TASK 3

1. Find 3 websites vulnerable to OPEN REDIRECT/URL REDIRECTION

2. Find 3 websites vulnerable to XSS/Cross site Scripting.

All types of XSS vulnerability are accepted.

1. Self XSS/Reflected XSS

2. Persistent XSS/Stored XSS

3. DOM based XSS

payload list:

https://github.com/ihebski/XSS-Payloads

3. Find 3 websites vulnerable to HTML INJECTION vulnerability.

Open redirection/ URL redirection vulnerability:

Open redirection, also known as URL redirection vulnerability, is a security flaw that occurs when a web application redirects users to an external URL specified by an attacker. This vulnerability can be exploited by manipulating the application's redirect parameters or input fields.

The impact of an open redirection vulnerability can be significant. Attackers can trick users into visiting malicious websites or phishing pages, leading to various consequences such as stolen sensitive information, malware installation, or unauthorized access to user accounts. Open redirection vulnerabilities can also be used in combination with other attacks, making them even more dangerous.

Impacts of Open Redirection Vulnerability:

Phishing attacks

Malware and virus infections

Identity theft

Reputation damage

Financial losses

Remedial Actions for Open Redirection Vulnerability:

Input validation and sanitization

Whitelisting

Avoid using user-supplied data for redirection.

Implement secure session management.

Regular security assessments

Educate users.

Payloads used:

<https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/Open%20Redirect>

1.telekom.de

<https://www.telekom.de/start#suche?%3cscript%3ewindow.location%20%3d%20%22http%3aevil.com%22%3c%2fscript%3e>

A picture containing text, screenshot, brand, multimedia

Description automatically generated

Payload used:

<script>window.location = "http:evil.com"</script>

This redirects us to evil.com.

A screenshot of a computer

Description automatically generated with medium confidence

2.

<https://account.flexihub.com/site/verify?returnUrl=///evil.com>

A screenshot of a computer

Description automatically generated with medium confidence

Here we change the return url value to evil.com and it redirects to evil.com

A screenshot of a computer screen

Description automatically generated with low confidence

XSS/Cross site Scripting:

Cross-Site Scripting (XSS) is a type of web security vulnerability that allows attackers to inject malicious scripts into web pages viewed by other users. This vulnerability arises when a website or web application does not properly validate or sanitize user input, allowing the attacker to insert arbitrary code that is executed by the victim's browser.

The impact of XSS attacks

Data theft

Session hijacking

Defacement and content manipulation

Malware distribution

Remedial actions

Input validation and sanitization

Output encoding

Content Security Policy (CSP)

Web Application Firewall (WAF)

Regular security updates and patches

Security testing and code reviews

1.

<https://library.usask.ca/srsd/gaycanada/search.php>

A screenshot of a computer

Description automatically generated with medium confidence

Enter the following js command in the Search box:

<script>alert(123);</script>

A picture containing text, screenshot

Description automatically generated

That will generate an alert box resulting in the prescene of XSS vulnerability.

2.

<https://www.englishforum.ch/search2.php>

A screenshot of a computer

Description automatically generated with medium confidence

Enter the following js command in the Search box:

<script>alert(123);</script>

A picture containing text, screenshot, software, multimedia

Description automatically generated

That will generate an alert box resulting in the prescene of XSS vulnerability.

3.

<https://www.hfsjg.ch/wordpress/ResearchProjects/Gornergrat/search.php>

A screenshot of a computer

Description automatically generated with low confidence

Enter the following js command in the Search box:

“> <script>alert(123);</script>

A screenshot of a computer

Description automatically generated with medium confidence

That will generate an alert box resulting in the prescene of XSS vulnerability.

HTML INJECTION:

HTML injection, also known as cross-site scripting (XSS), is a web security vulnerability that allows attackers to inject malicious code into web pages viewed by other users. This vulnerability occurs when a web application does not properly validate or sanitize user input before including it in the output HTML.

Impacts:

Unauthorized access

Phishing attacks

Defacement

Malware distribution

Remedial actions:

Input validation and sanitization

Output encoding

Content Security Policy (CSP)

Regular security audits

Educate developers

1.

<https://www.stevemadden.com/>

A picture containing graphics, font, graphic design, colorfulness

Description automatically generated

We have a search bar, If we give html commands in the search bar the HTML commands will get executed and will be displayed on the webpage.

Let us use the command:

<h1><b>Hello! World</b></h1>

A picture containing text, font, screenshot

Description automatically generated

This indicates the presence of HTML injection.

2.

<https://www.ucgraben.ac.cd/views/page-search.php>

A screenshot of a computer

Description automatically generated with medium confidence

We have a search bar, If we give html commands in the search bar the HTML commands will get executed and will be displayed on the webpage.

Let us use the command:

<h1><b>123</b></h1>

A screenshot of a computer

Description automatically generated with medium confidence

This indicates the presence of HTML injection.

3.

<https://www.debet.az/en/search.php>

A screenshot of a web page

Description automatically generated with medium confidence

We have a search bar, If we give html commands in the search bar the HTML commands will get executed and will be displayed on the webpage.

Let us use the command:

<h1><b>123</b></h1>

A screenshot of a computer

Description automatically generated with medium confidence

This indicates the presence of HTML injection.

Conclusion:

In conclusion, the vulnerabilities caused by HTML injection, SQL injection, and XSS (Cross-Site Scripting) represent serious risks to online applications and can have negative effects if not properly fixed.

When an attacker can enter harmful code into the HTML content of a website, HTML injection happens. This can result in several security problems, including the presentation of fake information, user redirection to malicious websites, or even the theft of important user data. Developers should use safe coding techniques like input validation and output encoding to avoid the insertion of malicious HTML code to mitigate this issue.

Using malicious SQL injection, attackers can take advantage of holes in a web application's database layer. This may lead to data alteration, data deletion, or even the execution of arbitrary instructions. It may also result in unauthorised access to the database. SQL injection threats must be avoided by the proper usage of prepared statements with parameters, input validation, and least privilege access constraints.

When a hacker can insert malicious scripts into an online application, which the victim's browser subsequently executes, XSS vulnerabilities take place. This enables the attacker to deface the website, hijack user sessions, or steal critical information. To mitigate XSS vulnerabilities, it is essential to implement correct input validation and output encoding, make use of Content Security Policy (CSP), and use safe coding methods.