Active Directory Enumeration with PowerShell

By Haboob Team
Research@haboob.sa

Table of Contents

INTRODUCTION

Nowadays, most of the environments are using Active Directory to manage their networks and resources. And over the past years, the attackers have been focused to abuse and attack the Active Directory environments using different techniques and methodologies. So in this research paper, we are going to use the power of the PowerShell to enumerate the resources of the Active Directory, like enumerating the domains, users, groups, ACL, GPOs, domain trusts also hunting the users and the domain admins. With this valuable information, we can increase our attack surface to abuse the AD like Privilege escalation, lateral movements and persistence and so on.

WHY POWERSHELL?

Penetration Tests and Red Team operations for secured environments need altered approaches. You cannot afford to touch disk, throw executable and use memory corruption exploits without the risk of being ineffective as a simulated adversary. To enhance offensive tactics and methodologies, PowerShell is the tool of choice.

PowerShell has changed the way Windows networks are attacked. It is Microsoft's shell and scripting language available by default in all modern Windows computers. It could interact with .Net, WMI, COM, Windows API, Registry and other computers on a Windows Domain. This makes it imperative for Penetration Testers and Red Teamers to learn PowerShell.

ATTACK DEMONSTRATION

In the attack demonstration, we are going to use the tool PowerView. PowerView is a PowerShell script which was developed by Will Schroeder and is part of PowerSploit framework. The script relies solely on PowerShell and WMI (Windows Management Instrumentation) queries.

We have built an Active Directory lab that simulates a real world environment with a set of machines, users, domains, misconfigurations. In this lab, we will simulate the attack as we have a limited shell on a Windows machine (joined-domain). From there, we will enumerate the domain using only PowerShell and we will not rely on any exploits or attack platform (like Kali Linux).

DOMAIN ENUMERATION

Let's start with enumerating the domains, like enumerating the users, groups, some interesting fields and resources.

Get-NetDomain

This command will give us information about the current domain like the domain name and the domain controller:

```
PS C:\Users\yasser\Desktop> Get-NetDomain

Forest                  : Fanzy.com
DomainControllers       : {DC-01.Fanzy.com}
Children                : {USH.Fanzy.com}
DomainMode              :
Parent                  :
PdcRoleOwner            : DC-01.Fanzy.com
RidRoleOwner            : DC-01.Fanzy.com
InfrastructureRoleOwner : DC-01.Fanzy.com
Name                    : Fanzy.com
```

As shown above, the domain name is (**Fanzy.com**) and the DC is (**DC-01.Fanzy.com**)

Get-NetDomain -domain "Domain Name"

If you want to get the same results for another domain, use the above command.

```
PS C:\Users\yasser\Desktop> Get-NetDomain -Domain Dampy

Forest                  : Dampy.com
DomainControllers       : {DC-02.Dampy.com}
Children                : {}
DomainMode              :
Parent                  :
PdcRoleOwner            : DC-02.Dampy.com
RidRoleOwner            : DC-02.Dampy.com
InfrastructureRoleOwner : DC-02.Dampy.com
Name                    : Dampy.com
```

Get-DomainSID

Use this command to get the domain SID (Security IDentifier is a unique ID number that a computer or domain controller uses to identify you).

```
PS C:\Users\yasser\Desktop> Get-DomainSID
S-1-5-21-3156372763-3995679764-3492621305
```

Get-DomainPolicy
(Get-DomainPolicy)."system access"

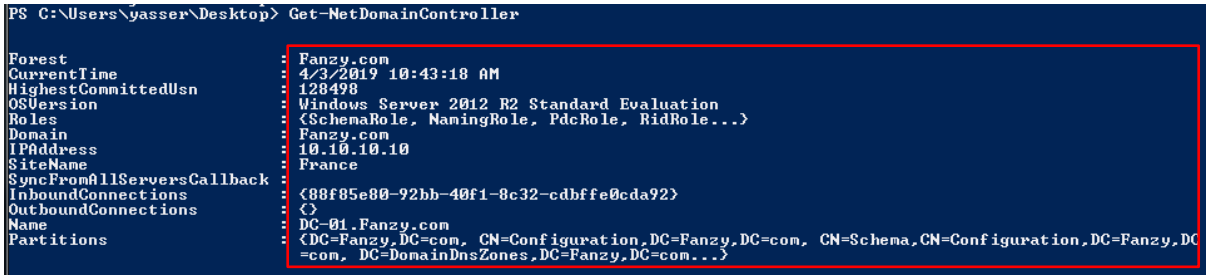
Use this command to get the policy of the current domain.

```
PS C:\Users\yasser\Desktop> (Get-DomainPolicy)."system access"

Name                                Value
----                                -----
PasswordComplexity                  {1}
MaximumPasswordAge                  {42}
MinimumPasswordLength               {7}
ForceLogoffWhenHourExpire           {0}
LSAAnonymousNameLookup              {0}
LockoutBadCount                     {0}
PasswordHistorySize                 {24}
MinimumPasswordAge                  {1}
ClearTextPassword                   {0}
RequireLogonToChangePassword        {0}
```

