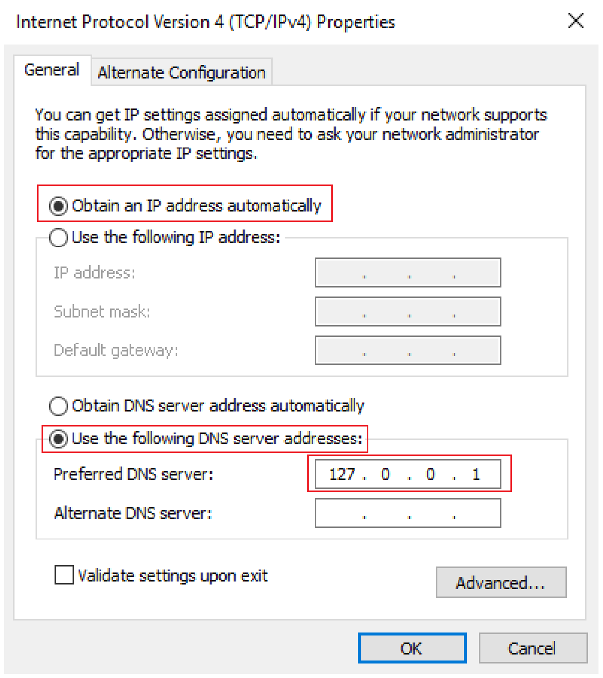
Lab - Enumerating Active Directory Using RPCClient

Overview

In this lab, you will learn enumeration of the domain via the SMB and RPC channels using the rpcclient. The rpcclient issues administrative commands using Microsoft Remote Procedure Calls (RPCs), which provide access to the Windows administration graphical user interfaces (GUIs) for systems management. The RPC run-time stubs and libraries manage most of the processes relating to network protocols and communication and support communication between Windows applications.

Lab Requirements

- One install of VirtualBox, the latest version with the extension pack.
- One virtual install of Kali Linux, latest version.
- One virtual install of Windows Server 2012, 2016, or 2019.
- Ensure all VirtualBox network adapters are set to Nat Network.
- Ensure your IPv4 settings for your Server 2016 DC are set for DHCP. Set the DNS address for manual and use 127.0.0.1 for the primary DNS server.


Internet Protocol Version 4 (TCP/IPv4) Properties

General | Alternate Configuration

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

[✓ Obtain an IP address automatically]
Use the following IP address:
IP address: . . .
Subnet mask: . . .
Default gateway: . . .

Obtain DNS server address automatically
[✓ Use the following DNS server addresses:]
Preferred DNS server: 127 . 0 . 0 . 1
Alternate DNS server: . . .

Validate settings upon exit

Begin the lab!

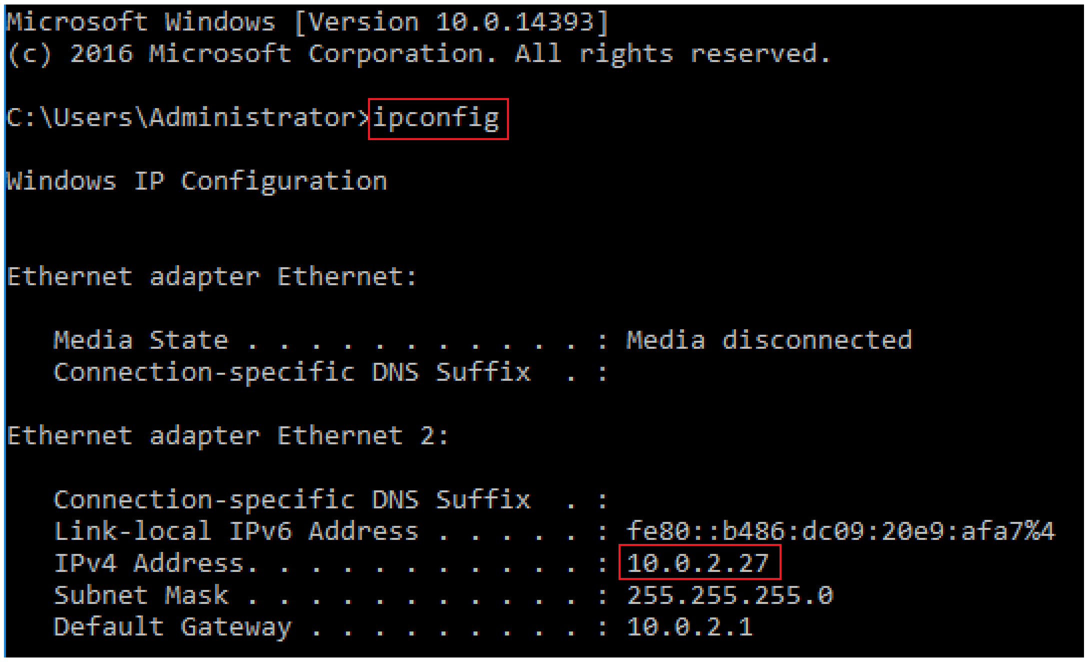
You need to have the IP address of your target machine. For this lab, I will be using Server 2016 configured as a root domain controller as my target.

