2. Active Directory LDAP

Active Directory Overview

`Active Directory` (`AD`) is a directory service for Windows network environments. It is a distributed, hierarchical structure that allows for centralized management of an organization's resources, including users, computers, groups, network devices and file shares, group policies, servers and workstations, and trusts. AD provides authentication and authorization functions within a Windows domain environment. It was first shipped with Windows Server 2000; it has come under increasing attack in recent years. Designed to be backward-compatible, and many features are arguably not "secure by default," and it can be easily misconfigured.

This can be leveraged to move laterally and vertically within a network and gain unauthorized access. AD is essentially a large database accessible to all users within the domain, regardless of their privilege level. A basic AD user account with no added privileges can be used to enumerate the majority of objects contained within AD, including but not limited to:

- Domain Computers
- Domain Users
- Domain Group Information
- Default Domain Policy
- Domain Functional Levels
- Password Policy
- Group Policy Objects (GPOs)
- Kerberos Delegation
- Domain Trusts
- Access Control Lists (ACLs)

This data will paint a clear picture of the overall security posture of an Active Directory environment. It can be used to quickly identify misconfigurations, overly permissive policies, and other ways of escalating privileges within an AD environment. Many attacks exist that merely leverage AD misconfigurations, bad practices, or poor administration, such as:

- Kerberoasting / ASREPRoasting
- NTLM Relaying
- Network traffic poisoning
- Password spraying

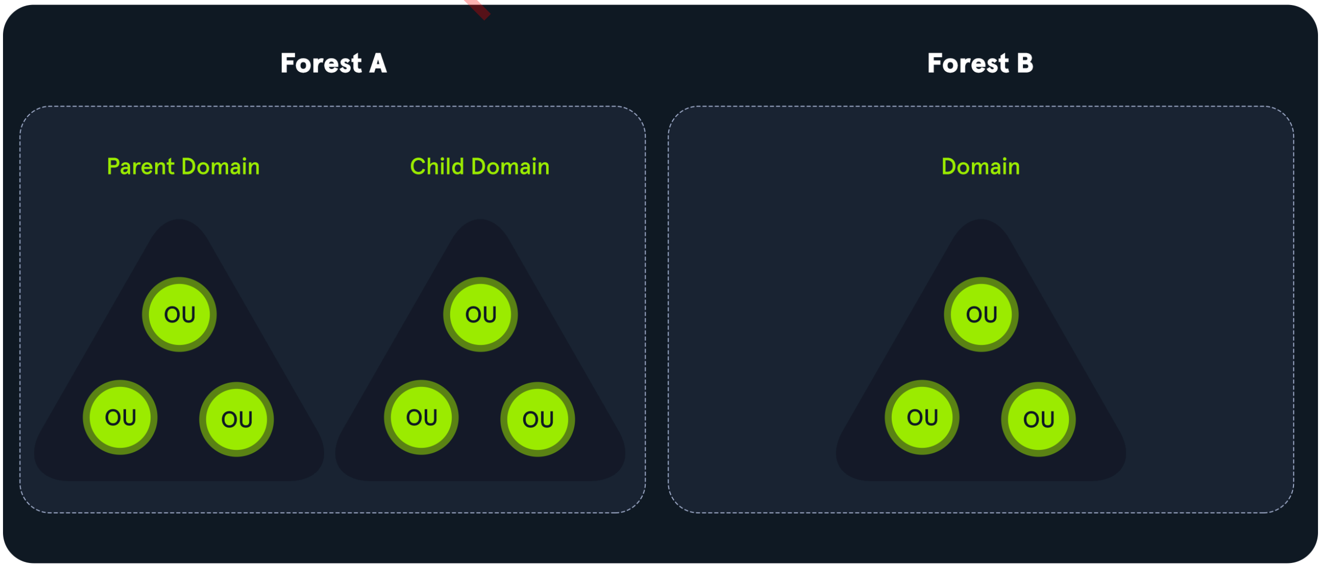
- Kerberos delegation abuse
- Domain trust abuse
- Credential theft
- Object control

Hardening Active Directory, along with a strong patching and configuration management policy, and proper network segmentation should be prioritized. If an environment is tightly managed and an adversary can gain a foothold and bypass EDR or other protections, proper management of AD can prevent them from escalating privileges, moving laterally, and getting to the crown jewels. Proper controls will help slow down an attacker and potentially force them to become noisier and risk detection.

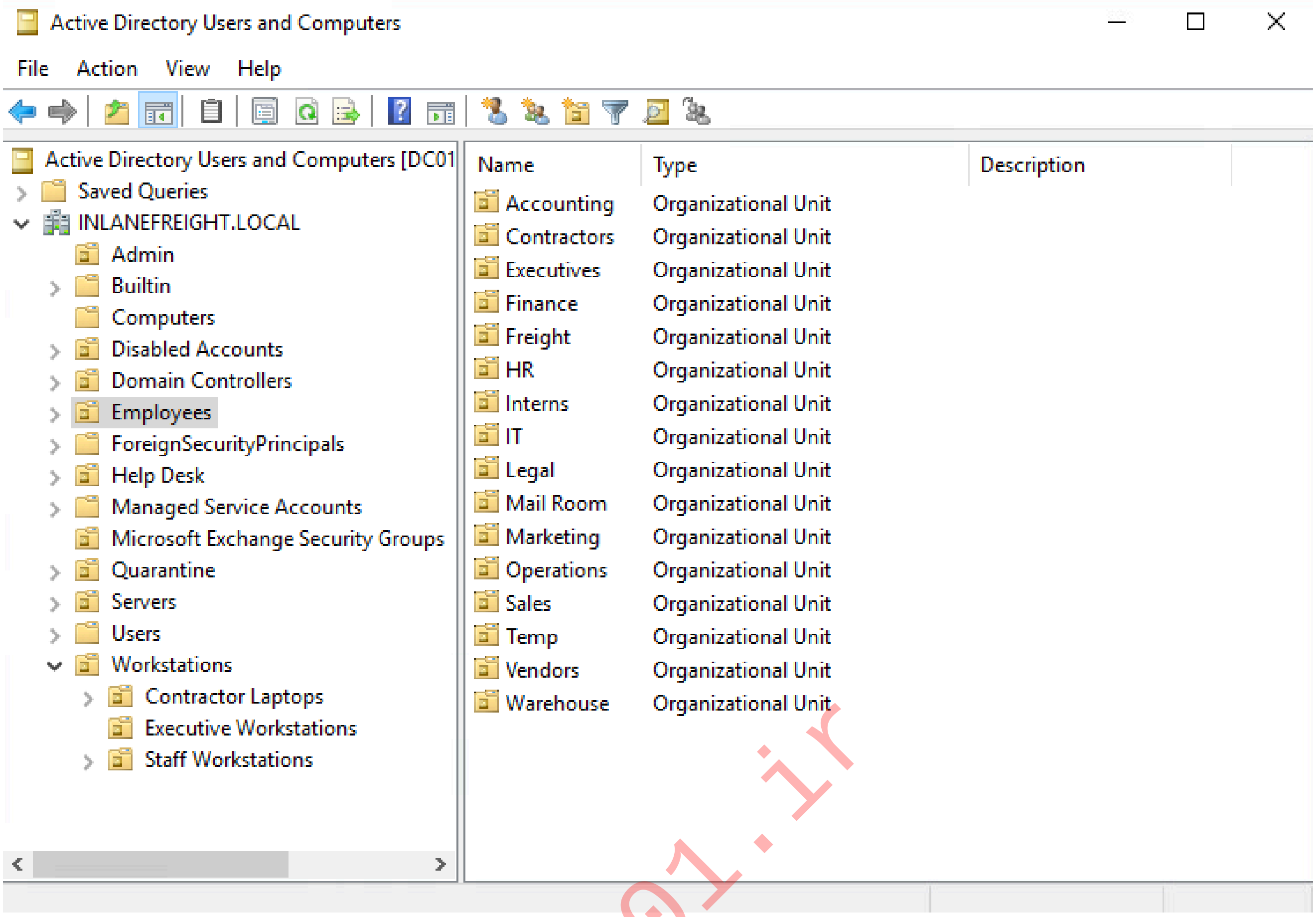
Active Directory Structure

Active Directory is arranged in a hierarchical tree structure, with a forest at the top containing one or more domains, which can themselves contain nested subdomains. A forest is the **security boundary** within which all objects are under administrative control. A forest may contain multiple domains, and a domain may contain further child or sub-domains. A domain is a structure within which contained objects (users, computers, and groups) are accessible. Objects are the most basic unit of data in AD.

It contains many built-in `Organizational Units` (`OU` s), such as “Domain Controllers,” “Users,” and “Computers,” and new `OU` s can be created as required. `OU` s may contain objects and sub-OUs, allowing for assignment of different group policies.



We can see this structure graphically by opening `Active Directory Users and Computers` on a Domain Controller. In our lab domain `INLANEFREIGHT.LOCAL`, we see various OUs such as `Admin`, `Employees`, `Servers`, `Workstations`, etc. Many of these OUs have OUs nested within them, such as the `Mail Room` OU under `Employees`. This helps maintain a clear and coherent structure within Active Directory, which is especially important as we add Group Policy Objects (GPOs) to enforce settings throughout the domain.



```
Active Directory Users and Computers [DC01
  > Saved Queries
  v INLANEFREIGHT.LOCAL
      Admin
    > Builtin
      Computers
    > Disabled Accounts
    > Domain Controllers
    > Employees
    > ForeignSecurityPrincipals
    > Help Desk
    > Managed Service Accounts
      Microsoft Exchange Security Groups
    > Quarantine
    > Servers
    > Users
    v Workstations
      > Contractor Laptops
        Executive Workstations
      > Staff Workstations
```

Name	Type	Description
Accounting	Organizational Unit	
Contractors	Organizational Unit	
Executives	Organizational Unit	
Finance	Organizational Unit	
Freight	Organizational Unit	
HR	Organizational Unit	
Interns	Organizational Unit	
IT	Organizational Unit	
Legal	Organizational Unit	
Mail Room	Organizational Unit	
Marketing	Organizational Unit	
Operations	Organizational Unit	
Sales	Organizational Unit	
Temp	Organizational Unit	
Vendors	Organizational Unit	
Warehouse	Organizational Unit	

Understanding the structure of Active Directory is paramount to perform proper enumeration and uncover the flaws and misconfigurations that sometimes have gone missed in an environment for many years.

Module Exercises

Throughout this module, you will connect to various target hosts via the Remote Desktop Protocol (RDP) to complete the exercises. Any necessary credentials will be provided with each exercise, and the RDP connection can be made via `xfreerdp` from the Pwnbox as follows:

```
xfreerdp /v:<target IP address> /u:htb-student /p:<password> /cert-ignore
```

Any necessary tools can be found in the `c:\tools` directory after logging in to the target host.

Why Enumerate AD?

As penetration testers, `enumeration` is one of, if not the most important, skills we must master. When starting an assessment in a new network gaining a comprehensive inventory of the environment is extremely important. The information gathered during this phase will inform our later attacks and even post-exploitation. Given the prevalence of AD in corporate networks, we will likely find ourselves in AD environments regularly, and therefore, it is important to hone our enumeration process. There are many tools and techniques to help with AD enumeration, which we will cover in-depth in this module and subsequent modules; however, before using these tools, it is important to understand the reason for performing detailed AD enumeration.

Whether we perform a penetration test or targeted AD assessment, we can always go above and beyond and provide our clients with extra value by giving them a detailed picture of their AD strengths and weaknesses. Corporate environments go through many changes over the years, adding and removing employees and hosts, installing software and applications that require changes in AD, or corporate policies that require GPO changes. These changes can introduce security flaws through misconfiguration, and it is our job as assessors to find these flaws, exploit them, and help our clients fix them.

Getting Started

Once we have a foothold in an AD environment, we should start by gathering several key pieces of information, including but not limited to:

- The domain functional level
- The domain password policy
- A full inventory of AD users
- A full inventory of AD computers
- A full inventory of AD groups and memberships
- Domain trust relationships
- Object ACLs
- Group Policy Objects (GPO) information
- Remote access rights

With this information in hand, we can look for any "quick wins" such as our current user or the entire `Domain Users` group having RDP and/or local administrator access to one or more hosts. This is common in large environments for many reasons, one being the improper use of jump hosts and another being Citrix server Remote Desktop Services (RDS) misconfigurations. We should also check what rights our current user has in the domain. Are they a member of any privileged groups? Do they have any special rights delegated? Do they have any control over another domain object such as a user, computer, or GPO?

The enumeration process is iterative. As we move through the AD environment, compromising hosts and users, we will need to perform additional enumeration to see if we have gained any further access to help us reach our goal.

Rights and Privileges in AD

AD contains many groups that grant their members powerful rights and privileges. Many of these can be abused to escalate privileges within a domain and ultimately gain Domain Admin or SYSTEM privileges on a Domain Controller (DC). Some of these groups are listed below.

Group	Description
Default Administrators	Domain Admins and Enterprise Admins "super" groups.
Server Operators	Members can modify services, access SMB shares, and backup files.
Backup Operators	Members are allowed to log onto DCs locally and should be considered Domain Admins. They can make shadow copies of the SAM/NTDS database, read the registry remotely, and access the file system on the DC via SMB. This group is sometimes added to the local Backup Operators group on non-DCs.
Print Operators	Members are allowed to logon to DCs locally and "trick" Windows into loading a malicious driver.
Hyper-V Administrators	If there are virtual DCs, any virtualization admins, such as members of Hyper-V Administrators, should be considered Domain Admins.
Account Operators	Members can modify non-protected accounts and groups in the domain.
Remote Desktop Users	Members are not given any useful permissions by default but are often granted additional rights such as *Allow Login Through Remote Desktop Services* and can move laterally using the RDP protocol.
Remote Management Users	Members are allowed to logon to DCs with PSRemoting (This group is sometimes added to the local remote management group on non-DCs).
Group Policy Creator Owners	Members can create new GPOs but would need to be delegated additional permissions to link GPOs to a container such as a domain or OU.
Schema Admins	Members can modify the Active Directory schema structure and can backdoor any to-be-created Group/GPO by adding a compromised account to the default object ACL.

Group	Description
DNS Admins	Members have the ability to load a DLL on a DC but do not have the necessary permissions to restart the DNS server. They can load a malicious DLL and wait for a reboot as a persistence mechanism. Loading a DLL will often result in the service crashing. A more reliable way to exploit this group is to create a WPAD record.

Members of "Schema Admins"

```
PS C:\htb> Get-ADGroup -Identity "Schema Admins" -Properties *

adminCount                          : 1
CanonicalName                       : INLANEFREIGHT.LOCAL/Users/Schema Admins
CN                                  : Schema Admins
Created                             : 7/26/2020 4:14:37 PM
createTimeStamp                     : 7/26/2020 4:14:37 PM
Deleted                             :
Description                         : Designated administrators of the schema
DisplayName                         :
DistinguishedName                   : CN=Schema
Admins,CN=Users,DC=INLANEFREIGHT,DC=LOCAL
dSCorePropagationData               : {7/29/2020 11:52:30 PM, 7/29/2020
11:09:16 PM, 7/27/2020 9:45:00 PM, 7/27/2020
                                      9:34:13 PM...}
GroupCategory                       : Security
GroupScope                          : Universal
groupType                           : -2147483640
HomePage                            :
instanceType                        : 4
isCriticalSystemObject              : True
isDeleted                           :
LastKnownParent                     :
ManagedBy                           :
member                              : {CN=Jenna Smith,OU=Server
Team,OU=IT,OU=Employees,DC=INLANEFREIGHT,DC=LOCAL,

CN=Administrator,CN=Users,DC=INLANEFREIGHT,DC=LOCAL}
MemberOf                            : {CN=Denied RODC Password Replication
Group,CN=Users,DC=INLANEFREIGHT,DC=LOCAL}
Members                             : {CN=Jenna Smith,OU=Server
Team,OU=IT,OU=Employees,DC=INLANEFREIGHT,DC=LOCAL,

CN=Administrator,CN=Users,DC=INLANEFREIGHT,DC=LOCAL}
Modified                            : 7/30/2020 2:04:05 PM
modifyTimeStamp                     : 7/30/2020 2:04:05 PM
Name                                : Schema Admins
nTSecurityDescriptor                :
```

```
System.DirectoryServices.ActiveDirectorySecurity
ObjectCategory                  :
CN=Group,CN=Schema,CN=Configuration,DC=INLANEFREIGHT,DC=LOCAL
ObjectClass                     : group
ObjectGUID                      : 36eef5cb-92b1-47d2-a25d-b9d73783ed1e
objectSid                       : S-1-5-21-2974783224-3764228556-
2640795941-518
ProtectedFromAccidentalDeletion : False
SamAccountName                  : Schema Admins
sAMAccountType                  : 268435456
sDRightsEffective               : 15
SID                             : S-1-5-21-2974783224-3764228556-
2640795941-518
SIDHistory                      : {}
uSNChanged                      : 66825
uSNCreated                      : 12336
whenChanged                     : 7/30/2020 2:04:05 PM
whenCreated                     : 7/26/2020 4:14:37 PM
```

User Rights Assignment

Depending on group membership, and other factors such as privileges assigned via Group Policy, users can have various rights assigned to their account. This Microsoft article on User Rights Assignment provides a detailed explanation of each of the user rights that can be set in Windows.

Typing the command `whoami /priv` will give you a listing of all user rights assigned to your current user. Some rights are only available to administrative users and can only be listed/leveraged when running an elevated cmd or PowerShell session. These concepts of elevated rights and User Account Control (UAC) are security features introduced with Windows Vista to default to restricting applications from running with full permissions unless absolutely necessary. If we compare and contrast the rights available to us as an admin in a non-elevated console vs. an elevated console, we will see that they differ drastically. Let's try this out as the `htb-student` user on the lab machine.

Below are the rights available to a Domain Admin user.

User Rights Non-Elevated

We can see the following in a non-elevated console:

```
PS C:\htb> whoami /priv

PRIVILEGES INFORMATION
```

```
----------------------

Privilege Name                Description                          State
============================= ====================================
========
SeShutdownPrivilege           Shut down the system
Disabled
SeChangeNotifyPrivilege       Bypass traverse checking             Enabled
SeUndockPrivilege             Remove computer from docking station
Disabled
SeIncreaseWorkingSetPrivilege Increase a process working set
Disabled
SeTimeZonePrivilege           Change the time zone
Disabled
```

User Rights Elevated

If we run an elevated command (our htb-student user has local admin rights via nested group membership; the Domain Users group is in the local Administrators group), we can see the complete listing of rights available to us:

```
PS C:\htb> whoami /priv

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

Privilege Name                            Description
State
=========================================
==================================================================
========
SeIncreaseQuotaPrivilege                  Adjust memory quotas for a
process                                 Disabled
SeMachineAccountPrivilege                 Add workstations to domain
Disabled
SeSecurityPrivilege                       Manage auditing and security log
Disabled
SeTakeOwnershipPrivilege                  Take ownership of files or other
objects                           Disabled
SeLoadDriverPrivilege                     Load and unload device drivers
Disabled
SeSystemProfilePrivilege                  Profile system performance
Disabled
SeSystemtimePrivilege                     Change the system time
Disabled
SeProfileSingleProcessPrivilege           Profile single process
Disabled
```

```
SeIncreaseBasePriorityPrivilege           Increase scheduling priority
Disabled
SeCreatePagefilePrivilege                 Create a pagefile
Disabled
SeBackupPrivilege                         Back up files and directories
Disabled
SeRestorePrivilege                        Restore files and directories
Disabled
SeShutdownPrivilege                       Shut down the system
Disabled
SeDebugPrivilege                          Debug programs
Enabled
SeSystemEnvironmentPrivilege              Modify firmware environment
values                                 Disabled
SeChangeNotifyPrivilege                   Bypass traverse checking
Enabled
SeRemoteShutdownPrivilege                 Force shutdown from a remote
system                                Disabled
SeUndockPrivilege                         Remove computer from docking
station                               Disabled
SeEnableDelegationPrivilege               Enable computer and user
accounts to be trusted for delegation     Disabled
SeManageVolumePrivilege                   Perform volume maintenance tasks
Disabled
SeImpersonatePrivilege                    Impersonate a client after
authentication                          Enabled
SeCreateGlobalPrivilege                   Create global objects
Enabled
SeIncreaseWorkingSetPrivilege             Increase a process working set
Disabled
SeTimeZonePrivilege                       Change the time zone
Disabled
SeCreateSymbolicLinkPrivilege             Create symbolic links
Disabled
SeDelegateSessionUserImpersonatePrivilege Obtain an impersonation token
for another user in the same session Disabled
```

A standard domain user, in contrast, has drastically fewer rights.

Domain User Rights

```
PS C:\htb> whoami /priv

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

Privilege Name                Description                    State
============================= ============================== ========
SeChangeNotifyPrivilege       Bypass traverse checking       Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set Disabled
```

User rights increase based on the groups they are placed in and/or their assigned privileges. Below is an example of the rights granted to users in the `Backup Operators` group. Users in this group do have other rights that are currently restricted by UAC. Still, we can see from this command that they have the `SeShutdownPrivilege`, which means that they can shut down a domain controller that could cause a massive service interruption should they log onto a domain controller locally (not via RDP or WinRM).

Backup Operator Rights

```
PS C:\htb> whoami /priv

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

Privilege Name                Description                    State
============================= ============================== ========
SeShutdownPrivilege           Shut down the system           Disabled
SeChangeNotifyPrivilege       Bypass traverse checking       Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set Disabled
```

As attackers and defenders, we need to review the membership of these groups. It's not uncommon to find seemingly low privileged users added to one or more of these groups, which can be used to further access or compromise the domain.

Note: When spawning your target, we ask you to wait for 3 minutes until the whole lab with all the configurations is set up so that the connection to your target works flawlessly.

