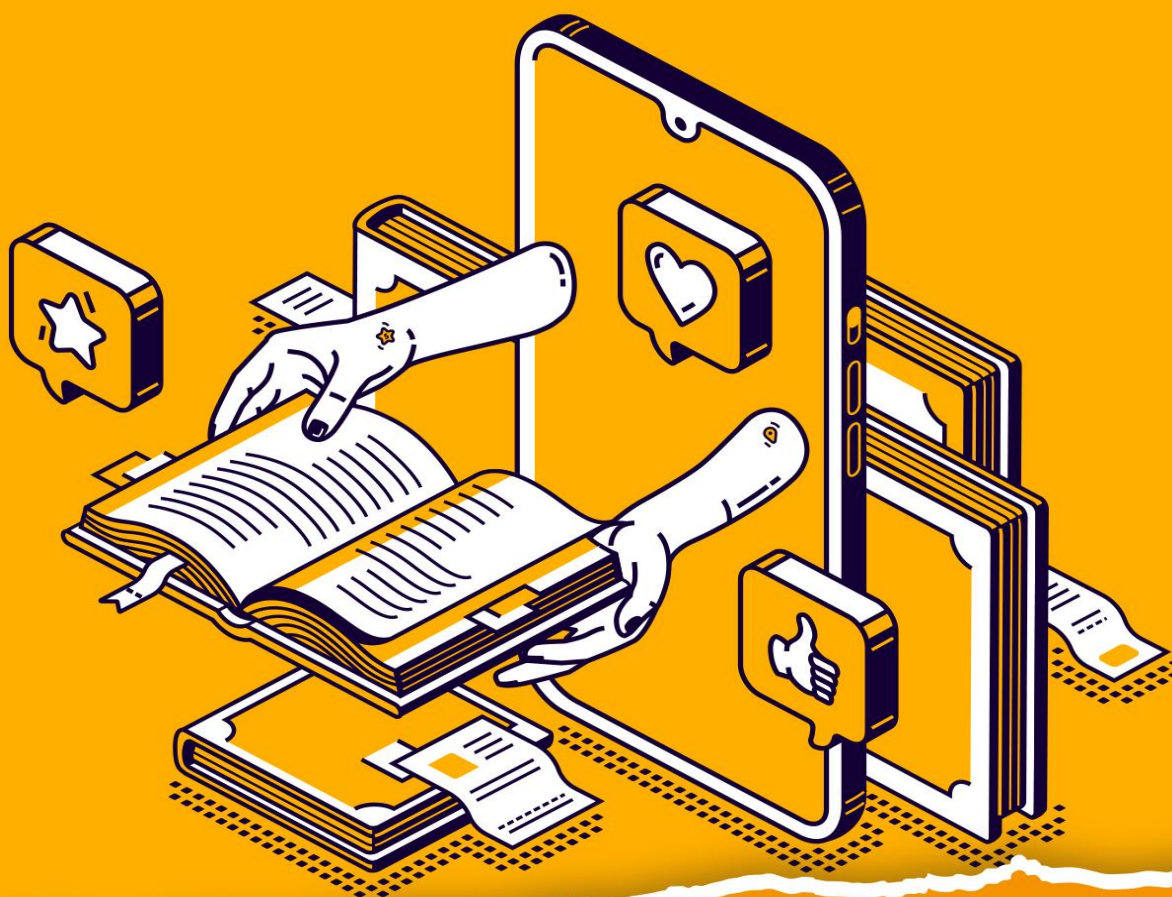


# Active **DIRECTORY ENUMERATION** RPCClient



## Contents

Introduction .....	4
Logging and Server Information.....	4
Domain Information Query.....	4
Enumerating Domain Users .....	5
Enumerating Domain Groups.....	5
Group Queries .....	6
User Queries.....	6
Enumerating Privileges .....	7
Get Domain Password Information .....	8
Get User Domain Password Information .....	9
Enumerating SID from LSA .....	9
Creating Domain User .....	10
Lookup Names.....	11
Enumerating Alias Groups.....	11
Delete Domain User .....	12
Net Share Enumeration.....	12
Net Share Get Information.....	13
Enumerating Domains.....	14
Enumerating Domain Groups.....	14
Display Query Information.....	15
Change Password of User.....	16
Create Domain Group .....	16
Delete Domain Group .....	16
Domain Lookup .....	17

<b>SAM Lookup .....</b>	<b>17</b>
<b>SID Lookup.....</b>	<b>18</b>
<b>LSA Query .....</b>	<b>18</b>
<b>LSA Create Account .....</b>	<b>19</b>
<b>Enumerating LSA Group Privileges.....</b>	<b>19</b>
<b>Enumerating LSA Account Privileges .....</b>	<b>21</b>
<b>LSA Query Security Objects.....</b>	<b>22</b>
<b>Conclusion .....</b>	<b>22</b>