If you do not know the IP address of your remote target on your target machine, open a command prompt and at the prompt type, **`ipconfig`**.

This is my target's IP address. Yours will differ!

Caveat!

Both your attack and your target machines must be part of the same network. If you cannot connect from your Kali machine to your target, ensure both machines have the VirtualBox adapters set to **NAT network**. Both devices need the first three octets of their IP address to match. This is the network portion of the IP address assigned.



```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Ethernet adapter Ethernet 2:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::b486:dc09:20e9:afa7%4
   IPv4 Address. . . . . . . . . . . : 10.0.2.27
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.0.2.1
```

In our previous lab, we brute-forced the SMB password of our Server 2016 using Metasploit using the **`auxiliary/scanner/smb/smb_login`** module, which can be configured to do password guessing using a wordlist.

Connecting To Your Remote Target Using Rpcclient

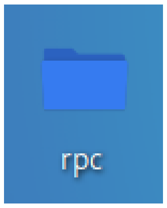
The easiest way to connect to a remote target using rpcclient is to use an authentication file.

This option allows you to specify the path of a text file which contains the username, password, and domain name needed for the connection. The format of the file is as follows. This my information; yours will differ!

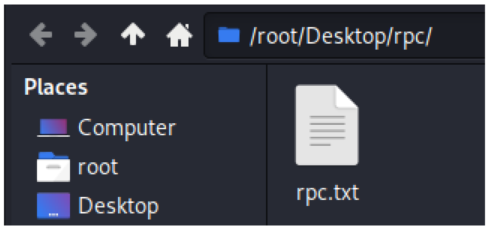
```
username = administrator
password = Password123!
domain   = us.syberoffense.com
```

Create A Working Directory

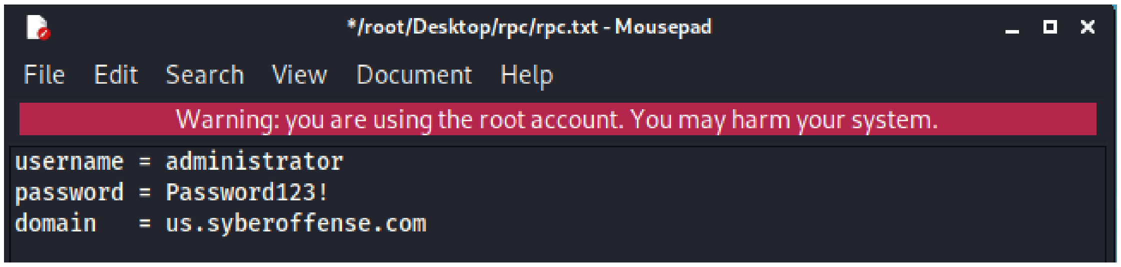
On your Kali desktop, create a new directory, name the new directory **`rpc`** all lower case.



Open the directory and in the right windowpane, right-click and from the context menu select, **Create Document** and name the document, **`rpc.txt`**.

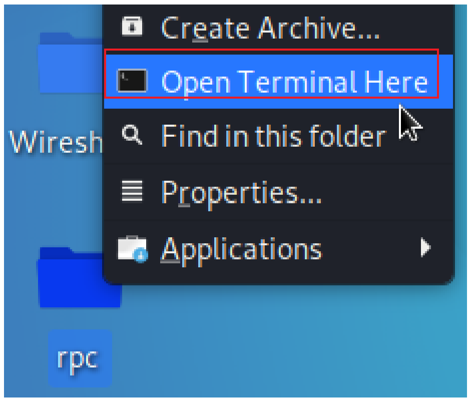


x2 click the file to open and paste the three values into the text files.


*/root/Desktop/rpc/rpc.txt - Mousepad
File Edit Search View Document Help
Warning: you are using the root account. You may harm your system.

