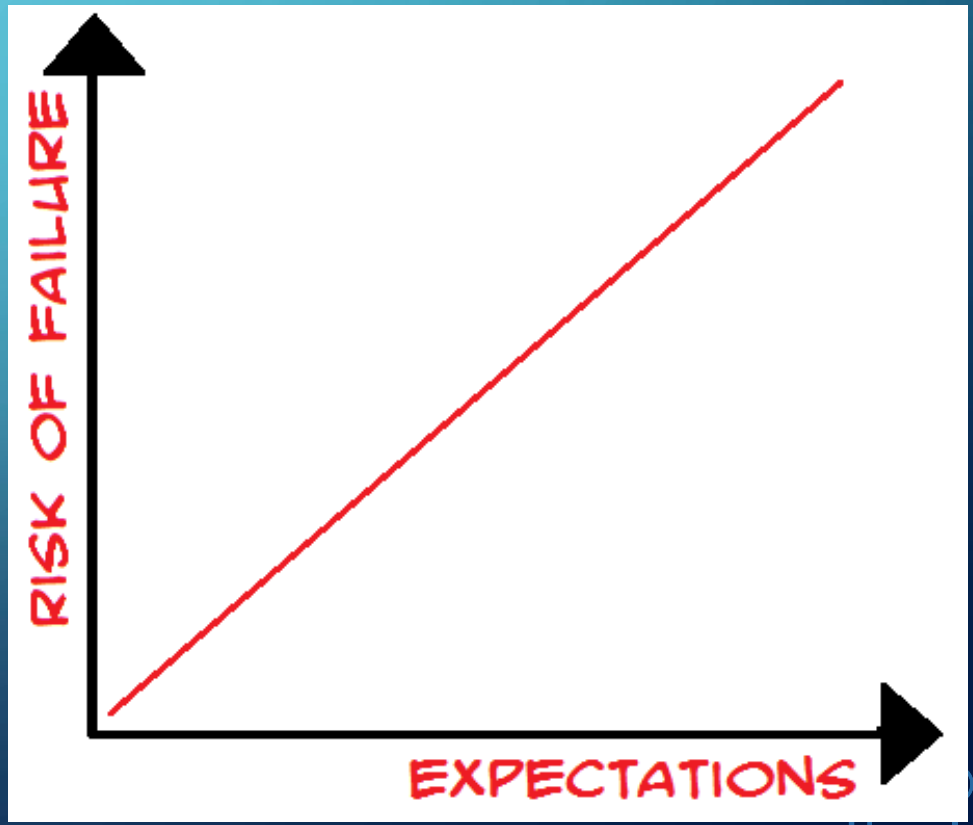
MASTERING ACTIVE DIRECTORY WITH POWERSHELL

NoVA PowerShell User Group
January 2015

SEAN METCALF
CTO
DAN SOLUTIONS
SEAN [@] DANSOLUTIONS . COM
DANSOLUTIONS.COM
ADSECURITY.ORG

EXPECTATIONS

- This is not Active Directory PowerShell Training (that would take hours/days).
- Meant to spark ideas on how to work with AD better.
- Lots of PowerShell example code – how it's used is up to you! ☺
- This session is interactive - Please ask questions!



AGENDA

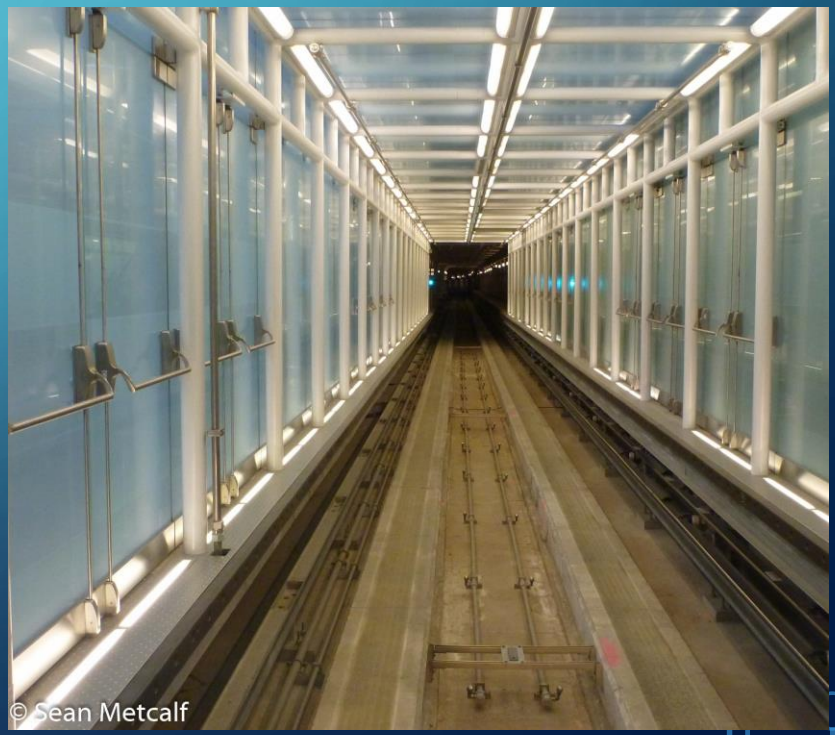
- Interfacing with Active Directory through PowerShell.
- PowerShell Active Directory Module Cmdlets
- Forest & Domain Discovery
- Useful AD Cmdlets
- Computers, Users, & Groups, Oh My!
- Interesting AD Config Data
- Service Accounts
- DCs & GCs
- AD Replication Power
- Tips & Tricks
- References

POWERSHELL & ACTIVE DIRECTORY

- PowerShell v1: NET & ADSI
- PowerShell v2 & newer: PowerShell Active Directory Module
  - Import-module servermanager;
    add-windowsfeature rsat-ad-tools
  - Import-module servermanager;
    add-windowsfeature rsat-ad-PowerShell

.NET

".NET Framework is a **software framework** developed by Microsoft that runs primarily on Microsoft Windows. It includes a **large class library** known as Framework Class Library (FCL) and **provides language interoperability** (each language can use code written in other languages) **across several programming languages**. Programs written for **.NET Framework execute in a software environment** (as contrasted to hardware environment), known as **Common Language Runtime (CLR)**, an application virtual machine that provides services such as security, memory management, and exception handling. FCL and CLR together constitute .NET Framework."
-Wikipedia



ACTIVE DIRECTORY .NET

- Get the Current Domain:
  - [System.DirectoryServices.ActiveDirectory.Domain]::GetCurrentDomain().Name
  - [System.DirectoryServices.ActiveDirectory.Domain]::GetComputerDomain().Name
- Get the Computer's Site:
  - [System.DirectoryServices.ActiveDirectory.ActiveDirectorySite]::GetComputerSite()
- List All Domain Controllers in a Domain:
  - [System.DirectoryServices.ActiveDirectory.Domain]::GetCurrentDomain().DomainControllers
- Get Active Directory Domain Mode:
  - [System.DirectoryServices.ActiveDirectory.Domain]::GetCurrentDomain().DomainMode
- List Active Directory FSMOs:
  - ([System.DirectoryServices.ActiveDirectory.Forest]::GetCurrentForest()).SchemaRoleOwner
  - ([System.DirectoryServices.ActiveDirectory.Forest]::GetCurrentForest()).NamingRoleOwner
  - ([System.DirectoryServices.ActiveDirectory.Domain]::GetCurrentDomain()).InfrastructureRoleOwner
  - ([System.DirectoryServices.ActiveDirectory.Domain]::GetCurrentDomain()).PdcRoleOwner
  - ([System.DirectoryServices.ActiveDirectory.Domain]::GetCurrentDomain()).RidRoleOwner

