Users Enumeration

After Make UserEnum you have to know all users and properties and all Machines

Get the list of users

Get-NetUser

#Get user in a Domain

Get-NetUser *admin* -Domain grey.wargrey.mon

#Get Admins in a Domain

Get-NetUser -AdminCount

Get-NetUser -AdminCount -Domain wargrey.mon

Fitler by username

Get-DomainUser -Domain wargrey.mon | ?{$_.name -match "Grey Mon"}

Grab the cn (common-name) from the list of users

Get-NetUser | select cn

Get actively logged users on a computer (needs local admin rights on the target)

Get-NetLoggedon -ComputerName

List all properties

Get-UserProperty

Display when the passwords were set last time

Get-UserProperty -Properties pwdlastset

Display when the accounts were created

Get-UserProperty -Properties whencreated

Get the list of users

Get-ADUser -Filter *

Get the list of users with properties

Get-ADUser -Filter *-Properties*

List samaccountname and description for users

Get-ADUser -Filter *-Properties* | select Samaccountname,Description

Get the list of users from cn common-name

Get-ADUser -Filter *-Properties* | select cn

Get the list of users from name

Get-ADUser -Filter *-Properties* | select name

Displays when the password was set

Get-ADUser -Filter *-Properties* | select name,@{expression= {[datetime]::fromFileTime($_.pwdlastset)}}

Get-NetUser | select samaccountname, lastlogon, pwdlastset

Get-NetUser | select samaccountname, lastlogon, pwdlastset | Sort-Object -Property lastlogon

#Get list of usernames and their groups

Get-NetUser | select samaccountname, memberof

#Get descripton field from the user

Find-UserField -SearchField Description -SearchTerm "built"

Get-netuser | Select-Object samaccountname,description

#Get SID for users

WMIC.exe useraccount get name,sid

#Basic user enabled info

Get-NetUser -UACFilter NOT_ACCOUNTDISABLE | select samaccountname, description, pwdlastset, logoncount, badpwdcount

#Find users with sidHistory set

Get-NetUser -LDAPFilter '(sidHistory=*)'

#search if you have local admin on any machine joined domain or not

#Find all machines on the current domain where the current user has local admin access

Find-LocalAdminAccess -Verbose

#This can also be done with the help of remote administration tools like WMI and PowerShell remoting. Pretty useful in cases ports (RPC and SMB) used by Find-LocalAdminAccess are blocked.

Find-WMILocalAdminAccess.ps1

Find-PSRemotingLocalAdminAccess.ps1

Kerberoasting Eumeration

#ASREPRoastable users

Get-NetUser -PreauthNotRequired

#Kerberoastable users

Get-NetUser -SPN

Get-NetComputer -SPN

#Kerberospolicy

(Get-DomainPolicyData).kerberospolicy

#Groups info

Get-NetGroup | select samaccountname, admincount, description

#Get AdminSDHolders

Get-DomainObjectAcl -SearchBase 'CN=AdminSDHolder,CN=System,DC=wargrey,DC=mon' | %{ $_.SecurityIdentifier } | Convert-SidToName

---

Computer

#basic

Get-NetComputer

Get-ADComputer -Filter *

#Get Computer name and OS

Get-NetComputer | select samaccountname, operatingsystem

Get-NetComputer -Domain wargrey.mon | select samaccountname, operatingsystem

Get-NetComputer -OperatingSystem "*Server 2016*"

#DCs always appear but aren't useful for privesc

Get-NetComputer -Unconstrained | select samaccountname

#Find computers with Constrined Delegation

Get-NetComputer -TrustedToAuth | select samaccountname

#Find any machine accounts in privileged groups

Get-DomainGroup -AdminCount | Get-DomainGroupMember -Recurse | ? {$_.MemberName -like '*'}

Get-NetGroupMember -Identity "Domain Admins" -Recurse | select MemberName

#List all the local groups on a machine (needs admin privs on non dc machines)