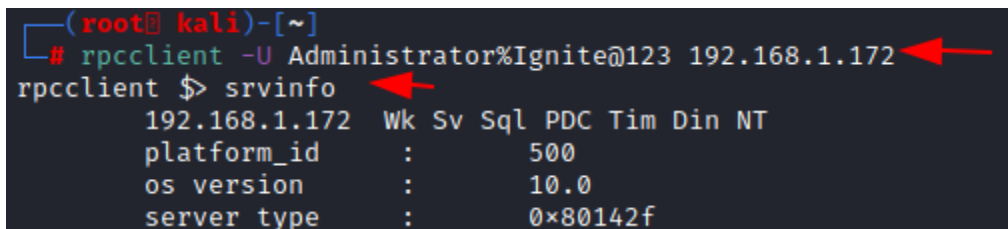
## Introduction

RPC, or Remote Procedure Call, is a service that helps establish and maintain communication between different Windows applications. The RPC service works on the RPC protocols that form the low-level inter-process communication between different applications. In this communication, the child process can make requests to the parent process. The child-parent relationship here can also be depicted as a client-server relationship. RPC is built on Microsoft's COM and DCOM technologies. In general, the rpcclient can be used to connect to the SMB protocol as well. rpcclient is a part of the Samba suite on Linux distributions. The rpcclient was designed to perform debugging and troubleshooting tasks on a Windows Samba configuration. During that time, the designers of the rpcclient might be clueless about the importance of this tool as a penetration testing tool. There are multiple methods to connect to a remote RPC service. However, for this particular demonstration, we are using rpcclient.

## Logging and Server Information

To begin the enumeration, a connection needs to be established. This can be done by providing the username and password followed by the target IP address of the server. After establishing the connection, to get the grasp of various commands that can be used you can run the help. One of the first enumeration commands to be demonstrated here is the srvinfo command. It can be used in the rpcclient shell that was generated to enumerate information about the server. It can be observed that the OS version seems to be 10.0. That narrows the versions that the attacker might be looking at to Windows 10, Windows Server 2016, and Windows Server 2019. Learn more about the OS versions.

```
rpcclient -U Administrator%Ignite@123 192.168.1.172
```



```
(root@kali)~# rpcclient -U Administrator%Ignite@123 192.168.1.172
rpcclient $> srvinfo
192.168.1.172 Wk Sv Sql PDC Tim Din NT
platform_id : 500
os version : 10.0
server type : 0x80142f
```

## Domain Information Query

The next command that can be used via rpcclient is querydominfo. This command retrieves the domain, server, users on the system, and other relevant information. From the demonstration, it can be observed that the domain that is being enumerated is IGNITE. It has a total of 67 users. There was a forced logoff on the server and other important information.

```
querydominfo
```

```
rpcclient $> querydomaininfo
Domain:      IGNITE
Server:
Comment:
Total Users: 67
Total Groups: 0
Total Aliases: 0
Sequence No: 1
Force Logoff: -1
Domain Server State: 0x1
Server Role:  ROLE_DOMAIN_PDC
Unknown 3:    0x1
```

## Enumerating Domain Users

Another command to use is the enumdomusers. The name is derived from the enumeration of domain users. Upon running this on the rpcclient shell, it will extract the usernames with their RID. RID is a suffix of the long SID in a hexadecimal format. In this specific demonstration, there are a bunch of users that include Administrator, yashika, aarti, raj, Pavan, etc.

enumdomusers

```
rpcclient $> enumdomusers
user:[Administrator] rid:[0x1f4]
user:[Guest] rid:[0x1f5]
user:[krbtgt] rid:[0x1f6]
user:[DefaultAccount] rid:[0x1f7]
user:[yashika] rid:[0x44f]
user:[geet] rid:[0x450]
user:[aarti] rid:[0x451]
user:[raj] rid:[0x642]
user:[pavan] rid:[0x643]
user:[SVC_SQLService] rid:[0x838]
user:[jeenali] rid:[0x83a]
user:[japneet] rid:[0x83b]
user:[ignite] rid:[0x83c]
```

## Enumerating Domain Groups

Since we performed enumeration on different users, it is only fair to extend this to various groups as well. The group information helps the attacker plan their way to the administrator or elevated access. The policies that are applied to a domain are also dictated by the various groups that exist. Many groups are created for a specific service. So, it is also a good way to enumerate what kind of services might be running on the server. This can be done using enumdomgroup. The name is derived from the enumeration of domain groups. Upon running this on the rpcclient shell, it will extract the groups with their RID.

enumdomgroups

```
rpcclient $> enumdomgroups
group:[Enterprise Read-only Domain Controllers] rid:[0x1f2]
group:[Domain Admins] rid:[0x200]
group:[Domain Users] rid:[0x201]
group:[Domain Guests] rid:[0x202]
group:[Domain Computers] rid:[0x203]
group:[Domain Controllers] rid:[0x204]
group:[Schema Admins] rid:[0x206]
group:[Enterprise Admins] rid:[0x207]
group:[Group Policy Creator Owners] rid:[0x208]
group:[Read-only Domain Controllers] rid:[0x209]
group:[Cloneable Domain Controllers] rid:[0x20a]
group:[Protected Users] rid:[0x20d]
group:[Key Admins] rid:[0x20e]
group:[Enterprise Key Admins] rid:[0x20f]
group:[DnsUpdateProxy] rid:[0x44e]
group:[Finance] rid:[0x839]
```

## Group Queries

After enumerating groups, it is possible to extract details about a particular group from the list. This information includes the Group Name, Description, Attributes, and the number of members in that group. It is possible to target the group using the RID that was extracted while running the enumdomgroup. For the demonstration here, RID 0x200 was used to find that it belongs to the Domain Admin group. This group constitutes 7 attributes, and 2 users are members of this group.