ACTIVE DIRECTORY .NET

- Get Active Directory Forest Name:
  - [System.DirectoryServices.ActiveDirectory.Forest]::GetCurrentForest().Name
- Get a List of Sites in the Active Directory Forest:
  - [array] $ADSites = [System.DirectoryServices.ActiveDirectory.Forest]::GetCurrentForest().Sites
- Get Active Directory Forest Domains:
  - [System.DirectoryServices.ActiveDirectory.Forest]::GetCurrentForest().Domains
- Get Active Directory Forest Global Catalogs:
  - [System.DirectoryServices.ActiveDirectory.Forest]::GetCurrentForest().GlobalCatalogs
- Get Active Directory Forest Mode:
  - [System.DirectoryServices.ActiveDirectory.Forest]::GetCurrentForest().ForestMode
- Get Active Directory Forest Root Domain:
  - [System.DirectoryServices.ActiveDirectory.Forest]::GetCurrentForest().RootDomain

OLD SCHOOL - ADSI

- Active Directory Service Interface (ADSI)
  - *"Active Directory Service Interfaces (ADSI) is a set of* ***COM interfaces used to access the features of directory services from different network providers****. ADSI is used in a distributed computing environment to present a single set of directory service interfaces for managing network resources.* ***Administrators and developers can use ADSI services to enumerate and manage the resources in a directory service, no matter which network environment contains the resource****."*
- ADSI Example:
  - $UserID = "JoeUser"
  - $root = [ADSI]''
  - $searcher = new-object System.DirectoryServices.DirectorySearcher($root)
  - $searcher.filter = "(&(objectClass=user)(sAMAccountName= $UserID))"
  - $user = $searcher.findall()
  - $user

POWERSHELL ACTIVE DIRECTORY MODULE

- Requires AD Web Services (ADWS) running on targeted DC (TCP 9389)
  - Get-ADDomainController –Discover –Service “ADWS”
- SOAP XML message(s) over HTTP translated on DC
- PowerShell AD Cmdlet Example:
  - Import-module ActiveDirectory
  - $UserID = “JoeUser”
  - Get-ADUser $UserID –property *

```
PS C:\temp> import-module servermanager ; add-windowsfeature rsat-ad-powershell

Success Restart Needed Exit Code      Feature Result
------- -------------- ---------      --------------
True    No             NoChangeNeeded {}
```

ACTIVE DIRECTORY DRIVE

```
PS C:\Users\LukeSkywalker> import-module activedirectory
PS C:\Users\LukeSkywalker> dir ad:

Name                 ObjectClass           DistinguishedName
----                 -----------           -----------------
lab                  domainDNS             DC=lab,DC=adsecurity,DC=org
Configuration        configuration         CN=Configuration,DC=lab,DC=adsecurity,DC=org
Schema               dMD                   CN=Schema,CN=Configuration,DC=lab,DC=adsecurity,DC=org
DomainDnsZones       domainDNS             DC=DomainDnsZones,DC=lab,DC=adsecurity,DC=org
ForestDnsZones       domainDNS             DC=ForestDnsZones,DC=lab,DC=adsecurity,DC=org


PS C:\Users\LukeSkywalker> set-location ad:
PS AD:\> set-location "dc=lab,dc=adsecurity,dc=org"
PS AD:\dc=lab,dc=adsecurity,dc=org> dir

Name                 ObjectClass           DistinguishedName
----                 -----------           -----------------
Admin Groups         organizationalUnit    OU=Admin Groups,DC=lab,DC=adsecurity,DC=org
Builtin              builtinDomain         CN=Builtin,DC=lab,DC=adsecurity,DC=org
Computers            container             CN=Computers,DC=lab,DC=adsecurity,DC=org
CorpOU               organizationalUnit    OU=CorpOU,DC=lab,DC=adsecurity,DC=org
Domain Controllers   organizationalUnit    OU=Domain Controllers,DC=lab,DC=adsecurity,DC=org
Domain Management    organizationalUnit    OU=Domain Management,DC=lab,DC=adsecurity,DC=org
ForeignSecurityPr... container             CN=ForeignSecurityPrincipals,DC=lab,DC=adsecurity,DC=org
Infrastructure       infrastructureUpdate  CN=Infrastructure,DC=lab,DC=adsecurity,DC=org
LostAndFound         lostAndFound          CN=LostAndFound,DC=lab,DC=adsecurity,DC=org
Managed Service A... container             CN=Managed Service Accounts,DC=lab,DC=adsecurity,DC=org
                                           CN=NTDS Quotas,DC=lab,DC=adsecurity,DC=org
Program Data         container             CN=Program Data,DC=lab,DC=adsecurity,DC=org
Service Accounts     organizationalUnit    OU=Service Accounts,DC=lab,DC=adsecurity,DC=org
System               container             CN=System,DC=lab,DC=adsecurity,DC=org
                                           CN=TPM Devices,DC=lab,DC=adsecurity,DC=org
Users                container             CN=Users,DC=lab,DC=adsecurity,DC=org


PS AD:\dc=lab,dc=adsecurity,dc=org>
```

FINDING USEFUL AD COMMANDS

- Get-Module -ListAvailable
- Get-Command -module ActiveDirectory

- PowerShell AD Module Cmdlets:
  - Windows Server 2008 R2: **76** cmdlets
  - Windows Server 2012: **135** cmdlets
  - Windows Server 2012 R2: **147** cmdlets

POPULAR CMDLETS: WINDOWS SERVER 2008 R2

- Get/Set-ADForest
- Get/Set-ADDomain
- Get/Set-ADDomainController
- Get/Set-**ADUser**
- Get/Set-ADComputer
- Get/Set-ADGroup
- Get/Set-ADGroupMember
- Get/Set-**ADObject**
- Get/Set-ADOrganizationalUnit
- Enable-ADOptionalFeature
- Disable/Enable-ADAccount
- Move-ADDirectoryServerOperationMasterRole
- New-ADUser
- New-ADComputer
- New-ADGroup
- New-ADObject
- New-ADOrganizationalUnit

(SOME) NEW CMDLETS: WINDOWS SERVER 2012+

- *-ADResourcePropertyListMember
- *-ADAuthenticationPolicy
- *-ADAuthenticationPolicySilo
- *-ADCentralAccessPolicy
- *-ADCentralAccessRule
- *-ADResourceProperty
- *-ADResourcePropertyList
- *-ADResourcePropertyValueType
- *-ADDCCloneConfigFile
- *-ADReplicationAttributeMetadata
- *-ADReplicationConnection
- *-ADReplicationFailure
- *-ADReplicationPartnerMetadata
- *-ADReplicationQueueOperation
- *-ADReplicationSite
- *-ADReplicationSiteLink
- *-ADReplicationSiteLinkBridge
- *-ADReplicationSubnet
- *-ADReplicationUpToDatenessVectorTable
- Sync-ADObject