Microsoft Remote Server Administration Tools (RSAT)

RSAT Background

The `Remote Server Administration Tools` (`RSAT`) have been part of Windows since the days of Windows 2000. RSAT allows systems administrators to remotely manage Windows Server roles and features from a workstation running Windows 10, Windows 8.1, Windows 7, or Windows Vista. `RSAT` can only be installed on Professional or Enterprise editions of Windows. In an enterprise environment, RSAT can remotely manage Active Directory, DNS, and DHCP. RSAT also allows us to manage installed server roles and features, File Services, and Hyper-V. The full listing of tools included with `RSAT` is:

- SMTP Server Tools
- Hyper-V Management Tools
- Hyper-V Module for Windows PowerShell
- Hyper-V GUI Management Tools
- Windows Server Update Services Tools
- API and PowerShell cmdlets
- User Interface Management Console
- Active Directory Users and Computers Snap-in
- Active Directory Sites and Services Snap-in
- Active Directory Domains and Trusts Snap-in
- Active Directory Administrative Center Snap-in
- ADSI Edit Snap-in
- Active Directory Schema Snap-in (Not Registered)
- Active Directory Command Line Tools
- Active Directory Module for Windows PowerShell
- IIS Management Tools
- IIS Management Console
- IIS Management Compatibility
- Feature Tools
- Remote Desktop Services Tools
- Role Tools
- Update Services Tools
- Group Policy Tools

This script can be used to install RSAT in Windows 10 1809, 1903, and 1909. Installation instructions for other versions of Windows, as well as additional information about RSAT, can be found here. RSAT can be installed easily with PowerShell as well.

We can check which, if any, RSAT tools are installed using PowerShell.

PowerShell - Available RSAT Tools

```
PS C:\htb>  Get-WindowsCapability -Name RSAT* -Online | Select-Object -Property Name, State

Name                                                            State
----                                                            -----
Rsat.ActiveDirectory.DS-LDS.Tools~~~~0.0.1.0               NotPresent
Rsat.BitLocker.Recovery.Tools~~~~0.0.1.0                   NotPresent
Rsat.CertificateServices.Tools~~~~0.0.1.0                  NotPresent
```

```
Rsat.DHCP.Tools~~~~0.0.1.0                               NotPresent
Rsat.Dns.Tools~~~~0.0.1.0                                NotPresent
Rsat.FailoverCluster.Management.Tools~~~~0.0.1.0         NotPresent
Rsat.FileServices.Tools~~~~0.0.1.0                       NotPresent
Rsat.GroupPolicy.Management.Tools~~~~0.0.1.0             NotPresent
Rsat.IPAM.Client.Tools~~~~0.0.1.0                        NotPresent
Rsat.LLDP.Tools~~~~0.0.1.0                               NotPresent
Rsat.NetworkController.Tools~~~~0.0.1.0                  NotPresent
Rsat.NetworkLoadBalancing.Tools~~~~0.0.1.0               NotPresent
Rsat.RemoteAccess.Management.Tools~~~~0.0.1.0            NotPresent
Rsat.RemoteDesktop.Services.Tools~~~~0.0.1.0             NotPresent
Rsat.ServerManager.Tools~~~~0.0.1.0                      NotPresent
Rsat.Shielded.VM.Tools~~~~0.0.1.0                        NotPresent
Rsat.StorageMigrationService.Management.Tools~~~~0.0.1.0 NotPresent
Rsat.StorageReplica.Tools~~~~0.0.1.0                     NotPresent
Rsat.SystemInsights.Management.Tools~~~~0.0.1.0          NotPresent
Rsat.VolumeActivation.Tools~~~~0.0.1.0                   NotPresent
Rsat.WSUS.Tools~~~~0.0.1.0                               NotPresent
```

From here, we can choose to install all available tools using the following command:

PowerShell - Install All Available RSAT Tools

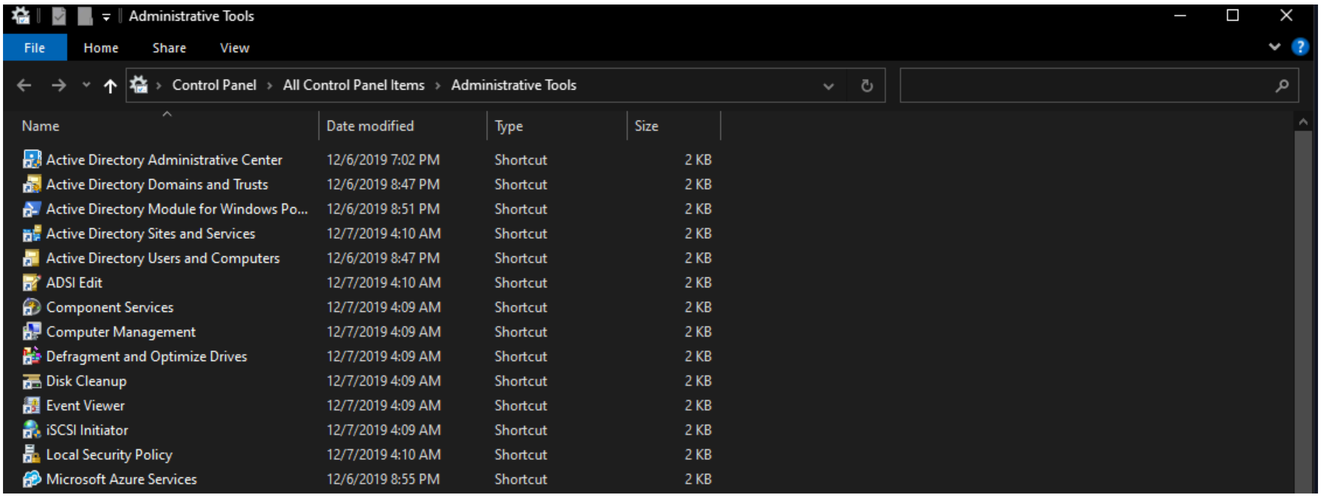
```
PS C:\htb> Get-WindowsCapability -Name RSAT* -Online | Add-WindowsCapability –Online
```

We can also install tools one at a time as needed.

PowerShell - Install an RSAT Tool

```
PS C:\htb> Add-WindowsCapability -Name Rsat.ActiveDirectory.DS-LDS.Tools~~~~0.0.1.0  –Online
```

Once installed, all of the tools will be available under `Administrative Tools` in the `Control Panel`.


Administrative Tools
File
Home
Share
View
Control Panel › All Control Panel Items › Administrative Tools

Name	Date modified	Type	Size
Active Directory Administrative Center	12/6/2019 7:02 PM	Shortcut	2 KB
Active Directory Domains and Trusts	12/6/2019 8:47 PM	Shortcut	2 KB
Active Directory Module for Windows Po...	12/6/2019 8:51 PM	Shortcut	2 KB
Active Directory Sites and Services	12/7/2019 4:10 AM	Shortcut	2 KB
Active Directory Users and Computers	12/6/2019 8:47 PM	Shortcut	2 KB
ADSI Edit	12/7/2019 4:10 AM	Shortcut	2 KB
Component Services	12/7/2019 4:09 AM	Shortcut	2 KB
Computer Management	12/7/2019 4:09 AM	Shortcut	2 KB
Defragment and Optimize Drives	12/7/2019 4:09 AM	Shortcut	2 KB
Disk Cleanup	12/7/2019 4:09 AM	Shortcut	2 KB
Event Viewer	12/7/2019 4:09 AM	Shortcut	2 KB
iSCSI Initiator	12/7/2019 4:09 AM	Shortcut	2 KB
Local Security Policy	12/7/2019 4:10 AM	Shortcut	2 KB
Microsoft Azure Services	12/6/2019 8:55 PM	Shortcut	2 KB

Domain Context for Enumeration

Many tools are missing credential and context parameters and instead get those values directly from the current context. There are a few ways to alter a user's context in Windows if you have access to a password or a hash, such as:

Using " `runas /netonly` " to leverage the built-in runas.exe command line tool.

CMD - Runas User

```
C:\htb> runas /netonly /user:htb.local\jackie.may powershell
```

Other tools that we will discuss in later modules, such as Rubeus and mimikatz can be passed cleartext credentials or an NTLM password hash.

CMD - Rubeus.exe Cleartext Credentials

```
C:\htb> rubeus.exe asktgt /user:jackie.may /domain:htb.local
/dc:10.10.110.100 /rc4:ad11e823e1638def97afa7cb08156a94
```

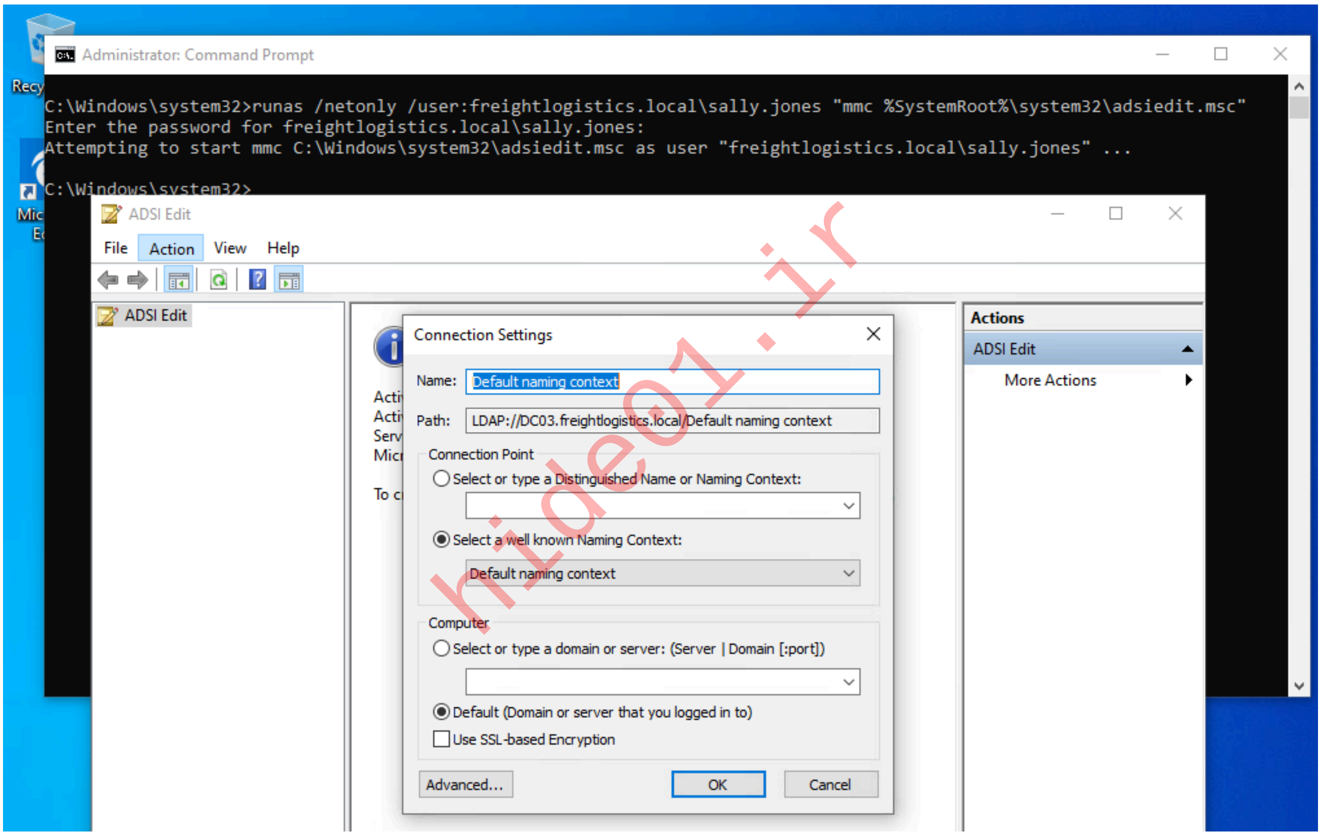
CMD - Mimikatz.exe Cleartext Credentials

```
C:\htb> mimikatz.exe sekurlsa::pth /domain:htb.local /user:jackie.may
/rc4:ad11e823e1638def97afa7cb08156a94
```

Enumeration with RSAT

If we compromise a domain-joined system (or a client has you perform an AD assessment from one of their workstations), we can leverage RSAT to enumerate AD. While RSAT will make GUI tools such as `Active Directory Users and Computers` and `ADSI Edit` available to us, the most important tool we have seen throughout this module is the PowerShell [Active Directory module](#).

Alternatively, we can enumerate the domain from a non-domain joined host (provided that it is in a subnet that communicates with a domain controller) by launching any RSAT snap-ins using " `runas` " from the command line. This is particularly useful if we find ourselves performing an internal assessment, gain valid AD credentials, and would like to perform enumeration from a Windows VM.



```
C:\Windows\system32>runas /netonly /user:freightlogistics.local\sally.jones "mmc %SystemRoot%\system32\adsiedit.msc"
Enter the password for freightlogistics.local\sally.jones:
Attempting to start mmc C:\Windows\system32\adsiedit.msc as user "freightlogistics.local\sally.jones" ...

C:\Windows\system32>
```

Name: Default naming context

Path: LDAP://DC03.freightlogistics.local/Default naming context

Connection Point

○ Select or type a Distinguished Name or Naming Context:

● Select a well known Naming Context: Default naming context

Computer

○ Select or type a domain or server: (Server | Domain [:port])

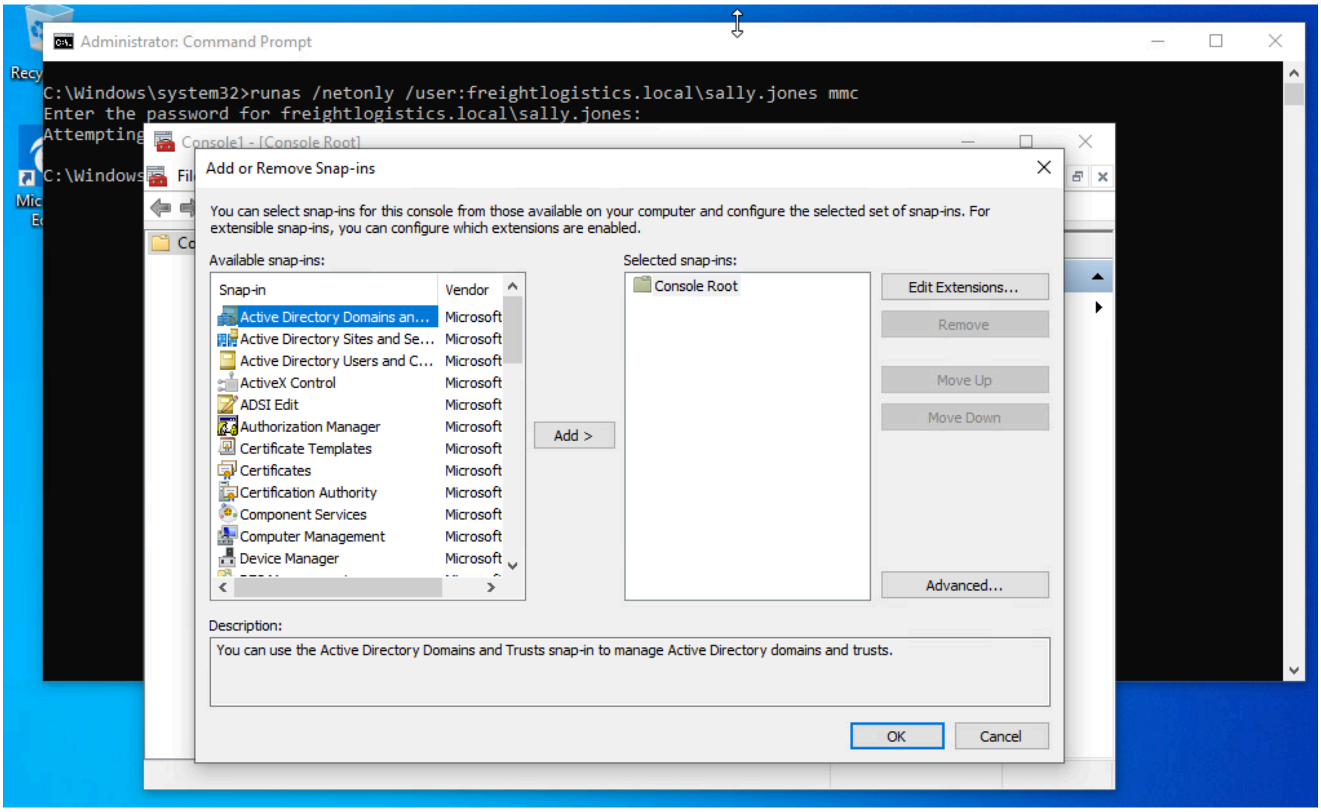
● Default (Domain or server that you logged in to)

□ Use SSL-based Encryption

We can also open the `MMC Console` from a non-domain joined computer using the following command syntax:

CMD - MMC Runas Domain User

```
C:\htb> runas /netonly /user:Domain_Name\Domain_USER mmc
```



```
C:\Windows\system32>runas /netonly /user:freightlogistics.local\sally.jones mmc
Enter the password for freightlogistics.local\sally.jones:
Attempting
C:\Windows
```

Add or Remove Snap-ins

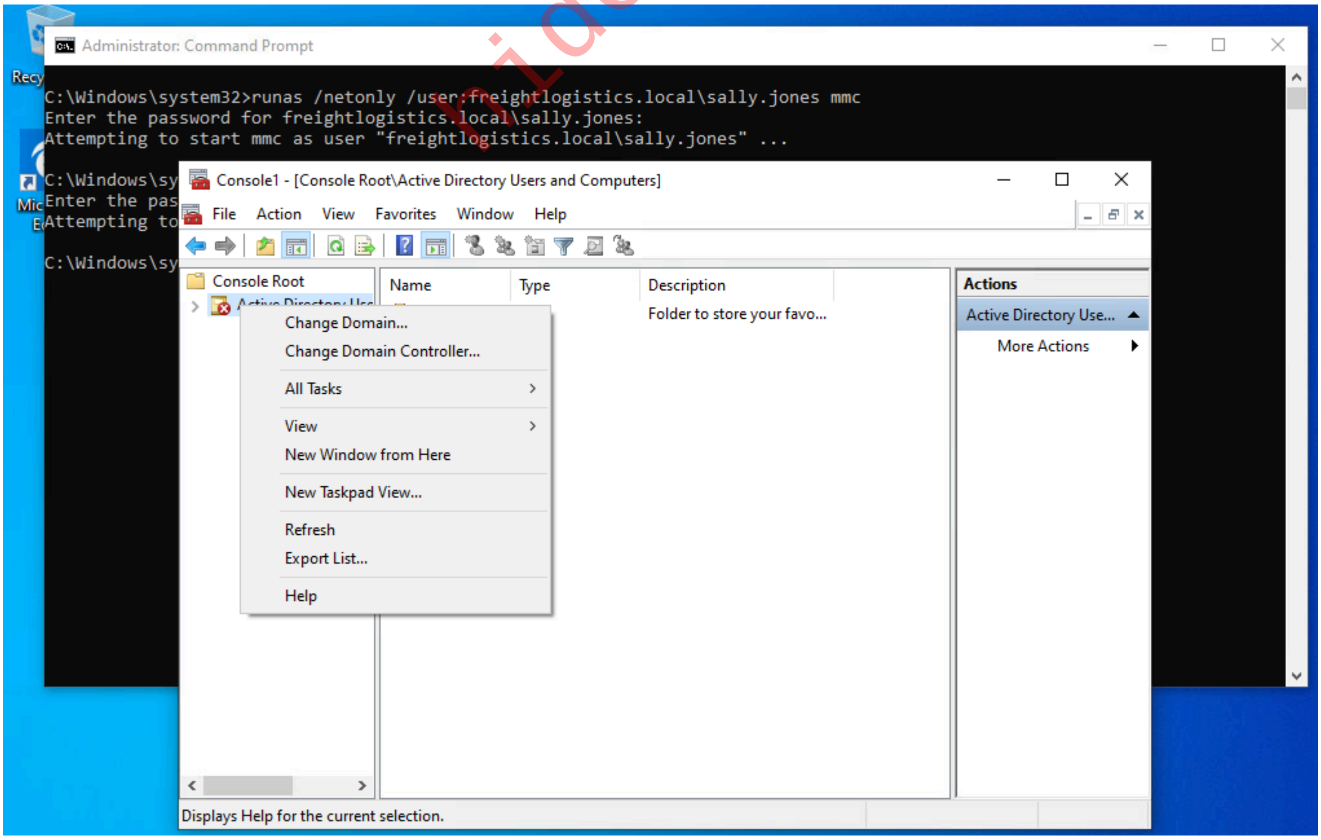
You can select snap-ins for this console from those available on your computer and configure the selected set of snap-ins. For extensible snap-ins, you can configure which extensions are enabled.

Snap-in	Vendor
Active Directory Domains an...	Microsoft
Active Directory Sites and Se...	Microsoft
Active Directory Users and C...	Microsoft
ActiveX Control	Microsoft
ADSI Edit	Microsoft
Authorization Manager	Microsoft
Certificate Templates	Microsoft
Certificates	Microsoft
Certification Authority	Microsoft
Component Services	Microsoft
Computer Management	Microsoft
Device Manager	Microsoft

Selected snap-ins: Console Root

Description:
You can use the Active Directory Domains and Trusts snap-in to manage Active Directory domains and trusts.

We can add any of the RSAT snap-ins and enumerate the target domain in the context of the target user `sally.jones` in the `freightlogistics.local` domain. After adding the snap-ins, we will get an error message that the "specified domain either does not exist or could not be contacted." From here, we have to right-click on the `Active Directory Users and Computers` snap-in (or any other chosen snap-in) and choose `Change Domain`.