**querygroup 0x200**

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

## User Queries

The ability to enumerate individually doesn't limit to the groups but also extends to the users. To enumerate a particular user from rpcclient, the queryuser command must be used. When provided the username, it extracts information such as the username, Full name, Home Drive, Profile Path, Description, Logon Time, Logoff Time, Password set time, Password Change Frequency, RID, Groups, etc. In the demonstration, it can be observed that the user has stored their credentials in the Description. Hence, the credentials were successfully enumerated and the account can be taken over now.

**queryuser yashika**

```
rpcclient $> queryuser yashika
User Name      : yashika
Full Name      : yashika
Home Drive     :
Dir Drive      :
Profile Path    :
Logon Script    :
Description    : pass Password@1
Workstations    :
Comment        :
Remote Dial     :
Logon Time      : Sun, 18 Apr 2021 14:54:32 EDT
Logoff Time     : Wed, 31 Dec 1969 19:00:00 EST
Kickoff Time    : Wed, 13 Sep 30828 22:48:05 EDT
Password last set Time : Mon, 29 Jun 2020 13:08:50 EDT
Password can change Time : Tue, 30 Jun 2020 13:08:50 EDT
Password must change Time: Wed, 13 Sep 30828 22:48:05 EDT
unknown_2[0..31] ...
user_rid       : 0x44f
group_rid      : 0x201
acb_info       : 0x00000210
fields_present : 0x00ffffff
logon_divs      : 168
bad_password_count: 0x00000000
logon_count     : 0x00000046
padding1[0..7] ...
logon_hrs[0..21] ...
```

## Enumerating Privileges

After the user details and the group details, another piece of information that can help an attacker that has retained the initial foothold on the domain is the privileges. These privileges can help the attacker plan for elevating privileges on the domain. The privileges can be enumerated using the `enumprivs` command on `rpcclient`. In the demonstration, it can be observed that the current user has been allocated 35 privileges.

**enumprivs**



```

rpcclient $> enumprivs
found 35 privileges

SeCreateTokenPrivilege          0:2 (0x0:0x2)
SeAssignPrimaryTokenPrivilege   0:3 (0x0:0x3)
SeLockMemoryPrivilege          0:4 (0x0:0x4)
SeIncreaseQuotaPrivilege        0:5 (0x0:0x5)
SeMachineAccountPrivilege       0:6 (0x0:0x6)
SeTcbPrivilege                  0:7 (0x0:0x7)
SeSecurityPrivilege             0:8 (0x0:0x8)
SeTakeOwnershipPrivilege        0:9 (0x0:0x9)
SeLoadDriverPrivilege          0:10 (0x0:0xa)
SeSystemProfilePrivilege        0:11 (0x0:0xb)
SeSystemtimePrivilege          0:12 (0x0:0xc)
SeProfileSingleProcessPrivilege 0:13 (0x0:0xd)
SeIncreaseBasePriorityPrivilege 0:14 (0x0:0xe)
SeCreatePagefilePrivilege       0:15 (0x0:0xf)
SeCreatePermanentPrivilege      0:16 (0x0:0x10)
SeBackupPrivilege              0:17 (0x0:0x11)
SeRestorePrivilege             0:18 (0x0:0x12)
SeShutdownPrivilege            0:19 (0x0:0x13)
SeDebugPrivilege               0:20 (0x0:0x14)
SeAuditPrivilege               0:21 (0x0:0x15)
SeSystemEnvironmentPrivilege    0:22 (0x0:0x16)
SeChangeNotifyPrivilege        0:23 (0x0:0x17)
SeRemoteShutdownPrivilege      0:24 (0x0:0x18)
SeUndockPrivilege              0:25 (0x0:0x19)
SeSyncAgentPrivilege           0:26 (0x0:0x1a)
SeEnableDelegationPrivilege     0:27 (0x0:0x1b)
SeManageVolumePrivilege        0:28 (0x0:0x1c)
SeImpersonatePrivilege          0:29 (0x0:0x1d)
SeCreateGlobalPrivilege        0:30 (0x0:0x1e)
SeTrustedCredManAccessPrivilege 0:31 (0x0:0x1f)
SeRelabelPrivilege             0:32 (0x0:0x20)
SeIncreaseWorkingSetPrivilege   0:33 (0x0:0x21)
SeTimeZonePrivilege            0:34 (0x0:0x22)
SeCreateSymbolicLinkPrivilege   0:35 (0x0:0x23)
SeDelegateSessionUserImpersonatePrivilege 0:36 (0x0:0x24)

```

## Get Domain Password Information

The getdompwininfo command can be used to enumerate the password properties on the domain. This is made from the words "get domain password information." This will help in getting the information, such as the kind of password policies that have been enforced by the domain administrator. It is possible to enumerate the minimum password length and the enforcement of complex password rules. If these kinds of features are not enabled on the domain, then it is possible to brute force the credentials on the domain.

getdompwininfo



```
rpcclient $> getdompwinfo  
min_password_length: 7  
password_properties: 0x00000001  
DOMAIN_PASSWORD_COMPLEX
```

## Get User Domain Password Information

In the previous command, we used `getdompwinfo` to get the password properties of the domain administrated by the policies. But it is also possible to get the password properties of individual users using the `getusrdompwinfo` command with the user's RID. In the demonstration, the user with RID 0x1f4 was enumerated regarding their password properties.