ACTIVE DIRECTORY DISCOVERY: GET-ADROOTDSE

```
PS C:\Windows\system32> get-adrootdse


configurationNamingContext    : CN=Configuration,DC=lab,DC=adsecurity,DC=org
currentTime                   : 1/18/2015 9:07:52 PM
defaultNamingContext          : DC=lab,DC=adsecurity,DC=org
dnsHostName                   : ADSDC05.lab.adsecurity.org
domainControllerFunctionality : Windows2012R2
domainFunctionality           : Windows2003Domain
dsServiceName                 : CN=NTDS Settings,CN=ADSDC05,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=lab,DC
                                =adsecurity,DC=org
forestFunctionality           : Windows2003Forest
highestCommittedUSN           : 110986
isGlobalCatalogReady          : {TRUE}
isSynchronized                : {TRUE}
ldapServiceName               : lab.adsecurity.org:adsdc05$@LAB.ADSECURITY.ORG
namingContexts                : {DC=lab,DC=adsecurity,DC=org, CN=Configuration,DC=lab,DC=adsecurity,DC=org,
                                CN=Schema,CN=Configuration,DC=lab,DC=adsecurity,DC=org,
                                DC=DomainDnsZones,DC=lab,DC=adsecurity,DC=org...}
rootDomainNamingContext       : DC=lab,DC=adsecurity,DC=org
schemaNamingContext           : CN=Schema,CN=Configuration,DC=lab,DC=adsecurity,DC=org
serverName                    : CN=ADSDC05,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=lab,DC=adsecurity,DC=or
                                g
subschemaSubentry             : CN=Aggregate,CN=Schema,CN=Configuration,DC=lab,DC=adsecurity,DC=org
supportedCapabilities         : {1.2.840.113556.1.4.800 (LDAP_CAP_ACTIVE_DIRECTORY_OID), 1.2.840.113556.1.4.1670
                                (LDAP_CAP_ACTIVE_DIRECTORY_V51_OID), 1.2.840.113556.1.4.1791
                                (LDAP_CAP_ACTIVE_DIRECTORY_LDAP_INTEG_OID), 1.2.840.113556.1.4.1935
                                (LDAP_CAP_ACTIVE_DIRECTORY_V61_OID)...}
supportedControl              : {1.2.840.113556.1.4.319 (LDAP_PAGED_RESULT_OID_STRING), 1.2.840.113556.1.4.801
                                (LDAP_SERVER_SD_FLAGS_OID), 1.2.840.113556.1.4.473 (LDAP_SERVER_SORT_OID), 1.2.840.113556.1.4.528
                                (LDAP_SERVER_NOTIFICATION_OID)...}
supportedLDAPPolicies         : {MaxPoolThreads, MaxPercentDirSyncRequests, MaxDatagramRecv, MaxReceiveBuffer...}
supportedLDAPVersion          : {3, 2}
supportedSASLMechanisms       : {GSSAPI, GSS-SPNEGO, EXTERNAL, DIGEST-MD5}
```

ACTIVE DIRECTORY DISCOVERY: GET-ADFOREST

```
PS C:\Windows\system32> get-adforest


ApplicationPartitions : {DC=DomainDnsZones,DC=lab,DC=adsecurity,DC=org, DC=ForestDnsZones,DC=lab,DC=adsecurity,DC=org}
CrossForestReferences : {}
DomainNamingMaster    : ADSDC01.lab.adsecurity.org
Domains               : {lab.adsecurity.org}
ForestMode            : Windows2003Forest
GlobalCatalogs        : {ADSDC01.lab.adsecurity.org, ADSDC02.lab.adsecurity.org, ADSDC04.lab.adsecurity.org,
                        ADSDC05.lab.adsecurity.org}
Name                  : lab.adsecurity.org
PartitionsContainer   : CN=Partitions,CN=Configuration,DC=lab,DC=adsecurity,DC=org
RootDomain            : lab.adsecurity.org
SchemaMaster          : ADSDC01.lab.adsecurity.org
Sites                 : {Default-First-Site-Name}
SPNSuffixes           : {}
UPNSuffixes           : {}
```

ACTIVE DIRECTORY DISCOVERY: GET-ADDOMAIN

```
PS C:\Windows\system32> Get-ADDomain



AllowedDNSSuffixes                : {}
ChildDomains                      : {}
ComputersContainer                : CN=Computers,DC=lab,DC=adsecurity,DC=org
DeletedObjectsContainer           : CN=Deleted Objects,DC=lab,DC=adsecurity,DC=org
DistinguishedName                 : DC=lab,DC=adsecurity,DC=org
DNSRoot                           : lab.adsecurity.org
DomainControllersContainer        : OU=Domain Controllers,DC=lab,DC=adsecurity,DC=org
DomainMode                        : Windows2003Domain
DomainSID                         : S-1-5-21-1473643419-774954089-2222329127
ForeignSecurityPrincipalsContainer : CN=ForeignSecurityPrincipals,DC=lab,DC=adsecurity,DC=org
Forest                            : lab.adsecurity.org
InfrastructureMaster              : ADSDC01.lab.adsecurity.org
LastLogonReplicationInterval      :
LinkedGroupPolicyObjects          : {cn={ABDBA081-F312-4F2A-9F95-143800450BB8},cn=policies,cn=system,DC=lab,DC=adsecurity,DC=org,
                                    cn={19DB3FB7-0098-4F85-8E24-B03050C6B6DE},cn=policies,cn=system,DC=lab,DC=adsecurity,DC=org,
                                    CN={31B2F340-016D-11D2-945F-00C04FB984F9},CN=Policies,CN=System,DC=lab,DC=adsecurity,DC=org}
LostAndFoundContainer             : CN=LostAndFound,DC=lab,DC=adsecurity,DC=org
ManagedBy                         :
Name                              : lab
NetBIOSName                       : ADSECLAB
ObjectClass                       : domainDNS
ObjectGUID                        : f6d46828-b721-463d-9696-3b3714e2676a
ParentDomain                      :
PDCEmulator                       : ADSDC01.lab.adsecurity.org
QuotasContainer                   : CN=NTDS Quotas,DC=lab,DC=adsecurity,DC=org
ReadOnlyReplicaDirectoryServers   : {}
ReplicaDirectoryServers           : {ADSDC01.lab.adsecurity.org, ADSDC02.lab.adsecurity.org, ADSDC04.lab.adsecurity.org,
                                    ADSDC05.lab.adsecurity.org}
RIDMaster                         : ADSDC02.lab.adsecurity.org
SubordinateReferences             : {DC=ForestDnsZones,DC=lab,DC=adsecurity,DC=org, DC=DomainDnsZones,DC=lab,DC=adsecurity,DC=org,
                                    CN=Configuration,DC=lab,DC=adsecurity,DC=org}
SystemsContainer                  : CN=System,DC=lab,DC=adsecurity,DC=org
UsersContainer                    : CN=Users,DC=lab,DC=adsecurity,DC=org
```

GET-ADDOMAINCONTROLLER

```
PS C:\Windows\system32> Get-ADDomainController



ComputerObjectDN           : CN=ADSDC05,OU=Domain Controllers,DC=lab,DC=adsecurity,DC=org
DefaultPartition           : DC=lab,DC=adsecurity,DC=org
Domain                     : lab.adsecurity.org
Enabled                    : True
Forest                     : lab.adsecurity.org
HostName                   : ADSDC05.lab.adsecurity.org
InvocationId               : 2df64259-f56d-4e61-acde-3b67548a0977
IPv4Address                : 172.16.11.15
IPv6Address                :
IsGlobalCatalog            : True
IsReadOnly                 : False
LdapPort                   : 389
Name                       : ADSDC05
NTDSSettingsObjectDN       : CN=NTDS Settings,CN=ADSDC05,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=lab,DC=adse
                             curity,DC=org
OperatingSystem            : Windows Server 2012 R2 Datacenter
OperatingSystemHotfix      :
OperatingSystemServicePack :
OperatingSystemVersion     : 6.3 (9600)
OperationMasterRoles       : {}
Partitions                 : {DC=ForestDnsZones,DC=lab,DC=adsecurity,DC=org, DC=DomainDnsZones,DC=lab,DC=adsecurity,DC=org,
                             CN=Schema,CN=Configuration,DC=lab,DC=adsecurity,DC=org, CN=Configuration,DC=lab,DC=adsecurity,DC=org...}
ServerObjectDN             : CN=ADSDC05,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=lab,DC=adsecurity,DC=org
ServerObjectGuid           : d68af971-b5af-4a32-9531-7f61f95e15cf
Site                       : Default-First-Site-Name
SslPort                    : 636
```

GET-ADCOMPUTER

```
PS C:\Windows\system32> get-adcomputer adsdc05



DistinguishedName : CN=ADSDC05,OU=Domain Controllers,DC=lab,DC=adsecurity,DC=org
DNSHostName       : ADSDC05.lab.adsecurity.org
Enabled           : True
Name              : ADSDC05
ObjectClass       : computer
ObjectGUID        : 72b0c16d-a1b6-4f31-bd36-901744a699ec
SamAccountName    : ADSDC05$
SID               : S-1-5-21-1473643419-774954089-2222329127-1602
UserPrincipalName :
```