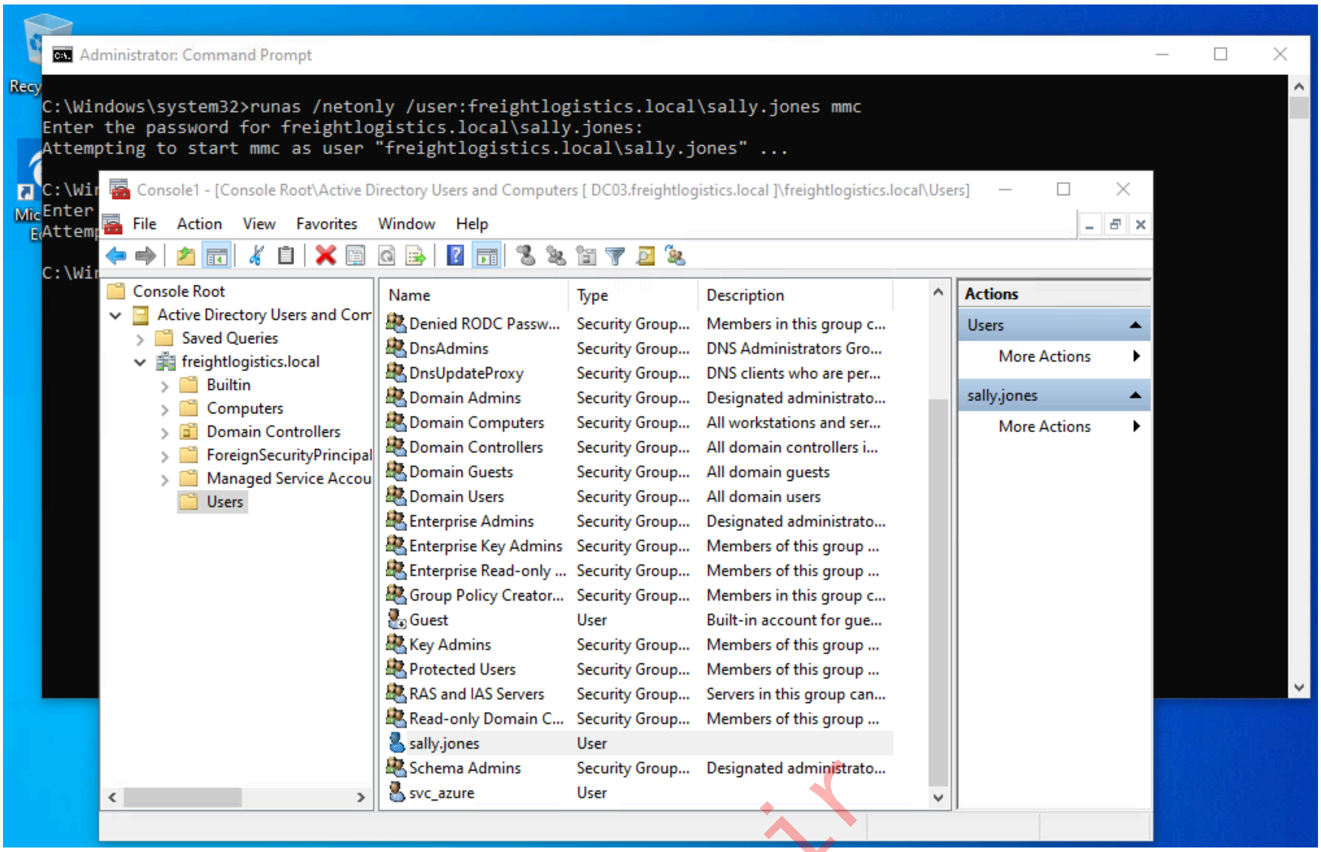


```
C:\Windows\system32>runas /netonly /user:freightlogistics.local\sally.jones mmc
Enter the password for freightlogistics.local\sally.jones:
Attempting to start mmc as user "freightlogistics.local\sally.jones" ...

C:\Windows\sy
Enter the pas
Attempting to

C:\Windows\sy
```

Type the target domain into the `Change domain` dialogue box, here `freightlogistics.local`. From here, we can now freely enumerate the domain using any of the AD RSAT snapins.



```
C:\Windows\system32>runas /netonly /user:freightlogistics.local\sally.jones mmc
Enter the password for freightlogistics.local\sally.jones:
Attempting to start mmc as user "freightlogistics.local\sally.jones" ...
```

Console1 - [Console Root\Active Directory Users and Computers [DC03.freightlogistics.local]\freightlogistics.local\Users]

Name	Type	Description
Denied RODC Passw...	Security Group...	Members in this group c...
DnsAdmins	Security Group...	DNS Administrators Gro...
DnsUpdateProxy	Security Group...	DNS clients who are per...
Domain Admins	Security Group...	Designated administrato...
Domain Computers	Security Group...	All workstations and ser...
Domain Controllers	Security Group...	All domain controllers i...
Domain Guests	Security Group...	All domain guests
Domain Users	Security Group...	All domain users
Enterprise Admins	Security Group...	Designated administrato...
Enterprise Key Admins	Security Group...	Members of this group ...
Enterprise Read-only ...	Security Group...	Members of this group ...
Group Policy Creator...	Security Group...	Members in this group c...
Guest	User	Built-in account for gue...
Key Admins	Security Group...	Members of this group ...
Protected Users	Security Group...	Members of this group ...
RAS and IAS Servers	Security Group...	Servers in this group can...
Read-only Domain C...	Security Group...	Members of this group ...
sally.jones	User	
Schema Admins	Security Group...	Designated administrato...
svc_azure	User	

While these graphical tools are useful and easy to use, they are very inefficient when trying to enumerate a large domain. In the next few sections, we will introduce `LDAP` and various types of search filters that we can use to enumerate AD using PowerShell. The topics that we cover in these sections will help us gain a better understanding of how AD works and how to search for information efficiently, which will ultimately better inform us on the usage of the more "automated" tools and scripts that we will cover in the next two `AD Enumeration` modules.

The Power of NT AUTHORITY\SYSTEM

The LocalSystem account `NT AUTHORITY\SYSTEM` is a built-in account in Windows operating systems, used by the service control manager. It has the highest level of access in the OS (and can be made even more powerful with Trusted Installer privileges). This account has more privileges than a local administrator account and is used to run most Windows services. It is also very common for third-party services to run in the context of this account by default. The SYSTEM account has the following privileges:

Privilege	Default State
SE_ASSIGNPRIMARYTOKEN_NAME	disabled
SE_AUDIT_NAME	enabled

Privilege	Default State
SE_BACKUP_NAME	disabled
SE_CHANGE_NOTIFY_NAME	enabled
SE_CREATE_GLOBAL_NAME	enabled
SE_CREATE_PAGEFILE_NAME	enabled
SE_CREATE_PERMANENT_NAME	enabled
SE_CREATE_TOKEN_NAME	disabled
SE_DEBUG_NAME	enabled
SE_IMPERSONATE_NAME	enabled
SE_INC_BASE_PRIORITY_NAME	enabled
SE_INCREASE_QUOTA_NAME	disabled
SE_LOAD_DRIVER_NAME	disabled
SE_LOCK_MEMORY_NAME	enabled
SE_MANAGE_VOLUME_NAME	disabled
SE_PROF_SINGLE_PROCESS_NAME	enabled
SE_RESTORE_NAME	disabled
SE_SECURITY_NAME	disabled
SE_SHUTDOWN_NAME	disabled
SE_SYSTEM_ENVIRONMENT_NAME	disabled
SE_SYSTEMTIME_NAME	disabled
SE_TAKE_OWNERSHIP_NAME	disabled
SE_TCB_NAME	enabled
SE_UNDOCK_NAME	disabled

The SYSTEM account on a domain-joined host can enumerate Active Directory by impersonating the computer account, which is essentially a special user account. If you land on a domain-joined host with SYSTEM privileges during an assessment and cannot find any useful credentials in memory or other data on the machine, there are still many things you can do. Having SYSTEM-level access within a domain environment is nearly equivalent to having a domain user account. The only real limitation is not being able to perform cross-trust Kerberos attacks such as Kerberoasting.

There are several ways to gain SYSTEM-level access on a host, including but not limited to:

- Remote Windows exploits such as EternalBlue or BlueKeep.
- Abusing a service running in the context of the SYSTEM account.
- Abusing SeImpersonate privileges using RottenPotatoNG against older Windows systems, Juicy Potato, or PrintSpoofer if targeting Windows 10/Windows Server 2019.

- Local privilege escalation flaws in Windows operating systems such as the Windows 10 Task Scheduler 0day.
- PsExec with the `-s` flag

By gaining SYSTEM-level access on a domain-joined host, we will be able to:

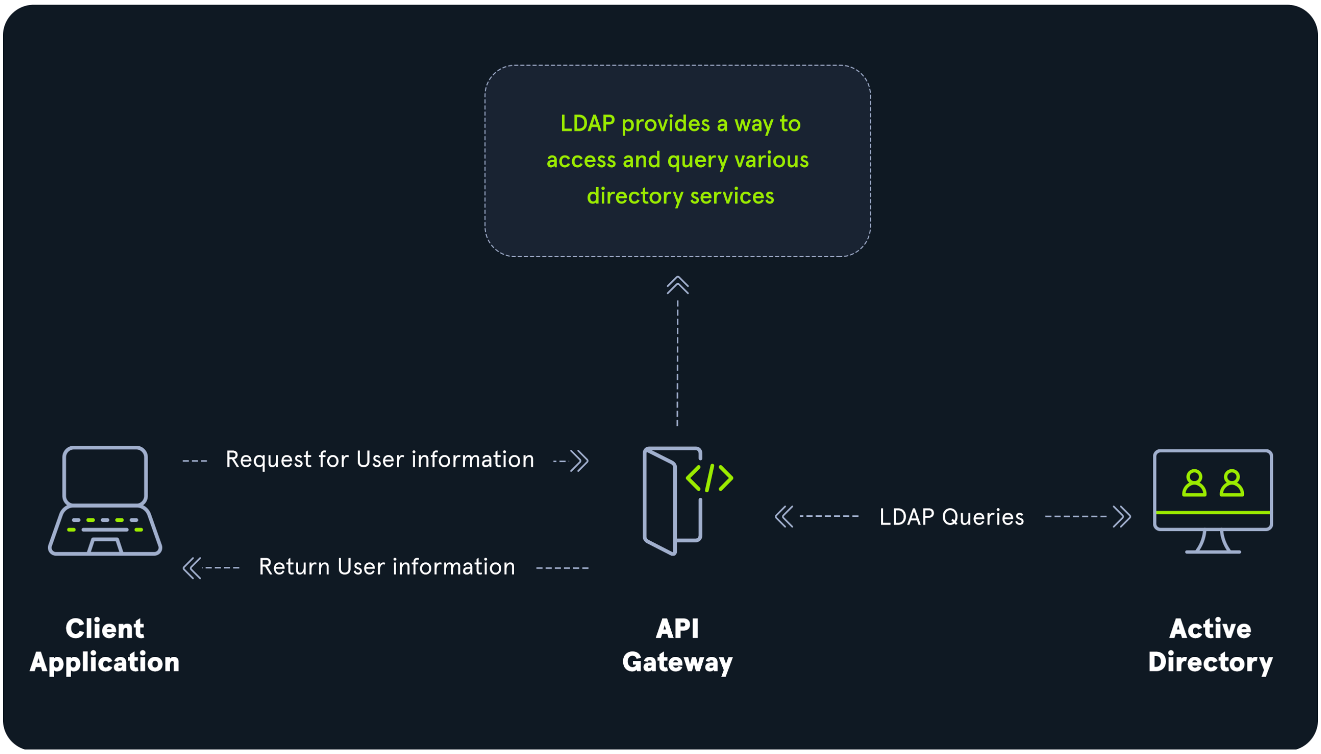
- Enumerate the domain and gather data such as information about domain users and groups, local administrator access, domain trusts, ACLs, user and computer properties, etc., using `BloodHound`, and `PowerView`/ `SharpView`.
- Perform Kerberoasting / ASREPRoasting attacks.
- Run tools such as Inveigh to gather Net-NTLM-v2 hashes or perform relay attacks.
- Perform token impersonation to hijack a privileged domain user account.
- Carry out ACL attacks.

LDAP Overview

`Lightweight Directory Access Protocol` (`LDAP`) is an integral part of Active Directory (AD). The latest LDAP specification is Version 3, which is published as RFC 4511. A firm understanding of how LDAP works in an AD environment is crucial for both attackers and defenders.

`LDAP` is an open-source and cross-platform protocol used for authentication against various directory services (such as AD). As discussed in the previous section, AD stores user account information and security information such as passwords and facilitates sharing this information with other devices on the network. `LDAP` is the language that applications use to communicate with other servers that also provide directory services. In other words, `LDAP` is a way that systems in the network environment can "speak" to AD.

An `LDAP` session begins by first connecting to an `LDAP` server, also known as a `Directory System Agent`. The Domain Controller in AD actively listens for `LDAP` requests, such as security authentication requests.


LDAP provides a way to access and query various directory services
Request for User information
Return User information
LDAP Queries
Client Application
API Gateway
Active Directory

The relationship between AD and `LDAP` can be compared to Apache and HTTP. The same way Apache is a web server that uses the HTTP protocol, Active Directory is a directory server that uses the `LDAP` protocol.

While uncommon, you may come across organizations while performing an assessment that does not have AD but does have LDAP, meaning that they most likely use another type of `LDAP` server such as OpenLDAP.

AD LDAP Authentication

`LDAP` is set up to authenticate credentials against AD using a "BIND" operation to set the authentication state for an `LDAP` session. There are two types of `LDAP` authentication.

1. **Simple Authentication:** This includes anonymous authentication, unauthenticated authentication, and username/password authentication. Simple authentication means that a username and password create a BIND request to authenticate to the LDAP server.
2. **SASL Authentication:** The Simple Authentication and Security Layer (SASL) framework uses other authentication services, such as Kerberos, to bind to the `LDAP` server and then uses this authentication service (Kerberos in this example) to authenticate to `LDAP`. The `LDAP` server uses the `LDAP` protocol to send an `LDAP` message to the authorization service which initiates a series of challenge/response messages resulting in either successful or unsuccessful authentication. SASL can provide further security due to the separation of authentication methods from application protocols.

LDAP authentication messages are sent in cleartext by default so anyone can sniff out LDAP messages on the internal network. It is recommended to use TLS encryption or similar to safeguard this information in transit.

LDAP Queries

We can communicate with the directory service using `LDAP` queries to ask the service for information. For example, the following query can be used to find all workstations in a network `(objectCategory=computer)` while this query can be used to find all domain controllers: `(&(objectCategory=Computer)(userAccountControl:1.2.840.113556.1.4.803:=8192))`.

LDAP queries can be used to perform user-related searches, such as " `(&(objectCategory=person)(objectClass=user))` " which searches for all users, as well as group related searches such as " `(objectClass=group)` " which returns all groups. Here is one example of a simple query to find all AD groups using the " `Get-ADObject` " cmdlet and the " `LDAPFilter parameter` ".

LDAP Query - User Related Search

```
PS C:\htb> Get-ADObject -LDAPFilter '(objectClass=group)' | select name

name
--
Administrators
Users
Guests
Print Operators
Backup Operators
Replicator
Remote Desktop Users
Network Configuration Operators
Performance Monitor Users
Performance Log Users
Distributed COM Users
IIS_IUSRS
Cryptographic Operators
Event Log Readers
Certificate Service DCOM Access
RDS Remote Access Servers
RDS Endpoint Servers
RDS Management Servers
Hyper-V Administrators
Access Control Assistance Operators
Remote Management Users
```

```
<SNIP>
```

We can also use LDAP queries to perform more detailed searches. This query searches the domain for all administratively disabled accounts.

LDAP Query - Detailed Search

```
PS C:\htb> Get-ADObject -LDAPFilter '(&(objectCategory=person)
(objectClass=user)(userAccountControl:1.2.840.113556.1.4.803:=2))' -
Properties * | select samaccountname,useraccountcontrol

samaccountname
useraccountcontrol
--------------                                                       ---
---------------
Guest                    ACCOUNTDISABLE, PASSWD_NOTREQD, NORMAL_ACCOUNT,
DONT_EXPIRE_PASSWORD
DefaultAccount           ACCOUNTDISABLE, PASSWD_NOTREQD, NORMAL_ACCOUNT,
DONT_EXPIRE_PASSWORD
krbtgt                                   ACCOUNTDISABLE, NORMAL_ACCOUNT,
DONT_EXPIRE_PASSWORD
caroline.ali                         ACCOUNTDISABLE, PASSWD_NOTREQD,
NORMAL_ACCOUNT
$SH2000-FPNHUU487JP0                 ACCOUNTDISABLE, PASSWD_NOTREQD,
NORMAL_ACCOUNT
SM_00390f38b41e488ab                                     ACCOUNTDISABLE,
NORMAL_ACCOUNT
SM_e081bc60d79c4597b                                     ACCOUNTDISABLE,
NORMAL_ACCOUNT
SM_a9a4eed7ad2d4369a                                     ACCOUNTDISABLE,
NORMAL_ACCOUNT
SM_d836f82078bf4cf89                                     ACCOUNTDISABLE,
NORMAL_ACCOUNT
SM_6a24f488535649558                                     ACCOUNTDISABLE,
NORMAL_ACCOUNT
SM_08a2324990674a87b                                     ACCOUNTDISABLE,
NORMAL_ACCOUNT
SM_d1fea2710dc146b1b                                     ACCOUNTDISABLE,
NORMAL_ACCOUNT
SM_b56189681baa441db                                     ACCOUNTDISABLE,
NORMAL_ACCOUNT
SM_b72a918d27554863b                                     ACCOUNTDISABLE,
NORMAL_ACCOUNT
```

More examples of basic and more advanced `LDAP` queries for AD can be found at the following links:

- LDAP queries related to AD computers
- LDAP queries related to AD users
- LDAP queries related to AD groups

`LDAP` queries are extremely powerful tools for querying Active Directory. We can harness their power to gather a wide variety of information, map out the AD environment, and hunt for misconfigurations. LDAP queries can be combined with filters to perform even more granular searches. The next two sections will cover both AD and LDAP search filters in-depth to prepare us for introducing a variety of AD enumeration tools in subsequent modules.

Note: When spawning your target, we ask you to wait for 3 minutes until the whole lab with all the configurations is set up so that the connection to your target works flawlessly.

Active Directory Search Filters

The next two sections will cover the `Filter` and `LDAPFilter` parameters used by the ActiveDirectory PowerShell module cmdlets. It is important to know how to build proper filter syntax for querying Active Directory using `PowerShell`. This knowledge gives us a deeper understanding of how our tools such as `PowerView` function under the hood and how we can further harness their power when enumerating Active Directory. It is also useful to understand how to formulate filters if you find yourself in a situation during an assessment without any of your tools available to you. Armed with this knowledge, you will be able to effectively "live off the land" and utilize built-in PowerShell cmdlets to perform your enumeration tasks (albeit slower than using many of the tools we will cover in this module).

PowerShell Filters

Filters in PowerShell allow you to process piped output more efficiently and retrieve exactly the information you need from a command. Filters can be used to narrow down specific data in a large result or retrieve data that can then be piped to another command.

We can use filters with the `Filter` parameter. A basic example is querying a computer for installed software:

PowerShell - Filter Installed Software

```
PS C:\htb> get-ciminstance win32_product | fl
```

```
IdentifyingNumber : {7FED75A1-600C-394B-8376-712E2A8861F2}
Name              : Microsoft Visual C++ 2017 x86 Additional Runtime -
14.12.25810
Vendor            : Microsoft Corporation
Version           : 14.12.25810
Caption           : Microsoft Visual C++ 2017 x86 Additional Runtime -
14.12.25810

IdentifyingNumber : {748D3A12-9B82-4B08-A0FF-CFDE83612E87}
Name              : VMware Tools
Vendor            : VMware, Inc.
Version           : 10.3.2.9925305
Caption           : VMware Tools

IdentifyingNumber : {EA8CB806-C109-4700-96B4-F1F268E5036C}
Name              : Local Administrator Password Solution
Vendor            : Microsoft Corporation
Version           : 6.2.0.0
Caption           : Local Administrator Password Solution

IdentifyingNumber : {2CD849A7-86A1-34A6-B8F9-D72F5B21A9AE}
Name              : Microsoft Visual C++ 2017 x64 Additional Runtime -
14.12.25810
Vendor            : Microsoft Corporation
Version           : 14.12.25810
Caption           : Microsoft Visual C++ 2017 x64 Additional Runtime -
14.12.25810

<SNIP>
```

The above command can provide considerable output. We can use the `Filter` parameter with the `notlike` operator to filter out all Microsoft software (which may be useful when enumerating a system for local privilege escalation vectors).

PowerShell - Filter Out Microsoft Software

```
PS C:\htb> get-ciminstance win32_product -Filter "NOT Vendor like
'%Microsoft%'" | fl

IdentifyingNumber : {748D3A12-9B82-4B08-A0FF-CFDE83612E87}
Name              : VMware Tools
Vendor            : VMware, Inc.
Version           : 10.3.2.9925305
Caption           : VMware Tools
```

Operators

The `Filter` operator requires at least one operator, which can help narrow down search results or reduce a large amount of command output to something more digestible. Filtering properly is important, especially when enumerating large environments and looking for very specific information in the command output. The following operators can be used with the `Filter` parameter:

Filter	Meaning
-eq	Equal to
-le	Less than or equal to
-ge	Greater than or equal to
-ne	Not equal to
-lt	Less than
-gt	Greater than
-approx	Approximately equal to
-bor	Bitwise OR
-band	Bitwise AND
-recursivematch	Recursive match
-like	Like
-notlike	Not like
-and	Boolean AND
-or	Boolean OR
-not	Boolean NOT

Filter Examples: AD Object Properties

The filter can be used with operators to compare, exclude, search for, etc., a variety of AD object properties. Filters can be wrapped in curly braces, single quotes, parentheses, or double-quotes. For example, the following simple search filter using `Get-ADUser` to find information about the user `Sally Jones` can be written as follows:

PowerShell - Filter Examples

```
Get-ADUser -Filter "name -eq 'sally jones'"
Get-ADUser -Filter {name -eq 'sally jones'}
```

```
Get-ADUser -Filter 'name -eq "sally jones"'
```

As seen above, the property value (here, `sally jones`) can be wrapped in single or double-quotes. The asterisk (`*`) can be used as a wildcard when performing queries. The command `Get-ADUser -filter {name -like "joe*"}` using a wildcard would return all domain users whose name start with `joe` (joe, joel, etc.). When using filters, certain characters must be escaped:

Character	Escaped As	Note
“	`”	Only needed if the data is enclosed in double-quotes.
‘	\’	Only needed if the data is enclosed in single quotes.
NUL	\00	Standard LDAP escape sequence.
\	\5c	Standard LDAP escape sequence.
*	\2a	Escaped automatically, but only in -eq and -ne comparisons. Use -like and -notlike operators for wildcard comparison.
(	/28	Escaped automatically.
)	/29	Escaped automatically.
/	/2f	Escaped automatically.