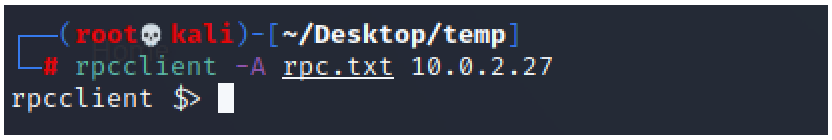
```
username = administrator
password = Password123!
domain   = us.syberoffense.com
```

Once you have ensured you have connectivity between your Kali and your target, you can begin the lab by opening a terminal on your Kali Linux using your **`rpc`** working directory as your root. From the desktop, find your rpc directory, right-click the folder icon and from the context menu select, **Open Terminal Here**.



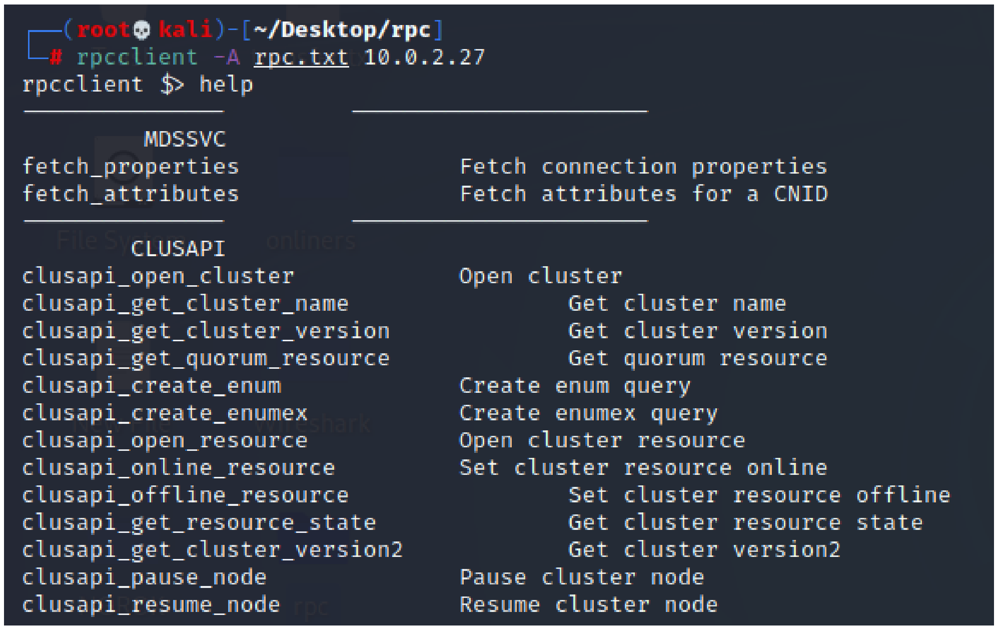
To authenticate with the target, at the prompt, type the following command. Notice your prompt changes.

`rpcclient -A rpc.txt 10.0.2.27`



```
┌──(root💀 kali)-[~/Desktop/temp]
└─# rpcclient -A rpc.txt 10.0.2.27
rpcclient $> 
```

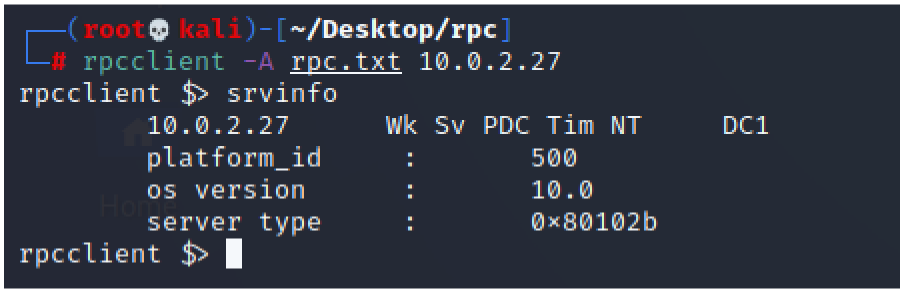
To demonstrate how powerful this utility is, at the prompt type, **`help`**. You could spend hours going through each of these command options.



```
┌──(root💀 kali)-[~/Desktop/rpc]
└─# rpcclient -A rpc.txt 10.0.2.27
rpcclient $> help
───────────────         ──────────────────────
         MDSSVC
fetch_properties                Fetch connection properties
fetch_attributes                Fetch attributes for a CNID
───────────────         ──────────────────────
        CLUSAPI
clusapi_open_cluster            Open cluster
clusapi_get_cluster_name                Get cluster name
clusapi_get_cluster_version             Get cluster version
clusapi_get_quorum_resource             Get quorum resource
clusapi_create_enum             Create enum query
clusapi_create_enumex           Create enumex query
clusapi_open_resource           Open cluster resource
clusapi_online_resource         Set cluster resource online
clusapi_offline_resource                Set cluster resource offline
clusapi_get_resource_state              Get cluster resource state
clusapi_get_cluster_version2            Get cluster version2
clusapi_pause_node              Pause cluster node
clusapi_resume_node             Resume cluster node
```

`srvinfo`

When enumerating the target, we can use the **`srvinfo`** command to get the Windows operating system version.