**getusrdompwinfo 0x1f4**

```
rpcclient $> getusrdompwinfo 0x1f4  
&info: struct samr_PwInfo  
min_password_length : 0x0007 (7)  
password_properties : 0x00000001 (1)  
1: DOMAIN_PASSWORD_COMPLEX  
0: DOMAIN_PASSWORD_NO_ANON_CHANGE  
0: DOMAIN_PASSWORD_NO_CLEAR_CHANGE  
0: DOMAIN_PASSWORD_LOCKOUT_ADMINS  
0: DOMAIN_PASSWORD_STORE_CLEARTXT  
0: DOMAIN_REFUSE_PASSWORD_CHANGE
```

## Enumerating SID from LSA

Learning about various kinds of compromises that can be performed using Mimikatz we know that the SID of a user is the security Identifier that can be used for a lot of elevating privileges and minting tickets attacks. It can be enumerated through `rpcclient` using the `Isaenumsid` command. In the demonstration, it can be observed that `Isaenumsid` has enumerated 20 SIDs within the Local Security Authority or LSA.

**Isaenumsid**

```
rpcclient $> lsaenumsid   
found 20 SIDs  
  
S-1-5-9  
S-1-5-82-3006700770-424185619-1745488364-794895919-4004696415  
S-1-5-82-2887664442-61840710-2462234416-3208743808-4134260289  
S-1-5-82-1407536422-3657846629-613172646-645089302-3793875275  
S-1-5-80-3139157870-2983391045-3678747466-658725712-1809340420  
S-1-5-80-0  
S-1-5-6  
S-1-5-32-568  
S-1-5-32-559  
S-1-5-32-554  
S-1-5-32-551  
S-1-5-32-550  
S-1-5-32-549  
S-1-5-32-548  
S-1-5-32-545  
S-1-5-32-544  
S-1-5-20  
S-1-5-19  
S-1-5-11  
S-1-1-0
```

## Creating Domain User

While having some privileges, it is also possible to create a user within the domain using the rpcclient. It can be done with the help of the createdomuser command with the username that you want to create as a parameter. In the demonstration, a user named "hacker" is created with the help of a createdomuser and then a password is provided to it using the setuserinfo2 command. Finally, it can be verified using the enumdomusers command.

```
createdomuser hacker  
setuserinfo2 hacker 24 Password@1  
enumdomusers
```

```

rpcclient $> createdomuser hacker
rpcclient $> setuserinfo2 hacker 24 Password@1
rpcclient $> enumdomusers
user:[Administrator] rid:[0x1f4]
user:[Guest] rid:[0x1f5]
user:[krbtgt] rid:[0x1f6]
user:[DefaultAccount] rid:[0x1f7]
user:[yashika] rid:[0x44f]
user:[geet] rid:[0x450]
user:[aarti] rid:[0x451]
user:[raj] rid:[0x642]
user:[pavan] rid:[0x643]
user:[SVC_SQLService] rid:[0x838]
user:[jeenali] rid:[0x83a]
user:[japneet] rid:[0x83b]
user:[ignite] rid:[0x83c]
user:[hacker] rid:[0x83d]

```

## Lookup Names

We can also check if the user we created has been assigned a SID or not by using the `lookupnames` command on the `rpcclient`. As with the `lsaenumsid`, it was possible to extract the SID, but it was not possible to tell which user has that SID. This problem is solved using `lookupnames`, whereupon providing a username, the SID of that particular user can be extracted with ease.

```
lookupnames hacker
```

```

rpcclient $> lookupnames hacker
hacker S-1-5-21-501555289-2168925624-2051597760-2109 (User: 1)

```

## Enumerating Alias Groups

The next command that can be used is `enumalsgroups`. It enumerates alias groups in the domain. An alias is an alternate name that can be used to reference an object or element. When used with the built-in parameter, it shows all the builtin groups by their alias names, as demonstrated below.

```
enumalsgroups builtin
```

```
rpcclient $> enumalsgroups builtin
group:[Account Operators] rid:[0x224]
group:[Pre-Windows 2000 Compatible Access] rid:[0x22a]
group:[Incoming Forest Trust Builders] rid:[0x22d]
group:[Windows Authorization Access Group] rid:[0x230]
group:[Terminal Server License Servers] rid:[0x231]
group:[Administrators] rid:[0x220]
group:[Users] rid:[0x221]
group:[Guests] rid:[0x222]
group:[Print Operators] rid:[0x226]
group:[Backup Operators] rid:[0x227]
group:[Replicator] rid:[0x228]
group:[Remote Desktop Users] rid:[0x22b]
group:[Network Configuration Operators] rid:[0x22c]
group:[Performance Monitor Users] rid:[0x22e]
group:[Performance Log Users] rid:[0x22f]
group:[Distributed COM Users] rid:[0x232]
group:[IIS_IUSRS] rid:[0x238]
group:[Cryptographic Operators] rid:[0x239]
group:[Event Log Readers] rid:[0x23d]
group:[Certificate Service DCOM Access] rid:[0x23e]
group:[RDS Remote Access Servers] rid:[0x23f]
group:[RDS Endpoint Servers] rid:[0x240]
group:[RDS Management Servers] rid:[0x241]
group:[Hyper-V Administrators] rid:[0x242]
group:[Access Control Assistance Operators] rid:[0x243]
group:[Remote Management Users] rid:[0x244]
group:[System Managed Accounts Group] rid:[0x245]
group:[Storage Replica Administrators] rid:[0x246]
group:[Server Operators] rid:[0x225]
```

## Delete Domain User

The ability to manipulate a user doesn't end with creating a user or changing the password of a user. If proper privileges are assigned, it is also possible to delete a user using the rpcclient. The `deletedomuser` command is used to perform this action.