Let's try out some of these filters to enumerate the `INLANEFREIGHT.LOCAL` domain. We can search all domain computers for interesting hostnames. SQL servers are a particularly juicy target on internal assessments. The below command searches all hosts in the domain using `Get-ADComputer`, filtering on the `DNSHostName` property that contains the word `SQL`.

PowerShell - Filter For SQL

```
PS C:\htb> Get-ADComputer  -Filter "DNSHostName -like 'SQL*'"

DistinguishedName : CN=SQL01,OU=SQL
Servers,OU=Servers,DC=INLANEFREIGHT,DC=LOCAL
DNSHostName       : SQL01.INLANEFREIGHT.LOCAL
Enabled           : True
Name              : SQL01
ObjectClass       : computer
ObjectGUID        : 42cc9264-1655-4bfa-b5f9-21101afb33d0
SamAccountName    : SQL01$
SID               : S-1-5-21-2974783224-3764228556-2640795941-1104
```

```
UserPrincipalName :
```

Next, let's search for administrative groups. We can do this by filtering on the `adminCount` attribute. The group with this attribute set to `1` are protected by AdminSDHolder and known as protected groups. `AdminSDHolder` is owned by the Domain Admins group. It has the privileges to change the permissions of objects in Active Directory. As discussed above, we can pipe the filtered command output and select just the group names.

PowerShell - Filter Administrative Groups

```
PS C:\htb> Get-ADGroup -Filter "adminCount -eq 1" | select Name

Name
----
Administrators
Print Operators
Backup Operators
Replicator
Domain Controllers
Schema Admins
Enterprise Admins
Domain Admins
Server Operators
Account Operators
Read-only Domain Controllers
Security Operations
```

We can also combine filters. Let's search for all administrative users with the `DoesNotRequirePreAuth` attribute set, meaning that they can be ASREPRoasted (this attack will be covered in-depth in later modules). Here we are selecting all domain users and specifying two conditions with the `-eq` operator.

PowerShell - Filter Administrative Users

```
PS C:\htb> Get-ADUser -Filter {adminCount -eq '1' -and
DoesNotRequirePreAuth -eq 'True'}

DistinguishedName : CN=Jenna Smith,OU=Server
Team,OU=IT,OU=Employees,DC=INLANEFREIGHT,DC=LOCAL
GivenName         : jenna
Name              : Jenna Smith
ObjectClass       : user
ObjectGUID        : ea3c930f-aa8e-4fdc-987c-4a9ee1a75409
SamAccountName    : jenna.smith
SID               : S-1-5-21-2974783224-3764228556-2640795941-1999
```

```
Surname           : smith
UserPrincipalName : jenna.smith@inlanefreight
```

Finally, let's see an example of combining filters and piping output multiple times to find our desired information. The following command can be used to find all administrative users with the " `servicePrincipalName` " attribute set, meaning that they can likely be subject to a Kerberoasting attack. This example applies the `Filter` parameter to find accounts with the `adminCount` attribute set to `1`, pipes this output to find all accounts with a Service Principal Name (SPN), and finally selects a few attributes about the accounts, including the account name, group membership, and the SPN.

PowerShell - Find Administrative Users with the ServicePrincipalName

```
PS C:\htb> Get-ADUser -Filter "adminCount -eq '1'" -Properties * | where
servicePrincipalName -ne $null | select
SamAccountName,MemberOf,ServicePrincipalName | fl

SamAccountName       : krbtgt
MemberOf             : {CN=Denied RODC Password Replication
Group,CN=Users,DC=INLANEFREIGHT,DC=LOCAL}
ServicePrincipalName : {kadmin/changepw}

SamAccountName       : sqlqa
MemberOf             : {CN=Domain
Admins,CN=Users,DC=INLANEFREIGHT,DC=LOCAL}
ServicePrincipalName : {MSSQL_svc_qa/inlanefreight.local:1443}
```

It would take an extremely long time to enumerate an Active Directory environment using many combinations of the commands above. This last example could be performed quickly and easily with tools such as `PowerView` or `Rubeus`. Nevertheless, it is important to apply filters competently when enumerating AD as the output from tools like `PowerView` can even be further filtered to provide us with precise results.

Note: When spawning your target, we ask you to wait for 3 minutes until the whole lab with all the configurations is set up so that the connection to your target works flawlessly.

LDAP Search Filters

Basic LDAP Filter Syntax and Operators

The `LDAPFilter` parameter with the same cmdlets lets us use LDAP search filters when searching for information. The syntax for these filters is defined in RFC 4515 - Lightweight Directory Access Protocol (LDAP): String Representation of Search Filters.

LDAP filters must have one or more criteria. If more than one criteria exist, they can be concatenated together using logical `AND` or `OR` operators. These operators are always placed in the front of the criteria (operands), which is also referred to as Polish Notation.

Filter rules are enclosed in parentheses and can be grouped by surrounding the group in parentheses and using one of the following comparison operators:

Operator	Function
`&`	and
`	`
`!`	not

Some example `AND` and `OR` operations are as follows:

`AND` Operation:

- One criteria: `(& (..C1..) (..C2..))`
- More than two criteria: `(& (..C1..) (..C2..) (..C3..))`

`OR` Operation:

- One criteria: `(| (..C1..) (..C2..))`
- More than two criteria: `(| (..C1..) (..C2..) (..C3..))`

We can also have nested operations, for example " `(|(& (..C1..) (..C2..))(& (..C3..) (..C4..)))` " translates to " `(C1 AND C2) OR (C3 AND C4)` ".

Search Criteria

When writing an LDAP search filter, we need to specify a rule requirement for the LDAP attribute in question (i.e. " `(displayName=william)` "). The following rules can be used to specify our search criteria:

Criteria	Rule	Example
Equal to	(attribute=123)	(&(objectclass=user)(displayName=Smith)
Not equal to	(!(attribute=123))	!objectClass=group)

Criteria	Rule	Example
Present	(attribute=*)	(department=*)
Not present	(!(attribute=*))	(!homeDirectory=*)
Greater than	(attribute>=123)	(maxStorage=100000)
Less than	(attribute<=123)	(maxStorage<=100000)
Approximate match	(attribute~=123)	(sAMAccountName~=Jason)
Wildcards	(attribute=*A)	(givenName=*Sam)

This link contains a large listing of User Attributes, and the below is a list of all Base Attributes.

Full list of Base Attributes

LDAP Display Name	CN	Attribut
`accountExpires`	Account-Expires	1.2.840.
`accountNameHistory`	Account-Name-History	1.2.840.
`aCSAggregateTokenRatePerUser`	ACS-Aggregate-Token-Rate-Per-User	1.2.840.
`aCSAllocableRSVPBandwidth`	ACS-Allocable-RSVP-Bandwidth	1.2.840.
`aCSCacheTimeout`	ACS-Cache-Timeout	1.2.840.
`aCSDirection`	ACS-Direction	1.2.840.
`aCSDSBMDeadTime`	ACS-DSBM-DeadTime	1.2.840.
`aCSDSBMPriority`	ACS-DSBM-Priority	1.2.840.
`aCSDSBMRefresh`	ACS-DSBM-Refresh	1.2.840.
`aCSEnableACSService`	ACS-Enable-ACS-Service	1.2.840.
`aCSEnableRSVPAccounting`	ACS-Enable-RSVP-Accounting	1.2.840.
`aCSEnableRSVPMessageLogging`	ACS-Enable-RSVP-Message-Logging	1.2.840.
`aCSEventLogLevel`	ACS-Event-Log-Level	1.2.840.
`aCSIdentityName`	ACS-Identity-Name	1.2.840.
`aCSMaxAggregatePeakRatePerUser`	ACS-Max-Aggregate-Peak-Rate-Per-User	1.2.840.
`aCSMaxDurationPerFlow`	ACS-Max-Duration-Per-Flow	1.2.840.
`aCSMaximumSDUSize`	ACS-Maximum-SDU-Size	1.2.840.
`aCSMaxNoOfAccountFiles`	ACS-Max-No-Of-Account-Files	1.2.840.
`aCSMaxNoOfLogFiles`	ACS-Max-No-Of-Log-Files	1.2.840.

LDAP Display Name	CN	Attribut
`aCSMaxPeakBandwidth`	ACS-Max-Peak-Bandwidth	1.2.840.
`aCSMaxPeakBandwidthPerFlow`	ACS-Max-Peak-Bandwidth-Per-Flow	1.2.840.
`aCSMaxSizeOfRSVPAccountFile`	ACS-Max-Size-Of-RSVP-Account-File	1.2.840.
`aCSMaxSizeOfRSVPLogFile`	ACS-Max-Size-Of-RSVP-Log-File	1.2.840.
`aCSMaxTokenBucketPerFlow`	ACS-Max-Token-Bucket-Per-Flow	1.2.840.
`aCSMaxTokenRatePerFlow`	ACS-Max-Token-Rate-Per-Flow	1.2.840.
`aCSMinimumDelayVariation`	ACS-Minimum-Delay-Variation	1.2.840.
`aCSMinimumLatency`	ACS-Minimum-Latency	1.2.840.
`aCSMinimumPolicedSize`	ACS-Minimum-Policed-Size	1.2.840.
`aCSNonReservedMaxSDUSize`	ACS-Non-Reserved-Max-SDU-Size	1.2.840.
`aCSNonReservedMinPolicedSize`	ACS-Non-Reserved-Min-Policed-Size	1.2.840.
`aCSNonReservedPeakRate`	ACS-Non-Reserved-Peak-Rate	1.2.840.
`aCSNonReservedTokenSize`	ACS-Non-Reserved-Token-Size	1.2.840.
`aCSNonReservedTxLimit`	ACS-Non-Reserved-Tx-Limit	1.2.840.
`aCSNonReservedTxSize`	ACS-Non-Reserved-Tx-Size	1.2.840.
`aCSPermissionBits`	ACS-Permission-Bits	1.2.840.
`aCSPolicyName`	ACS-Policy-Name	1.2.840.
`aCSPriority`	ACS-Priority	1.2.840.
`aCSRSVPAccountFilesLocation`	ACS-RSVP-Account-Files-Location	1.2.840.
`aCSRSVPLogFilesLocation`	ACS-RSVP-Log-Files-Location	1.2.840.
`aCSServerList`	ACS-Server-List	1.2.840.
`aCSServiceType`	ACS-Service-Type	1.2.840.
`aCSTimeOfDay`	ACS-Time-Of-Day	1.2.840.
`aCSTotalNoOfFlows`	ACS-Total-No-Of-Flows	1.2.840.
`additionalTrustedServiceNames`	Additional-Trusted-Service-Names	1.2.840.
`addressBookRoots`	Address-Book-Roots	1.2.840.
`addressEntryDisplayTable`	Address-Entry-Display-Table	1.2.840.
`addressEntryDisplayTableMSDOS`	Address-Entry-Display-Table-MSDOS	1.2.840.
`addressSyntax`	Address-Syntax	1.2.840.
`addressType`	Address-Type	1.2.840.
`adminContextMenu`	Admin-Context-Menu	1.2.840.
`adminCount`	Admin-Count	1.2.840.

LDAP Display Name	CN	Attribut
adminDescription	Admin-Description	1.2.840.
adminDisplayName	Admin-Display-Name	1.2.840.
adminPropertyPages	Admin-Property-Pages	1.2.840.
allowedAttributes	Allowed-Attributes	1.2.840.
allowedAttributesEffective	Allowed-Attributes-Effective	1.2.840.
allowedChildClasses	Allowed-Child-Classes	1.2.840.
allowedChildClassesEffective	Allowed-Child-Classes-Effective	1.2.840.
altSecurityIdentities	Alt-Security-Identities	1.2.840.
aNR	ANR	1.2.840.
applicationName	Application-Name	1.2.840.
appliesTo	Applies-To	1.2.840.
appSchemaVersion	App-Schema-Version	1.2.840.
assetNumber	Asset-Number	1.2.840.
assistant	Assistant	1.2.840.
assocNTAccount	Assoc-NT-Account	1.2.840.
attributeDisplayNames	Attribute-Display-Names	1.2.840.
attributeID	Attribute-ID	1.2.840.
attributeSecurityGUID	Attribute-Security-GUID	1.2.840.
attributeSyntax	Attribute-Syntax	1.2.840.
attributeTypes	Attribute-Types	2.5.21.5
auditingPolicy	Auditing-Policy	1.2.840.
authenticationOptions	Authentication-Options	1.2.840.
authorityRevocationList	Authority-Revocation-List	2.5.4.38
auxiliaryClass	Auxiliary-Class	1.2.840.
badPasswordTime	Bad-Password-Time	1.2.840.
badPwdCount	Bad-Pwd-Count	1.2.840.
birthLocation	Birth-Location	1.2.840.
bridgeheadServerListBL	Bridgehead-Server-List-BL	1.2.840.
bridgeheadTransportList	Bridgehead-Transport-List	1.2.840.
builtinCreationTime	Builtin-Creation-Time	1.2.840.
builtinModifiedCount	Builtin-Modified-Count	1.2.840.
businessCategory	Business-Category	2.5.4.15
bytesPerMinute	Bytes-Per-Minute	1.2.840.
c	Country-Name	2.5.4.6

LDAP Display Name	CN	Attribut
`cACertificate`	CA-Certificate	2.5.4.37
`cACertificateDN`	CA-Certificate-DN	1.2.840.
`cAConnect`	CA-Connect	1.2.840.
`canonicalName`	Canonical-Name	1.2.840.
`canUpgradeScript`	Can-Upgrade-Script	1.2.840.
`catalogs`	Catalogs	1.2.840.
`categories`	Categories	1.2.840.
`categoryId`	Category-Id	1.2.840.
`cAUsages`	CA-Usages	1.2.840.
`cAWEBURL`	CA-WEB-URL	1.2.840.
`certificateAuthorityObject`	Certificate-Authority-Object	1.2.840.
`certificateRevocationList`	Certificate-Revocation-List	2.5.4.39
`certificateTemplates`	Certificate-Templates	1.2.840.
`classDisplayName`	Class-Display-Name	1.2.840.
`cn`	Common-Name	2.5.4.3
`co`	Text-Country	1.2.840.
`codePage`	Code-Page	1.2.840.
`cOMClassID`	COM-ClassID	1.2.840.
`cOMCLSID`	COM-CLSID	1.2.840.
`cOMInterfaceID`	COM-InterfaceID	1.2.840.
`comment`	User-Comment	1.2.840.
`cOMOtherProgId`	COM-Other-Prog-Id	1.2.840.
`company`	Company	1.2.840.
`cOMProgID`	COM-ProgID	1.2.840.
`cOMTreatAsClassId`	COM-Treat-As-Class-Id	1.2.840.
`cOMTypelibId`	COM-Typelib-Id	1.2.840.
`cOMUniqueLIBID`	COM-Unique-LIBID	1.2.840.
`contentIndexingAllowed`	Content-Indexing-Allowed	1.2.840.
`contextMenu`	Context-Menu	1.2.840.
`controlAccessRights`	Control-Access-Rights	1.2.840.
`cost`	Cost	1.2.840.
`countryCode`	Country-Code	1.2.840.
`createDialog`	Create-Dialog	1.2.840.
`createTimeStamp`	Create-Time-Stamp	2.5.18.1

LDAP Display Name	CN	Attribut
`createWizardExt`	Create-Wizard-Ext	1.2.840.
`creationTime`	Creation-Time	1.2.840.
`creationWizard`	Creation-Wizard	1.2.840.
`creator`	Creator	1.2.840.
`cRLObject`	CRL-Object	1.2.840.
`cRLPartitionedRevocationList`	CRL-Partitioned-Revocation-List	1.2.840.
`crossCertificatePair`	Cross-Certificate-Pair	2.5.4.40
`currentLocation`	Current-Location	1.2.840.
`currentParentCA`	Current-Parent-CA	1.2.840.
`currentValue`	Current-Value	1.2.840.
`currMachineId`	Curr-Machine-Id	1.2.840.
`dBCSPwd`	DBCS-Pwd	1.2.840.
`dc`	Domain-Component	0.9.234
`defaultClassStore`	Default-Class-Store	1.2.840.
`defaultGroup`	Default-Group	1.2.840.
`defaultHidingValue`	Default-Hiding-Value	1.2.840.
`defaultLocalPolicyObject`	Default-Local-Policy-Object	1.2.840.
`defaultObjectCategory`	Default-Object-Category	1.2.840.
`defaultPriority`	Default-Priority	1.2.840.
`defaultSecurityDescriptor`	Default-Security-Descriptor	1.2.840.
`deltaRevocationList`	Delta-Revocation-List	2.5.4.53
`department`	Department	1.2.840.
`description`	Description	2.5.4.13
`desktopProfile`	Desktop-Profile	1.2.840.
`destinationIndicator`	Destination-Indicator	2.5.4.27
`dhcpClasses`	dhcp-Classes	1.2.840.
`dhcpFlags`	dhcp-Flags	1.2.840.
`dhcpIdentification`	dhcp-Identification	1.2.840.
`dhcpMask`	dhcp-Mask	1.2.840.
`dhcpMaxKey`	dhcp-MaxKey	1.2.840.
`dhcpObjDescription`	dhcp-Obj-Description	1.2.840.
`dhcpObjName`	dhcp-Obj-Name	1.2.840.
`dhcpOptions`	dhcp-Options	1.2.840.
`dhcpProperties`	dhcp-Properties	1.2.840.

LDAP Display Name	CN	Attribut
`dhcpRanges`	dhcp-Ranges	1.2.840.
`dhcpReservations`	dhcp-Reservations	1.2.840.
`dhcpServers`	dhcp-Servers	1.2.840.
`dhcpSites`	dhcp-Sites	1.2.840.
`dhcpState`	dhcp-State	1.2.840.
`dhcpSubnets`	dhcp-Subnets	1.2.840.
`dhcpType`	dhcp-Type	1.2.840.
`dhcpUniqueKey`	dhcp-Unique-Key	1.2.840.
`dhcpUpdateTime`	dhcp-Update-Time	1.2.840.
`directReports`	Reports	1.2.840.
`displayName`	Display-Name	1.2.840.
`displayNamePrintable`	Display-Name-Printable	1.2.840.
`distinguishedName`	Obj-Dist-Name	2.5.4.49
`dITContentRules`	DIT-Content-Rules	2.5.21.2
`division`	Division	1.2.840.
`dMDLocation`	DMD-Location	1.2.840.
`dmdName`	DMD-Name	1.2.840.
`dNReferenceUpdate`	DN-Reference-Update	1.2.840.
`dnsAllowDynamic`	Dns-Allow-Dynamic	1.2.840.
`dnsAllowXFR`	Dns-Allow-XFR	1.2.840.
`dNSHostName`	DNS-Host-Name	1.2.840.
`dnsNotifySecondaries`	Dns-Notify-Secondaries	1.2.840.
`dNSProperty`	DNS-Property	1.2.840.
`dnsRecord`	Dns-Record	1.2.840.
`dnsRoot`	Dns-Root	1.2.840.
`dnsSecureSecondaries`	Dns-Secure-Secondaries	1.2.840.
`dNSTombstoned`	DNS-Tombstoned	1.2.840.
`domainCAs`	Domain-Certificate-Authorities	1.2.840.
`domainCrossRef`	Domain-Cross-Ref	1.2.840.
`domainID`	Domain-ID	1.2.840.
`domainIdentifier`	Domain-Identifier	1.2.840.
`domainPolicyObject`	Domain-Policy-Object	1.2.840.
`domainPolicyReference`	Domain-Policy-Reference	1.2.840.
`domainReplica`	Domain-Replica	1.2.840.

LDAP Display Name	CN	Attribut
`domainWidePolicy`	Domain-Wide-Policy	1.2.840.
`driverName`	Driver-Name	1.2.840.
`driverVersion`	Driver-Version	1.2.840.
`dSASignature`	DSA-Signature	1.2.840.
`dSCorePropagationData`	DS-Core-Propagation-Data	1.2.840.
`dSHeuristics`	DS-Heuristics	1.2.840.
`dSUIAdminMaximum`	DS-UI-Admin-Maximum	1.2.840.
`dSUIAdminNotification`	DS-UI-Admin-Notification	1.2.840.
`dSUIShellMaximum`	DS-UI-Shell-Maximum	1.2.840.
`dynamicLDAPServer`	Dynamic-LDAP-Server	1.2.840.
`eFSPolicy`	EFSPolicy	1.2.840.
`employeeID`	Employee-ID	1.2.840.
`employeeNumber`	Employee-Number	1.2.840.
`employeeType`	Employee-Type	1.2.840.
`Enabled`	Enabled	1.2.840.
`enabledConnection`	Enabled-Connection	1.2.840.
`enrollmentProviders`	Enrollment-Providers	1.2.840.
`extendedAttributeInfo`	Extended-Attribute-Info	1.2.840.
`extendedCharsAllowed`	Extended-Chars-Allowed	1.2.840.
`extendedClassInfo`	Extended-Class-Info	1.2.840.
`extensionName`	Extension-Name	1.2.840.
`facsimileTelephoneNumber`	Facsimile-Telephone-Number	2.5.4.23
`fileExtPriority`	File-Ext-Priority	1.2.840.
`flags`	Flags	1.2.840.
`flatName`	Flat-Name	1.2.840.
`forceLogoff`	Force-Logoff	1.2.840.
`foreignIdentifier`	Foreign-Identifier	1.2.840.
`friendlyNames`	Friendly-Names	1.2.840.
`fromEntry`	From-Entry	1.2.840.
`fromServer`	From-Server	1.2.840.
`frsComputerReference`	Frs-Computer-Reference	1.2.840.
`frsComputerReferenceBL`	Frs-Computer-Reference-BL	1.2.840.
`fRSControlDataCreation`	FRS-Control-Data-Creation	1.2.840.
`fRSControlInboundBacklog`	FRS-Control-Inbound-Backlog	1.2.840.

