School of Computing and Information Systems

CS440: Foundation of Cybersecurity

Lab 9: Web Security Lab

**Part 1: Prerequisites**

1. **Mutillidae Setup - [** [**Video Walkthrough**](https://www.youtube.com/watch?v=mtD0hw4ptik) **]**

There is an error in a configuration file in Metasploitable.

1. In Metasploitable, navigate to Mutillidae's folder using the command

cd /var/www/mutillidae/

1. Edit **config.inc** (you will be asked for your password) using the command

sudo vim config.inc

1. Replace metasploit with owasp10 , save and exit.
2. Restart the **Apache** webserver using the command

sudo /etc/init.d/apache2 reload

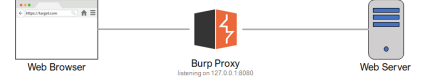
1. In Kali Linux, open Firefox and go to the webpage

\_<\_youripaddress>\_\_/mutillidae/

1. On the navigation bar of Mutillidae, find the button for **Reset DB**, **click it**, then select **ok**.
2. You can now Register and Login into Mutillidae
3. **Burp Suite Setup - [** [**Video Walkthrough**](https://www.youtube.com/watch?v=Bs5vwRlgG-0) **] (For Week 12 Lab)**

Burp Suite is a set of tools used for penetration testing of web applications. We are using Burp Suite its proxy, which allows us to intercept all requests and responses between the browser and the target application.

**Note:** Only setup Burp Suite if the section requires the use of Burp Suite.



In Firefox, Select **Preferences** and search for **Network Settings**

1. To **enable** Burp Suite:
2. Select the Radio Button **Manual proxy configuration**
3. Under **HTTP Proxy**, enter 127.0.0.1 and Under **Port** on the same line, enter 8080
4. To **disable** Burp Suite:
5. Select the Radio Button **No proxy**

3. Working with Burp Suite (**Proxy** tab) :

1. Toggle **Intercept on/off** to allow/disallow Burp Suite to intercept requests
2. When a request is intercepted, you can make changes its Body and Parameters.
3. Press **Drop** to terminate the request and **Forward** to proceed with the request

**Part 2: Session Hijacking**

Session hijacking is an attack where a user's **session** is taken over by an attacker.

**More info:** https://owasp.org/www-community/attacks/Session\_hijacking\_attack

1. **Installing Cookie Editor - [** [**Video Walkthrough**](https://www.youtube.com/watch?v=bIPQQ6xkdEs) **]**

We will use Cookie Editor to modifiy Mutillidae's Cookies.

1. Search for **Cookie Editor Extension** on Firefox
2. Click **+ Add to Firefox **, Firefox will ask if you want to add the extension,

select **Add**

1. You should see a small **Cookie Icon** appear in the top right , if your window is small, you can access the extension by clicking the **Arrow Icon** at the top right.
2. **Gaining Administrator Access - [** [**Video Walkthrough**](https://www.youtube.com/watch?v=bIPQQ6xkdEs&t=6s) **]**

We will create a standard user account, login with it and then modify its cookie to that of the Administrator's.

1. In Mutillidae, register a new user and login with the account. (or just login if you have already created an account)
2. After logging in, click the Cookie Icon on the top right of the screen
3. Click into the row with **uid**, modify the value to 1, then click the **Save Icon **
4. **Refresh** the page, you should see you are logged in as the Administrator

**Part 3: SQL Injection**

An SQL Injection involves the insertion of an SQL query via an input field in the client, which is sent to the webserver. The input modifies the existing SQL query made to the database, allowing an attacker to perform actions or view data that they are normally not able to do.

**More info:** https://owasp.org/www-community/attacks/SQL\_Injection

1. **SQL Injecting Mutillidae (Easy) - [** [**Video Walkthrough**](https://www.youtube.com/watch?v=sOmTJeid2Hg) **]**

Mutillidae takes in the **Name** ($username) and **Password** ($password) and uses it to construct an SQL statement.

The SQL statement is used to query the database to see if such a user exists.

**A look at the statement:**

SELECT \* FROM accounts

WHERE username='$username'

AND password='$password'

In the input box for **Name**, enter admin' OR 1=1# , leave **Password** blank

**A look at the new statement:**

SELECT \* FROM accounts

WHERE username='admin'

OR 1=1#

AND password=''

Notice that this will always return a value even if the username admin does not exist.

You should be logged in as the Administrator

1. **SQL Injecting Mutillidae (Medium) - [** [**Video Walkthrough**](https://www.youtube.com/watch?v=sOmTJeid2Hg&t=30s) **]**

**Note:** Burp Suite is used here!

In this exercise, there is **Client-Side Validation**. In order to bypass the client-side validation, we will send a legitimate request, use Burp Suite to intercept the request. After which, we can modify the legitimate values with our payload.