`deletedomuser hacker`

```
rpcclient $> deletedomuser hacker
```

## Net Share Enumeration

When dealing with SMB an attacker is bound to be dealt with the Network Shares on the Domain. Most corporate offices don't want their employees to use USB sticks or other mediums to share files and data among themselves. Hence, they usually set up a network share. There are times when these shared folders may contain sensitive or confidential information that can be used to compromise the target. To enumerate these shares, the attacker can use `netshareenum` on the rpcclient. If you want to enumerate all the shares, then use `netshareenumall`.

`netshareenum`  
`netshareenumall`

```

rpcclient $> netshareenum ←
netname: SYSVOL
    remark: Logon server share
    path:   C:\Windows\SYSVOL\sysvol
    password: (null)
netname: NETLOGON
    remark: Logon server share
    path:   C:\Windows\SYSVOL\sysvol\ignite.local\SCRIPTS
    password: (null)
netname: Users
    remark:
    path:   C:\Users
    password: (null)
netname: Confidential
    remark:
    path:   C:\Confidential
    password: (null)
rpcclient $> netshareenumall ←
netname: ADMIN$
    remark: Remote Admin
    path:   C:\Windows
    password: (null)
netname: C$
    remark: Default share
    path:   C:\
    password: (null)
netname: Confidential
    remark:
    path:   C:\Confidential
    password: (null)
netname: IPC$
    remark: Remote IPC
    path:
    password: (null)
netname: NETLOGON
    remark: Logon server share
    path:   C:\Windows\SYSVOL\sysvol\ignite.local\SCRIPTS
    password: (null)
netname: SYSVOL
    remark: Logon server share
    path:   C:\Windows\SYSVOL\sysvol
    password: (null)
netname: Users
    remark:
    path:   C:\Users
    password: (null)

```

## Net Share Get Information

As with the previous commands, the share enumeration command also comes with the feature of targeting a specific entity. The command `netsharegetinfo` followed by the name of the share you are trying to enumerate will extract details about that particular share. This detail includes the path of the share,

remarks, and it will indicate if the share has a password for access. It will also tell the number of users accessing the share and what kind of access is allowed on the share.

**netsharegetinfo Confidential**

```
rpcclient $> netsharegetinfo Confidential
netname: Confidential
remark:
path: C:\Confidential
password: (null)
type: 0x0
perms: 0
max_uses: -1
num_uses: 0
revision: 1
type: 0x8004: SEC_DESC_DACL_PRESENT SEC_DESC_SELF_RELATIVE
DACL
  ACL      Num ACEs:      1      revision:      2
  ---
  ACE
    type: ACCESS_ALLOWED (0) flags: 0x03 SEC_ACE_FLAG_OBJECT_INHERIT SEC_ACE_FLAG_CONTAINER_INHERIT
    Specific bits: 0x1ff
    Permissions: 0x1f01ff: SYNCHRONIZE_ACCESS WRITE_OWNER_ACCESS WRITE_DAC_ACCESS READ_CONTROL_ACCESS DELETE_ACCESS
    SID: S-1-1-0

  Owner SID: S-1-5-32-544
  Group SID: S-1-5-21-501555289-2168925624-2051597760-513
rpcclient $>
```

## Enumerating Domains

In scenarios where there is a possibility of multiple domains in the network, the attacker can use `enumdomains` to enumerate all the domains that might be deployed in that network. In the demonstration presented, there are two domains: IGNITE and Builtin.

**enumdomains**

```
rpcclient $> enumdomains
name:[IGNITE] idx:[0x0]
name:[Builtin] idx:[0x0]
```

## Enumerating Domain Groups

Next, we have two query-oriented commands. These commands can enumerate the users and groups in a domain. Since we already performed the enumeration of such data before in the article, we will enumerate using `enumdomgroup` and `enumdomusers` and the query-oriented commands in this demonstration. When using the `enumdomgroup`, we see that we have different groups with their respective RID and when this RID is used with the `queryusergroups`, it reveals information about that particular holder or RID. In the case of `queryusergroups`, the group will be enumerated. When using `querygroupmem`, it will reveal information about that group member specific to that particular RID.