LDAP Display Name	CN	Attribut
`fRSControlOutboundBacklog`	FRS-Control-Outbound-Backlog	1.2.840.
`fRSDirectoryFilter`	FRS-Directory-Filter	1.2.840.
`fRSDSPoll`	FRS-DS-Poll	1.2.840.
`fRSExtensions`	FRS-Extensions	1.2.840.
`fRSFaultCondition`	FRS-Fault-Condition	1.2.840.
`fRSFileFilter`	FRS-File-Filter	1.2.840.
`fRSFlags`	FRS-Flags	1.2.840.
`fRSLevelLimit`	FRS-Level-Limit	1.2.840.
`fRSMemberReference`	FRS-Member-Reference	1.2.840.
`fRSMemberReferenceBL`	FRS-Member-Reference-BL	1.2.840.
`fRSPartnerAuthLevel`	FRS-Partner-Auth-Level	1.2.840.
`fRSPrimaryMember`	FRS-Primary-Member	1.2.840.
`fRSReplicaSetGUID`	FRS-Replica-Set-GUID	1.2.840.
`fRSReplicaSetType`	FRS-Replica-Set-Type	1.2.840.
`fRSRootPath`	FRS-Root-Path	1.2.840.
`fRSRootSecurity`	FRS-Root-Security	1.2.840.
`fRSServiceCommand`	FRS-Service-Command	1.2.840.
`fRSServiceCommandStatus`	FRS-Service-Command-Status	1.2.840.
`fRSStagingPath`	FRS-Staging-Path	1.2.840.
`fRSTimeLastCommand`	FRS-Time-Last-Command	1.2.840.
`fRSTimeLastConfigChange`	FRS-Time-Last-Config-Change	1.2.840.
`fRSUpdateTimeout`	FRS-Update-Timeout	1.2.840.
`fRSVersion`	FRS-Version	1.2.840.
`fRSVersionGUID`	FRS-Version-GUID	1.2.840.
`fRSWorkingPath`	FRS-Working-Path	1.2.840.
`fSMORoleOwner`	FSMO-Role-Owner	1.2.840.
`garbageCollPeriod`	Garbage-Coll-Period	1.2.840.
`generatedConnection`	Generated-Connection	1.2.840.
`generationQualifier`	Generation-Qualifier	2.5.4.44
`givenName`	Given-Name	2.5.4.42
`globalAddressList`	Global-Address-List	1.2.840.
`governsID`	Governs-ID	1.2.840.
`gPCFileSysPath`	GPC-File-Sys-Path	1.2.840.
`gPCFunctionalityVersion`	GPC-Functionality-Version	1.2.840.

LDAP Display Name	CN	Attribut
`gPCMachineExtensionNames`	GPC-Machine-Extension-Names	1.2.840.
`gPCUserExtensionNames`	GPC-User-Extension-Names	1.2.840.
`gPLink`	GP-Link	1.2.840.
`gPOptions`	GP-Options	1.2.840.
`groupAttributes`	Group-Attributes	1.2.840.
`groupMembershipSAM`	Group-Membership-SAM	1.2.840.
`groupPriority`	Group-Priority	1.2.840.
`groupsToIgnore`	Groups-to-Ignore	1.2.840.
`groupType`	Group-Type	1.2.840.
`hasMasterNCs`	Has-Master-NCs	1.2.840.
`hasPartialReplicaNCs`	Has-Partial-Replica-NCs	1.2.840.
`helpData16`	Help-Data16	1.2.840.
`helpData32`	Help-Data32	1.2.840.
`helpFileName`	Help-File-Name	1.2.840.
`homeDirectory`	Home-Directory	1.2.840.
`homeDrive`	Home-Drive	1.2.840.
`homePhone`	Phone-Home-Primary	0.9.2342
`homePostalAddress`	Address-Home	1.2.840.
`iconPath`	Icon-Path	1.2.840.
`implementedCategories`	Implemented-Categories	1.2.840.
`indexedScopes`	IndexedScopes	1.2.840.
`info`	Comment	1.2.840.
`initialAuthIncoming`	Initial-Auth-Incoming	1.2.840.
`initialAuthOutgoing`	Initial-Auth-Outgoing	1.2.840.
`initials`	Initials	2.5.4.43
`installUiLevel`	Install-Ui-Level	1.2.840.
`instanceType`	Instance-Type	1.2.840.
`internationalISDNNumber`	International-ISDN-Number	2.5.4.25
`interSiteTopologyFailover`	Inter-Site-Topology-Failover	1.2.840.
`interSiteTopologyGenerator`	Inter-Site-Topology-Generator	1.2.840.
`interSiteTopologyRenew`	Inter-Site-Topology-Renew	1.2.840.
`invocationId`	Invocation-Id	1.2.840.
`ipPhone`	Phone-Ip-Primary	1.2.840.
`ipsecData`	Ipsec-Data	1.2.840.

LDAP Display Name	CN	Attribut
`ipsecDataType`	Ipsec-Data-Type	1.2.840.
`ipsecFilterReference`	Ipsec-Filter-Reference	1.2.840.
`ipsecID`	Ipsec-ID	1.2.840.
`ipsecISAKMPReference`	Ipsec-ISAKMP-Reference	1.2.840.
`ipsecName`	Ipsec-Name	1.2.840.
`iPSECNegotiationPolicyAction`	IPSEC-Negotiation-Policy-Action	1.2.840.
`ipsecNegotiationPolicyReference`	Ipsec-Negotiation-Policy-Reference	1.2.840.
`iPSECNegotiationPolicyType`	IPSEC-Negotiation-Policy-Type	1.2.840.
`ipsecNFAReference`	Ipsec-NFA-Reference	1.2.840.
`ipsecOwnersReference`	Ipsec-Owners-Reference	1.2.840.
`ipsecPolicyReference`	Ipsec-Policy-Reference	1.2.840.
`isCriticalSystemObject`	Is-Critical-System-Object	1.2.840.
`isDefunct`	Is-Defunct	1.2.840.
`isDeleted`	Is-Deleted	1.2.840.
`isEphemeral`	Is-Ephemeral	1.2.840.
`isMemberOfPartialAttributeSet`	Is-Member-Of-Partial-Attribute-Set	1.2.840.
`isPrivilegeHolder`	Is-Privilege-Holder	1.2.840.
`isSingleValued`	Is-Single-Valued	1.2.840.
`keywords`	Keywords	1.2.840.
`knowledgeInformation`	Knowledge-Information	2.5.4.2
`l`	Locality-Name	2.5.4.7
`lastBackupRestorationTime`	Last-Backup-Restoration-Time	1.2.840.
`lastContentIndexed`	Last-Content-Indexed	1.2.840.
`lastKnownParent`	Last-Known-Parent	1.2.840.
`lastLogoff`	Last-Logoff	1.2.840.
`lastLogon`	Last-Logon	1.2.840.
`lastSetTime`	Last-Set-Time	1.2.840.
`lastUpdateSequence`	Last-Update-Sequence	1.2.840.
`lDAPAdminLimits`	LDAP-Admin-Limits	1.2.840.
`lDAPDisplayName`	LDAP-Display-Name	1.2.840.
`lDAPIPDenyList`	LDAP-IPDeny-List	1.2.840.
`legacyExchangeDN`	Legacy-Exchange-DN	1.2.840.
`linkID`	Link-ID	1.2.840.
`linkTrackSecret`	Link-Track-Secret	1.2.840.

LDAP Display Name	CN	Attribut
lmPwdHistory	Lm-Pwd-History	1.2.840.
localeID	Locale-ID	1.2.840.
localizationDisplayId	Localization-Display-Id	1.2.840.
localizedDescription	Localized-Description	1.2.840.
localPolicyFlags	Local-Policy-Flags	1.2.840.
localPolicyReference	Local-Policy-Reference	1.2.840.
location	Location	1.2.840.
lockoutDuration	Lockout-Duration	1.2.840.
lockOutObservationWindow	Lock-Out-Observation-Window	1.2.840.
lockoutThreshold	Lockout-Threshold	1.2.840.
lockoutTime	Lockout-Time	1.2.840.
logonCount	Logon-Count	1.2.840.
logonHours	Logon-Hours	1.2.840.
logonWorkstation	Logon-Workstation	1.2.840.
lSACreationTime	LSA-Creation-Time	1.2.840.
lSAModifiedCount	LSA-Modified-Count	1.2.840.
machineArchitecture	Machine-Architecture	1.2.840.
machinePasswordChangeInterval	Machine-Password-Change-Interval	1.2.840.
machineRole	Machine-Role	1.2.840.
machineWidePolicy	Machine-Wide-Policy	1.2.840.
mail	E-mail-Addresses	0.9.234
mailAddress	SMTP-Mail-Address	1.2.840.
managedBy	Managed-By	1.2.840.
managedObjects	Managed-Objects	1.2.840.
manager	Manager	0.9.234
mAPIID	MAPI-ID	1.2.840.
marshalledInterface	Marshalled-Interface	1.2.840.
masteredBy	Mastered-By	1.2.840.
maxPwdAge	Max-Pwd-Age	1.2.840.
maxRenewAge	Max-Renew-Age	1.2.840.
maxStorage	Max-Storage	1.2.840.
maxTicketAge	Max-Ticket-Age	1.2.840.
mayContain	May-Contain	1.2.840.
meetingAdvertiseScope	meetingAdvertiseScope	1.2.840.

LDAP Display Name	CN	Attribut
meetingApplication	meetingApplication	1.2.840.
meetingBandwidth	meetingBandwidth	1.2.840.
meetingBlob	meetingBlob	1.2.840.
meetingContactInfo	meetingContactInfo	1.2.840.
meetingDescription	meetingDescription	1.2.840.
meetingEndTime	meetingEndTime	1.2.840.
meetingID	meetingID	1.2.840.
meetingIP	meetingIP	1.2.840.
meetingIsEncrypted	meetingIsEncrypted	1.2.840.
meetingKeyword	meetingKeyword	1.2.840.
meetingLanguage	meetingLanguage	1.2.840.
meetingLocation	meetingLocation	1.2.840.
meetingMaxParticipants	meetingMaxParticipants	1.2.840.
meetingName	meetingName	1.2.840.
meetingOriginator	meetingOriginator	1.2.840.
meetingOwner	meetingOwner	1.2.840.
meetingProtocol	meetingProtocol	1.2.840.
meetingRating	meetingRating	1.2.840.
meetingRecurrence	meetingRecurrence	1.2.840.
meetingScope	meetingScope	1.2.840.
meetingStartTime	meetingStartTime	1.2.840.
meetingType	meetingType	1.2.840.
meetingURL	meetingURL	1.2.840.
member	Member	2.5.4.31
memberOf	Is-Member-Of-DL	1.2.840.
mhsORAddress	MHS-OR-Address	1.2.840.
middleName	Other-Name	2.16.84
minPwdAge	Min-Pwd-Age	1.2.840.
minPwdLength	Min-Pwd-Length	1.2.840.
minTicketAge	Min-Ticket-Age	1.2.840.
mobile	Phone-Mobile-Primary	0.9.234
modifiedCount	Modified-Count	1.2.840.
modifiedCountAtLastProm	Modified-Count-At-Last-Prom	1.2.840.
modifyTimeStamp	Modify-Time-Stamp	2.5.18.2

LDAP Display Name	CN	Attribut
`moniker`	Moniker	1.2.840.
`monikerDisplayName`	Moniker-Display-Name	1.2.840.
`moveTreeState`	Move-Tree-State	1.2.840.
`mscopeId`	Mscope-Id	1.2.840.
`mS-DS-ConsistencyChildCount`	MS-DS-Consistency-Child-Count	1.2.840.
`mS-DS-ConsistencyGuid`	MS-DS-Consistency-Guid	1.2.840.
`mS-DS-CreatorSID`	MS-DS-Creator-SID	1.2.840.
`ms-DS-MachineAccountQuota`	MS-DS-Machine-Account-Quota	1.2.840.
`mS-DS-ReplicatesNCReason`	MS-DS-Replicates-NC-Reason	1.2.840.
`msiFileList`	Msi-File-List	1.2.840.
`msiScript`	Msi-Script	1.2.840.
`msiScriptName`	Msi-Script-Name	1.2.840.
`msiScriptPath`	Msi-Script-Path	1.2.840.
`msiScriptSize`	Msi-Script-Size	1.2.840.
`mSMQAuthenticate`	MSMQ-Authenticate	1.2.840.
`mSMQBasePriority`	MSMQ-Base-Priority	1.2.840.
`mSMQComputerType`	MSMQ-Computer-Type	1.2.840.
`mSMQComputerTypeEx`	MSMQ-Computer-Type-Ex	1.2.840.
`mSMQCost`	MSMQ-Cost	1.2.840.
`mSMQCSPName`	MSMQ-CSP-Name	1.2.840.
`mSMQDependentClientService`	MSMQ-Dependent-Client-Service	1.2.840.
`mSMQDependentClientServices`	MSMQ-Dependent-Client-Services	1.2.840.
`mSMQDigests`	MSMQ-Digests	1.2.840.
`mSMQDigestsMig`	MSMQ-Digests-Mig	1.2.840.
`mSMQDsService`	MSMQ-Ds-Service	1.2.840.
`mSMQDsServices`	MSMQ-Ds-Services	1.2.840.
`mSMQEncryptKey`	MSMQ-Encrypt-Key	1.2.840.
`mSMQForeign`	MSMQ-Foreign	1.2.840.
`mSMQInRoutingServers`	MSMQ-In-Routing-Servers	1.2.840.
`mSMQInterval1`	MSMQ-Interval1	1.2.840.
`mSMQInterval2`	MSMQ-Interval2	1.2.840.
`mSMQJournal`	MSMQ-Journal	1.2.840.
`mSMQJournalQuota`	MSMQ-Journal-Quota	1.2.840.
`mSMQLabel`	MSMQ-Label	1.2.840.

LDAP Display Name	CN	Attribut
`mSMQLabelEx`	MSMQ-Label-Ex	1.2.840.
`mSMQLongLived`	MSMQ-Long-Lived	1.2.840.
`mSMQMigrated`	MSMQ-Migrated	1.2.840.
`mSMQNameStyle`	MSMQ-Name-Style	1.2.840.
`mSMQNt4Flags`	MSMQ-Nt4-Flags	1.2.840.
`mSMQNt4Stub`	MSMQ-Nt4-Stub	1.2.840.
`mSMQOSType`	MSMQ-OS-Type	1.2.840.
`mSMQOutRoutingServers`	MSMQ-Out-Routing-Servers	1.2.840.
`mSMQOwnerID`	MSMQ-Owner-ID	1.2.840.
`mSMQPrevSiteGates`	MSMQ-Prev-Site-Gates	1.2.840.
`mSMQPrivacyLevel`	MSMQ-Privacy-Level	1.2.840.
`mSMQQMID`	MSMQ-QM-ID	1.2.840.
`mSMQQueueJournalQuota`	MSMQ-Queue-Journal-Quota	1.2.840.
`mSMQQueueNameExt`	MSMQ-Queue-Name-Ext	1.2.840.
`mSMQQueueQuota`	MSMQ-Queue-Quota	1.2.840.
`mSMQQueueType`	MSMQ-Queue-Type	1.2.840.
`mSMQQuota`	MSMQ-Quota	1.2.840.
`mSMQRoutingService`	MSMQ-Routing-Service	1.2.840.
`mSMQRoutingServices`	MSMQ-Routing-Services	1.2.840.
`mSMQServices`	MSMQ-Services	1.2.840.
`mSMQServiceType`	MSMQ-Service-Type	1.2.840.
`mSMQSignCertificates`	MSMQ-Sign-Certificates	1.2.840.
`mSMQSignCertificatesMig`	MSMQ-Sign-Certificates-Mig	1.2.840.
`mSMQSignKey`	MSMQ-Sign-Key	1.2.840.
`mSMQSite1`	MSMQ-Site-1	1.2.840.
`mSMQSite2`	MSMQ-Site-2	1.2.840.
`mSMQSiteForeign`	MSMQ-Site-Foreign	1.2.840.
`mSMQSiteGates`	MSMQ-Site-Gates	1.2.840.
`mSMQSiteGatesMig`	MSMQ-Site-Gates-Mig	1.2.840.
`mSMQSiteID`	MSMQ-Site-ID	1.2.840.
`mSMQSiteName`	MSMQ-Site-Name	1.2.840.
`mSMQSiteNameEx`	MSMQ-Site-Name-Ex	1.2.840.
`mSMQSites`	MSMQ-Sites	1.2.840.
`mSMQTransactional`	MSMQ-Transactional	1.2.840.

LDAP Display Name	CN	Attribut
mSMQUserSid	MSMQ-User-Sid	1.2.840.
mSMQVersion	MSMQ-Version	1.2.840.
msNPAllowDialin	msNPAllowDialin	1.2.840.
msNPCalledStationID	msNPCalledStationID	1.2.840.
msNPCallingStationID	msNPCallingStationID	1.2.840.
msNPSavedCallingStationID	msNPSavedCallingStationID	1.2.840.
msRADIUSCallbackNumber	msRADIUSCallbackNumber	1.2.840.
msRADIUSFramedIPAddress	msRADIUSFramedIPAddress	1.2.840.
msRADIUSFramedRoute	msRADIUSFramedRoute	1.2.840.
msRADIUSServiceType	msRADIUSServiceType	1.2.840.
msRASSavedCallbackNumber	msRASSavedCallbackNumber	1.2.840.
msRASSavedFramedIPAddress	msRASSavedFramedIPAddress	1.2.840.
msRASSavedFramedRoute	msRASSavedFramedRoute	1.2.840.
msRRASAttribute	ms-RRAS-Attribute	1.2.840.
msRRASVendorAttributeEntry	ms-RRAS-Vendor-Attribute-Entry	1.2.840.
mS-SQL-Alias	MS-SQL-Alias	1.2.840.
mS-SQL-AllowAnonymousSubscription	MS-SQL-AllowAnonymousSubscription	1.2.840.
mS-SQL-AllowImmediateUpdatingSubscription	MS-SQL-AllowImmediateUpdatingSubscription	1.2.840.
mS-SQL-AllowKnownPullSubscription	MS-SQL-AllowKnownPullSubscription	1.2.840.
mS-SQL-AllowQueuedUpdatingSubscription	MS-SQL-AllowQueuedUpdatingSubscription	1.2.840.
mS-SQL-AllowSnapshotFilesFTPDownloading	MS-SQL-AllowSnapshotFilesFTPDownloading	1.2.840.
mS-SQL-AppleTalk	MS-SQL-AppleTalk	1.2.840.
mS-SQL-Applications	MS-SQL-Applications	1.2.840.
mS-SQL-Build	MS-SQL-Build	1.2.840.
mS-SQL-CharacterSet	MS-SQL-CharacterSet	1.2.840.
mS-SQL-Clustered	MS-SQL-Clustered	1.2.840.
mS-SQL-ConnectionURL	MS-SQL-ConnectionURL	1.2.840.
mS-SQL-Contact	MS-SQL-Contact	1.2.840.
mS-SQL-CreationDate	MS-SQL-CreationDate	1.2.840.
mS-SQL-Database	MS-SQL-Database	1.2.840.

LDAP Display Name	CN	Attribut
`mS-SQL-Description`	MS-SQL-Description	1.2.840.
`mS-SQL-GPSHeight`	MS-SQL-GPSHeight	1.2.840.
`mS-SQL-GPSLatitude`	MS-SQL-GPSLatitude	1.2.840.
`mS-SQL-GPSLongitude`	MS-SQL-GPSLongitude	1.2.840.
`mS-SQL-InformationDirectory`	MS-SQL-InformationDirectory	1.2.840.
`mS-SQL-InformationURL`	MS-SQL-InformationURL	1.2.840.
`mS-SQL-Keywords`	MS-SQL-Keywords	1.2.840.
`mS-SQL-Language`	MS-SQL-Language	1.2.840.
`mS-SQL-LastBackupDate`	MS-SQL-LastBackupDate	1.2.840.
`mS-SQL-LastDiagnosticDate`	MS-SQL-LastDiagnosticDate	1.2.840.
`mS-SQL-LastUpdatedDate`	MS-SQL-LastUpdatedDate	1.2.840.
`mS-SQL-Location`	MS-SQL-Location	1.2.840.
`mS-SQL-Memory`	MS-SQL-Memory	1.2.840.
`mS-SQL-MultiProtocol`	MS-SQL-MultiProtocol	1.2.840.
`mS-SQL-Name`	MS-SQL-Name	1.2.840.
`mS-SQL-NamedPipe`	MS-SQL-NamedPipe	1.2.840.
`mS-SQL-PublicationURL`	MS-SQL-PublicationURL	1.2.840.
`mS-SQL-Publisher`	MS-SQL-Publisher	1.2.840.
`mS-SQL-RegisteredOwner`	MS-SQL-RegisteredOwner	1.2.840.
`mS-SQL-ServiceAccount`	MS-SQL-ServiceAccount	1.2.840.
`mS-SQL-Size`	MS-SQL-Size	1.2.840.
`mS-SQL-SortOrder`	MS-SQL-SortOrder	1.2.840.
`mS-SQL-SPX`	MS-SQL-SPX	1.2.840.
`mS-SQL-Status`	MS-SQL-Status	1.2.840.
`mS-SQL-TCPIP`	MS-SQL-TCPIP	1.2.840.
`mS-SQL-ThirdParty`	MS-SQL-ThirdParty	1.2.840.
`mS-SQL-Type`	MS-SQL-Type	1.2.840.
`mS-SQL-UnicodeSortOrder`	MS-SQL-UnicodeSortOrder	1.2.840.
`mS-SQL-Version`	MS-SQL-Version	1.2.840.
`mS-SQL-Vines`	MS-SQL-Vines	1.2.840.
`mustContain`	Must-Contain	1.2.840.
`name`	RDN	1.2.840.
`nameServiceFlags`	Name-Service-Flags	1.2.840.
`nCName`	NC-Name	1.2.840.