QUICK AD COMPUTER COUNT

- $Time = (Measure-Command `
  {[array] $AllComputers = Get-ADComputer -filter * -properties Name,CanonicalName,Enabled,passwordLastSet,SAMAccountName,LastLogonTimeStamp,DistinguishedName,OperatingSystem }).TotalMinutes
  $AllComputersCount = $AllComputers.Count
  Write-Output “There were $AllComputersCount Computers discovered in $DomainDNS in $Time minutes… `r “

FINDING INACTIVE COMPUTER ACCOUNTS

```
PS C:\Windows\system32> $InactiveDate = (get-date).AddDays(-10)
Get-ADComputer -filter {(LastLogonDate -le $InactiveDate) -AND (PasswordLastSet -le $InactiveDate)} -property Name,IPv4Address,`
LastLogonDate,PasswordLastSet,Description,Created,DNSHostName


Created           : 12/7/2014 12:13:35 PM
Description       :
DistinguishedName : CN=ADSWKWIN8,CN=Computers,DC=lab,DC=adsecurity,DC=org
DNSHostName       : ADSWKWin8.lab.adsecurity.org
Enabled           : True
IPv4Address       : 172.16.11.202
LastLogonDate     : 1/6/2015 2:31:23 PM
Name              : ADSWKWIN8
ObjectClass       : computer
ObjectGUID        : ff423c3c-842c-41a2-ba02-0d035364a249
PasswordLastSet   : 1/7/2015 10:58:35 AM
SamAccountName    : ADSWKWIN8$
SID               : S-1-5-21-1473643419-774954089-2222329127-1109
UserPrincipalName :
```

GET-ADUSER

```
PS C:\Windows\system32> get-aduser "hansolo"



DistinguishedName : CN=Han Solo,CN=Users,DC=lab,DC=adsecurity,DC=org
Enabled           : True
GivenName         : Han
Name              : Han Solo
ObjectClass       : user
ObjectGUID        : 8239fdc4-f82a-4346-a6bb-fac16b4b7bbf
SamAccountName    : HanSolo
SID               : S-1-5-21-1473643419-774954089-2222329127-1107
Surname           : Solo
UserPrincipalName : HanSolo@lab.adsecurity.org
```

AD DOMAIN USER STATISTICS

```
Import-Module ActiveDirectory
$DomainDNS = [System.DirectoryServices.ActiveDirectory.Domain]::GetCurrentDomain().Name
[array]$AllUsers = Get-ADUser -filter * -properties
Name,DistinguishedName,Enabled,LastLogonDate,LastLogonTimeStamp,LockedOut,msExchHom
eServerName,SAMAccountName
$AllUsersCount = $AllUsers.Count
Write-Output “There were $AllUsersCount user objects discovered in $ADDomainDNSRoot … “

[array] $DisabledUsers = $AllUsers | Where-Object { $_.Enabled -eq $False }
$DisabledUsersCount = $DisabledUsers.Count

[array] $EnabledUsers = $AllUsers | Where-Object { $_.Enabled -eq $True }
$EnabledUsersCount = $EnabledUsers.Count
Write-Output “There are $EnabledUsersCount Enabled users and there are $DisabledUsersCount
Disabled users in $DomainDNS “
```

FINDING INACTIVE USER ACCOUNTS

```
PS C:\Windows\system32> $InactiveDate = (get-date).AddDays(-15)
Get-ADUser -filter {(LastLogonDate -le $InactiveDate) -AND (PasswordLastSet -le $InactiveDate)} -property SAMAccountName,DisplayName,`
LastLogonDate,PasswordLastSet,Description,Created,UserPrincipalName



Created           : 12/28/2014 7:15:49 PM
Description       :
DisplayName       : svc-SQLAgent01
DistinguishedName : CN=svc-SQLAgent01,OU=Service Accounts,DC=lab,DC=adsecurity,DC=org
Enabled           : True
GivenName         :
LastLogonDate     : 12/28/2014 7:18:02 PM
Name              : svc-SQLAgent01
ObjectClass       : user
ObjectGUID        : eba3c611-6ea6-46bc-b68c-c8f28685e7f5
PasswordLastSet   : 1/3/2015 1:42:01 PM
SamAccountName    : svc-SQLAgent01
SID               : S-1-5-21-1473643419-774954089-2222329127-1606
Surname           :
UserPrincipalName : svc-SQLAgent01@lab.adsecurity.org

Created           : 12/28/2014 7:16:23 PM
Description       :
DisplayName       : svc-SQLDBEngine01
DistinguishedName : CN=svc-SQLDBEngine01,OU=Service Accounts,DC=lab,DC=adsecurity,DC=org
Enabled           : True
GivenName         :
LastLogonDate     : 12/28/2014 7:18:02 PM
Name              : svc-SQLDBEngine01
ObjectClass       : user
ObjectGUID        : 9f05af08-4f2c-4e95-8064-ad7a690ee495
PasswordLastSet   : 1/3/2015 1:43:26 PM
SamAccountName    : svc-SQLDBEngine01
SID               : S-1-5-21-1473643419-774954089-2222329127-1607
Surname           :
UserPrincipalName : svc-SQLDBEngine01@lab.adsecurity.org
```

FINDING USERS USING ANR

- Ambiguos Name Resolution (ANR) used by Outlook to find users
- Import-Module ActiveDirectory
  Get-ADObject -LDAPFilter { (&(ObjectClass=User)(ANR=T
- ANR queries are compared to indexed attributes such as:
  - sAMAccountName
  - displayName
  - Name (cn)
  - givenName (first name)
  - sn (surname aka last name)
  - legacyExchangeDN
  - proxyAddresses (Exchange attribute)

GET & SET AD ATTRIBUTES

- **Find all users and display $AttributeName**
  - Get-ADUser -filter * -SearchBase $SourceOU -properties *,$AttributeName
- **Find all users with $AttributeName =** $AttributeValue
  - Get-ADUser -filter { $_.”$AttributeName” -eq $AttributeValue } -properties $AttributeName
- **Find all users where $AttributeName has a value**
  - Get-ADUser -filter { $AttributeName –like “*” } –prop $AttributeName
- **Update $User $AttributeName to** "$AttributeValue
  - Set-ADUser $User -replace @{ "$AttributeName" = "$AttributeValue" }

GET-ADGROUP

```
PS C:\Windows\system32> get-adgroup "Administrators"



DistinguishedName : CN=Administrators,CN=Builtin,DC=lab,DC=adsecurity,DC=org
GroupCategory     : Security
GroupScope        : DomainLocal
Name              : Administrators
ObjectClass       : group
ObjectGUID        : db5e60b4-9e61-4712-a518-ce7d06a9db24
SamAccountName    : Administrators
SID               : S-1-5-32-544
```

GET AD DOMAIN GROUP STATISTICS

[array]$AllADGroups = **Get-ADGroup -Filter * -Properties ***
$AllADGroupsCount = $AllADGroups.Count
Write-Output “There are $AllADGroupsCount Total groups in AD `r “

[array]$ADUniversalGroups = **$AllADGroups | Where {$_.GroupScope -eq “Universal” }**
[int]$ADUniversalGroupsCount = $ADUniversalGroups.Count
Write-Output “There are $ADUniversalGroupsCount Universal groups in AD “

[array]$ADSecurityGroups = **$AllADGroups | Where {$_.GroupCategory -eq “Security” }**
$ADSecurityGroupsCount = $ADSecurityGroups.Count
Write-Output “There are $ADSecurityGroupsCount Security groups in AD “

