

Lab 05 Requirements

- Internet connectivity & VMware Workstation version 15.5.7 or above
- Ensure that you can browse from Kali's Firefox to *FOLusername-uws/mutillidae*

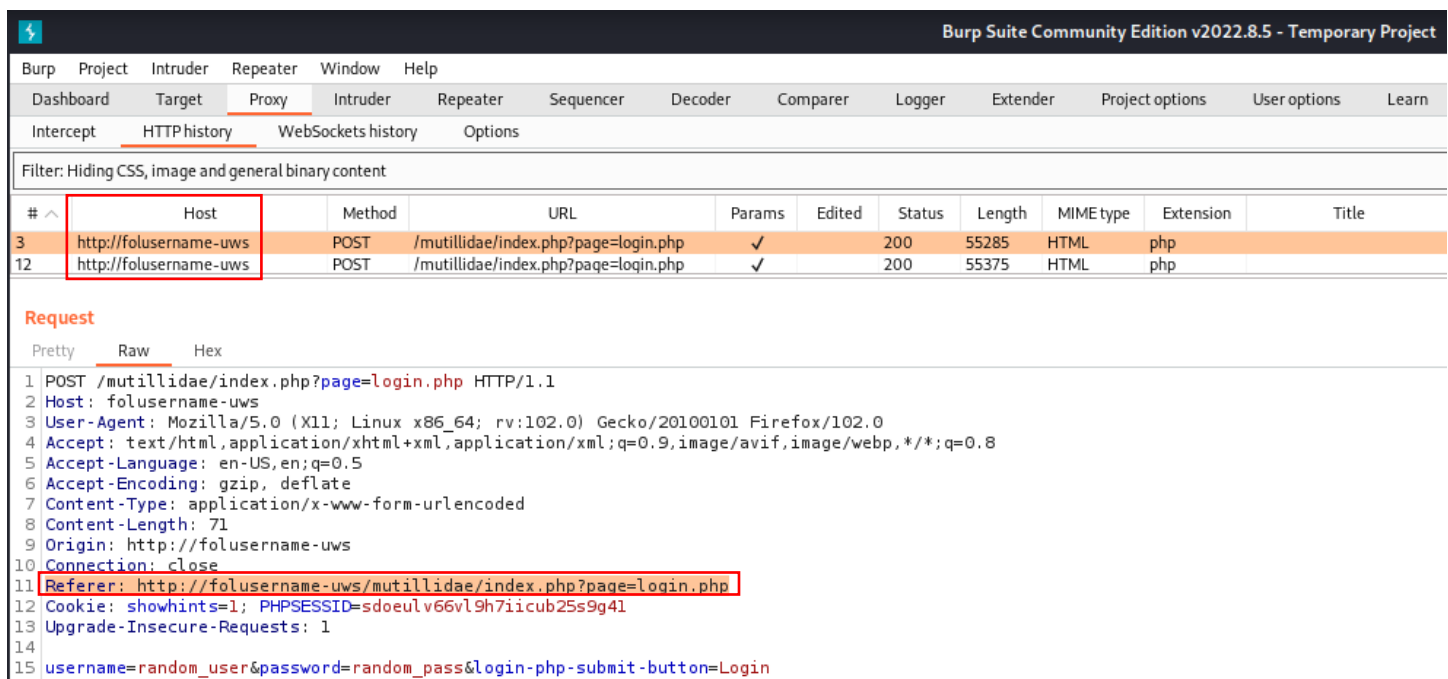
Part 01: Capture HTTP Requests and Responses

Screenshot Details: (you only get credit for the screenshot if you meet these requirements)

- You need to highlight ONLY the requested information in each Request **OR** Response
- All screenshots need to be taken from the History section, showing the Raw data
- All screenshots must include your hostname in the Host column
- The following example screenshot is provided as a guide

Example Screenshot

- The example below shows the raw **Request** data, viewed via the HTTP History Tab
- The host column shows the page was accessed via FOLusername
- The value for Referrer is highlighted



Burp Suite Community Edition v2022.8.5 - Temporary Project

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn

Intercept HTTP history WebSockets history Options

Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title
3	http://folusername-uws	POST	/mutillidae/index.php?page=login.php	✓		200	55285	HTML	php	
12	http://folusername-uws	POST	/mutillidae/index.php?page=login.php	✓		200	55375	HTML	php	

Request

Pretty Raw Hex

```

1 POST /mutillidae/index.php?page=login.php HTTP/1.1
2 Host: folusername-uws
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 71
9 Origin: http://folusername-uws
10 Connection: close
11 Referer: http://folusername-uws/mutillidae/index.php?page=login.php
12 Cookie: showhints=1; PHPSESSID=sdoeu1v66v19h7iicub25s9g41
13 Upgrade-Insecure-Requests: 1
14
15 username=random_user&password=random_pass&login-php-submit-button=Login
  
```

Based on the example screenshot shown above, below are the slides you need to provide:

Slide 01:

- HTTP Request showing the User-Agent
- Highlight the User-Agent

Slide 02:

- HTTP Response showing a status code of 404
- Highlight the status code

Take Over a Session

Create two new users in Mutillidae: your FOLusername-01 and your FOLusername-02

Login as your FOLusername-01 and gather the **Set-Cookie** information from Burp Suite

Slide 03:

- Highlight the **Set-Cookie** field
 - ✓ You will use this cookie information in the next step

Slide 04 & 05:

- Login as your FOLusername-02
- Use Burp Suite to edit the cookie information, while you are navigating to the Add to Your Blog Page, so that you end up at the Add to Your Blog page as your FOLusername-01
 - If you don't end up logged in as FOLusername-01 it didn't work
- Take a screenshot of original request for screenshot 04 (found in HTTP History)
- Take a screenshot of the edited request for screenshot 05 (found in HTTP History)

Based on the UIDs you have seen and manipulated so far intercept another request to establish a session as the Mutillidae Admin

Slide 06:

- Take a screenshot of the edited request for Admin Login (found in HTTP History)

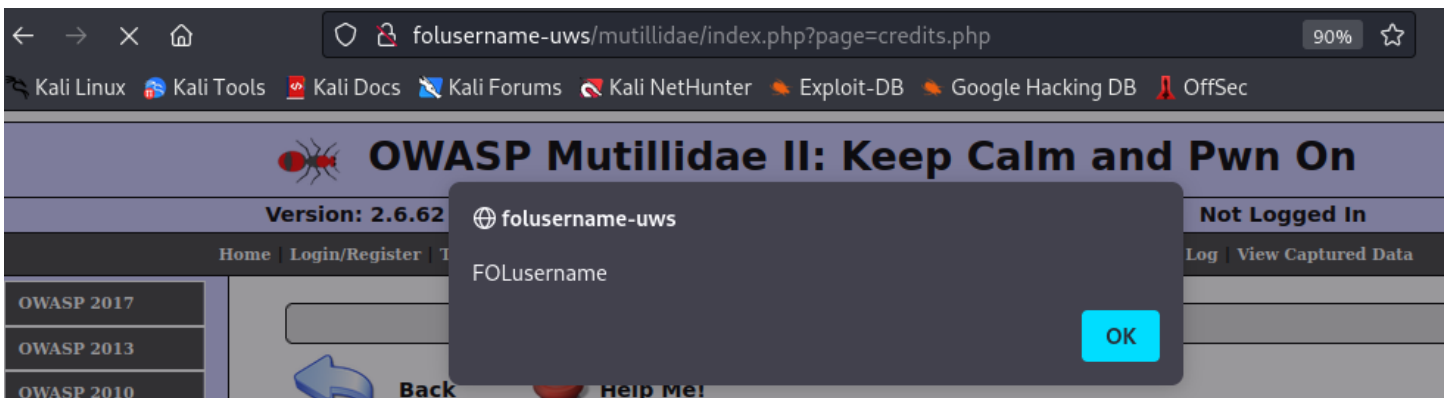
Part 02: Change the User-Agent HTTP Header in Firefox

Reset the DB in Mutillidae and load the home page. Turn your Burp Suite intercept on and capture the packets as you navigate to another page... (The example is using folusername-uws/mutillidae/index.php?page=credits.php)

Modify the User-Agent Header in the captured HTTP request packet to reflect your FOLusername as the page loads by typing in the following JavaScript into the User-Agent field in the header:

```
<script type="text/javascript">alert("FOLusername");</script>
```

Once you have modified the packet, click **Forward**

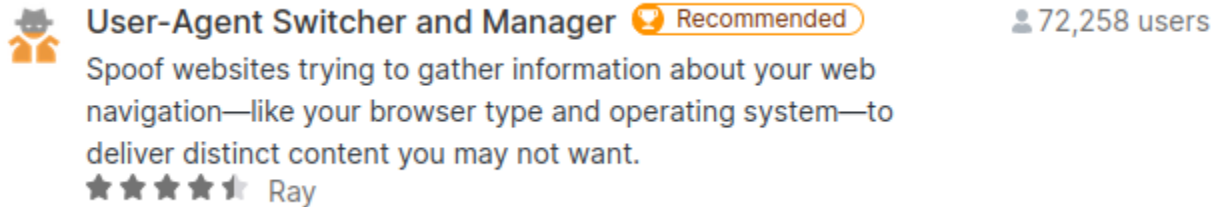


If you have done everything correctly, you will receive an alert box once the page loads as shown

You will also notice that Mutillidae uses the user-agent information to populate the content in the footer on every page. Could this pose a problem?