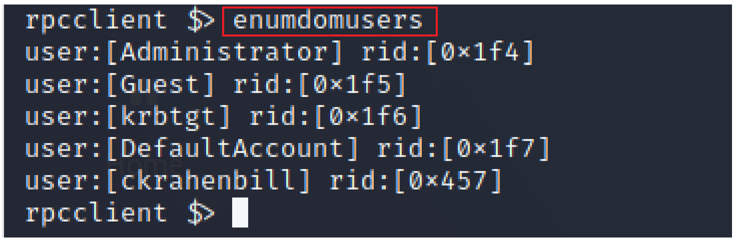


```
┌──(root💀 kali)-[~/Desktop/rpc]
└─# rpcclient -A rpc.txt 10.0.2.27
rpcclient $> srvinfo
        10.0.2.27       Wk Sv PDC Tim NT     DC1
        platform_id     :       500
        os version      :       10.0
        server type     :       0x80102b
rpcclient $> 
```

We can determine by the information that our target is running the OS version 10, meaning the machine is either a Windows 10 or Server 2016. We can also determine that since the target is configured as a PDC, it is running Windows Server 2016.

`enumdomusers`

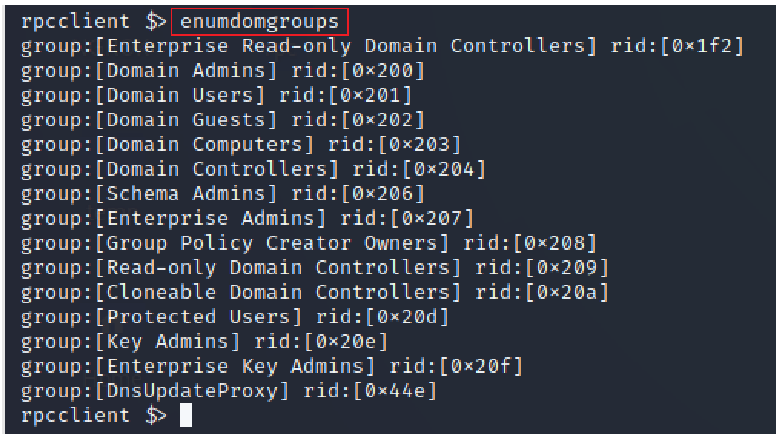
We can next move on to enumerating domain users using the **`enumdomusers`** command.



```
rpcclient $> enumdomusers
user:[Administrator] rid:[0×1f4]
user:[Guest] rid:[0×1f5]
user:[krbtgt] rid:[0×1f6]
user:[DefaultAccount] rid:[0×1f7]
user:[ckrahenbill] rid:[0×457]
rpcclient $>
```

`enumdomgroups`

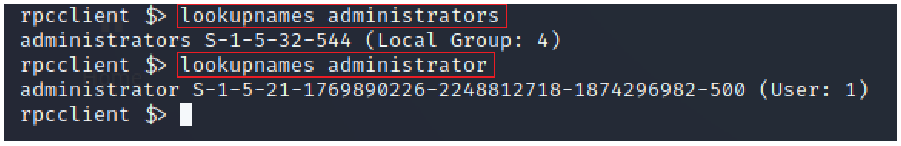
Enumerating groups helps in planning how best to elevate privilege access. The different policies that are applied on a domain are also dictated by the various groups that exist. Many groups are created for a specific service which helps identify which services are running on the target.



```
rpcclient $> enumdomgroups
group:[Enterprise Read-only Domain Controllers] rid:[0×1f2]
group:[Domain Admins] rid:[0×200]
group:[Domain Users] rid:[0×201]
group:[Domain Guests] rid:[0×202]
group:[Domain Computers] rid:[0×203]
group:[Domain Controllers] rid:[0×204]
group:[Schema Admins] rid:[0×206]
group:[Enterprise Admins] rid:[0×207]
group:[Group Policy Creator Owners] rid:[0×208]
group:[Read-only Domain Controllers] rid:[0×209]
group:[Cloneable Domain Controllers] rid:[0×20a]
group:[Protected Users] rid:[0×20d]
group:[Key Admins] rid:[0×20e]
group:[Enterprise Key Admins] rid:[0×20f]
group:[DnsUpdateProxy] rid:[0×44e]
rpcclient $>
```

`lookupnames`

For both the users and the groups, we are given the RID for each user or group. To see the actual SID for the user or the group, we can use the **`lookupnames`** command.



