**Domain-Analysis CLI Tools:**

**Get-AD**<TAB>

Get-ADDomain | FL NetBIOSName

Get-ADTrust | -Filter \*

Get-ADDomainControler -Filter | {Domain -eq "army.warriors"} | Select Name

Get-ADRootDSE

Get-ADUser -Identity "Administrator" -Properties \*

Get-ADObject -Filter ((ObjectClass -eq "user") -and (ObjectCategory -eq "person")) | Measure-Object | FL count

more on Get-AD\* : <http://ss64.com/ps/ad.html>

dsadd

dsquery \* DC=army,DC=warrriors -Filter "(&(ObjectClass=user)(ObjectCategory=person))" -attr sAMAccountName

Get-ADObject -Filter {(objectClass -eq "user") -and (objectcategory -eq "person")}

.. finds object in the domain

dsquery \* DC=army,DC=warrriors -limit 0 -filter <filter> -attr <atribList>

Get-ADObject -SearchBase "CN=users,DC=army,DC=warrriors" -Filter \* \_resultSetSize 12000

.. search for large number of objects in domain

adfind -b "CN=group,CN=users,DC=army,DC=warrriors" -asq member -f objectClass=user

Get-ADGroup "Domain Admins" | Get-ADGroupMember | Get-ADUser -Properties sAMAccountname | Select samaccountname

Get-ADUser -Properties sAMAccountName | Select samaccountname

.. search domain w/ attribute scoped query

dsquery \* -s <target ip> -u <username> -p ReallyStrongPassword!! DC=army,DC=warrriors -scope subtree -attr "name" -filter (&(objectclass=group)(objectCategory=group) (groupType:1.2.840.113556.1.4.804:=8))"

.. finds universal groups in the 'army.warriors' domain by using bitwise AND filter (804).

dsquery \* -s <target ip> -u <username> -p ReallyStrongPassword!! DC=army,DC=warrriors -attr "name" -filter (&(objectclass=user)(objectCategory=person) useraccountcontrol:1.2.840.113556.1.4.803:=**2**))"

UAC Code **2**

.. finds disabled user-accounts in the 'army.warriors' domain by using bitwise AND filter (803).

**UAC (User Account Control) Codes:**

1

**2**

16

32

64

**512**

2048

4096

8192

65536

524288

1048576

Login script will execute

**Account is dissabled**

Account Locked out

Password not required

Password can not be changed

**Normal user account**

Interdomain trust account

Domain ws or member server

Domain controler

Password never expires

Trusted for impersonation

Account may not be impersonated

UAC Codes can be added when querying for more than one UAC attribute, ie:

Account Disabled + Normal user account

= 2 + 512

= 514 UAC code

adfind -default -bit -f groupType:AND:=8 (finds same as above using 'adfind')

adfind -default -bit -f useraccountcontrol:AND:=2 (finds same as above using 'adfind')

admod -b "CN=Joe Blow,OU=ORG,DC=army,DC=warrriors" description::"Brigade Commander"

Set-ADObject -Identity "CN=Joe Blow,OU=ORG,DC=army,DC=warrriors" -Replace @{Description="Brigade CDR"}

.. modifies an object's atributes

dsmove "CN=Bad Guy,OU=ORG,DC=army,DC=warrriors" -newparent "OU=JAIL,DC=army,DC=warrriors"

admod -b "CN=Bad Guy,OU=ORG,DC=army,DC=warrriors" -move "OU=JAIL,DC=army,DC=warrriors"

Move-ADObject -Identity "CN=Bad Guy,OU=ORG,DC=army,DC=warrriors" -TargetPath "OU=JAIL,DC=army,DC=warrriors"

.. moves object (user, computer) to different OU / container

dsquery \* "CN=Jimbo,OU=ORG,DC=army,DC=warrriors" -attr name createTimestamp modifyTimestamp

adfind -default -rb CN=Users -f "cn=Jimbo" createTimestamp modifyTimestamp

.. viewing MAC times of an object (user, computer)

dsquery ou

adfind -default -f "objectcategory=organizationalUnit" -dn

adfind -default -sc oudmp messy

Get-ADOrganizationalUnit -Filter \* | Select DistinguishedName

.. enumerating the OUs in the domain

dsquery \* "OU=ORG,CN=users,DC=army,DC=warrriors" -limit 0 -scope onelevel

adfind -b "OU=ORG,CN=users,DC=army,DC=warrriors" -s one -dn

Get-ADObject -SearchBase "CN=users,DC=army,DC=warrriors " -Filter \*

.. enumerating the objects (user, computer)in an OU

dsquery \* "OU=ORG,CN=users,DC=army,DC=warrriors" -scope base -attr msDS-Approx-Immed-Subordinates

adfind -b "OU=ORG,CN=users,DC=army,DC=warrriors" -s base msDS-Approx-Immed-Subordinates