Download a Firefox add-on in Kali that will automatically change the User-Agent field in the HTTP header

I will be using the **User-Agent Switcher and Manager** by Ray for my examples, but you can feel free to use another add-on as long as it accomplishes the same task

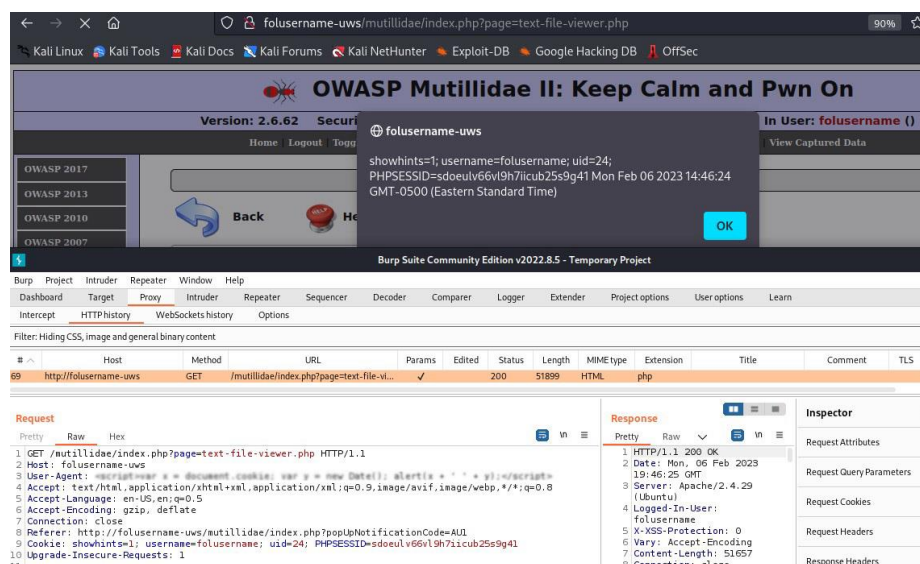


This allows you to customize your User-Agent header field. Start by customizing it to some JavaScript code that you can inject. Intercept the packets in Burp Suite to ensure that the add-on is working properly

Create a new user in Mutillidae with the username of your FOLusername. Login as that user and edit the add-on to use a custom script as opposed to the default user agent information

Set the User-Agent field to use JavaScript to display today's date and time and the cookie information for the session

Capture the **Request** packet using the **POST** method in Burp Suite



Slide 07:

- Take a screenshot showing the Original Request packet
- Your FOLusername user logged in
- Alert box displaying the time, date, and cookie information

Part 03: Burp Suite Intruder

On your Kali VM, ensure that you have turned intercept off in Burp suite. Navigate to the Mutillidae web application in FireFox and create a new account in Mutillidae with the username of **folusername** and set a password of **foobar**

Make sure that your Firefox browser is set to use BurpSuite as a proxy, return to the login page on Mutillidae and set intercept to **on**

Enter your FOLusername into the username field, and a random password on the Mutillidae login screen to capture the traffic

Capture the request and then right click on the raw packet and select **Send to Intruder**

Click on the **Intruder** tab in Burp Suite. You will see the captured packet with some of the fields automatically highlighted as Payload Positions. Anything between the two meta characters is part of the variable field.

Configure the **Positions** sub-tab so that it only has one variable field as shown in the image below. You will need to *clear* all the fields except for the password value. This is the only field that will be using words from the payloads tab.

```
1 POST /mutillidae/index.php?page=login.php HTTP/1.1
2 Host: folusername-uws
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 71
9 Origin: http://folusername-uws
10 Connection: close
11 Referer: http://folusername-uws/mutillidae/index.php?page=login.php
12 Cookie: showhints=1; PHPSESSID=sdoeulv66vl9h7iicub25s9g41
13 Upgrade-Insecure-Requests: 1
14
15 username=folusername&password=$random_pass$&login-php-submit-button=Login
```

To configure the **Payloads** sub-tab, navigate to **random.org/passwords** in Firefox and use the Random Password Generator to create a password list containing 15 passwords. Copy and paste the results into a text file and append it to include the word **foobar**

Remove the spaces in front of the passwords, copy the list, and **Paste** it into *Payload Options*

Now that the *Positions* and *Payloads* tabs are configured, click on **Start attack**
Let Intruder run and look for a 302 redirect response from the server

If it worked properly, you should see that Mutillidae now has you logged in as the folusername user (You might need to refresh the page in Firefox)

⚡
2. Intruder attack of http://folusername-uws - Temporary attack - Not saved to project file

Attack Save Columns							
Results	Positions	Payloads	Resource Pool	Options			
Filter: Showing all items							
Request ^	Payload	Status	Error	Timeout	Length	Comment	
13	xp3hA7Mf	200	<input type="checkbox"/>	<input type="checkbox"/>	55285		
14	AXw4Mp4U	200	<input type="checkbox"/>	<input type="checkbox"/>	55285		
15	y4K6ewpJ	200	<input type="checkbox"/>	<input type="checkbox"/>	55285		
16	foobar	302	<input type="checkbox"/>	<input type="checkbox"/>	380		

Request
Response

Pretty
Raw
Hex

```

1 POST /mutillidae/index.php?page=login.php HTTP/1.1
2 Host: folusername-uws
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 66
9 Origin: http://folusername-uws
10 Connection: close
11 Referer: http://folusername-uws/mutillidae/index.php?page=login.php
12 Cookie: showhints=1; PHPSESSID=sdoeulv66vl9h7iicub25s9g4l
13 Upgrade-Insecure-Requests: 1
14
15 username=folusername&password=foobar&login-php-submit-button=Login

```

Slide 08:

- Show the raw tab used in the request that received the 302 status code

*** Take a snapshot of all the VMs named **After Lab 05** ***