GET-ADGROUPMEMBER

```
PS C:\Windows\system32> get-adgroupmember "Administrators"



distinguishedName : CN=svc-SQLReporting,OU=Service Accounts,DC=lab,DC=adsecurity,DC=org
name              : svc-SQLReporting
objectClass       : user
objectGUID        : d85ccfa7-bec2-43a8-bf3e-cbf7760b90bc
SamAccountName    : svc-SQLReporting
SID               : S-1-5-21-1473643419-774954089-2222329127-1609

distinguishedName : CN=admin,OU=Domain Management,DC=lab,DC=adsecurity,DC=org
name              : admin
objectClass       : user
objectGUID        : f608ef24-72b8-4013-9dda-03008d6fd56a
SamAccountName    : admin
SID               : S-1-5-21-1473643419-774954089-2222329127-1000

distinguishedName : CN=Domain Admins,CN=Users,DC=lab,DC=adsecurity,DC=org
name              : Domain Admins
objectClass       : group
objectGUID        : 66bbe7dd-1a23-4df1-9904-4ea276cdf303
SamAccountName    : Domain Admins
SID               : S-1-5-21-1473643419-774954089-2222329127-512

distinguishedName : CN=Enterprise Admins,CN=Users,DC=lab,DC=adsecurity,DC=org
name              : Enterprise Admins
objectClass       : group
objectGUID        : 833a5827-5d7c-44a7-b5a6-b1b5f6f1d4b1
SamAccountName    : Enterprise Admins
SID               : S-1-5-21-1473643419-774954089-2222329127-519

distinguishedName : CN=Administrator,OU=Domain Management,DC=lab,DC=adsecurity,DC=org
name              : Administrator
objectClass       : user
objectGUID        : bc70c1fd-9513-40d9-9e29-264cface3fcf
SamAccountName    : Administrator
SID               : S-1-5-21-1473643419-774954089-2222329127-500
```

GET LOGONTIMESYNCINTERVAL VALUE

```
PS C:\Windows\system32> $DomainDistinguishedName = (Get-ADDomain).DistinguishedName
$DirectoryServicesNamingContext = Get-ADObject -Identity "$DomainDistinguishedName" -Properties *
$LLTReplicationValue = $DirectoryServicesNamingContext."msDS-LogonTimeSyncInterval"

IF ($LLTReplicationValue -ge 1)
    { Write-Output "The msDS-LogonTimeSyncInterval attribute value on $DomainDNS was changed from the default value of 14 to $LLTReplicationVa
 ELSE
    { $LLTReplicationValue = 14 ; Write-Output "The msDS-LogonTimeSyncInterval attribute value on $DomainDNS is configured with the default va

The msDS-LogonTimeSyncInterval attribute value on  is configured with the default value of 14 (value is blank)
```

GET ACTIVE DIRECTORY INSTANTIATION DATE

```
PS C:\Windows\system32> Get-ADObject -SearchBase (Get-ADForest).PartitionsContainer `
-LDAPFilter "(&(objectClass=crossRef)(systemFlags=3))" `
-Property dnsRoot, nETBIOSName, whenCreated | Sort-Object whenCreated | Format-Table dnsRoot, nETBIOSName, whenCreated -AutoSize


dnsRoot             nETBIOSName whenCreated
-------             ----------- -----------
{lab.adsecurity.org} ADSECLAB    12/7/2014 11:16:54 AM
```

GET AD PASSWORD POLICY

```
PS C:\Windows\system32> Get-ADDefaultDomainPasswordPolicy


ComplexityEnabled           : True
DistinguishedName           : DC=lab,DC=adsecurity,DC=org
LockoutDuration             : 00:30:00
LockoutObservationWindow    : 00:30:00
LockoutThreshold            : 0
MaxPasswordAge              : 42.00:00:00
MinPasswordAge              : 1.00:00:00
MinPasswordLength           : 7
objectClass                 : {domainDNS}
objectGuid                  : f6d46828-b721-463d-9696-3b3714e2676a
PasswordHistoryCount        : 24
ReversibleEncryptionEnabled : False
```

GET AD TOMBSTONE LIFETIME

```
PS C:\Windows\system32> $ADForestconfigurationNamingContext = (Get-ADRootDSE).configurationNamingContext
$DirectoryServicesConfigPartition = Get-ADObject -Identity `
"CN=Directory Service,CN=Windows NT,CN=Services,$ADForestconfigurationNamingContext" `
 -Partition $ADForestconfigurationNamingContext -Properties *
$TombstoneLifetime = $DirectoryServicesConfigPartition.tombstoneLifetime
Write-Output "Active Directory's Tombstone Lifetime is set to $TombstoneLifetime days `r "

Active Directory's Tombstone Lifetime is set to 180 days
```

THE AD RECYCLE BIN

- Requires Forest Functional Mode = Windows Server 2008 R2
- **Enable the Recycle Bin (as Enterprise Admin)**
  Enable-ADOptionalFeature –Identity 'CN=Recycle Bin Feature,CN=Optional Features,CN=Directory Service,CN=Windows NT,CN=Services,CN=Configuration,DC=DOMAIN,DC=COM' –Scope ForestOrConfigurationSet –Target 'DOMAIN.COM'
- **Find all Deleted Users**
  $DeletedUsers = Get-ADObject -SearchBase "CN=Deleted Objects,DC=DOMAIN,DC=COM" -Filter {ObjectClass -eq "user"} -IncludeDeletedObjects -Properties lastKnownParent
- **Restore all Deleted Users**
  $DeletedUsers | Restore-ADObject
- **Restore users deleted on a specific date**
  $ChangeDate = Get-Date ("1/1/2015")
  Get-ADObject -Filter { (whenChanged -eq $changeDate) -and (isDeleted -eq $true) -and (name -ne "Deleted Objects") -and (ObjectClass -eq "user") } -IncludeDeletedObjects -Properties * | Restore-ADObject

GET DOMAIN RID STATS

```
PS C:\Windows\system32> $DomainDistinguishedName = (Get-ADDomain).DistinguishedName
$RIDManagerProperty = Get-ADObject "cn=rid manager$,cn=system,$DomainDistinguishedName" -property RIDAvailablePool `
-server ((Get-ADDomain).RIDMaster)
$RIDInfo = $RIDManagerProperty.RIDAvailablePool
[int32]$TotalSIDS = $RIDInfo / ([math]::Pow(2,32))
[int64]$Temp64val = $TotalSIDS * ([math]::Pow(2,32))
[int32]$CurrentRIDPoolCount = $RIDInfo - $Temp64val
$RIDsRemaining = $TotalSIDS - $CurrentRIDPoolCount

$RIDsIssuedPcntOfTotal = ( $CurrentRIDPoolCount / $TotalSIDS )
$RIDsIssuedPercentofTotal = "{0:P2}" -f $RIDsIssuedPcntOfTotal
$RIDsRemainingPcntOfTotal = ( $RIDsRemaining / $TotalSIDS )
$RIDsRemainingPercentofTotal = "{0:P2}" -f $RIDsRemainingPcntOfTotal

Write-Output "RIDs Issued: $CurrentRIDPoolCount ($RIDsIssuedPercentofTotal of total) `r "
Write-Output "RIDs Remaining: $RIDsRemaining ($RIDsRemainingPercentofTotal of total) `r "

RIDs Issued: 3101 (0.00 % of total)
RIDs Remaining: 1073738722 (100.00 % of total)
```

ENUMERATE DOMAIN TRUSTS

```
PS C:\Windows\system32> $DomainDNS = [System.DirectoryServices.ActiveDirectory.Domain]::GetCurrentDomain().Name
[array]$ADDomainTrusts = Get-ADObject -Filter {ObjectClass -eq "trustedDomain"} -Properties *
[int]$ADDomainTrustsCount = $ADDomainTrusts.Count

Write-Output "Discovered $ADDomainTrustsCount Trust(s) in $DomainDNS `r"
$ADDomainTrusts | select Name,Created,flatName,instanceType,trustAttributes,trustDirection,securityIdentifier | format-table -auto
Discovered 1 Trust(s) in lab.adsecurity.org

Name              Created              flatName instanceType trustAttributes trustDirection securityIdentifier
----              -------              -------- ------------ --------------- -------------- -------------------
rd.adsecurity.org 1/11/2015 5:09:45 PM ADSECRD             4               8              2 S-1-5-21-3834807805-851291830-904607491
```

GET AD SITES

```
PS C:\Windows\system32> $ADSites = [System.DirectoryServices.ActiveDirectory.Forest]::GetCurrentForest().Sites
[int]$ADSitesCount = $ADSites.Count
Write-Output "There are $ADSitesCount AD Sites `r"

$ADSites | select-object Name,Domains,Subnets,AdjacentSites,SiteLinks | format-table -AutoSize

There are 1 AD Sites

Name                    Domains             Subnets AdjacentSites SiteLinks
----                    -------             ------- ------------- ---------
Default-First-Site-Name {lab.adsecurity.org} {}      {}            {DEFAULTIPSITELINK}
```

BACKUP DOMAIN GPOS… FOR FREE!

