Vulnerability Detection:

In this module, we aim find the vulnerabilities present in the target. We implemented this vulnerability detection module by analyzing the response from webpages. The technique we use in responses analysis is Semantic Matching. It looks for server response that it contains error statement for inputs supplied along with url.

Algorithm: Vulnerability Detection with Response Analysis

Input:
ListofPages [],
ListofGetPostVariables [],
ExtraInputData[

Method:
1. Get page url, variable name, attack vulnerability specific input and create a WebRequest and send it to webserver.
   TargetURL=Url.Append("ExtraInputData");
2. Collect and analyse the WebResponse,
   If(WebResponse.contains(ErrorStatement))
   Then
     “Not Vulnerable”
   Else
     “Vulnerability Exists”
3. Repeat

This algorithm tries to inject some extra information to the URL. If the server compiles the extra data that means that vulnerability exists. In SQLi Vulnerability testing we added (" " "") to the URL and observed the response. In case of XSS we added scripts to URL like <script>alert();</script> to the URL and observed the response from the server. If the script is compiled successfully by server then the page is XSS vulnerable.
Figure 2 shows Response Analysis mechanism in vulnerability detection. In this list of webpages, variables, and extra data are taken as an input WebRequest is created with url, Extra Input Data. This request is sent to server and the WebResponse is analysed, using the algorithm. If the response do not contains any error statement then vulnerability exists in that page otherwise it is not vulnerable.

Table 1 shows the sample dataset used in vulnerability detection. This extra data is appended in url to create web request for SQLi and XSS vulnerability detection.

<table>
<thead>
<tr>
<th>Sr. No</th>
<th>Pattern</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>= 1' or '1' = '1</td>
<td>SQLi</td>
</tr>
<tr>
<td>2</td>
<td>:and 1=1 and 1=2</td>
<td>SQLi</td>
</tr>
<tr>
<td>3</td>
<td>&quot;!'-&quot;&lt;XSS&gt;=&amp;{()</td>
<td>XSS</td>
</tr>
<tr>
<td>4</td>
<td>&lt;script src=<a href="http://xss.rocks/xss.js">http://xss.rocks/xss.js</a>&lt;/script&gt;</td>
<td>XSS</td>
</tr>
</tbody>
</table>

4. Exploitation:
In this part, exploitation of the identified set of vulnerabilities is done in the same manner as an attacker would do. The goal of launching exploits is to gain full access to the target system. We implemented this module for XSS Exploitation which exploits XSS vulnerabilities.

Target URL= URL.Append(Attack Vector); For e.g.
welcome.php?name="<script>alert(document.cookie)</script>"

We have a list of attack vectors, vulnerable links, we will use these attack vectors to exploit each vulnerable link identified in earlier phase.

5. Report Generation
This is the last phase of proposed model. It collects the results and artifacts from other modules. Automatically drafts a well formed VAPT report which contains list of vulnerable links for each attack test, total time taken to perform all these operations, total pages scanned.

IV. CONCLUSION
In this paper, the VAPT tool is proposed for both to detection and exploitation vulnerabilities present in web applications. Vulnerability assessment is done by analyzing server response and described with CWE numbers. Exploitation of identified XSS vulnerable links is done with a set XSS attack vectors. Also, we have reporting platform which contains the summary of results. The proposed tool is able to identify SQL Injection and Cross-site Scripting (XSS) attack vulnerabilities also Exploitation module works effectively for XSS vulnerabilities. The future scope of this work may include increasing types of vulnerability checks and provide possible remediation for vulnerabilities identified.

V. REFERENCES