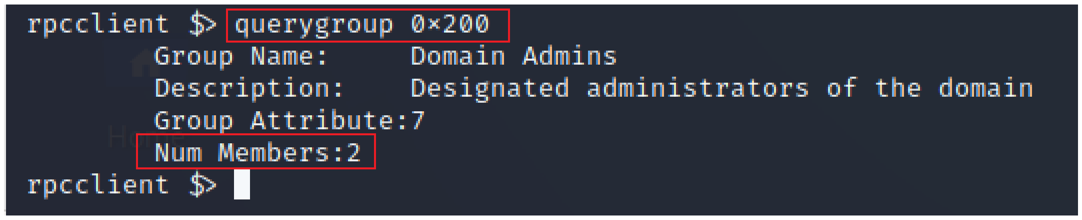
```
rpcclient $> lookupnames administrators
administrators S-1-5-32-544 (Local Group: 4)
rpcclient $> lookupnames administrator
administrator S-1-5-21-1769890226-2248812718-1874296982-500 (User: 1)
rpcclient $>
```

In the above example, you can see the SID for the administrator's group, and on the following line, you can see the SID for the individual administrator user account.

query group <rid>

Once we have enumerated the groups, we can gather more information about each group using the **querygroup <rid>** command. Let dig up some information about the administrators group.

querygroup 0x200

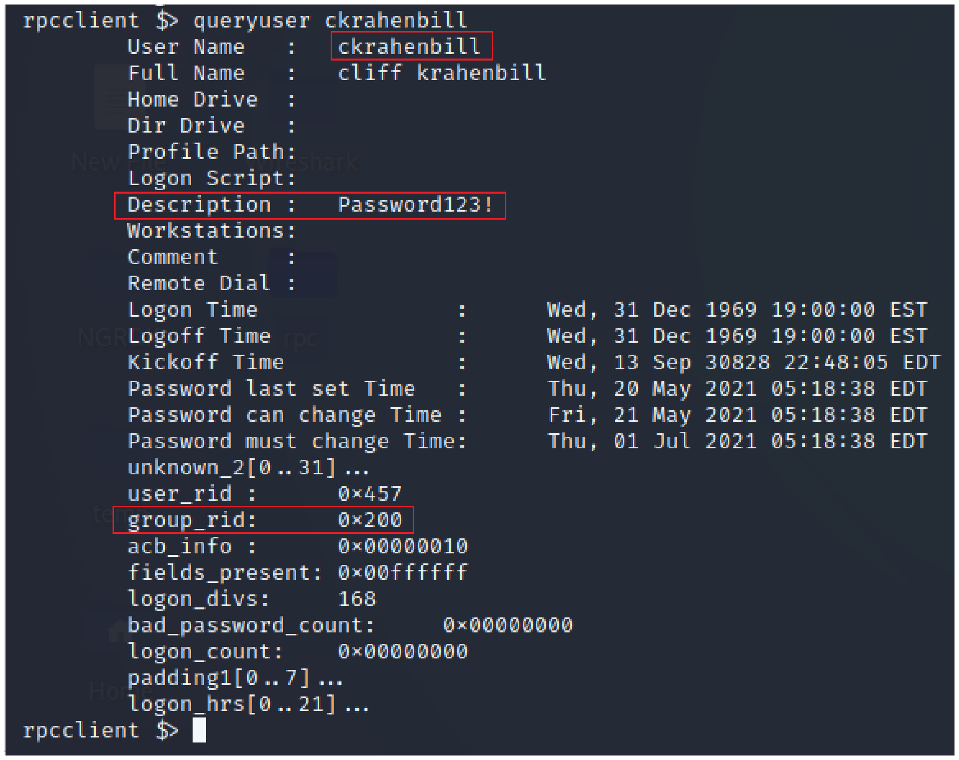


```
rpcclient $> querygroup 0x200
        Group Name:     Domain Admins
        Description:    Designated administrators of the domain
        Group Attribute:7
        Num Members:2
rpcclient $>
```