Get-ADObject -SearchBase "OU=ORG,CN=users,DC=army,DC=warrriors" -Filter \* | Measure-Object | FL Count

.. Determining Approximately How Many Child Objects an OU Has

User Account Attributes:

**homeDirectory** Local or UNC path of user’s home directory.

**homeDrive** Defines the drive letter to map the user’s home directory to.

**lastLogon** The last time that a user logged on to a particular DC. This information is not replicated among domain controllers in a forest.

**lastLogonTimestamp** Approximate last logon timestamp, which is replicated among domain controllers.

**managedObjects** Multivalued, linked attribute (with managedBy) that contains a list of DNs of objects the user manages.

**lockoutTime** Large integer representation of the timestamp for when a user was locked out.

**memberOf** Backlink listing of DNs of the groups the user is a member of.

**objectSid** Octet string representing the SID of the user.

**primaryGroupID** ID of the primary group for the user.

**profilePath** UNC path to profile directory. pwdLastSet Large integer denoting the last time the user’s password was set.

**sAMAccountName** NetBIOS-style name of the user; limited to 20 characters to support legacy applications.

**sIDHistory** Multivalued attribute that contains a list of SIDs that are associated w ith the user.

**scriptPath** Path and filename of logon script.

**sn** Last name of user.

**tokenGroups** List of SIDs for the groups in the domain the user is a member of (both directly and via nesting).

**unicodePwd** Octet string that contains a hash of a user’s password. This attribute cannot be directly queried.

**userAccountControl** Account flags that define such things as account status and password change status.

**userPrincipalName** Internet-style account name for a user, which the user can use to log on to a computer; in most cases this should map to the user’s email address, but this does not always need to be the case.

**userWorkstations** List of computers a user can log on to, stored as a Unicode string.

**msDS-PSOApplied** A backlink that lists the Password Settings Objects that are applied to a user object.

**msDS-ResultantPSO** A constructed attribute that indicates which PSO is in effect for a user object.

**msDS-UserPasswordExpiryTimeCompu**

**te**d A constructed attribute that indicates when a user’s password is going to expire.

**msDS-FailedInteractiveLogonCount**

Indicates the number of failed interactive logons for a user account since the Interactive Logon Count feature was enabled.

**msDS-FailedInteractiveLogonCountAtLastSuccessfulLogon**

Indicates the number of failed interactive logons for a user account since the last time the user successfully logged on interactively.

**msDSLastFailedInteractiveLogonTime**

Indicates the last time and date that the user performed an unsuccessful interactive logon.

**msDSLastSuccessfulInteractiveLogonTime**

Indicates the last time and date that the user performed a successful interactive logon.

**msDS-AuthenticatedAtDC** A multivalued attribute listing the RODCs through which a user has successfully authenticated to a full DC.

**msDS-RevealedDSAs** Backlink indicating which RODCs have cached a user’s password secrets.

**msDS-ManagedPassword** Contains password information for group-managed service accounts.

.. user account attributes

dsmove "CN=Susan,OU=OLD,DC=army,DC=warrriors" -newparent "CN=Susan,OU=NEW,DC=army,DC=warrriors"

admod -b "CN=Susan,OU=OLD,DC=army,DC=warrriors" -move "CN=Susan,OU=NEW,DC=army,DC=warrriors"

Move-ADObject -Identity "CN=Susan,OU=OLD,DC=army,DC=warrriors" -TargetPath "CN=Susan,OU=NEW,DC=army,DC=warrriors"

.. moving a user

Get-ADObject " DC=army,DC=warrriors " -Properties \* | FL minPwdLength,pwdHistoryLength,pwdProperties,lockoutThreshold