**enumdomgroups**  
**enumdomusers**  
**queryusersgroups 0x44f**  
**querygroupmem 0x201**

```

rpcclient $> enumdomgroups
group:[Enterprise Read-only Domain Controllers] rid:[0x1f2]
group:[Domain Admins] rid:[0x200]
group:[Domain Users] rid:[0x201]
group:[Domain Guests] rid:[0x202]
group:[Domain Computers] rid:[0x203]
group:[Domain Controllers] rid:[0x204]
group:[Schema Admins] rid:[0x206]
group:[Enterprise Admins] rid:[0x207]
group:[Group Policy Creator Owners] rid:[0x208]
group:[Read-only Domain Controllers] rid:[0x209]
group:[Cloneable Domain Controllers] rid:[0x20a]
group:[Protected Users] rid:[0x20d]
group:[Key Admins] rid:[0x20e]
group:[Enterprise Key Admins] rid:[0x20f]
group:[DnsUpdateProxy] rid:[0x44e]
rpcclient $> enumdomusers
user:[Administrator] rid:[0x1f4]
user:[Guest] rid:[0x1f5]
user:[krbtgt] rid:[0x1f6]
user:[DefaultAccount] rid:[0x1f7]
user:[raj] rid:[0x44f]
user:[john] rid:[0x450]
user:[jeenaly] rid:[0x452]
rpcclient $> queryusergroups 0x44f
group rid:[0x201] attr:[0x7]
rpcclient $> querygroupmem 0x201
rid:[0x1f4] attr:[0x7]
rid:[0x1f7] attr:[0x7]
rid:[0x1f6] attr:[0x7]
rid:[0x44f] attr:[0x7]
rid:[0x450] attr:[0x7]
rid:[0x452] attr:[0x7]
rpcclient $>

```

## Display Query Information

From the enumdomusers command, it was possible to obtain the users of the domain as well as the RID. This information can be elaborated on using the querydispinfo. This will extend the amount of information about the users and their descriptions.

**querydispinfo**

```

rpcclient $> querydispinfo
index: 0xfbc RID: 0x1f4 acb: 0x00000010 Account: Administrator Name: (null) Desc: Built-in account for administering the computer/domain
index: 0xfbe RID: 0x1f7 acb: 0x000000215 Account: DefaultAccount Name: (null) Desc: A user account managed by the system.
index: 0xfbd RID: 0x1f5 acb: 0x000000215 Account: Guest Name: (null) Desc: Built-in account for guest access to the computer/domain
index: 0x109b RID: 0x452 acb: 0x00020010 Account: jeenaly Name: jeenaly Desc: (null)
index: 0x1094 RID: 0x450 acb: 0x000000210 Account: john Name: john Desc: (null)
index: 0xff4 RID: 0x1f6 acb: 0x00020011 Account: krbtgt Name: (null) Desc: Key Distribution Center Service Account
index: 0x1093 RID: 0x44f acb: 0x000000210 Account: raj Name: raj Desc: (null)
rpcclient $>

```



## Change Password of User

From the previous commands, we saw that it is possible to create a user through rpcclient. Depending on the user privilege, it is possible to change the password using the chgpaswd command.

```
chgpaswd raj Password@1 Password@987
```

```
rpcclient $> chgpaswd raj Password@1 Password@987
rpcclient $> █
```

## Create Domain Group

After creating the users and changing their passwords, it's time to manipulate the groups. Using rpcclient it is possible to create a group. The createdomgroup command is to be used to create a group. It accepts the group name as a parameter. After creating the group, it is possible to see the newly created group using the enumdomgroups command.

```
createdomgroup newgroup
enumdomgroups
```

```
rpcclient $> createdomgroup newgroup
rpcclient $> enumdomgroups
group:[Enterprise Read-only Domain Controllers] rid:[0x1f2]
group:[Domain Admins] rid:[0x200]
group:[Domain Users] rid:[0x201]
group:[Domain Guests] rid:[0x202]
group:[Domain Computers] rid:[0x203]
group:[Domain Controllers] rid:[0x204]
group:[Schema Admins] rid:[0x206]
group:[Enterprise Admins] rid:[0x207]
group:[Group Policy Creator Owners] rid:[0x208]
group:[Read-only Domain Controllers] rid:[0x209]
group:[Cloneable Domain Controllers] rid:[0x20a]
group:[Protected Users] rid:[0x20d]
group:[Key Admins] rid:[0x20e]
group:[Enterprise Key Admins] rid:[0x20f]
group:[DnsUpdateProxy] rid:[0x44e]
group:[newgroup] rid:[0x453]
rpcclient $> █
```

## Delete Domain Group

The manipulation of the groups is not limited to the creation of a group. If the permissions allow, an attacker can delete a group as well. The command to be used to delete a group using deletedomgroup. This can be verified using the enumdomgroups command.

```
deletedomgroup newgroup
enumdomgroups
```

```

rpcclient $> deletedomgroup newgroup
rpcclient $> enumdomgroups
group:[Enterprise Read-only Domain Controllers] rid:[0x1f2]
group:[Domain Admins] rid:[0x200]
group:[Domain Users] rid:[0x201]
group:[Domain Guests] rid:[0x202]
group:[Domain Computers] rid:[0x203]
group:[Domain Controllers] rid:[0x204]
group:[Schema Admins] rid:[0x206]
group:[Enterprise Admins] rid:[0x207]
group:[Group Policy Creator Owners] rid:[0x208]
group:[Read-only Domain Controllers] rid:[0x209]
group:[Cloneable Domain Controllers] rid:[0x20a]
group:[Protected Users] rid:[0x20d]
group:[Key Admins] rid:[0x20e]
group:[Enterprise Key Admins] rid:[0x20f]
group:[DnsUpdateProxy] rid:[0x44e]
rpcclient $>

```

## Domain Lookup

We have enumerated the users and groups in the domain but have not enumerated the domain itself. To extract information about the domain, the attacker can provide the domain name as a parameter to the command `lookupdomain`, as demonstrated.