LDAP Display Name	CN	Attribut
`nETBIOSName`	NETBIOS-Name	1.2.840.
`netbootAllowNewClients`	netboot-Allow-New-Clients	1.2.840.
`netbootAnswerOnlyValidClients`	netboot-Answer-Only-Valid-Clients	1.2.840.
`netbootAnswerRequests`	netboot-Answer-Requests	1.2.840.
`netbootCurrentClientCount`	netboot-Current-Client-Count	1.2.840.
`netbootGUID`	Netboot-GUID	1.2.840.
`netbootInitialization`	Netboot-Initialization	1.2.840.
`netbootIntelliMirrorOSes`	netboot-IntelliMirror-OSes	1.2.840.
`netbootLimitClients`	netboot-Limit-Clients	1.2.840.
`netbootLocallyInstalledOSes`	netboot-Locally-Installed-OSes	1.2.840.
`netbootMachineFilePath`	Netboot-Machine-File-Path	1.2.840.
`netbootMaxClients`	netboot-Max-Clients	1.2.840.
`netbootMirrorDataFile`	Netboot-Mirror-Data-File	1.2.840.
`netbootNewMachineNamingPolicy`	netboot-New-Machine-Naming-Policy	1.2.840.
`netbootNewMachineOU`	netboot-New-Machine-OU	1.2.840.
`netbootSCPBL`	netboot-SCP-BL	1.2.840.
`netbootServer`	netboot-Server	1.2.840.
`netbootSIFFile`	Netboot-SIF-File	1.2.840.
`netbootTools`	netboot-Tools	1.2.840.
`networkAddress`	Network-Address	1.2.840.
`nextLevelStore`	Next-Level-Store	1.2.840.
`nextRid`	Next-Rid	1.2.840.
`nonSecurityMember`	Non-Security-Member	1.2.840.
`nonSecurityMemberBL`	Non-Security-Member-BL	1.2.840.
`notes`	Additional-Information	1.2.840.
`notificationList`	Notification-List	1.2.840.
`nTGroupMembers`	NT-Group-Members	1.2.840.
`nTMixedDomain`	NT-Mixed-Domain	1.2.840.
`ntPwdHistory`	Nt-Pwd-History	1.2.840.
`nTSecurityDescriptor`	NT-Security-Descriptor	1.2.840.
`o`	Organization-Name	2.5.4.10
`objectCategory`	Object-Category	1.2.840.
`objectClass`	Object-Class	2.5.4.0
`objectClassCategory`	Object-Class-Category	1.2.840.

LDAP Display Name	CN	Attribut
`objectClasses`	Object-Classes	2.5.21.6
`objectCount`	Object-Count	1.2.840.
`objectGUID`	Object-Guid	1.2.840.
`objectSid`	Object-Sid	1.2.840.
`objectVersion`	Object-Version	1.2.840.
`oEMInformation`	OEM-Information	1.2.840.
`oMObjectClass`	OM-Object-Class	1.2.840.
`oMSyntax`	OM-Syntax	1.2.840.
`oMTGuid`	OMT-Guid	1.2.840.
`oMTIndxGuid`	OMT-Indx-Guid	1.2.840.
`operatingSystem`	Operating-System	1.2.840.
`operatingSystemHotfix`	Operating-System-Hotfix	1.2.840.
`operatingSystemServicePack`	Operating-System-Service-Pack	1.2.840.
`operatingSystemVersion`	Operating-System-Version	1.2.840.
`operatorCount`	Operator-Count	1.2.840.
`optionDescription`	Option-Description	1.2.840.
`options`	Options	1.2.840.
`optionsLocation`	Options-Location	1.2.840.
`originalDisplayTable`	Original-Display-Table	1.2.840.
`originalDisplayTableMSDOS`	Original-Display-Table-MSDOS	1.2.840.
`otherFacsimileTelephoneNumber`	Phone-Fax-Other	1.2.840.
`otherHomePhone`	Phone-Home-Other	1.2.840.
`otherIpPhone`	Phone-Ip-Other	1.2.840.
`otherLoginWorkstations`	Other-Login-Workstations	1.2.840.
`otherMailbox`	Other-Mailbox	1.2.840.
`otherMobile`	Phone-Mobile-Other	1.2.840.
`otherPager`	Phone-Pager-Other	1.2.840.
`otherTelephone`	Phone-Office-Other	1.2.840.
`otherWellKnownObjects`	Other-Well-Known-Objects	1.2.840.
`ou`	Organizational-Unit-Name	2.5.4.11
`owner`	Owner	2.5.4.32
`packageFlags`	Package-Flags	1.2.840.
`packageName`	Package-Name	1.2.840.
`packageType`	Package-Type	1.2.840.

LDAP Display Name	CN	Attribut
`pager`	Phone-Pager-Primary	0.9.2342
`parentCA`	Parent-CA	1.2.840.
`parentCACertificateChain`	Parent-CA-Certificate-Chain	1.2.840.
`parentGUID`	Parent-GUID	1.2.840.
`partialAttributeDeletionList`	Partial-Attribute-Deletion-List	1.2.840.
`partialAttributeSet`	Partial-Attribute-Set	1.2.840.
`pekKeyChangeInterval`	Pek-Key-Change-Interval	1.2.840.
`pekList`	Pek-List	1.2.840.
`pendingCACertificates`	Pending-CA-Certificates	1.2.840.
`pendingParentCA`	Pending-Parent-CA	1.2.840.
`perMsgDialogDisplayTable`	Per-Msg-Dialog-Display-Table	1.2.840.
`perRecipDialogDisplayTable`	Per-Recip-Dialog-Display-Table	1.2.840.
`personalTitle`	Personal-Title	1.2.840.
`physicalDeliveryOfficeName`	Physical-Delivery-Office-Name	2.5.4.19
`physicalLocationObject`	Physical-Location-Object	1.2.840.
`pKICriticalExtensions`	PKI-Critical-Extensions	1.2.840.
`pKIDefaultCSPs`	PKI-Default-CSPs	1.2.840.
`pKIDefaultKeySpec`	PKI-Default-Key-Spec	1.2.840.
`pKIEnrollmentAccess`	PKI-Enrollment-Access	1.2.840.
`pKIExpirationPeriod`	PKI-Expiration-Period	1.2.840.
`pKIExtendedKeyUsage`	PKI-Extended-Key-Usage	1.2.840.
`pKIKeyUsage`	PKI-Key-Usage	1.2.840.
`pKIMaxIssuingDepth`	PKI-Max-Issuing-Depth	1.2.840.
`pKIOverlapPeriod`	PKI-Overlap-Period	1.2.840.
`pKT`	PKT	1.2.840.
`pKTGuid`	PKT-Guid	1.2.840.
`policyReplicationFlags`	Policy-Replication-Flags	1.2.840.
`portName`	Port-Name	1.2.840.
`possibleInferiors`	Possible-Inferiors	1.2.840.
`possSuperiors`	Poss-Superiors	1.2.840.
`postalAddress`	Postal-Address	2.5.4.16
`postalCode`	Postal-Code	2.5.4.17
`postOfficeBox`	Post-Office-Box	2.5.4.18
`preferredDeliveryMethod`	Preferred-Delivery-Method	2.5.4.28

LDAP Display Name	CN	Attribut
`preferredOU`	Preferred-OU	1.2.840.
`prefixMap`	Prefix-Map	1.2.840.
`presentationAddress`	Presentation-Address	2.5.4.29
`previousCACertificates`	Previous-CA-Certificates	1.2.840.
`previousParentCA`	Previous-Parent-CA	1.2.840.
`primaryGroupID`	Primary-Group-ID	1.2.840.
`primaryGroupToken`	Primary-Group-Token	1.2.840.
`primaryInternationalISDNNumber`	Phone-ISDN-Primary	1.2.840.
`primaryTelexNumber`	Telex-Primary	1.2.840.
`printAttributes`	Print-Attributes	1.2.840.
`printBinNames`	Print-Bin-Names	1.2.840.
`printCollate`	Print-Collate	1.2.840.
`printColor`	Print-Color	1.2.840.
`printDuplexSupported`	Print-Duplex-Supported	1.2.840.
`printEndTime`	Print-End-Time	1.2.840.
`printerName`	Printer-Name	1.2.840.
`printFormName`	Print-Form-Name	1.2.840.
`printKeepPrintedJobs`	Print-Keep-Printed-Jobs	1.2.840.
`printLanguage`	Print-Language	1.2.840.
`printMACAddress`	Print-MAC-Address	1.2.840.
`printMaxCopies`	Print-Max-Copies	1.2.840.
`printMaxResolutionSupported`	Print-Max-Resolution-Supported	1.2.840.
`printMaxXExtent`	Print-Max-X-Extent	1.2.840.
`printMaxYExtent`	Print-Max-Y-Extent	1.2.840.
`printMediaReady`	Print-Media-Ready	1.2.840.
`printMediaSupported`	Print-Media-Supported	1.2.840.
`printMemory`	Print-Memory	1.2.840.
`printMinXExtent`	Print-Min-X-Extent	1.2.840.
`printMinYExtent`	Print-Min-Y-Extent	1.2.840.
`printNetworkAddress`	Print-Network-Address	1.2.840.
`printNotify`	Print-Notify	1.2.840.
`printNumberUp`	Print-Number-Up	1.2.840.
`printOrientationsSupported`	Print-Orientations-Supported	1.2.840.
`printOwner`	Print-Owner	1.2.840.

LDAP Display Name	CN	Attribut
`printPagesPerMinute`	Print-Pages-Per-Minute	1.2.840.
`printRate`	Print-Rate	1.2.840.
`printRateUnit`	Print-Rate-Unit	1.2.840.
`printSeparatorFile`	Print-Separator-File	1.2.840.
`printShareName`	Print-Share-Name	1.2.840.
`printSpooling`	Print-Spooling	1.2.840.
`printStaplingSupported`	Print-Stapling-Supported	1.2.840.
`printStartTime`	Print-Start-Time	1.2.840.
`printStatus`	Print-Status	1.2.840.
`priority`	Priority	1.2.840.
`priorSetTime`	Prior-Set-Time	1.2.840.
`priorValue`	Prior-Value	1.2.840.

Object Identifiers (OIDs)

We can also use matching rule Object Identifiers (OIDs) with LDAP filters as listed in this Search Filter Syntax document from Microsoft:

Matching rule OID	String identifier	Description
1.2.840.113556.1.4.803	LDAP_MATCHING_RULE_BIT_AND	A match is found only if all bits from the attribute match the value. This rule is equivalent to a bitwise **AND** operator.
1.2.840.113556.1.4.804	LDAP_MATCHING_RULE_BIT_OR	A match is found if any bits from the attribute match the value. This rule is equivalent to a bitwise **OR** operator.

Matching rule OID	String identifier	Description
1.2.840.113556.1.4.1941	LDAP_MATCHING_RULE_IN_CHAIN	This rule is limited to filters that apply to the DN. This is a special "extended" match operator that walks the chain of ancestry in objects all the way to the root until it finds a match.

We can clarify the above OIDs with some examples. Let's take the following LDAP query:

```
(&(objectCategory=person)(objectClass=user)
(userAccountControl:1.2.840.113556.1.4.803:=2))
```

This query will return all administratively disabled user accounts, or ACCOUNTDISABLE (2). We can combine this query as an LDAP search filter with the " `Get-ADUser` " cmdlet against our target domain. The LDAP query can be shortened as follows:

LDAP Query - Filter Disabled User Accounts

```
PS C:\htb> Get-ADUser -LDAPFilter
'(userAccountControl:1.2.840.113556.1.4.803:=2)' | select name

name
----
Guest
DefaultAccount
krbtgt
Exchange Online-ApplicationAccount
SystemMailbox{1f05a927-35b9-4cc9-bbe1-11e28cddb180}
SystemMailbox{bb558c35-97f1-4cb9-8ff7-d53741dc928c}
SystemMailbox{e0dc1c29-89c3-4034-b678-e6c29d823ed9}
DiscoverySearchMailbox {D919BA05-46A6-415f-80AD-7E09334BB852}
Migration.8f3e7716-2011-43e4-96b1-aba62d229136
FederatedEmail.4c1f4d8b-8179-4148-93bf-00a95fa1e042
SystemMailbox{D0E409A0-AF9B-4720-92FE-AAC869B0D201}
SystemMailbox{2CE34405-31BE-455D-89D7-A7C7DA7A0DAA}
SystemMailbox{8cc370d3-822a-4ab8-a926-bb94bd0641a9}
```

Now let's look at an example of the extensible match rule " `1.2.840.113556.1.4.1941` ". Consider the following query:

```
(member:1.2.840.113556.1.4.1941:=CN=Harry Jones,OU=Network
Ops,OU=IT,OU=Employees,DC=INLANEFREIGHT,DC=LOCAL)
```

This matching rule will find all groups that the user `Harry Jones` (" `CN=Harry Jones,OU=Network Ops,OU=IT,OU=Employees,DC=INLANEFREIGHT,DC=LOCAL` ") is a member of. Using this filter with the " `Get-ADGroup` " cmdlet gives us the following output:

LDAP Query - Find All Groups

```
PS C:\htb> Get-ADGroup -LDAPFilter
'(member:1.2.840.113556.1.4.1941:=CN=Harry Jones,OU=Network
Ops,OU=IT,OU=Employees,DC=INLANEFREIGHT,DC=LOCAL)' | select Name

Name
----
Administrators
Backup Operators
Domain Admins
Denied RODC Password Replication Group
LAPS Admins
Security Operations
Help Desk
Network Team
```

Filter Types, Item Types & Escaped Characters

With LDAP search filters, we have the following four filter types:

Operator	Meaning
=	Equal to
~=	Approximately equal to
>=	Greater than or equal to
<=	Less than or equal to

And we have four item types:

Type	Meaning
=	Simple
=*	Present
=something*	Substring
Extensible	varies depending on type

Finally, the following characters must be escaped if used in an LDAP filter:

Character	Represented as Hex
*	\2a
(	\28
)	\29
\	\5c
NUL	\00

Example LDAP Filters

Let's build a few more LDAP filters to use against our test domain.

We can use the filter " `(&(objectCategory=user)(description=*))` " to find all user accounts that do not have a blank `description` field. This is a useful search that should be performed on every internal network assessment as it not uncommon to find passwords for users stored in the user description attribute in AD (which can be read by all AD users).

Combining this with the " `Get-ADUser` " cmdlet, we can search for all domain users that do not have a blank description field and, in this case, find a service account password!

LDAP Query - Description Field

```
PS C:\htb> Get-ADUser -Properties * -LDAPFilter '(&(objectCategory=user)
(description=*))' | select samaccountname,description

samaccountname description
-------------- -----------
Administrator  Built-in account for administering the computer/domain
Guest          Built-in account for guest access to the computer/domain
```

```
DefaultAccount A user account managed by the system.
krbtgt         Key Distribution Center Service Account
svc-sccm       **Do not change password** 03/04/2015 N3ssu$_svc2014!
```

This filter " `(userAccountControl:1.2.840.113556.1.4.803:=524288)` " can be used to find all users or computers marked as `trusted for delegation`, or unconstrained delegation, which will be covered in a later module on Kerberos Attacks. We can enumerate users with the help of this LDAP filter:

LDAP Query - Find Trusted Users

```
PS C:\htb> Get-ADUser -Properties * -LDAPFilter
'(userAccountControl:1.2.840.113556.1.4.803:=524288)' | select
Name,memberof, servicePrincipalName,TrustedForDelegation | fl

Name                 : sqldev
memberof             : {CN=Protected
Users,CN=Users,DC=INLANEFREIGHT,DC=LOCAL}
servicePrincipalName : {MSSQL_svc_dev/inlanefreight.local:1443}
TrustedForDelegation : True
```

We can enumerate computers with this setting as well:

LDAP Query - Find Trusted Computers

```
PS C:\htb> Get-ADComputer -Properties * -LDAPFilter
'(userAccountControl:1.2.840.113556.1.4.803:=524288)' | select
DistinguishedName,servicePrincipalName,TrustedForDelegation | fl

DistinguishedName    : CN=DC01,OU=Domain
Controllers,DC=INLANEFREIGHT,DC=LOCAL
servicePrincipalName : {exchangeAB/DC01,
exchangeAB/DC01.INLANEFREIGHT.LOCAL, TERMSRV/DC01,
                       TERMSRV/DC01.INLANEFREIGHT.LOCAL...}
TrustedForDelegation : True

DistinguishedName    : CN=SQL01,OU=SQL
Servers,OU=Servers,DC=INLANEFREIGHT,DC=LOCAL
servicePrincipalName : {MSSQLsvc/SQL01.INLANEFREIGHT.LOCAL:1433,
TERMSRV/SQL01, TERMSRV/SQL01.INLANEFREIGHT.LOCAL,
                       RestrictedKrbHost/SQL01...}
TrustedForDelegation : True
```

Lastly, let's search for all users with the " `adminCount` " attribute set to `1` whose " `useraccountcontrol` " attribute is set with the flag " `PASSWD_NOTREQD` ," meaning that the account can have a blank password set. To do this, we must combine two LDAP search filters as follows:

```
(&(objectCategory=person)(objectClass=user)
(userAccountControl:1.2.840.113556.1.4.803:=32))(adminCount=1)
```

LDAP Query - Users With Blank Password

```
PS C:\htb> Get-AdUser -LDAPFilter '(&(objectCategory=person)
(objectClass=user)(userAccountControl:1.2.840.113556.1.4.803:=32))
(adminCount=1)' -Properties * | select name,memberof | fl

name     : Jenna Smith
memberof : CN=Schema Admins,CN=Users,DC=INLANEFREIGHT,DC=LOCAL

name     : Harry Jones
memberof : {CN=Network Team,CN=Users,DC=INLANEFREIGHT,DC=LOCAL, CN=Help
Desk,OU=Microsoft Exchange Security
           Groups,DC=INLANEFREIGHT,DC=LOCAL, CN=Security
Operations,CN=Users,DC=INLANEFREIGHT,DC=LOCAL, CN=LAPS
           Admins,CN=Users,DC=INLANEFREIGHT,DC=LOCAL...}
```

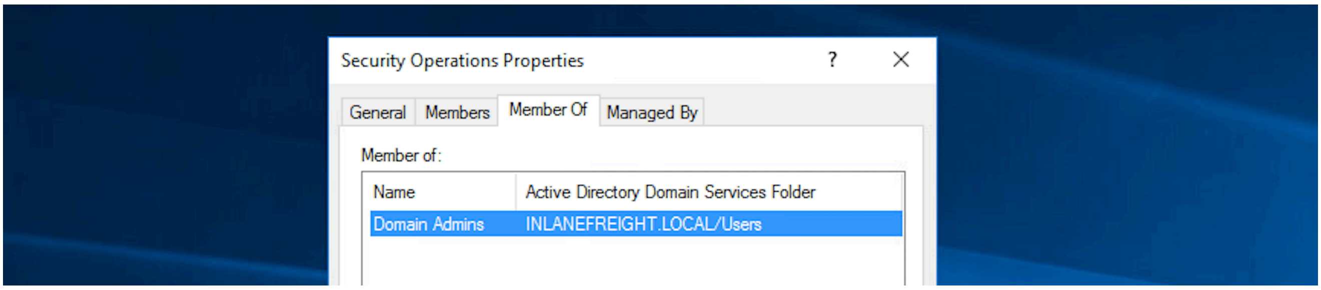
While uncommon, we find accounts without a password set from time to time, so it is always important to enumerate accounts with the `PASSWD_NOTREQD` flag set and check to see if they indeed do not have a password set. This could happen intentionally (perhaps as a timesaver) or accidentally if a user with this flag set changes their password via command line and accidentally presses enter before typing in a password. All organizations should perform periodic account audits and remove this flag from any accounts that have no valid business reason to have it set.

Try out building some filters of your own. This guide Active Directory: LDAP Syntax Filters is a great starting point.

Recursive Match

We can use the " `RecursiveMatch` " parameter in a similar way that we use the matching rule OID " `1.2.840.113556.1.4.1941` ". A good example of this is to find all of the groups that an AD user is a part of, both directly and indirectly. This is also known as "nested group membership." For example, the user `bob.smith` may not be a direct member of the `Domain`

`Admins` group but has `derivative` Domain Admin rights because the group `Security Operations` is a member of the `Domain Admins` group. We can see this graphically by looking at `Active Directory Computers and Users`.


Security Operations Properties

Member of:

Name	Active Directory Domain Services Folder
Domain Admins	INLANEFREIGHT.LOCAL/Users

We can enumerate this with PowerShell several ways, one way being the " `Get-ADGroupMember` " cmdlet.

PowerShell - Members Of Security Operations

```
PS C:\htb> Get-ADGroupMember -Identity "Security Operations"

distinguishedName : CN=Harry Jones,OU=Network
Ops,OU=IT,OU=Employees,DC=INLANEFREIGHT,DC=LOCAL
name              : Harry Jones
objectClass       : user
objectGUID        : f6d9b03e-7056-478b-a737-6c3298d18b9d
SamAccountName    : harry.jones
SID               : S-1-5-21-2974783224-3764228556-2640795941-2040
```

As we can see above, the `Security Operations` group is indeed "nested" within the `Domain Admins` group. Therefore any of its members are effectively Domain Admins.

Searching for a user's group membership using `Get-ADUser` focusing on the property `memberof` will not directly show this information.

PowerShell - User's Group Membership

```
PS C:\htb> Get-ADUser -Identity harry.jones -Properties * | select
memberof | ft -Wrap

memberof
--------
{CN=Network Team,CN=Users,DC=INLANEFREIGHT,DC=LOCAL, CN=Help
Desk,OU=Microsoft Exchange Security
Groups,DC=INLANEFREIGHT,DC=LOCAL, CN=Security
Operations,CN=Users,DC=INLANEFREIGHT,DC=LOCAL, CN=LAPS
Admins,CN=Users,DC=INLANEFREIGHT,DC=LOCAL...}
```

We can find nested group membership with the matching rule OID and the `RecursiveMatch` parameter, as seen in the following examples. The first example shows an AD filter and the `RecursiveMatch` to recursively query for all groups that the user `harry.jones` is a member of.

