Seal walkthrough

Index

List of pictures

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:

```
┌──(k14d1u5㉿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=node01qzk49nqjsvvy1iifjtp61bvf32.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=node01aeuyqcdaoqrz1c9cc9syw5vot0.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=node0aw4er9unbz3gy44aw2j7n2j91.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

```
|   HTTPOptions:
|     HTTP/1.1 200 OK
|     Date: Wed, 14 May 2025 11:25:26 GMT
|     Set-Cookie: JSESSIONID=node0aw4er9unbz3gy44aw2j7n2j91.node0; 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=0x80
|     Content-Type: text/html;charset=iso-8859-1
|     Content-Length: 71
|     Connection: close
|     <h1>Bad Message 400</h1><pre>reason: Illegal character OTEXT=0x80</pre>
|   RTSPRequest:
|     HTTP/1.1 505 Unknown Version
|     Content-Type: text/html;charset=iso-8859-1
|     Content-Length: 58
|     Connection: close
|     <h1>Bad Message 505</h1><pre>reason: Unknown Version</pre>
|   Socks4:
|     HTTP/1.1 400 Illegal character CNTL=0x4
|     Content-Type: text/html;charset=iso-8859-1
|     Content-Length: 69
|     Connection: close
|     <h1>Bad Message 400</h1><pre>reason: Illegal character CNTL=0x4</pre>
|   Socks5:
|     HTTP/1.1 400 Illegal character CNTL=0x5
|     Content-Type: text/html;charset=iso-8859-1
|     Content-Length: 69
|     Connection: close
|_    <h1>Bad Message 400</h1><pre>reason: Illegal character CNTL=0x5</pre>
|_http-title: Site doesn't have a title (text/html;charset=utf-8).
1 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:V=7.94SVN%I=7%D=5/14%Time=6824DA75%P=x86_64-pc-linux-gnu%r
SF:(GetRequest,F5,"HTTP/1\.1\x20401\x20Unauthorized\r\nDate:\x20Wed,\x2014
SF:\x20May\x202025\x2011:25:26\x20GMT\r\nSet-Cookie:\x20JSESSIONID=node01a
SF:euyqcdaoqrz1c9cc9syw5vot0\.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:\x20JSESSIONID=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:\x2058\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\
```

Figure 2 - nMap scan results 2

```
|_http-title: Site doesn't have a title (text/html;charset=utf-8).
1 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:V=7.94SVN%I=7%D=5/14%Time=68247DA7%P=x86_64-pc-linux-gnu%r
SF:(GetRequest,F5,"HTTP/1\.1\x20401\x20Unauthorized\r\nDate:\x20Wed,\x2014
SF:\x20May\x202025\x2011:25:26\x20GMT\r\nSet-Cookie:\x20JSESSIONID=node01a
SF:euyqcdaoqrz1c9cc9syw5vot0\.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:\x20JSESSIONID=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:\x2058\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:20JSESSIONID=node01qzk49nqjsvvy1iifjtp61bvf32\.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-0\*5\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-0\*5</pre>")%r(Socks4,C3,"HTTP/1\.1\x
SF:20400\x20Illegal\x20character\x20CNTL-0\*4\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-0\*4</pre>")%r(RPCCheck,C7,"HTTP/1\.1\x20400\x20Illegal\x20char
SF:acter\x20OTEXT-0\*80\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\x20OTEXT-0\*80</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 Linuz 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/:

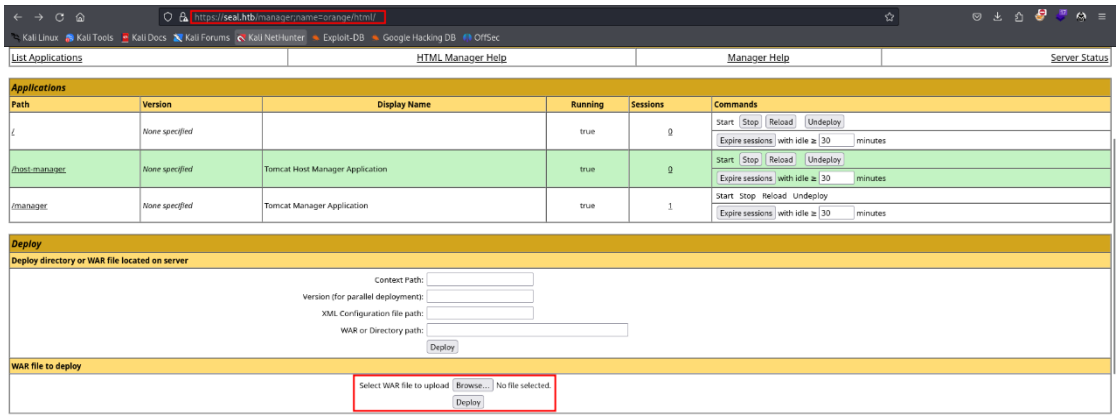

Figure 5 - Tomcat manager GUI

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

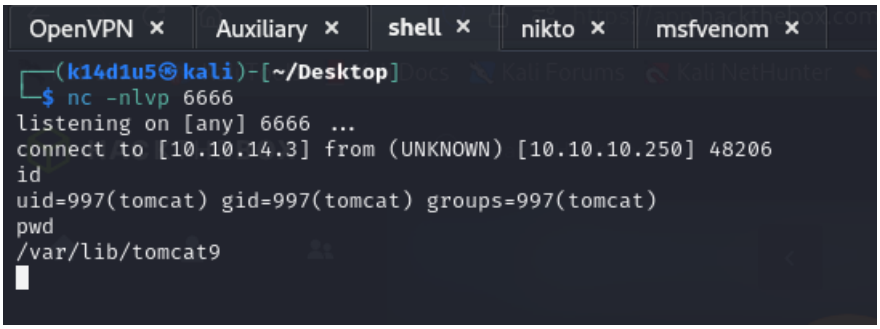

Figure 6 - First user shell

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):

```
cat /opt/backups/playbook/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/
tomcat@seal:/var/lib/tomcat9$
```

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:

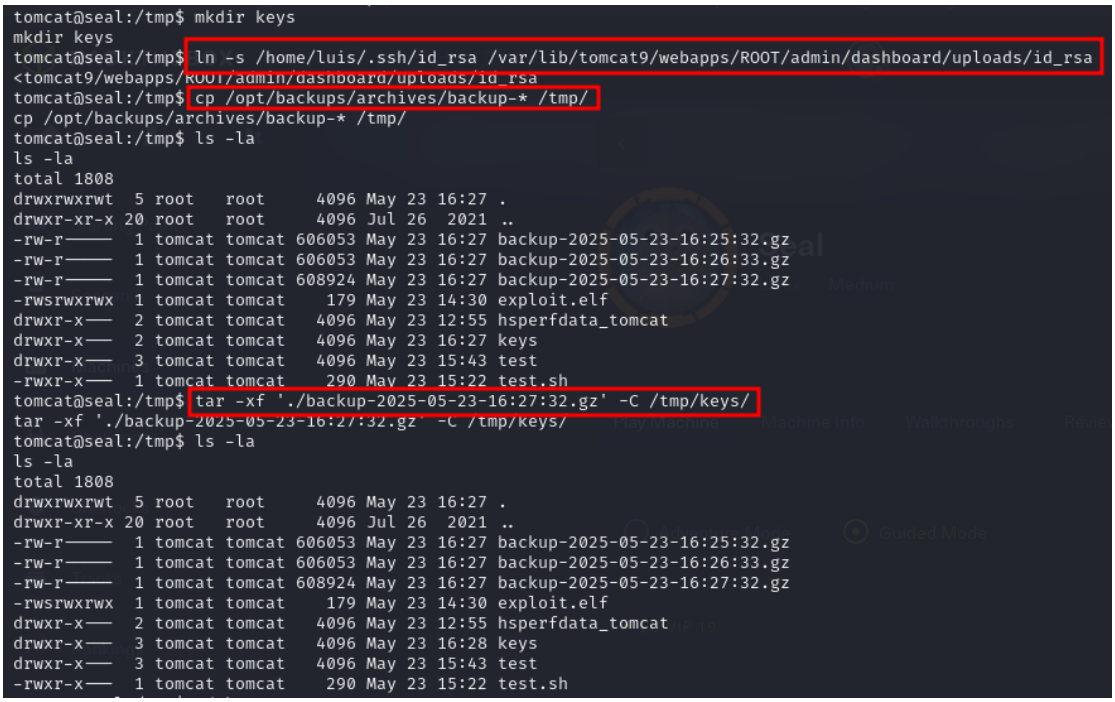

Figure 8 - Extracting Luis SSH private key

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

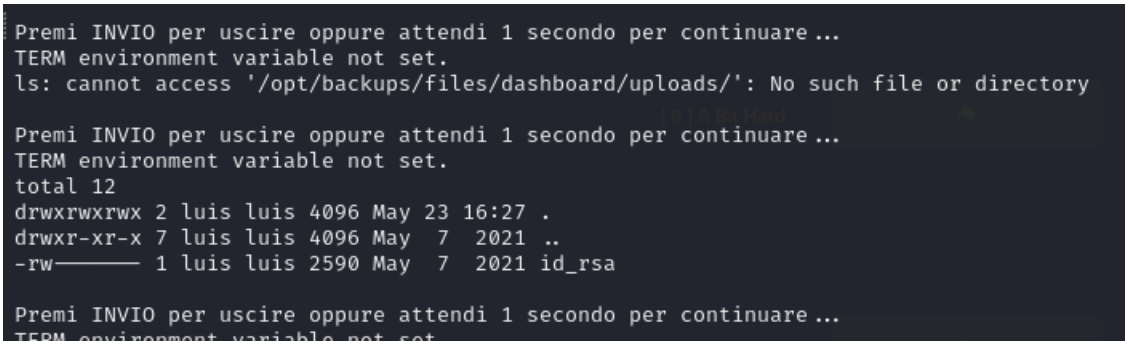

Figure 9 - Backup completed check

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:

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:

Figure 11 - Malicious reverse shell

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

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/.