**lookupdomain ignite**

```

rpcclient $> lookupdomain ignite
SAMR_LOOKUP_DOMAIN: Domain Name: ignite Domain SID: S-1-5-21-3232368669-2512470540-2741904768
rpcclient $>

```

## SAM Lookup

Since the user and password-related information is stored inside the SAM file of the server, It is possible to enumerate the SAM data through the `rpcclient` as well. When provided with the username to the `samlookupnames` command, it can extract the RID of that particular user. If the RID is the parameter, the `samlookuprids` command can extract the username relevant to that particular RID.

**samlookupnames domain raj**  
**samlookuprids domain 0x44f**

```

rpcclient $> samlookupnames domain raj
name raj: 0x44f (1)
rpcclient $> samlookuprids domain 0x44f
rid 0x44f: raj (1)
rpcclient $>

```

## SID Lookup

The next command to demonstrate is lookupsids. This command can be used to extract the details regarding the user to whom the SID belongs. In our previous attempt to enumerate SID, we used the lsaenumsid command. That command reveals the SIDs for different users on the domain. To extract further information about that user, or if during the other enumeration the attacker comes into contact with the SID of a user, then they have to use the lookupsids command to get more information about that particular user. In the demonstration, it can be observed that the SID that was enumerated belonged to the Administrator of the Builtin users.

**lsaenumsid**  
**lookupsids S-1-5-32-544**

```
rpcclient $> lsaenumsid
found 16 SIDs

S-1-5-9
S-1-5-80-3139157870-2983391045-3678747466-658725712-1809340420
S-1-5-80-0
S-1-5-6
S-1-5-32-559
S-1-5-32-554
S-1-5-32-551
S-1-5-32-550
S-1-5-32-549
S-1-5-32-548
S-1-5-32-545
S-1-5-32-544
S-1-5-20
S-1-5-19
S-1-5-11
S-1-1-0
rpcclient $> lookupsids S-1-5-32-544
S-1-5-32-544 BUILTIN\Administrators (4)
rpcclient $>
```

## LSA Query

The next command that can help with the enumeration is lsaquery. This command can help with the enumeration of the LSA Policy for that particular domain. In the demonstration, it can be observed that a query was generated for LSA which returned with information such as Domain Name and SID. Similarly, to enumerate the Primary Domain Information such as the role of the machine, native more of the domain can be done using the dsroledominfo command as demonstrated.

**lsaquery**  
**dsroledominfo**

```
rpcclient $> lsquery
Domain Name: IGNITE
Domain Sid: S-1-5-21-3232368669-2512470540-2741904768
rpcclient $> dsroledominfo
Machine Role = [5]
Directory Service is running.
Domain is in native mode.
rpcclient $> █
```

## LSA Create Account

An attacker can create an account object based on the SID of that user. For this particular demonstration, we will first need a SID. This can be extracted using the lookupnames command used earlier. Passing the SID as a parameter in the lsacreateaccount command will enable us as an attacker to create an account object as shown in the image below.

```
lookupnames raj
lsacreateaccount S-1-5-21-3232368669-2512470540-2741904768-1103
```

```
rpcclient $> lookupnames raj
raj S-1-5-21-3232368669-2512470540-2741904768-1103 (User: 1)
rpcclient $> lsacreateaccount S-1-5-21-3232368669-2512470540-2741904768-1103
Account for SID S-1-5-21-3232368669-2512470540-2741904768-1103 successfully created
rpcclient $> █
```

## Enumerating LSA Group Privileges

During our previous demonstrations, we were able to enumerate the permissions and privileges of users and groups based on the RID of that particular user. It is possible to perform an enumeration regarding the privileges of a group or a user based on their SID as well. To do this first, the attacker needs a SID. This can be obtained by running the lsenumsid command. In the demonstration below, the attacker chooses S-1-1-0 SID to enumerate. When it was passed as a parameter in the command lookupsids, the attacker was able to know that this belongs to the group Everyone. Further, when the attacker used the same SID as a parameter for lsenumprivsaccount, they were able to enumerate the levels of privileges such as high, low, and attribute. Then the attacker used the SID to enumerate the privileges using the lsenumacctrights command. This command was able to enumerate two specific privileges, such as SeChangeNotifyPrivilege and the SeNetworkLogonRight privilege.

```
lsenumsid
lookupsids S-1-1-0
lsenumprivsaccount S-1-1-0
lsenumacctrights S-1-1-0
```

```

rpcclient $> lsenumsid
found 17 SIDs

S-1-5-9
S-1-5-80-3139157870-2983391045-3678747466-658725712-1809340420
S-1-5-80-0
S-1-5-6
S-1-5-32-559
S-1-5-32-554
S-1-5-32-551
S-1-5-32-550
S-1-5-32-549
S-1-5-32-548
S-1-5-32-545
S-1-5-32-544
S-1-5-21-3232368669-2512470540-2741904768-1103
S-1-5-20
S-1-5-19
S-1-5-11
S-1-1-0
rpcclient $> lookupsids S-1-1-0
S-1-1-0 \Everyone (5)
rpcclient $> lsenumprivsaccount S-1-1-0
found 1 privileges for SID S-1-1-0

high    low    attribute
0       23      3
rpcclient $> lsenumacctrights S-1-1-0
found 2 privileges for SID S-1-1-0
SeChangeNotifyPrivilege
SeNetworkLogonRight
rpcclient $>