We now know that this group has full access to the domain (duh!), and the group has two members.

queryuser <username>

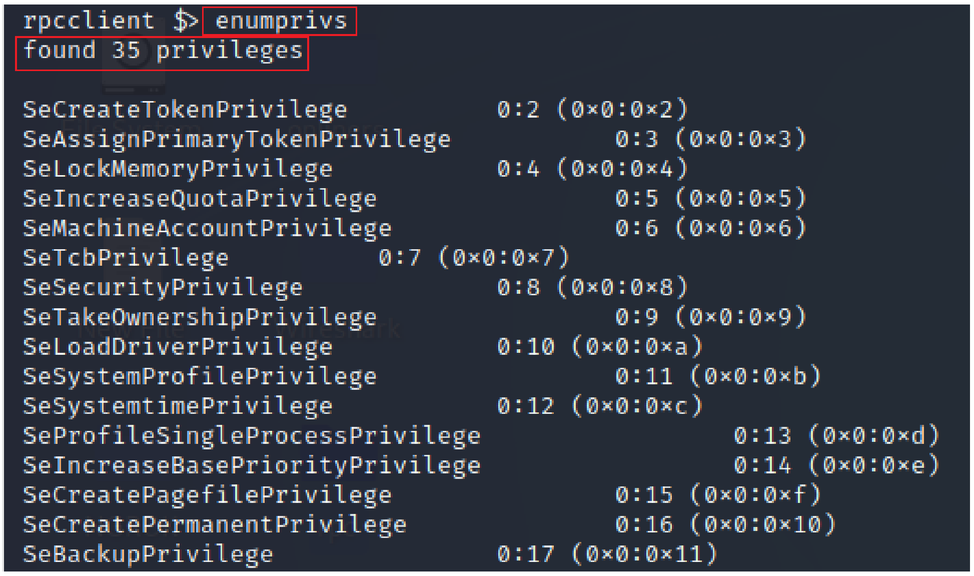
And of course, we can do the same with any user in the domain. In this example, we see that the user's description has his password provided. Nice! And we can also see that his primary group membership is the domain admins group, 0x200. The account can now be taken over.



```
rpcclient $> queryuser ckrahenbill
        User Name   :   ckrahenbill
        Full Name   :   cliff krahenbill
        Home Drive  :
        Dir Drive   :
        Profile Path:
        Logon Script:
        Description :   Password123!
        Workstations:
        Comment     :
        Remote Dial :
        Logon Time               :      Wed, 31 Dec 1969 19:00:00 EST
        Logoff Time              :      Wed, 31 Dec 1969 19:00:00 EST
        Kickoff Time             :      Wed, 13 Sep 30828 22:48:05 EDT
        Password last set Time   :      Thu, 20 May 2021 05:18:38 EDT
        Password can change Time :      Fri, 21 May 2021 05:18:38 EDT
        Password must change Time:      Thu, 01 Jul 2021 05:18:38 EDT
        unknown_2[0..31]...
        user_rid :      0x457
        group_rid:      0x200
        acb_info :      0x00000010
        fields_present: 0x00ffffff
        logon_divs:     168
        bad_password_count:     0x00000000
        logon_count:    0x00000000
        padding1[0..7]...
        logon_hrs[0..21]...
rpcclient $>
```

`enumprivs`

After enumerating the users and groups, the next step would be to enumerate the privileges. Understanding the privileges of the user we are currently logged as can help us determine the best path for elevating our privileges and access.



```
rpcclient $> enumprivs
found 35 privileges

SeCreateTokenPrivilege          0:2 (0×0:0×2)
SeAssignPrimaryTokenPrivilege           0:3 (0×0:0×3)
SeLockMemoryPrivilege           0:4 (0×0:0×4)
SeIncreaseQuotaPrivilege                0:5 (0×0:0×5)
SeMachineAccountPrivilege               0:6 (0×0:0×6)
SeTcbPrivilege          0:7 (0×0:0×7)
SeSecurityPrivilege             0:8 (0×0:0×8)
SeTakeOwnershipPrivilege                0:9 (0×0:0×9)
SeLoadDriverPrivilege           0:10 (0×0:0×a)
SeSystemProfilePrivilege                0:11 (0×0:0×b)
SeSystemtimePrivilege           0:12 (0×0:0×c)
SeProfileSingleProcessPrivilege                 0:13 (0×0:0×d)
SeIncreaseBasePriorityPrivilege                 0:14 (0×0:0×e)
SeCreatePagefilePrivilege               0:15 (0×0:0×f)
SeCreatePermanentPrivilege              0:16 (0×0:0×10)
SeBackupPrivilege               0:17 (0×0:0×11)
```