.. Viewing the Domain-Wide Account Lockout and Password Policies

dsmod user "CN=Susan,OU=NEW,DC=army,DC=warrriors" -pwd Passw0rD

admod -b "CN=Susan,OU=NEW,DC=army,DC=warrriors" #setpwd#::Passw0rD

Set-ADAccountPassword -Identity "CN=Sue,OU=NEW,DC=army,DC=warrriors" -NewPassword Passw0rD -Reset

.. Seeting a User's password

adfind -b "CN=Sue,OU=NEW,DC=army,DC=warrriors" lastLogonTimestamp -tdc

.. Determining a User’s Last Logon Time

dsquery user -inactive <NumWeeks>

.. Finding Users Who Have Not Logged On Recently

Get-ADPermission -Identity <Group Name>

adfind -gcb -f name=<Group Name> ntsecuritydescriptor -sddl++ -resolvesids

.. Viewing the Permissions of a Group

Get-ADGroupMember -Identity "Domain Admins" -recursive | Select Name

memberof -group "OU=group,DC=army,DC=warrriors"

dsget group "OU=group,DC=army,DC=warrriors" -members -expand

.. Viewing the Nested Members of a Group

dsquery user -stalepwd <NumDaysSinceLastPwdChange>

.. Finding Users Whose Passwords Are About to Expire

Add-ADPermission -Identity <Group Name> -User <User or Group Name> -AccessRights WriteProperty -Properties "members"

Set-ADGroup <Group Name> -ManagedBy "OU=group,DC=army,DC=warrriors"

dsacls "cn=Joe Blow,ou=sales,dc=adatum,dc=com" /G salesadmins@adatum.com:WP;member;

.. Delegating Control for Managing Membership of a Group

Attributes of computer objects ..

**cn** Relative distinguished name of computer objects.

**dnsHostName** Fully qualified DNS name of the computer.

**lastLogonTimestamp** The approximate timestamp of the last time the computer logged in to the domain.

**managedBy** The distinguished name (DN) of the user or group that manages the computer.

**memberOf** List of DNs of the groups the computer is a member of.

**msDSAuthenticatedToAccount**

**List**

It is a backlink attribute that identifies users who have successfully authenticated to a full DC via a particular RODC.

**msDSIsPrimaryComputerFor** It indirectly identifies the primary user of a computer.

**msDS-RevealedUsers**  This attribute was introduced in Windows Server 2008. It identifies the list of users and computers whose secrets have been replicated to an RODC.

**operatingSystem** Text description of the operating system running on the computer.

**operatingSystemHotFix** Currently not being used, but will hopefully be populated at some point.

**operatingSystemServicePack** Service pack version installed on the computer.

**operatingSystemVersion** Numeric version of the operating system installed on the computer.

**pwdLastSet** Large integer that can be translated into the last time the computer’s password was set.

**sAMAccountName** NetBIOS-style name of the computer. This is typically the name of the computer with a $ at the end.

**userAccountControl** Account flag that defines various account properties. In the case of a computer object, this specifies whether the computer is a member computer or a domain controller.

.. Attributes of computer objects

New-ADComputer -Name "<ComputerName>" -SamAccountName "<SamAccountName>" -Path "OU=group,DC=army,DC=warrriors"

admod -b "<ComputerDN>" objectclass::computer sAMAccountName::<ComputerName>$ userAccountControl::4096

description::"<Description>" -add

dsadd computer "<ComputerDN>" -desc "<Description>"

.. Creating a Computer

Move-ADObject -Identity "<ComputerDN>" -TargetPath "<NewParentDN>"

admod -b "<ComputerDN>" -move "<NewParentDN>"

dsmove "<ComputerDN>" -newparent "<NewParentDN>"

.. Moving a Computer Within the Same Domain

Get-ADComputer -Filter {OperatingSystem -eq "Windows Server 2012 Datacenter"} | Select Name

Get-ADComputer -Filter {OperatingSystem -eq "Windows Server 2008 R2 Enterprise" -and OperatingSystemVersion -eq "6.1 (7601)"} | Select Name

adfind -b <DomainDN> -f "(&(objectcategory=computer)(operatingSystem=Windows Server 2012 Datacenter))"

dsquery \* <DomainDN> -scope subtree -attr "\*" -filter "(&(objectcategory=computer)(operatingSystem=Windows Server 2012 Datacenter))"

.. Finding Computers with a Particular OS

Get-ADComputer -Filter \* | Select Name

adfind -default -f objectCategory=computer

.. Listing All the Computer Accounts in a Domain

net user

.. access shares remotely