```

The ability to interact with privileges doesn't end with the enumeration regarding the SID or privileges. It is also possible to manipulate the privileges of that SID to make them either vulnerable to a particular privilege or remove the privilege of a user altogether. To demonstrate this, the attacker first used the `Isaaddpriv` command to add the `SeCreateTokenPrivilege` to the SID and then used the `Isadelpriv` command to remove that privilege from that group as well. All this can be observed in the usage of the `Isaenumprivsaccount` command.

```

Isaaddpriv S-1-1-0 SeCreateTokenPrivilege
Isaenumprivsaccount S-1-1-0
Isadelpriv S-1-1-0 SeCreateTokenPrivilege
Isaenumprivsaccount S-1-1-0

```

```

rpcclient $> lsaddpriv S-1-1-0 SeCreateTokenPrivilege
rpcclient $> lsenumprivsaccount S-1-1-0
found 2 privileges for SID S-1-1-0

high    low    attribute
0       23     3
0       2      0

rpcclient $> lsdelpriv S-1-1-0 SeCreateTokenPrivilege
rpcclient $> lsenumprivsaccount S-1-1-0
found 1 privileges for SID S-1-1-0

high    low    attribute
0       23     3
rpcclient $>

```

## Enumerating LSA Account Privileges

In the previous demonstration, the attacker was able to provide and remove privileges to a group. It is also possible to add and remove privileges to a specific user as well. The `lsaaddacctrights` command can be used to add privileges to a user based on their SID. The SID was retrieved using the `lookupnames` command. After verifying that the privilege was added using the `lsaenumprivsaccount` command, we removed the privileges from the user using the `lsaremoveacctrights` command.

**lookupnames raj**

**lsaaddacctrights S-1-5-21-3232368669-2512470540-2741904768-1103 SeCreateTokenPrivilege**

**lsaenumprivsaccount S-1-5-21-3232368669-2512470540-2741904768-1103**

**lsaremoveacctrights S-1-5-21-3232368669-2512470540-2741904768-1103 SeCreateTokenPrivilege**

**lsaenumprivsaccount S-1-5-21-3232368669-2512470540-2741904768-1103**

```

rpcclient $> lookupnames raj
raj S-1-5-21-3232368669-2512470540-2741904768-1103 (User: 1)
rpcclient $> lsaaddacctrights S-1-5-21-3232368669-2512470540-2741904768-1103 SeCreateTokenPrivilege
rpcclient $> lsaenumprivsaccount S-1-5-21-3232368669-2512470540-2741904768-1103
found 1 privileges for SID S-1-5-21-3232368669-2512470540-2741904768-1103

high    low    attribute
0       2      0

rpcclient $> lsaremoveacctrights S-1-5-21-3232368669-2512470540-2741904768-1103 SeCreateTokenPrivilege
rpcclient $> lsaenumprivsaccount S-1-5-21-3232368669-2512470540-2741904768-1103
result was NT_STATUS_OBJECT_NAME_NOT_FOUND
rpcclient $>

```

After manipulating the privileges of the different users and groups, it is possible to enumerate the values of those specific privileges for a particular user using the `lsalookupprivvalue` command.

**lsalookupprivvalue SeCreateTokenPrivilege**

```
rpcclient $> lsalookupprivvalue SeCreateTokenPrivilege
0:2 (0x0:0x2)
rpcclient $>
```

## LSA Query Security Objects

The next command to observe is the `lsaquerysecobj` command. This command is made from the LSA Query Security Object. This command helps the attacker enumerate the security objects or permissions and privileges related to the security as demonstrated below.

`lsaquerysecobj`

```
rpcclient $> lsaquerysecobj
revision: 1
type: 0x8004: SEC_DESC_DACL_PRESENT SEC_DESC_SELF_RELATIVE
DACL
  ACL      Num ACEs:      8      revision:      2
  ---
  ACE
    type: ACCESS DENIED (1) flags: 0x00
    Specific bits: 0x800
    Permissions: 0x800:
    SID: S-1-5-7

  ACE
    type: ACCESS ALLOWED (0) flags: 0x00
    Specific bits: 0x1fff
    Permissions: 0xf1fff: WRITE_OWNER_ACCESS WRITE_DAC_ACCESS READ_CONTROL
    SID: S-1-5-32-544

  ACE
    type: ACCESS ALLOWED (0) flags: 0x00
    Specific bits: 0x801
    Permissions: 0x20801: READ_CONTROL_ACCESS
    SID: S-1-1-0

  ACE
    type: ACCESS ALLOWED (0) flags: 0x00
    Specific bits: 0x801
    Permissions: 0x801:
    SID: S-1-5-7

  ACE
    type: ACCESS ALLOWED (0) flags: 0x00
    Specific bits: 0x1000
    Permissions: 0x1000:
    SID: S-1-5-19
```

## Conclusion

Using the `rpcclient` tool, we were able to enumerate a wide range of information through the SMB and RPC channels inside a domain. This can serve as a reference for Red Team activists for attacking and enumerating the domain, but it can also be helpful for the Blue Team to understand and test the measures applied to the domain to protect the network and its users.