Import-module GroupPolicy

Backup-GPO –All –Domain “mlab.adsecurity.org” –Path “c:\GPOBackup”

```
PS C:\Users\Administrator.ADSECMLAB> Backup-GPO -All -Domain "mlab.adsecurity.org" -Path "C:\GPOBackup"


DisplayName      : Default Domain Policy
GpoId            : 31b2f340-016d-11d2-945f-00c04fb984f9
Id               : f64bc902-e7d0-45f5-a702-ac610cf04a4b
BackupDirectory  : C:\GPOBackup
CreationTime     : 1/27/2015 8:30:42 PM
DomainName       : mlab.adsecurity.org
Comment          :

DisplayName      : Default Domain Controllers Policy
GpoId            : 6ac1786c-016f-11d2-945f-00c04fb984f9
Id               : 33ddea3b-c539-4b2c-bfe5-2e080f47dea0
BackupDirectory  : C:\GPOBackup
CreationTime     : 1/27/2015 8:30:47 PM
DomainName       : mlab.adsecurity.org
Comment          :
```

FINDING SERVICE ACCOUNTS

```
PS C:\Windows\system32> Get-ADUser -filter {ServicePrincipalName -like "*"} -property serviceprincipalname



DistinguishedName     : CN=krbtgt,CN=Users,DC=lab,DC=adsecurity,DC=org
Enabled               : False
GivenName             :
Name                  : krbtgt
ObjectClass           : user
ObjectGUID            : 6fd9529f-0805-4f3c-bb4d-29ad2ac377ef
SamAccountName        : krbtgt
serviceprincipalname  : {kadmin/changepw}
SID                   : S-1-5-21-1473643419-774954089-2222329127-502
Surname               :
UserPrincipalName     :

DistinguishedName     : CN=svc-SQLAgent01,OU=Service Accounts,DC=lab,DC=adsecurity,DC=org
Enabled               : True
GivenName             :
Name                  : svc-SQLAgent01
ObjectClass           : user
ObjectGUID            : eba3c611-6ea6-46bc-b68c-c8f28685e7f5
SamAccountName        : svc-SQLAgent01
serviceprincipalname  : {MSSQLSvc/ADSAPPSQL03.lab.adsecurity.org:1433,
                        MSSQLSvc/ADSAPPSQL02.lab.adsecurity.org:1433,
                        MSSQLSvc/ADSAPPSQL01.lab.adsecurity.org:1433}
SID                   : S-1-5-21-1473643419-774954089-2222329127-1606
Surname               :
UserPrincipalName     : svc-SQLAgent01@lab.adsecurity.org

DistinguishedName     : CN=svc-MSSQLServer01,OU=Service Accounts,DC=lab,DC=adsecurity,DC=org
Enabled               : True
GivenName             :
Name                  : svc-MSSQLServer01
ObjectClass           : user
ObjectGUID            : 2260906f-6985-404b-b6ea-fbed5d573bff
SamAccountName        : svc-MSSQLServer01
serviceprincipalname  : {MSSQLSvc/adsmswin2k8r2:1433, MSSQLSvc/adsmswin2k8r2.lab.adsecurity.org:1433}
SID                   : S-1-5-21-1473643419-774954089-2222329127-1613
Surname               :
UserPrincipalName     : svc-MSSQLServer01@lab.adsecurity.org
```

SERVICE ACCOUNTS INVENTORY SCRIPT

```
Discovering service account SPNs in the AD Domain lab.adsecurity.org


Domain                : lab.adsecurity.org
UserID                : krbtgt
PasswordLastSet       : 12/07/2014 16:17:39
LastLogon             : 01/01/1601 00:00:00
Description           : Key Distribution Center Service Account
SPNServers            :
SPNTypes              : {kadmin}
ServicePrincipalNames : {kadmin/changepw}

Domain                : lab.adsecurity.org
UserID                : svc-SQLAgent01
PasswordLastSet       : 01/01/1601 00:00:00
LastLogon             : 01/01/1601 00:00:00
Description           :
SPNServers            : {ADSAPPSQL01.lab.adsecurity.org, ADSAPPSQL02.lab.adsecurity.org, ADSAPPSQL03.lab.adsecurity.org}
SPNTypes              : {MSSQLSvc}
ServicePrincipalNames : {MSSQLSvc/ADSAPPSQL01.lab.adsecurity.org:1433, MSSQLSvc/ADSAPPSQL02.lab.adsecurity.org:1433,
                        MSSQLSvc/ADSAPPSQL03.lab.adsecurity.org:1433}

Domain                : lab.adsecurity.org
UserID                : svc-MSSQLServer01
PasswordLastSet       : 01/01/1601 00:00:00
LastLogon             : 01/01/1601 00:00:00
Description           :
SPNServers            : {adsmswin2k8r2.lab.adsecurity.org}
SPNTypes              : {MSSQLSvc}
ServicePrincipalNames : {MSSQLSvc/adsmswin2k8r2.lab.adsecurity.org:1433, MSSQLSvc/adsmswin2k8r2:1433}
```

https://github.com/PyroTek3/PowerShell-AD-Recon/blob/master/Find-PSServiceAccounts

DISCOVERING SERVICES IN AD WITH SPNS: SQL

```
PS C:\Windows\system32> get-adobject -filter { ServicePrincipalName -like "*SQL*" } -Properties Name,userPrincipalName,servicePrincipalName


DistinguishedName    : CN=svc-SQLAgent01,OU=Service Accounts,DC=lab,DC=adsecurity,DC=org
Name                 : svc-SQLAgent01
ObjectClass          : user
ObjectGUID           : eba3c611-6ea6-46bc-b68c-c8f28685e7f5
servicePrincipalName : {MSSQLSvc/ADSAPPSQL03.lab.adsecurity.org:1433, MSSQLSvc/ADSAPPSQL02.lab.adsecurity.org:1433,
                       MSSQLSvc/ADSAPPSQL01.lab.adsecurity.org:1433}
userPrincipalName    : svc-SQLAgent01@lab.adsecurity.org

DistinguishedName    : CN=svc-MSSQLServer01,OU=Service Accounts,DC=lab,DC=adsecurity,DC=org
Name                 : svc-MSSQLServer01
ObjectClass          : user
ObjectGUID           : 2260906f-6985-404b-b6ea-fbed5d573bff
servicePrincipalName : {MSSQLSvc/adsmswin2k8r2:1433, MSSQLSvc/adsmswin2k8r2.lab.adsecurity.org:1433}
userPrincipalName    : svc-MSSQLServer01@lab.adsecurity.org
```