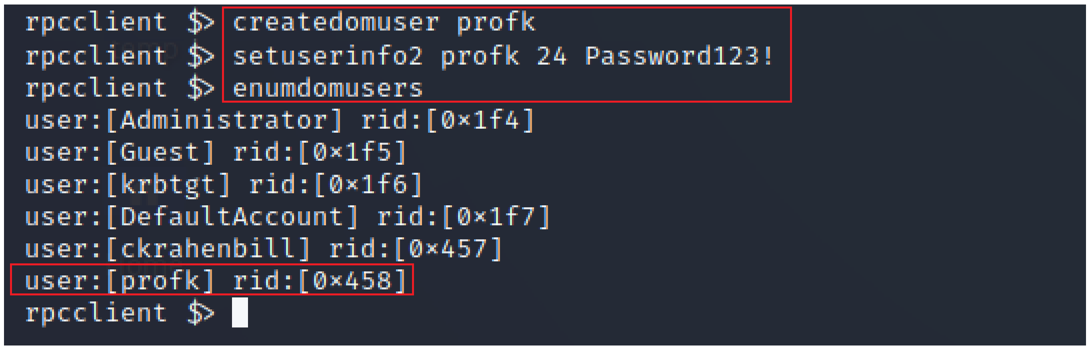
Creating Domain User

Now that we know what privileges we have, we can create a domain user using the three following commands. In this example, using the **`createdomuser`** command, I create a new user called **profk.**

Using the **`setuserinfo2`**, I set a level 24 password for the new user, and lastly, I can verify the user has been created using the **`enumdomusers`** command.

Information about different password levels can be found using this MSDN article.

```
createdomuser profk
setuserinfo2 profk 24 Password123!
Enumdomusers
```

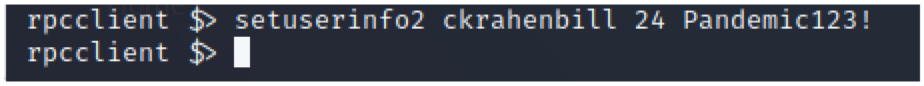


```
rpcclient $> createdomuser profk
rpcclient $> setuserinfo2 profk 24 Password123!
rpcclient $> enumdomusers
user:[Administrator] rid:[0×1f4]
user:[Guest] rid:[0×1f5]
user:[krbtgt] rid:[0×1f6]
user:[DefaultAccount] rid:[0×1f7]
user:[ckrahenbill] rid:[0×457]
user:[profk] rid:[0×458]
rpcclient $> 
```

Change Password When The Password Is Unknown

We are unable to change the password of anyone with **`AdminCount = 1`** (Domain Admins), but we can target users who have alternate admin accounts:

`setuserinfo2 ckrahenbill 24 Pandemic123!`



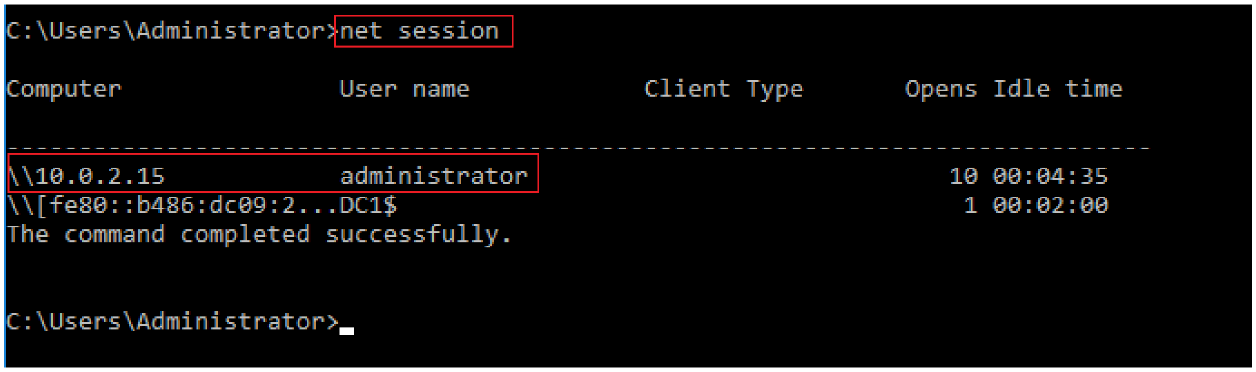
```
rpcclient $> setuserinfo2 ckrahenbill 24 Pandemic123!
rpcclient $> 
```

There is no confirmation, but we are returned to the prompt to let us know that the command was completed successfully. We know that the user crahenbill is a member of the domain admins, and so if we did not have his password given to us in his account description, we could change his domain password using this method.

