XSS Injection Attack Guide

By Vansh Khanna

Step 1: Set up the Environment

First, we need to set up the environment for the pentest. We will use the Damn Vulnerable Web Application (DVWA), which is an intentionally vulnerable web application designed for practicing and testing various web vulnerabilities.

To install DVWA, we need to run the following commands:

```bash
bash                                                                        Copy code

sudo apt-get update
sudo apt-get install -y apache2 mysql-server php libapache2-mod-php php-mysql
sudo git clone https://github.com/ethicalhack3r/DVWA.git /var/www/html/dvwa
sudo chown -R www-data:www-data /var/www/html/dvwa
sudo chmod -R 755 /var/www/html/dvwa
sudo mysql_secure_installation
```

Step 2: Configure DVWA

Once DVWA is installed, we need to configure it by accessing it through the browser. To do this, we need to open the browser and enter the following URL: `http://localhost/dvwa/setup.php`.

The setup page will prompt us to create a new database for DVWA. we need to enter the database details and click on the `Create/Reset Database` button.

After the database is created, we need to log in to DVWA using the default credentials:

- Username: admin
- Password: password

Step 3: Set up Burp Suite

Burp Suite is a popular web application testing tool that helps us in testing web applications for vulnerabilities. We can download Burp Suite from its official website and install it on our Kali Linux machine.

After installing Burp Suite, one needs to configure our browser to use Burp Suite as a proxy. To do this, we need to open the browser settings and set the proxy to `localhost` and port `8080`.

Step 4: Identify the Vulnerable Input Field

Now, we need to identify the vulnerable input field in DVWA. In this case, we will use the `DVWA/vulnerabilities/xss_r` page, which has a vulnerable input field that is susceptible to XSS injection attacks.

Step 5: Perform XSS Injection Attack

We will now use Burp Suite to perform the XSS injection attack. To do this, we need to follow these steps:

1. Open Burp Suite and click on the `Proxy` tab.
2. Start the proxy by clicking on the `Intercept is On` button.
3. Open the browser and navigate to the `DVWA/vulnerabilities/xss_r` page.
4. Enter any text in the `Reflected XSS` input field and click on the `Submit` button.
5. Burp Suite will intercept the request and display it in the `Proxy` tab.
6. Right-click on the request and select `Send to Repeater`.
7. In the `Repeater` tab, we will see the intercepted request, and we can modify it to perform the XSS injection attack.
8. Modify the payload by entering the following text in the `Reflected XSS` input field:

```
php                                                                Copy code

<script>alert('XSS Injection Attack');</script>
```

9. Click on the `Go` button to send the modified request to the server.
10. The server will process the request and execute the injected JavaScript code, displaying an alert box with the message "XSS Injection Attack."

Step 6: Validate the Results

Finally, we needs to validate the results of the XSS injection attack. We can do this by checking if the injected JavaScript code is executed on the server-side and displayed an alert box with the message "XSS Injection Attack."

In this case, the validation is successful,and we have successfully demonstrated how an XSS injection attack can be performed using Kali Linux and Burp Suite.

Step 7: Mitigation

To mitigate XSS injection attacks, we need to sanitize and validate all user input before using it in the application. We can also use a Content Security Policy (CSP) to prevent the execution of unauthorized JavaScript code.

In addition, developers can use frameworks and libraries that provide built-in protections against XSS attacks, such as escaping special characters and encoding user input.

Conclusion

In this tutorial, we have demonstrated how to perform an XSS injection attack using Kali Linux and Burp Suite. We have also discussed the mitigation techniques that can be used to prevent such attacks.

It is essential for developers and security professionals to understand the various web vulnerabilities and how they can be exploited. By conducting regular penetration testing and vulnerability assessments, one can identify and remediate potential security risks in their web applications.

Additional Resources

1. OWASP - Cross-Site Scripting (XSS): **https://owasp.org/www-community/attacks/xss/**
2. XSS Cheat Sheet: **https://portswigger.net/web-security/cross-site-scripting/cheat-sheet**
3. Pentesting with Burp Suite: **https://portswigger.net/burp/documentation/desktop/testing**
4. OWASP Testing Guide - Testing for Cross-Site Scripting (XSS): **https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/03-Input_Validation_Testing/01-Testing_for_Reflected_Cross_site_scripting/**