Active Directory SPN Directory:
http://adsecurity.org/?page_id=183

INVENTORY SQL SERVERS

```
Domain             : lab.adsecurity.org
ServerName         : adsmswin2k8r2.lab.adsecurity.org
Port               : 1433
Instance           :
ServiceAccountDN   : {CN=svc-MSSQLServer01,OU=Service Accounts,DC=lab,DC=adsecurity,DC=org}
OperatingSystem    : {Windows Server 2008 R2 Datacenter}
OSServicePack      : {Service Pack 1}
LastBootup         : 1/17/2015 7:15:36 PM
OSVersion          : {6.1 (7601)}
Description        :
SrvAcctUserID      : svc-MSSQLServer01
SrvAcctDescription :
```

https://github.com/PyroTek3/PowerShell-AD-Recon/blob/master/Discover-PSMSSQLServers

FINDING DOMAIN CONTROLLERS

- **Get-ADDomain**

  import-module ActiveDirectory

  $ADInfo = Get-ADDomain

  $ADDomainReadOnlyReplicaDirectoryServers =

  $ADInfo.ReadOnlyReplicaDirectoryServers

  $ADDomainReplicaDirectoryServers = **$ADInfo.ReplicaDirectoryServers**

  $DomainControllers = $ADDomainReadOnlyReplicaDirectoryServers + `

  ADDomainReplicaDirectoryServers

- **Get-ADDomainController**

  import-module ActiveDirectory

  $DomainControllers = Get-ADDomainController -filter * -DomainName $DOMAIN

DOMAIN CONTROLLER INVENTORY

```
Import-Module ActiveDirectory
Get-ADDomainController –filter * | `
 select hostname,IPv4Address,IsGlobalCatalog,IsReadOnly,OperatingSystem | `
 format-table -auto
```

```
hostname                        IPv4Address  IsGlobalCatalog IsReadOnly OperatingSystem
--------                        -----------  --------------- ---------- ---------------
adsmlabdc1.mlab.adsecurity.org 172.16.16.11            True      False Windows Server 2008 R2 Datacenter
adsmlabdc5.mlab.adsecurity.org 172.16.16.12            True      False Windows Server 2012 R2 Datacenter
```

DOMAIN CONTROLLERS DISCOVERY

- **Discover PDCe in domain:**
  Get-ADDomainController –Discover –**ForceDiscover** –Service “PrimaryDC” –DomainName “lab.adsecurity.org”

- **Discover DCs in a Site:**
  Get-ADDomainController –Discover –Site “HQ”

- **Find all Read-Only Domain Controllers that are GCs**
  Get-ADDomainController –filter `
  { (isGlobalCatalog –eq $True) –AND (isReadOnly –eq $True) }

DISCOVERING GLOBAL CATALOGS (GCS)

- **Forest GCs**

```
import-module ActiveDirectory
$ADForest = Get-ADForest
$ADForestGlobalCatalogs = $ADForest.GlobalCatalogs
```

- **Domain DCs that are GCs**

```
import-module ActiveDirectory
$DCsNotGCs = Get-ADDomainController -filter { IsGlobalCatalog -eq $True}
```

- **Domain DCs that are <u>not</u> GCs**

```
import-module ActiveDirectory
$DCsNotGCs = Get-ADDomainController -filter { IsGlobalCatalog -eq $False }
```

ACTIVE DIRECTORY DATABASE INTEGRITY CHECK

```
Write-Output "Checking the NTDS database for errors (semantic database analysis) `r "
Stop-Service ntds -force
$NTDSdbChecker = ntdsutil "activate instance ntds" "semantic database analysis" "verbose on" "Go" q q
Start-Service ntds
Write-Output "Results of Active Directory database integrity check: `r "
$NTDSdbChecker
```

FINDING FSMOS

- **AD Cmdlets**

  Import-Module ActiveDirectory

  (Get-ADForest).SchemaMaster

  (Get-ADForest).DomainNamingMaster

  (Get-ADDomain).InfrastructureMaster

  (Get-ADDomain).PDCEmulator

  (Get-ADDomain).RIDMaster

- **.Net**

  ([System.DirectoryServices.ActiveDirectory.Forest]::GetCurrentForest()).SchemaRoleOwner

  ([System.DirectoryServices.ActiveDirectory.Forest]::GetCurrentForest()).NamingRoleOwner

  ([System.DirectoryServices.ActiveDirectory.Domain]::GetCurrentDomain()).InfrastructureRoleOwner

  ([System.DirectoryServices.ActiveDirectory.Domain]::GetCurrentDomain()).PdcRoleOwner

  ([System.DirectoryServices.ActiveDirectory.Domain]::GetCurrentDomain()).RidRoleOwner

MOVING FSMOS

- **Can PowerShell move the FSMO role from one DC to another?**

  get-command -module activedirectory -noun *Master*

- **Moving FSMO Roles**

  Move-ADDirectoryServerOperationMasterRole -Identity $DCName -OperationMasterRole RIDMaster

  Move-ADDirectoryServerOperationMasterRole -Identity $DCName -OperationMasterRole DomainNamingMaster

  Move-ADDirectoryServerOperationMasterRole -Identity $DCName -OperationMasterRole PDCEmulator

- **Seizing FSMO Roles**

  Move-ADDirectoryServerOperationMasterRole -Identity $DCName -OperationMasterRole PDCEmulator –FORCE

REPADMIN VS. POWERSHELL

REPADMIN	PowerShell
	2012 Cmdlets
/FailCache	Get-ADReplicationFailure
/Queue	Get-ADReplicationQueueOperation
/ReplSingleObj	Sync-ADObject
/ShowConn	Get-ADReplicationConnection
/ShowObjMeta	Get-ADReplicationAttributeMetadata
/ShowRepl /ReplSum	Get-ADReplicationPartnerMetadata
/ShowUTDVec	Get-ADReplicationUpToDatenessVectorTable
/SiteOptions	Set-ADReplicationSite
	2008 R2 Cmdlets
/ShowAttr	Get-ADObject
/SetAttr	Set-ADObject
/PRP	Get-ADDomainControllerPasswordReplicationPolicy
	Add-ADDomainControllerPasswordReplicationPolicy
	Remove-ADDomainControllerPasswordReplicationPolicy
	Get-ADAccountResultantPasswordReplicationPolicy
	Get-ADDomainControllerPasswordReplicationPolicyUsage

REPLICATION CMDLETS (2012)
GET-ADREPLICATIONPARTNERMETADATA

```
Get-ADReplicationPartnerMetadata -Target "adsmlabdc1"


CompressChanges                : False
ConsecutiveReplicationFailures : 0
DisableScheduledSync           : False
IgnoreChangeNotifications      : False
IntersiteTransport             :
IntersiteTransportGuid         :
IntersiteTransportType         : IP
LastChangeUsn                  : 13042
LastReplicationAttempt         : 1/27/2015 9:14:54 PM
LastReplicationResult          : 0
LastReplicationSuccess         : 1/27/2015 9:14:54 PM
Partition                      : DC=mlab,DC=adsecurity,DC=org
PartitionGuid                  : e2e5fdb0-bd05-4c73-8ab8-c28d054a7a2b
Partner                        : CN=NTDS Settings,CN=ADSMLABDC5,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=ml
                                 ab,DC=adsecurity,DC=org
PartnerAddress                 : 326cb425-1ae0-4e46-8a08-9b526cacfaeb._msdcs.mlab.adsecurity.org
PartnerGuid                    : 326cb425-1ae0-4e46-8a08-9b526cacfaeb
PartnerInvocationId            : 86bafa17-f779-4233-9028-f3b688b56bef
PartnerType                    : Inbound
ScheduledSync                  : True
Server                         : adsmlabdc1.mlab.adsecurity.org
SyncOnStartup                  : True
TwoWaySync                     : False
UsnFilter                      : 13042
Writable                       : True
```