1. Login as per normal, Burp Suite should intercept the Login request
2. Once Burp Suite intercepts the request, click onto **Params **
3. Modify the **Username** and **Password** with the same values shown in **(1)** Proceed with the request by pressing **Forward**
4. Press Forward again to allow Metasploitable's **response** to return
5. You should be logged in as the Administrator
6. **Automated SQL Injection with sqlmap - [** [**Video Walkthrough**](https://www.youtube.com/watch?v=sOmTJeid2Hg&t=71s) **]**

**sqlmap** is an open-source penetration testing tool that automates the process of detecting and exploiting SQL injection flaws and taking over of database servers.

**Note:** The command here spans more than one line.

If you plan to copy this into the terminal, copy 1 line, paste, then copy the other.

1. In the terminal, enter the command (might be a bit long)

sqlmap -u "http://\_\_youripaddress\_\_/mutillidae/index.php?page=user info.php&username=admin&password=pass&user-info-php-submit

button=View+Account+Details" --dbs --dump

1. Let **sqlmap** run for a while, if **sqlmap** prompts you, enter their recommended value (usually capitalized)
2. **sqlmap** will be able to see the database contents in Mutillidae

**Part 4: Security Misconfiguration**

Failing to implement the necessary security controls or doing so with errors.

1. **Accessing the Metasplotiable Database from Kali Linux - [** [**Video Walkthrough**](https://www.youtube.com/watch?v=cR3Qf_KPWgo) **]**

The mySQL database in Metasploitable is open to the public without any form of authentication.

1. In the terminal, enter the command

mysql -u root -h <youripaddress>

1. You will be logged in as the database administrator
2. You can navigate and view the database contents using these commands and your mysql knowledge:

|  |  |
| --- | --- |
| **Command** | **Function** |
| show databases; | Shows all the Databases |
| use \_\_database\_\_; | "Enters" the Database, you can then see the tables inside |
| show tables; | Shows the Tables in a Database |

1. **Finding the DVWA Administrator Username and Password - [** [**Video Walkthrough**](https://www.youtube.com/watch?v=cR3Qf_KPWgo&t=22s) **]**

This account will be used for the XSS workshop!

Hint: The password is hashed using MD5 (but can MD5 be reversed? ) Find the **DVWA** administrator **username** and **password**!

Answer

|  |  |
| --- | --- |
| Username: | admin |
| Password: | password |

**Part 5: Cross Site Scripting (XSS)**

Cross Site Scripting as an attack whereby malicious scripts are injected into webpages. These scripts are executed when the webpage is viewed by other users. XSS attacks exploit the trust the **user has with the webserver**.

**Reflected XSS**

Reflected XSS is also known as Non - Persistent XSS

**More info:** https://owasp.org/www-community/attacks/xss/#Stored\_and\_Reflected\_XSS\_Attacks

1. **Reflected XSS on DVWA (Easy) - [** [**Video Walkthrough**](https://www.youtube.com/watch?v=wceO-omufSk) **]**

This webpage asks you for your name and then shows you Hello with whatever you entered.

1. In the input box, enter

<script>alert(document.cookie)</script>

1. After submitting, an **alert** will prompt and shows you all the cookie information
2. This means the script you just entered has been successfully executed on your browser.
3. **Reflected XSS on DVWA (Medium) - [** [**Video Walkthrough**](https://www.youtube.com/watch?v=wceO-omufSk&t=25s) **]**

Do this yourself!

**Hint:** The webpage filters input and removes everything within and including the **<>** tags. But! its filtering is not perfect.

In the input box, enter

<ScRiPt>alert(document.cookie)</ScRIPT>

**Stored XSS**

Stored XSS is also known as Persistent XSS

**More info:** https://owasp.org/www-community/attacks/xss/#Stored\_and\_Reflected\_XSS\_Att acks

1. **Stored XSS on DVWA (Easy) - [** [**Video Walkthrough**](https://www.youtube.com/watch?v=wceO-omufSk&t=64s) **]**

This webpage functions like a blog or forum, it asks you for a message, saves it in its database and shows it to you every time you revisit the page.

**Note:** If you want to clear the messages, you need to clear the database. You can do that from Kali Linux by following the steps shown in Security Misconfiguration and using the sql command delete from guestbook

1. Enter a random name, in the Message Textbox, enter

<script>alert(document.cookie)</script>

1. After submitting, there will be an **alert** prompt which shows you all the cookie information
2. When you revisit the page, or refresh it (don't refresh immediately after submitting, it will ask if you want to resubmit the form which can be quite troublesome)
3. **Stored XSS on DVWA (Medium) - [** [**Video Walkthrough**](https://www.youtube.com/watch?v=wceO-omufSk&t=123s) **]**

Do this yourself!

**Hint:** The webpage filters input and removes everything within and including the **<>** tags. Each textbox has different levels of filtering and input validation.

Inspect the input box for Name

Use its HTML editor to add a value into the input box

Add this within the input box tags value="<script>alert(document.cookie)</script>"

Make sure the input appears inside the input box

Submit!