PowerShell - All Groups of User

```
PS C:\htb> Get-ADGroup -Filter 'member -RecursiveMatch "CN=Harry
Jones,OU=Network Ops,OU=IT,OU=Employees,DC=INLANEFREIGHT,DC=LOCAL"' |
select name

name
----
Administrators
Backup Operators
Domain Admins
Denied RODC Password Replication Group
LAPS Admins
Security Operations
Help Desk
Network Team
```

Another way to return this same information is by using an `LDAPFilter` and the matching rule OID.

LDAP Query - All Groups of User

```
PS C:\htb> Get-ADGroup -LDAPFilter
'(member:1.2.840.113556.1.4.1941:=CN=Harry Jones,OU=Network
Ops,OU=IT,OU=Employees,DC=INLANEFREIGHT,DC=LOCAL)' |select Name

Name
----
Administrators
Backup Operators
Domain Admins
Denied RODC Password Replication Group
LAPS Admins
Security Operations
Help Desk
Network Team
```

As shown in the above examples, searching recursively in AD can help us enumerate information that standard search queries do not show. Enumerating nested group membership is very important. We may uncover serious misconfigurations within the target

AD environment that would otherwise go unnoticed, especially in large organizations with thousands of objects in AD. We will see other ways to enumerate this information and even ways of presenting it in a graphical format, but `RecursiveMatch` is a powerful search parameter that should not be overlooked.

SearchBase and SearchScope Parameters

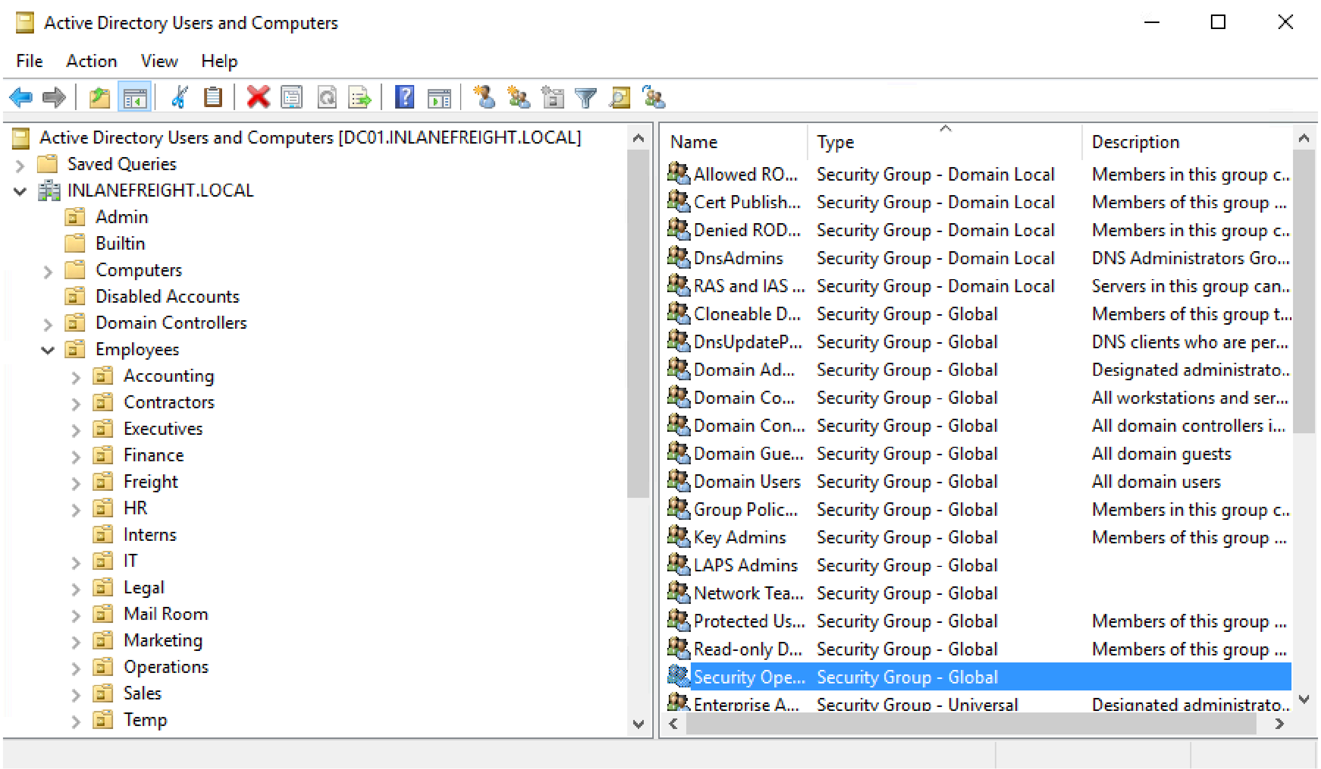
Even small Active Directory environments can contain hundreds if not thousands of objects. Active Directory can grow very quickly as users, groups, computers, OUs, etc., are added, and ACLs are set up, which creates an increasingly complex web of relationships. We may also find ourselves in a vast environment, 10-20 years old, with 10s of thousands of objects. Enumerating these environments can become an unwieldy task, so we need to refine our searches.

We can improve the performance of our enumeration commands and scripts and reduce the volume of objects returned by scoping our searches using the " `SearchBase` " parameter. This parameter specifies an Active Directory path to search under and allows us to begin searching for a user account in a specific OU. The " `SearchBase` " parameter accepts an OUs distinguished name (DN) such as `"OU=Employees,DC=INLANEFREIGHT,DC=LOCAL"` .

" `SearchScope` " allows us to define how deep into the OU hierarchy we would like to search. This parameter has three levels:

Name	Level	Description
Base	0	The object is specified as the `SearchBase` . For example, if we ask for all users in an OU defining a base scope, we get no results. If we specify a user or use `Get-ADObject` we get just that user or object returned.
OneLevel	1	Searches for objects in the container defined by the `SearchBase` but not in any sub-containers.
SubTree	2	Searches for objects contained by the `SearchBase` and all child containers, including their children, recursively all the way down the AD hierarchy.

When querying AD using " `SearchScope` " we can specify the name or the number (i.e., `SearchScope Onelevel` is interpreted the same as " `SearchScope 1` ".)



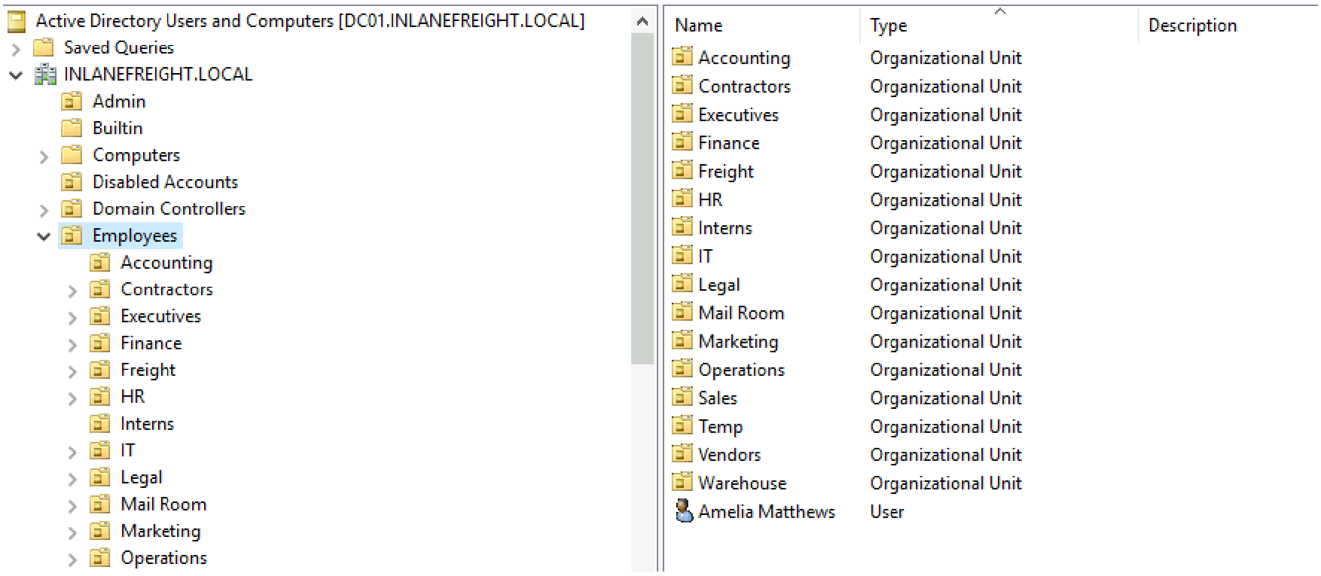
- Active Directory Users and Computers [DC01.INLANEFREIGHT.LOCAL]
  - Saved Queries
  - INLANEFREIGHT.LOCAL
    - Admin
    - Builtin
    - Computers
    - Disabled Accounts
    - Domain Controllers
    - Employees
      - Accounting
      - Contractors
      - Executives
      - Finance
      - Freight
      - HR
      - Interns
      - IT
      - Legal
      - Mail Room
      - Marketing
      - Operations
      - Sales
      - Temp

Name	Type	Description
Allowed RO...	Security Group - Domain Local	Members in this group c...
Cert Publish...	Security Group - Domain Local	Members of this group ...
Denied ROD...	Security Group - Domain Local	Members in this group c...
DnsAdmins	Security Group - Domain Local	DNS Administrators Gro...
RAS and IAS ...	Security Group - Domain Local	Servers in this group can...
Cloneable D...	Security Group - Global	Members of this group t...
DnsUpdateP...	Security Group - Global	DNS clients who are per...
Domain Ad...	Security Group - Global	Designated administrato...
Domain Co...	Security Group - Global	All workstations and ser...
Domain Con...	Security Group - Global	All domain controllers i...
Domain Gue...	Security Group - Global	All domain guests
Domain Users	Security Group - Global	All domain users
Group Polic...	Security Group - Global	Members in this group c...
Key Admins	Security Group - Global	Members of this group ...
LAPS Admins	Security Group - Global	
Network Tea...	Security Group - Global	
Protected Us...	Security Group - Global	Members of this group ...
Read-only D...	Security Group - Global	Members of this group ...
Security Ope...	Security Group - Global	
Enterprise A...	Security Group - Universal	Designated administrato...

In the above example, with the SearchBase set to OU=Employees,DC=INLANEFREIGHT,DC=LOCAL, a `SearchScope` set to `Base` would attempt to query the OU object (`Employees`) itself. A `SearchScope` set to `OneLevel` would search within the `Employees` OU only. Finally, a `SearchScope` set to `SubTree` would query the `Employees` OU and all of the OUs underneath it, such as `Accounting` , `Contractors` , etc. OUs under those OUs (child containers).

SearchBase and Search Scope Parameters Examples

Let's look at some examples to illustrate the difference between `Base` , `OneLevel` , and `Subtree` . For these examples, we will focus on the `Employees` OU. In the screenshot of `Active Directory Users and Computers` below `Employees` is the `Base` , and specifying it with `Get-ADUser` will return nothing. `OneLevel` will return just the user `Amelia Matthews` , and `SubTree` will return all users in all child containers under the `Employees` container.



Active Directory Users and Computers [DC01.INLANEFREIGHT.LOCAL]
- Saved Queries
- INLANEFREIGHT.LOCAL
  - Admin
  - Builtin
  - Computers
  - Disabled Accounts
  - Domain Controllers
  - Employees
    - Accounting
    - Contractors
    - Executives
    - Finance
    - Freight
    - HR
    - Interns
    - IT
    - Legal
    - Mail Room
    - Marketing
    - Operations

Name	Type	Description
Accounting	Organizational Unit	
Contractors	Organizational Unit	
Executives	Organizational Unit	
Finance	Organizational Unit	
Freight	Organizational Unit	
HR	Organizational Unit	
Interns	Organizational Unit	
IT	Organizational Unit	
Legal	Organizational Unit	
Mail Room	Organizational Unit	
Marketing	Organizational Unit	
Operations	Organizational Unit	
Sales	Organizational Unit	
Temp	Organizational Unit	
Vendors	Organizational Unit	
Warehouse	Organizational Unit	
Amelia Matthews	User	

We can confirm these results using PowerShell. For reference purposes, let's get a count of all AD users under the `Employees` OU, which shows 970 users.

PowerShell - Count of All AD Users

```
PS C:\htb> (Get-ADUser -SearchBase
"OU=Employees,DC=INLANEFREIGHT,DC=LOCAL" -Filter *).count

970
```

As expected, specifying a SearchScope of `Base` will return nothing.

PowerShell - SearchScope Base

```
PS C:\htb> Get-ADUser -SearchBase "OU=Employees,DC=INLANEFREIGHT,DC=LOCAL"
-SearchScope Base -Filter *
PS C:\htb>
```

However, if we specify " `Base` " with " `Get-ADObject` " we will get just the object (Employees OU) returned to us.

PowerShell - SearchScope Base OU Object

```
PS C:\htb> Get-ADObject -SearchBase
"OU=Employees,DC=INLANEFREIGHT,DC=LOCAL" -SearchScope Base -Filter *

DistinguishedName                       Name      ObjectClass
ObjectGUID
-----------------                       ----      -----------        ------
----
```

```
OU=Employees,DC=INLANEFREIGHT,DC=LOCAL Employees organizationalUnit
34f42767-8a2e-493f-afc6-556bdc0b1087
```

If we specify `OneLevel` as the SearchScope, we get one user returned to us, as expected per the image above.

PowerShell - Searchscope OneLevel

```
PS C:\htb> Get-ADUser -SearchBase "OU=Employees,DC=INLANEFREIGHT,DC=LOCAL"
-SearchScope OneLevel -Filter *

DistinguishedName : CN=Amelia
Matthews,OU=Employees,DC=INLANEFREIGHT,DC=LOCAL
Enabled           : True
GivenName         : amelia
Name              : Amelia Matthews
ObjectClass       : user
ObjectGUID        : 3f04328f-eb2e-487c-85fe-58dd598159c0
SamAccountName    : amelia.matthews
SID               : S-1-5-21-2974783224-3764228556-2640795941-1412
Surname           : matthews
UserPrincipalName : amelia.matthews@inlanefreight
```

As stated above, the `SearchScope` values are interchangeable, so the same result is returned when specifying `1` as the `SearchScope` value.

PowerShell - Searchscope 1

```
PS C:\htb> Get-ADUser -SearchBase "OU=Employees,DC=INLANEFREIGHT,DC=LOCAL"
-SearchScope 1 -Filter *

DistinguishedName : CN=Amelia
Matthews,OU=Employees,DC=INLANEFREIGHT,DC=LOCAL
Enabled           : True
GivenName         : amelia
Name              : Amelia Matthews
ObjectClass       : user
ObjectGUID        : 3f04328f-eb2e-487c-85fe-58dd598159c0
SamAccountName    : amelia.matthews
SID               : S-1-5-21-2974783224-3764228556-2640795941-1412
Surname           : matthews
UserPrincipalName : amelia.matthews@inlanefreight
```

Finally, if we specify `Subtree` as the SearchBase, we will get all objects within all child containers, which matches the user count we established above.

PowerShell - Searchscope Subtree

```
PS C:\htb> (Get-ADUser -SearchBase
"OU=Employees,DC=INLANEFREIGHT,DC=LOCAL" -SearchScope Subtree -Filter
*).count

970
```

Conclusion

This section, as well as the PowerShell Filters section, covered the many ways we can use search filters combined with built-in AD cmdlets to enhance our enumeration by "living off the land." In later sections, we will cover tools that make enumeration much quicker and easier and be combined with filters to be even more powerful. Regardless of if we are using built-in tools, custom scripts or, third-party tools, it is important to understand what they are doing and to be able to understand and use the output of our enumeration to help us achieve our goal.

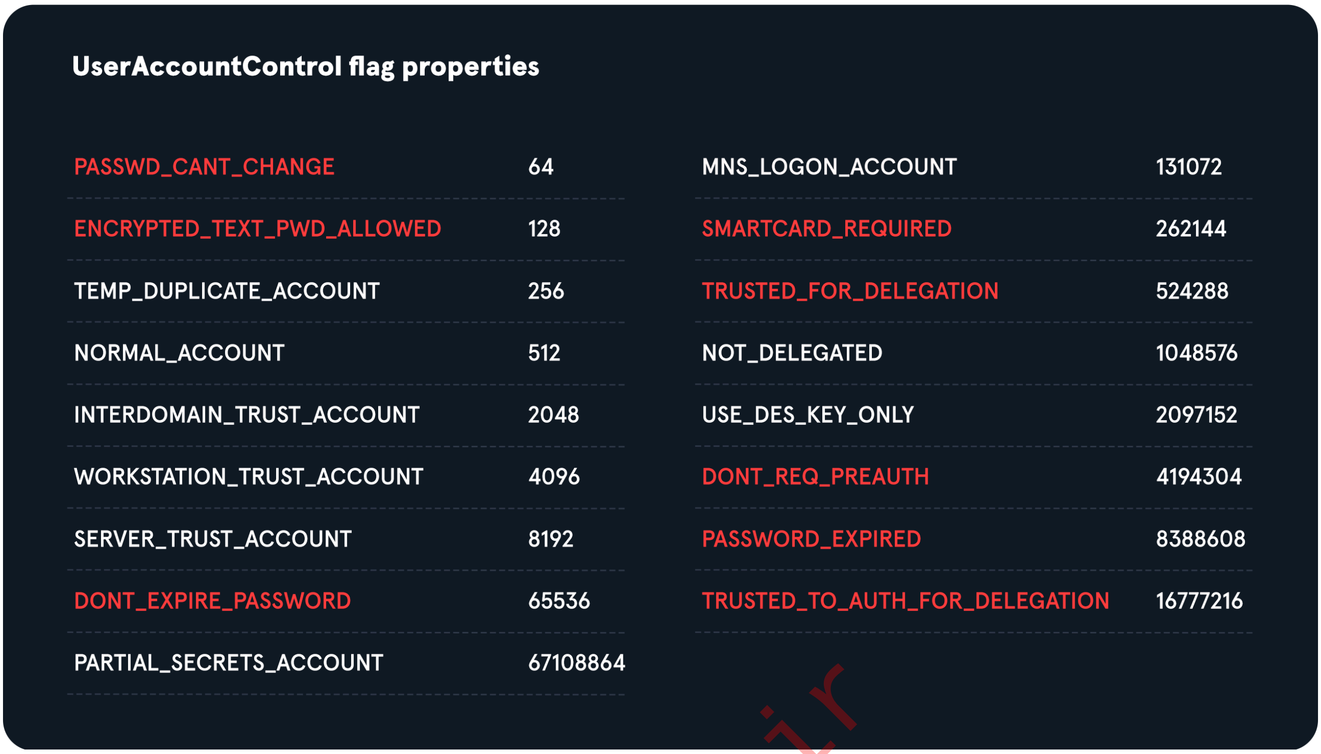
Note: When spawning your target, we ask you to wait for 3 minutes until the whole lab with all the configurations is set up so that the connection to your target works flawlessly.

Enumerating Active Directory with Built-in Tools

Proper enumeration is key for all penetration testing and red teaming assessments. Enumerating AD, especially large corporate environments with many hosts, users, and services, can be quite a daunting task and provide an overwhelming amount of data. Several built-in Windows tools can be used by sysadmins and pentesters to enumerate AD. Open source tools have been created based on the same enumeration techniques. Many of these tools (such as SharpView, BloodHound, and, PingCastle) can be utilized to expedite the enumeration process and accurately present the data in a consumable and actionable format. Knowledge of multiple tools and "offense in-depth" is important if you must live off the land on an assessment or detections are in place for certain tools.

User-Account-Control (UAC) Attributes

User-Account-Control Attributes control the behavior of domain accounts. These values are not to be confused with the Windows User Account Control technology. Many of these UAC attributes have security relevance:



UserAccountControl flag properties

PASSWD_CANT_CHANGE	64
ENCRYPTED_TEXT_PWD_ALLOWED	128
TEMP_DUPLICATE_ACCOUNT	256
NORMAL_ACCOUNT	512
INTERDOMAIN_TRUST_ACCOUNT	2048
WORKSTATION_TRUST_ACCOUNT	4096
SERVER_TRUST_ACCOUNT	8192
DONT_EXPIRE_PASSWORD	65536
PARTIAL_SECRETS_ACCOUNT	67108864
MNS_LOGON_ACCOUNT	131072
SMARTCARD_REQUIRED	262144
TRUSTED_FOR_DELEGATION	524288
NOT_DELEGATED	1048576
USE_DES_KEY_ONLY	2097152
DONT_REQ_PREAUTH	4194304
PASSWORD_EXPIRED	8388608
TRUSTED_TO_AUTH_FOR_DELEGATION	16777216

We can enumerate these values with built-in AD cmdlets:

PowerShell - Built-in AD Cmdlets

```
PS C:\htb> Get-ADUser -Filter {adminCount -gt 0} -Properties
admincount,useraccountcontrol | select Name,useraccountcontrol

Name               useraccountcontrol
----               ------------------
Administrator                   66048
krbtgt                          66050
daniel.carter                     512
sqlqa                             512
svc-backup                      66048
svc-secops                      66048
cliff.moore                     66048
svc-ata                           512
svc-sccm                          512
mrb3n                             512
sarah.lafferty                    512
Jenna Smith                   4260384
Harry Jones                     66080
pixis                             512
Cry0l1t3                          512
```

```
knightmare                    512
```

We still need to convert the `useraccountcontrol` values into their corresponding flags to interpret them. This can be done with this script. Let's take the user `Jenna Smith` with `useraccountcontrol` value `4260384` as an example.

PowerShell - UAC Values

```
PS C:\htb> .\Convert-UserAccountControlValues.ps1

Please provide the userAccountControl value: : 4260384

Name                           Value
----                           -----
PASSWD_NOTREQD                 32
NORMAL_ACCOUNT                 512
DONT_EXPIRE_PASSWORD           65536
DONT_REQ_PREAUTH               4194304
```

We can also use PowerView (which will be covered in-depth in subsequent modules) to enumerate these values. We can see that some of the users match the default value of `512` or `Normal_Account` while others would need to be converted. The value for `jenna.smith` does match what our conversion script provided.

`PowerView` can be found in the `c:\tools` directory on the target host. To load the tool, open a PowerShell console, navigate to the tools directory, and import `PowerView` using the command `Import-Module .\PowerView.ps1`.