REPLICATION CMDLETS (2012)
GET-ADREPLICATIONPARTNERFAILURE

```
Get-ADReplicationFailure -Target "adsmlabdc1"


FailureCount     : 14
FailureType      : Connection
FirstFailureTime : 1/27/2015 6:32:05 PM
LastError        : 8524
Partner          : CN=NTDS Settings,CN=ADSMLABDC2,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=mlab,DC=adsecuri
                   ty,DC=org
PartnerGuid      : 72a1a78e-c1b6-4d15-a066-c8e634220ab9
Server           : adsmlabdc1.mlab.adsecurity.org
```

REPLICATION CMDLETS (2012)
GET-ADREPLICATIONUPTODATENESSVECTORTABLE

```
PS C:\Users\Administrator.ADSECMLAB> Get-ADReplicationUpToDatenessVectorTable -Target "adsmlabdc1"


LastReplicationSuccess : 1/27/2015 9:14:54 PM
Partition              : DC=mlab,DC=adsecurity,DC=org
PartitionGuid          : e2e5fdb0-bd05-4c73-8ab8-c28d054a7a2b
Partner                : CN=NTDS Settings,CN=ADSMLABDC5,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=mlab,DC=ad
                         security,DC=org
PartnerInvocationId    : 86bafa17-f779-4233-9028-f3b688b56bef
Server                 : adsmlabdc1.mlab.adsecurity.org
UsnFilter              : 13042

LastReplicationSuccess : 1/27/2015 9:36:54 PM
Partition              : DC=mlab,DC=adsecurity,DC=org
PartitionGuid          : e2e5fdb0-bd05-4c73-8ab8-c28d054a7a2b
Partner                : CN=NTDS Settings,CN=ADSMLABDC1,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=mlab,DC=ad
                         security,DC=org
PartnerInvocationId    : 74a62edd-53bc-45e0-9ed4-cae49998cc9f
Server                 : adsmlabdc1.mlab.adsecurity.org
UsnFilter              : 21021
```

TIPS & TRICKS

- Schedule a Backup-GPO PowerShell script to run weekly (at least!)
- ADSI properties are often Case senSitive
- Properties with a space or special character require quotes:
  - $Object.”Property with spaces”
- PowerShell Remoting uses TCP 5985 (HTTP) or TCP 5986 (HTTPS)
- Don’t use credentials in Group Policy Preferences (MS14-025)

```
<?xml version=”1.0” encoding=”utf-8”?>
<Groups clsid=”{3125E937-EB16-4b4c-9934-544FC6D24D26}”>
<User clsid=”{DF5F1855-51E5-4d24-8B1A-D9BDE98BA1D1}” name=”LocalTestUser” image=”0” changed=”2013-07-04 00:07:13” uid=”{47F24835-4B58-4C48-A749-5747EAC84669}”>
<Properties action=”C” fullName=”” description=”” cpassword=”sFWOJZOU7bJICaqvmd+KAEN0o4RcpxxMLWnK7s7zgNR+JiJwoSa
+DLU3kAIdXc1WW5NKrIjIe9MIdBuJHvqFgbcNS873bDK2nbQBqpydkjbsPXV0HRPpQ96ph1e6N9tn4NF3KYyswokkDnj8gvuyZBXqoG94ML8M1Iq7/jhe37eHJiZGyi5IBoPuCfKpurj2” changeLogon=”0”
noChange=”0” neverExpires=”0” acctDisabled=”0” userName=”LocalTestUser”/>
</User>
</Groups>
```

REFERENCES

- ADSecurity.org – Filled with AD/Microsoft Security & PowerShell Goodness
- CMD to PowerShell Reference:
  http://blogs.technet.com/b/ashleymcglone/archive/2013/01/02/free-download-cmd-to-powershell-guide-for-ad.aspx
- PowerShell AD Cmdlet SOAP XML Messages
  http://blogs.msdn.com/b/adpowershell/archive/2009/10/05/how-to-view-soap-xml-messages-to-and-from-ad-webservices-and-powershell.aspx
- The AD: Drive
  http://blogs.technet.com/b/heyscriptingguy/archive/2013/03/18/playing-with-the-ad-drive-for-fun-and-profit.aspx
- Repadmin to PowerShell
  http://blogs.technet.com/b/ashleymcglone/archive/2012/10/17/ad-group-history-mystery-powershell-v3-repadmin.aspx
- Quest AD cmdlets are useful, though not as necessary.
  https://support.software.dell.com/download-install-detail/5024645

BONUS SLIDES FOLLOW…

VALIDATE INPUT AS IP ADDRESS

- $IPAddress = ’10.10.10.10’
  $IPAddressCheck = [System.Net.IPAddress]::parse($IPAddress)

- If there's data in $IPAddressCheck it's a valid IP.
  Example Result:
  Address : 168430090
  AddressFamily : InterNetwork
  ScopeId :
  IsIPv6Multicast : False
  IsIPv6LinkLocal : False
  IsIPv6SiteLocal : False
  IPAddressToString : 10.10.10.10

ENUMERATE DOMAIN DFS SHARES

```
$DomainDNS = [System.DirectoryServices.ActiveDirectory.Domain]::GetCurrentDomain().Name
$ADDomainDistinguishedName = (Get-ADDomain).DistinguishedName
$DFSConfigObjectDN = "CN=Dfs-Configuration,CN=System,$ADDomainDistinguishedName“
$DFSConfigurationObject = Get-ADObject $DFSConfigObjectDN
$DFSShareData = Get-ChildItem "AD:$DFSConfigObjectDN"

ForEach ($DFSShareDataItem in $DFSShareData)
 {  ## OPEN ForEach ($DFSShareDataItem in $DFSShareData)
$DFSShareDataItemDN = $DFSShareDataItem.DistinguishedName
$DFSShareDataItemNameArray = $DFSShareDataItemDN -split '='
$DFSShareDataItemNameArray2 = $DFSShareDataItemNameArray -split ','
$DFSShareDataItemName = $DFSShareDataItemNameArray2[1]
$DFSShareDataItemServerPath = Get-ADObject $DFSShareDataItemDN -property *,RemoteServerName
Write-Output "DFS Share Name: $DFSShareDataItemName `r “
Write-Output "$DFSShareDataItemDN `r “
$DFSShareDataItemServerPathName = ($DFSShareDataItemServerPath.RemoteServerName) -replace ('\*',"")
$DFSShareDataItemServerPathName
write-output " `r “
  }  ## CLOSE ForEach ($DFSShareDataItem in $DFSShareData)
```

CONVERT DOMAIN DN TO FQDN

```
$ADObjectDN = "CN=Object1,OU=OrgUnit1,DC=child,DC=domain,DC=com"
[array]$ADObjectDNArray = $ADObjectDN -Split(",DC=")
        [int]$DomainNameFECount = 0
        ForEach ($ADObjectDNArrayItem in $ADObjectDNArray)
            {
                IF ($DomainNameFECount -gt 0)
                { [string]$ADObjectDNArrayItemDN += $ADObjectDNArrayItem + "." }
                $DomainNameFECount++
            }
$ADObjectDNDomainName = $ADObjectDNArrayItemDN.Substring(0,$ADObjectDNArrayItemDN.Length-1)
```

CONVERT DOMAIN FQDN TO DN

```
$DomainFullyQualifiedDomainName = "child.domain.com"
$DomainFullyQualifiedDomainNameArray = $DomainFullyQualifiedDomainName -Split("\.")
[int]$DomainNameFECount = 0
ForEach ($DomainFullyQualifiedDomainNameArrayItem in $DomainFullyQualifiedDomainNameArray)
{
    IF ($DomainNameFECount -eq 0)
        { [string]$ADObjectDNArrayItemDomainName += "DC=" +$DomainFullyQualifiedDomainNameArrayItem }
    ELSE
        { [string]$ADObjectDNArrayItemDomainName += ",DC=" +$DomainFullyQualifiedDomainNameArrayItem }
    $DomainNameFECount++
}
$ADObjectDNArrayItemDomainName
```