Get-NetlocalGroup -Computername -ListGroups

#Get Member of all the local groups on a machine (needs admin privs on non dc machines)

Get-NetlocalGroup -Computername -Recurse

#Get actively logged users on a computer (needs local admin privs)

Get-NetLoggedon -Computername

#Get locally logged users on a computer (needs remote registry rights on the target)

Get-LoggedonLocal -Computername

#Get the last logged users on a computer (needs admin rights and remote registary on the target)

Get-LastLoggedOn -ComputerName

#get computer operating system and other important info

Get-ADComputer -Filter * -Property PrimaryGroupID

Get-ADComputer -Filter {PrimaryGroupID -eq ""} -Properties OperatingSystem,OperatingSystemVersion,OperatingSystemServicePack,PasswordLastSet,LastLogonDate,ServicePrincipalName,TrustedForDelegation,TrustedtoAuthForDelegation

---

#Find computers where a domain admin (or specified user/group) has sessions:

Find-DomainUserLocation -Verbose

Find-DomainUserLocation -UserGroupIdentity “RDPUsers”

Find computers where a domain admin session is available and current user has admin access (uses Test-AdminAccess).

#Find-DomainUserLocation -CheckAccess

Find computers (File Servers and Distributed File servers) where a domain admin session is available.

#Find-DomainUserLocation -Stealth

Shares

#Search readable shares

Find-DomainShare -CheckShareAccess

Groups and Members Enumeration

#baisc

Get-NetGroup

Get-NetLocalGroup

#Get all groups that contain the word "admin" in the group name

Get-NetGroup *Admin*

Get-NetGroupMember 'Domain Admins' -Recurse

Get-NetGroupMember 'Administrator' -Recurse

Get-NetGroupMember 'Remote Desktop Users' -Recurse

Get-NetGroupMember 'Remote Desktop' -Recurse

#Get all members of the "Domain Admins" group

Get-NetGroupMember -GroupName "Domain Admins" -Recurse

#Query the root domain as the "Enterprise Admins" group exists only in the root of a forest

Get-NetGroupMember -GroupName "Enterprise Admins" –Domain wargrey.mon

Get group membership for user "grey"

Get-NetGroup -UserName "grey"

Get-NetGroup -GroupName "Users" -Fulldata

Get all groups that contain the word "admin" in the group name

Get-ADGroup -Filter 'Name -like "*admin*"' | select Name

Get all members of the "Domain Admins" group

Get-ADGroupMember -Identity "Domain Admins" -Recursive

Get group membership for "grey"

Get-ADPrincipalGroupMembership -Identity grey

Get-ADComputer -Filter | *select Name Get-ADComputer -Filter 'OperatingSystem -like "*Server 2016*"" -Properties OperatingSystem | select Name,OperatingSystem

Get-ADComputer -Filter * -Properties DNSHostName | %{Test-Connection -Count 1 -ComputerName $_.DNSHostName}

Enum Domain Group

#Get all the groups in the current domain

Get-DomainGroup | select Name

Get-DomainGroup –Domain

Get-ADGroup -Filter * | select Name

Get-ADGroup -Filter *-Properties*

#Get all groups containing the word "admin" in group name

Get-DomainGroup *admin*

Get-ADGroup -Filter 'Name -like "*admin*"' | select Name

#Get all the members of the Domain Admins group

Get-DomainGroupMember -Identity "Domain Admins" -Recurse

Get-ADGroupMember -Identity "Domain Admins" -Recursive

#Get the group membership for a user:

Get-DomainGroup –UserName "grey"

Get-ADPrincipalGroupMembership -Identity grey

#Get Group admins

Get-NetGroup "*admins*" | Get-NetGroupMember -Recurse | ?{$_.MemberName -Like "*.*"}

#Get Clients on Host Domain

Get-NetGroup -ComputerName PDC

Get-NetGroup -ComputerName dc02

Password Policy

Get-DomainPolicyData

- SystemAccess
- KerberosPolicy