Detecting A Bad Smb Session

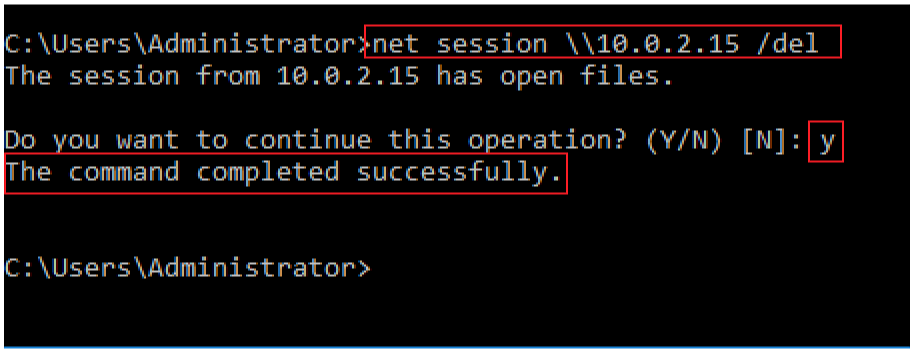
If you find yourself on the receiving end of an SMB attack, you can confirm your suspicions by seeing who or what has established an SMB session to your machine using the net session command.

From your Windows target, open a command prompt and at the prompt type, **`net session`**. Here we can see my SMB connection running from my Kali machine.



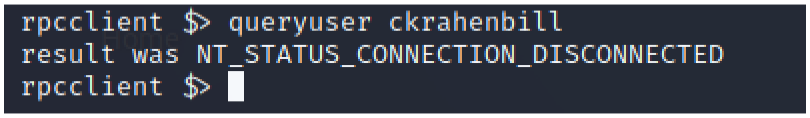
```
C:\Users\Administrator>net session

Computer                 User name           Client Type       Opens Idle time

-------------------------------------------------------------------------------
\\10.0.2.15              administrator                            10 00:04:35
\\[fe80::b486:dc09:2...DC1$                                        1 00:02:00
The command completed successfully.


C:\Users\Administrator>_
```

To kill the bad session, at the prompt, we can delete the connection using the following command **`net session \\10.0.2.15 /del`**



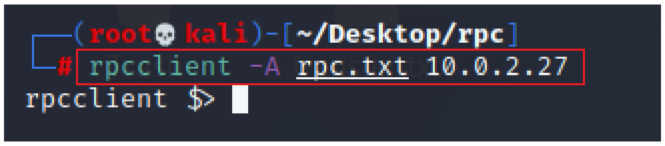
```
C:\Users\Administrator>net session \\10.0.2.15 /del
The session from 10.0.2.15 has open files.

Do you want to continue this operation? (Y/N) [N]: y
The command completed successfully.


C:\Users\Administrator>
```

Back at my Kali machine, we try and query a user; we see that we have been disconnected.



```
rpcclient $> queryuser ckrahenbill
result was NT_STATUS_CONNECTION_DISCONNECTED
rpcclient $>
```

I re-establish the connection.



```
┌──(root💀kali)-[~/Desktop/rpc]
└─# rpcclient -A rpc.txt 10.0.2.27
rpcclient $>
```

We can quickly re-establish the connection with our rpcclient. We could block the user by configuring the built-in Windows firewall to block TCP port 445, but that would stop all inbound SMB sessions. We can use a FOR loop on the Windows target to kill SMB sessions from this attacker every 1 second.

```
FOR /L %i in (1,0,2) do @net session \\10.0.2.15 /del /y & ping -n 2 127.0.0.1>nul
```

We use the FOR /L loop, which is a counter. It starts counting at 1, counts in steps of 0, up to 2. Thus, the command will keep running until we close the command prompt. At each iteration of the loop, I drop the SMB session and then ping localhost twice, introducing a 1-second delay. The first ping happens nearly instantly; the second one happens one second later. Our attacker will make a session that will last for one second and then get disconnected.

Summary

In this lab, you learned how to use the rpcclient to enumerate a wide range of information through the SMB and RPC channel inside a domain. This lab can serve as a reference for Red Team activists for attacking and enumerating the domain. Still, it can also be helpful for the Blue Team to understand and test the measures applied on the domain to protect the Network and its users.