PowerView - Domain Accounts

```
PS C:\htb> Get-DomainUser * -AdminCount | select
samaccountname,useraccountcontrol

samaccountname
useraccountcontrol
--------------                                                     -------
-----------
Administrator                                   NORMAL_ACCOUNT,
DONT_EXPIRE_PASSWORD
krbtgt                          ACCOUNTDISABLE, NORMAL_ACCOUNT,
DONT_EXPIRE_PASSWORD
daniel.carter
NORMAL_ACCOUNT
sqlqa
NORMAL_ACCOUNT
```

```
svc-backup                                       NORMAL_ACCOUNT,
DONT_EXPIRE_PASSWORD
svc-secops                                       NORMAL_ACCOUNT,
DONT_EXPIRE_PASSWORD
cliff.moore                                      NORMAL_ACCOUNT,
DONT_EXPIRE_PASSWORD
svc-ata
NORMAL_ACCOUNT
svc-sccm
NORMAL_ACCOUNT
mrb3n
NORMAL_ACCOUNT
sarah.lafferty
NORMAL_ACCOUNT
jenna.smith    PASSWD_NOTREQD, NORMAL_ACCOUNT, DONT_EXPIRE_PASSWORD,
DONT_REQ_PREAUTH
harry.jones                      PASSWD_NOTREQD, NORMAL_ACCOUNT,
DONT_EXPIRE_PASSWORD
pixis
NORMAL_ACCOUNT
Cry0l1t3
NORMAL_ACCOUNT
knightmare
NORMAL_ACCOUNT
```

Enumeration Using Built-In Tools

Tools that sysadmins are themselves likely to use, such as the PowerShell AD Module, the Sysinternals Suite, and AD DS Tools, are likely to be whitelisted and fly under the radar, especially in more mature environments. Several built-in tools can be leveraged for AD enumeration, including:

`DS Tools` is available by default on all modern Windows operating systems but required domain connectivity to perform enumeration activities.

DS Tools

```
C:\htb> dsquery user "OU=Employees,DC=inlanefreight,DC=local" -name * -
scope subtree -limit 0 | dsget user -samid -
pwdneverexpires | findstr /V no

  samid                  pwdneverexpires
  svc-backup             yes
  svc-scan               yes
  svc-secops             yes
  sql-test               yes
  cliff.moore            yes
  margaret.harris        yes

  <SNIP>

dsget succeeded
```

The `PowerShell Active Directory module` is a group of cmdlets used to manage Active Directory. The installation of the AD PowerShell module requires administrative access.

AD PowerShell Module

```
PS C:\htb> Get-ADUser -Filter * -SearchBase
'OU=Admin,DC=inlanefreight,dc=local'

DistinguishedName : CN=wilford.stewart,OU=Admin,DC=INLANEFREIGHT,DC=LOCAL
Enabled           : True
GivenName         :
Name              : wilford.stewart
ObjectClass       : user
ObjectGUID        : 1f54c02c-2fb4-48b6-a89c-38b6b0c54147
SamAccountName    : wilford.stewart
SID               : S-1-5-21-2974783224-3764228556-2640795941-2121
Surname           :
UserPrincipalName :

DistinguishedName : CN=trisha.duran,OU=Admin,DC=INLANEFREIGHT,DC=LOCAL
Enabled           : True
GivenName         :
Name              : trisha.duran
ObjectClass       : user
ObjectGUID        : 7a8db2bb-7b24-4f79-a3fe-7b49408bc7bf
SamAccountName    : trisha.duran
SID               : S-1-5-21-2974783224-3764228556-2640795941-2122
Surname           :
UserPrincipalName :

<SNIP>
```

`Windows Management Instrumentation` (WMI) can also be used to access and query objects in Active Directory. Many scripting languages can interact with the WMI AD provider, but PowerShell makes this very easy.

Windows Management Instrumentation (WMI)

```
PS C:\htb> Get-WmiObject -Class win32_group -Filter
"Domain='INLANEFREIGHT'" | Select Caption,Name

Caption                                               Name
-------                                               ----
INLANEFREIGHT\Cert Publishers                         Cert Publishers
INLANEFREIGHT\RAS and IAS Servers                     RAS and IAS Servers
INLANEFREIGHT\Allowed RODC Password Replication Group Allowed RODC
Password Replication Group
INLANEFREIGHT\Denied RODC Password Replication Group  Denied RODC Password
Replication Group
INLANEFREIGHT\DnsAdmins                               DnsAdmins
INLANEFREIGHT\$6I2000-MBUUOKUK1E1O                    $6I2000-MBUUOKUK1E1O
INLANEFREIGHT\Cloneable Domain Controllers            Cloneable Domain
Controllers
INLANEFREIGHT\Compliance Management                   Compliance
Management
INLANEFREIGHT\Delegated Setup                         Delegated Setup
INLANEFREIGHT\Discovery Management                    Discovery Management
INLANEFREIGHT\DnsUpdateProxy                          DnsUpdateProxy
INLANEFREIGHT\Domain Admins                           Domain Admins
INLANEFREIGHT\Domain Computers                        Domain Computers
INLANEFREIGHT\Domain Controllers                      Domain Controllers
INLANEFREIGHT\Domain Guests                           Domain Guests
INLANEFREIGHT\Domain Users                            Domain Users
INLANEFREIGHT\Enterprise Admins                       Enterprise Admins
INLANEFREIGHT\Enterprise Key Admins                   Enterprise Key
Admins
INLANEFREIGHT\Enterprise Read-only Domain Controllers Enterprise Read-only
Domain Controllers
INLANEFREIGHT\Exchange Servers                        Exchange Servers
INLANEFREIGHT\Exchange Trusted Subsystem              Exchange Trusted
Subsystem
INLANEFREIGHT\Exchange Windows Permissions            Exchange Windows
Permissions
INLANEFREIGHT\ExchangeLegacyInterop
ExchangeLegacyInterop
INLANEFREIGHT\Group Policy Creator Owners             Group Policy Creator
Owners
INLANEFREIGHT\Help Desk                               Help Desk
INLANEFREIGHT\Hygiene Management                      Hygiene Management
INLANEFREIGHT\Key Admins                              Key Admins
INLANEFREIGHT\LAPS Admins                             LAPS Admins
INLANEFREIGHT\Managed Availability Servers            Managed Availability
Servers
INLANEFREIGHT\Organization Management                 Organization
Management
INLANEFREIGHT\Protected Users                         Protected Users
```

```
<SNIP>
```

`Active Directory Service Interfaces` (ADSI) is a set of COM interfaces that can query Active Directory. PowerShell again provides an easy way to interact with it.

AD Service Interfaces (ADSI)

```
PS C:\htb> ([adsisearcher]"(&(objectClass=Computer))").FindAll() | select
Path

Path
----
LDAP://CN=DC01,OU=Domain Controllers,DC=INLANEFREIGHT,DC=LOCAL
LDAP://CN=EXCHG01,OU=Mail Servers,OU=Servers,DC=INLANEFREIGHT,DC=LOCAL
LDAP://CN=SQL01,OU=SQL Servers,OU=Servers,DC=INLANEFREIGHT,DC=LOCAL
LDAP://CN=WS01,OU=Staff
Workstations,OU=Workstations,DC=INLANEFREIGHT,DC=LOCAL
LDAP://CN=DC02,OU=Servers,DC=INLANEFREIGHT,DC=LOCAL
```

Note: When spawning your target, we ask you to wait for 3 minutes until the whole lab with all the configurations is set up so that the connection to your target works flawlessly.

LDAP Anonymous Bind

Lightweight Directory Access Protocol (LDAP) is a protocol that is used for accessing directory services.

Leveraging LDAP Anonymous Bind

LDAP anonymous binds allow unauthenticated attackers to retrieve information from the domain, such as a full listing of users, groups, computers, user account attributes, and the domain password policy. Linux hosts running open-source versions of LDAP and Linux vCenter appliances are often configured to allow anonymous binds.

When an LDAP server allows anonymous base binds, an attacker does not need to know a base object to query a considerable amount of information from the domain. This can also be leveraged to mount a password spraying attack or read information such as passwords stored in account description fields. Tools such as windapsearch and ldapsearch can be utilized to enumerate domain information via an anonymous LDAP bind. Information that we obtain from an anonymous LDAP bind can be leveraged to mount a password spraying or AS-REPRoasting attack, read information such as passwords stored in account description fields.

We can use `Python` to quickly check if we can interact with LDAP without credentials.

```
Python 3.8.5 (default, Aug  2 2020, 15:09:07)
[GCC 10.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from ldap3 import *
>>> s = Server('10.129.1.207',get_info = ALL)
>>> c =  Connection(s, '', '')
>>> c.bind()
True
>>> s.info
DSA info (from DSE):
  Supported LDAP versions: 3, 2
  Naming contexts:
    DC=INLANEFREIGHT,DC=LOCAL
    CN=Configuration,DC=INLANEFREIGHT,DC=LOCAL
    CN=Schema,CN=Configuration,DC=INLANEFREIGHT,DC=LOCAL
    DC=DomainDnsZones,DC=INLANEFREIGHT,DC=LOCAL
    DC=ForestDnsZones,DC=INLANEFREIGHT,DC=LOCAL
  Supported controls:

        <SNIP>

  dnsHostName:
    DC01.INLANEFREIGHT.LOCAL
  ldapServiceName:
    INLANEFREIGHT.LOCAL:[email protected]
  serverName:
    CN=DC01,CN=Servers,CN=Default-First-Site-
Name,CN=Sites,CN=Configuration,DC=INLANEFREIGHT,DC=LOCAL
  isSynchronized:
    TRUE
  isGlobalCatalogReady:
    TRUE
  domainFunctionality:
    7
  forestFunctionality:
    7
  domainControllerFunctionality:
    7
```

Using Ldapsearch

We can confirm anonymous LDAP bind with `ldapsearch` and retrieve all AD objects from LDAP.

```
ldapsearch -H ldap://10.129.1.207 -x -b "dc=inlanefreight,dc=local"
# extended LDIF
#
# LDAPv3
# base <dc=inlanefreight,dc=local> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#

# INLANEFREIGHT.LOCAL
dn: DC=INLANEFREIGHT,DC=LOCAL
objectClass: top
objectClass: domain
objectClass: domainDNS
distinguishedName: DC=INLANEFREIGHT,DC=LOCAL
instanceType: 5
whenCreated: 20200726201343.0Z
whenChanged: 20200827025341.0Z
subRefs: DC=LOGISTICS,DC=INLANEFREIGHT,DC=LOCAL
subRefs: DC=ForestDnsZones,DC=INLANEFREIGHT,DC=LOCAL
subRefs: DC=DomainDnsZones,DC=INLANEFREIGHT,DC=LOCAL
subRefs: CN=Configuration,DC=INLANEFREIGHT,DC=LOCAL
```

Using Windapsearch

`Windapsearch` is a Python script used to perform anonymous and authenticated LDAP enumeration of AD users, groups, and computers using LDAP queries. It is an alternative to tools such as `ldapsearch`, which require you to craft custom LDAP queries. We can use it to confirm LDAP NULL session authentication but providing a blank username with `-u ""` and add `--functionality` to confirm the domain functional level.

```
python3 windapsearch.py --dc-ip 10.129.1.207 -u "" --functionality
[+] No username provided. Will try anonymous bind.
[+] Using Domain Controller at: 10.129.1.207
[+] Getting defaultNamingContext from Root DSE
[+]     Found: DC=INLANEFREIGHT,DC=LOCAL
[+] Functionality Levels:
[+]      domainFunctionality: 2016
[+]      forestFunctionality: 2016
[+]      domainControllerFunctionality: 2016
[+] Attempting bind
[+]     ...success! Binded as:
[+]      None

[*] Bye!
```

We can pull a listing of all domain users to use in a password spraying attack.

```
python3 windapsearch.py --dc-ip 10.129.1.207 -u "" -U
[+] No username provided. Will try anonymous bind.
[+] Using Domain Controller at: 10.129.1.207
[+] Getting defaultNamingContext from Root DSE
[+]     Found: DC=INLANEFREIGHT,DC=LOCAL
[+] Attempting bind
[+]     ...success! Binded as:
[+]      None

[+] Enumerating all AD users
[+]     Found 1024 users:

cn: Guest
cn: DefaultAccount
cn: LOGISTICS$
cn: sqldev
cn: sqlprod
cn: svc-scan

<SNIP>
```

We can obtain information about all domain computers.

```
python3 windapsearch.py --dc-ip 10.129.1.207 -u "" -C
[+] No username provided. Will try anonymous bind.
[+] Using Domain Controller at: 10.129.1.207
[+] Getting defaultNamingContext from Root DSE
[+]     Found: DC=INLANEFREIGHT,DC=LOCAL
[+] Attempting bind
[+]     ...success! Binded as:
[+]      None

[+] Enumerating all AD computers
[+]     Found 5 computers:

cn: DC01
operatingSystem: Windows Server 2016 Standard
operatingSystemVersion: 10.0 (14393)
dNSHostName: DC01.INLANEFREIGHT.LOCAL

cn: EXCHG01
operatingSystem: Windows Server 2016 Standard
operatingSystemVersion: 10.0 (14393)
dNSHostName: EXCHG01.INLANEFREIGHT.LOCAL
```

```
cn: SQL01
operatingSystem: Windows Server 2016 Standard
operatingSystemVersion: 10.0 (14393)
dNSHostName: SQL01.INLANEFREIGHT.LOCAL

cn: WS01
operatingSystem: Windows Server 2016 Standard
operatingSystemVersion: 10.0 (14393)
dNSHostName: WS01.INLANEFREIGHT.LOCAL

cn: DC02
dNSHostName: DC02.INLANEFREIGHT.LOCAL

[*] Bye!
```

This process can be repeated to pull group information and more detailed information such as unconstrained users and computers, GPO information, user and computer attributes, and more.

Other Tools

There are many other tools and helper scripts for retrieving information from LDAP. This script ldapsearch-ad.py is similar to `windapsearch`.

```
python3 ldapsearch-ad.py -h
usage: ldapsearch-ad.py [-h] -l LDAP_SERVER [-ssl] -t REQUEST_TYPE [-d
DOMAIN] [-u USERNAME] [-p PASSWORD]
                        [-s SEARCH_FILTER] [-z SIZE_LIMIT] [-o
OUTPUT_FILE] [-v]
                        [search_attributes [search_attributes ...]]

Active Directory LDAP Enumerator

positional arguments:
  search_attributes     LDAP attributes to look for (default is all).

optional arguments:
  -h, --help            show this help message and exit
  -l LDAP_SERVER, --server LDAP_SERVER
                        IP address of the LDAP server.
  -ssl, --ssl           Force an SSL connection?.
  -t REQUEST_TYPE, --type REQUEST_TYPE
                        Request type: info, whoami, search, search-large,
trusts, pass-pols, show-admins,
                        show-user, show-user-list, kerberoast, all
  -d DOMAIN, --domain DOMAIN
                        Authentication account's FQDN. Example:
"contoso.local".
  -u USERNAME, --username USERNAME
                        Authentication account's username.
  -p PASSWORD, --password PASSWORD
                        Authentication account's password.
  -s SEARCH_FILTER, --search-filter SEARCH_FILTER
                        Search filter (use LDAP format).
  -z SIZE_LIMIT, --size_limit SIZE_LIMIT
                        Size limit (default is 100, or server' own limit).
  -o OUTPUT_FILE, --output OUTPUT_FILE
                        Write results in specified file too.
  -v, --verbose         Turn on debug mode
```

We can use it to pull domain information and confirm a NULL bind. This particular tool requires valid domain user credentials to perform additional enumeration.

```
python3 ldapsearch-ad.py -l 10.129.1.207 -t info

### Server infos ###
[+] Forest functionality level = Windows 2016
[+] Domain functionality level = Windows 2016
[+] Domain controller functionality level = Windows 2016
[+] rootDomainNamingContext = DC=INLANEFREIGHT,DC=LOCAL
[+] defaultNamingContext = DC=INLANEFREIGHT,DC=LOCAL
[+] ldapServiceName = INLANEFREIGHT.LOCAL:[email protected]
[+] naming_contexts = ['DC=INLANEFREIGHT,DC=LOCAL',
'CN=Configuration,DC=INLANEFREIGHT,DC=LOCAL',
'CN=Schema,CN=Configuration,DC=INLANEFREIGHT,DC=LOCAL',
'DC=DomainDnsZones,DC=INLANEFREIGHT,DC=LOCAL',
'DC=ForestDnsZones,DC=INLANEFREIGHT,DC=LOCAL']
```

Note: Tools necessary for completing this section can be found in the `/opt` directory on the Pwnbox.

Note: When spawning your target, we ask you to wait for 3 minutes until the whole lab with all the configurations is set up so that the connection to your target works flawlessly.

Credentialed LDAP Enumeration

As with SMB, once we have domain credentials, we can extract a wide variety of information from LDAP, including user, group, computer, trust, GPO info, the domain password policy, etc. `ldapsearch-ad.py` and `windapsearch` are useful for performing this enumeration.

Windapsearch

```
python3 windapsearch.py --dc-ip 10.129.1.207 -u inlanefreight\\james.cross
--da

Password for inlanefreight\james.cross:

[+] Using Domain Controller at: 10.129.1.207
[+] Getting defaultNamingContext from Root DSE
[+]     Found: DC=INLANEFREIGHT,DC=LOCAL
[+] Attempting bind
[+]     ...success! Binded as:
[+]      u:INLANEFREIGHT\james.cross
[+] Attempting to enumerate all Domain Admins
[+] Using DN: CN=Domain Admins,CN=Users.CN=Domain
Admins,CN=Users,DC=INLANEFREIGHT,DC=LOCAL
[+]     Found 14 Domain Admins:

cn: Administrator
userPrincipalName: [email protected]

cn: daniel.carter
cn: sqlqa
cn: svc-backup
cn: svc-secops
cn: cliff.moore
cn: svc-ata
cn: svc-sccm
cn: mrb3n
cn: sarah.lafferty

cn: Harry Jones
userPrincipalName: harry.jones@inlanefreight

cn: pixis
cn: Cry0l1t3
cn: knightmare

[+] Using DN: CN=Domain Admins,CN=Users.CN=Domain
Admins,CN=Users,DC=INLANEFREIGHT,DC=LOCAL
[+]     Found 14 Domain Admins:

cn: Administrator
userPrincipalName: [email protected]

cn: daniel.carter
cn: sqlqa
cn: svc-backup
cn: svc-secops
```

```
<SNIP>
```

Some additional useful options, including pulling users and computers with unconstrained delegation.

```
python3 windapsearch.py --dc-ip 10.129.1.207 -d inlanefreight.local -u
inlanefreight\\james.cross --unconstrained-users

Password for inlanefreight\james.cross:

[+] Using Domain Controller at: 10.129.1.207
[+] Getting defaultNamingContext from Root DSE
[+]     Found: DC=INLANEFREIGHT,DC=LOCAL
[+] Attempting bind
[+]     ...success! Binded as:
[+]      u:INLANEFREIGHT\james.cross
[+] Attempting to enumerate all user objects with unconstrained delegation
[+]     Found 1 Users with unconstrained delegation:

CN=sqldev,OU=Service Accounts,OU=IT,OU=Employees,DC=INLANEFREIGHT,DC=LOCAL

[*] Bye!
```

Ldapsearch-ad

This tool can perform all of the standard enumeration and a few built-in searches to simplify things. We can quickly obtain the password policy.

```
python3 ldapsearch-ad.py -l 10.129.1.207 -d inlanefreight -u james.cross -
p Summer2020 -t pass-pols

### Result of "pass-pols" command ###
Default password policy:
[+] |___Minimum password length = 7
[+] |___Password complexity = Disabled
[*] |___Lockout threshold = Disabled
[+] No fine grained password policy found (high privileges are required).
```

We can look for users who may be subject to a Kerberoasting attack.

```
python3 ldapsearch-ad.py -l 10.129.1.207 -d inlanefreight -u james.cross -p Summer2020 -t kerberoast | grep servicePrincipalName:

    servicePrincipalName: CIFS/roguecomputer.inlanefreight.local
    servicePrincipalName: MSSQLSvc/sql01:1433
    servicePrincipalName: MSSQL_svc_qa/inlanefreight.local:1443
    servicePrincipalName: MSSQL_svc_test/inlanefreight.local:1443
    servicePrincipalName: IIS_dev/inlanefreight.local:80
```

Also, it quickly retrieves users that can be ASREPRoasted.

```
python3 ldapsearch-ad.py -l 10.129.1.207 -d inlanefreight -u james.cross -p Summer2020 -t asreproast

### Result of "asreproast" command ###
[*] DN: CN=Amber Smith,OU=Contractors,OU=Employees,DC=INLANEFREIGHT,DC=LOCAL - STATUS: Read - READ TIME: 2020-09-02T17:11:45.572421
    cn: Amber Smith
    sAMAccountName: amber.smith

[*] DN: CN=Jenna Smith,OU=Server Team,OU=IT,OU=Employees,DC=INLANEFREIGHT,DC=LOCAL - STATUS: Read - READ TIME: 2020-09-02T17:11:45.572729
    cn: Jenna Smith
    sAMAccountName: jenna.smith
```

LDAP Wrap-up

We can use tools such as the two shown in this section to perform a considerable amount of AD enumeration using LDAP. The tools have many built-in queries to simplify searching and provide us with the most useful and actionable data. We can also combine these tools with the custom LDAP search filters that we learned about earlier in the module. These are great tools to keep in our arsenal, especially when we are in a position where most an AD assessment has to be performed from a Linux attack box.

Note: When spawning your target, we ask you to wait for 3 minutes until the whole lab with all the configurations is set up so that the connection to your target works flawlessly.

Active Directory LDAP - Skills Assessment

You have been contracted by the `INLANEFREIGHT` organization to perform an Active Directory security assessment to assess what flaws exist that could potentially be exploited by an attacker who gains internal network access with a standard Domain User account.

Connect to the target host and perform the enumeration tasks listed below to complete this module.

Note: When spawning your target, we ask you to wait for 3 minutes until the whole lab with all the configurations is set up so that the connection to your target works flawlessly.