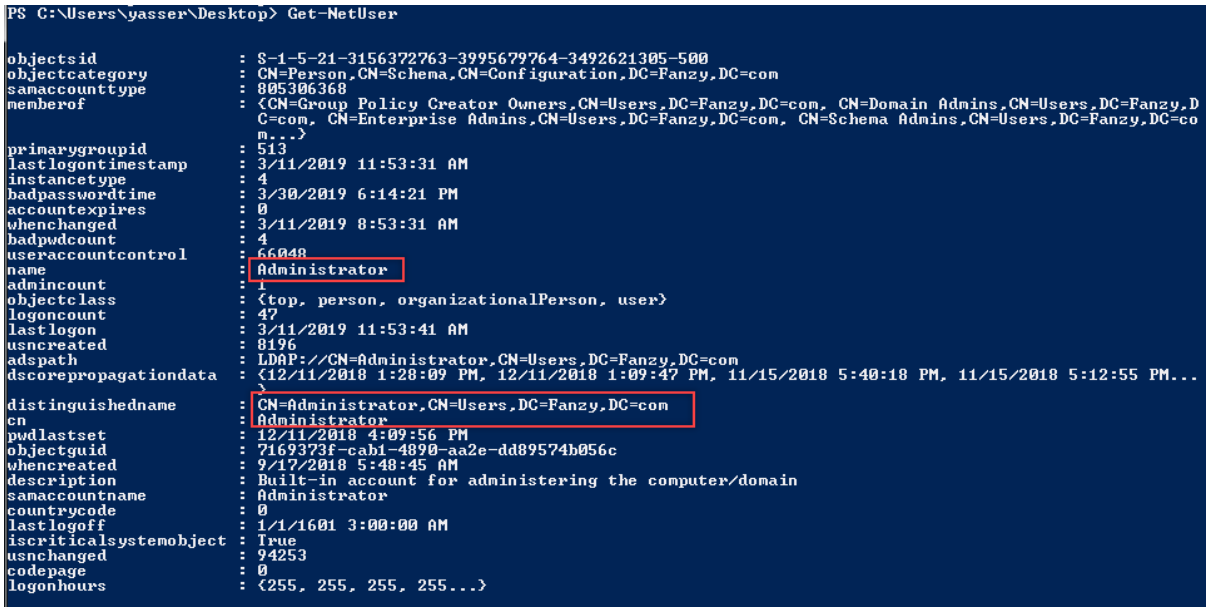
Get-NetDomainController

Use this command to get information about the current domain controller (DC).



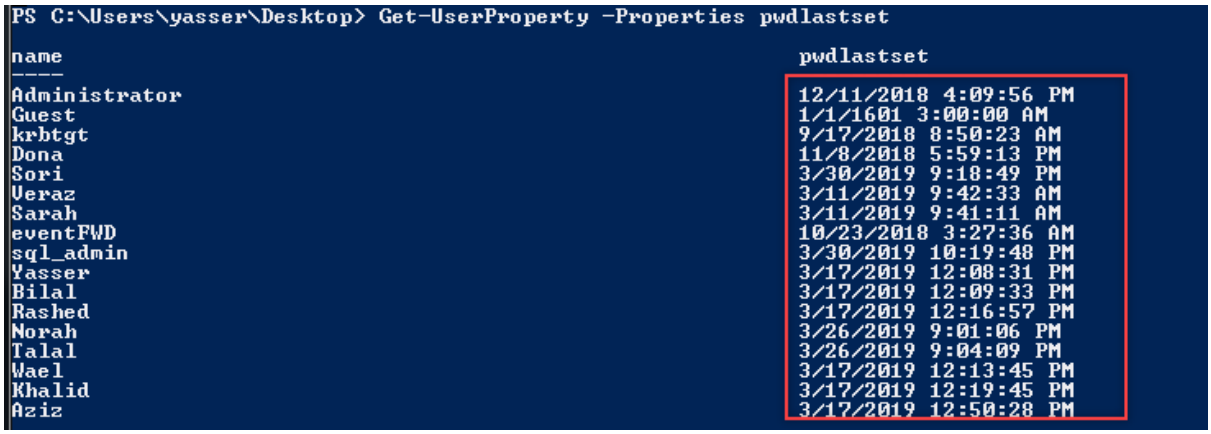
Get-NetUser

Use this command to list all the users in the current domain with information about each user.



Get-UserProperty –Properties pwdlastset

Use this command to see the last password set of each user in the current domain.



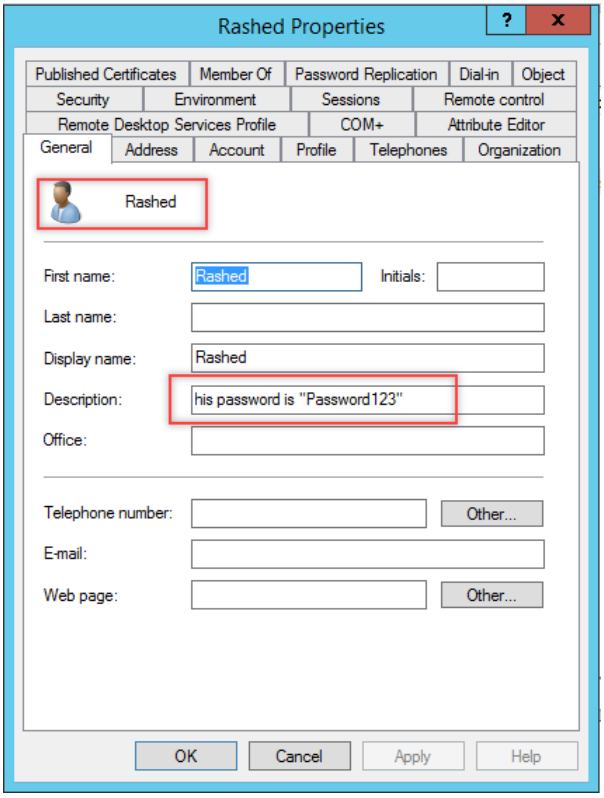
Find-UserField -SearchField Description –SearchTerm “pass”

Most of the system administrators are lazy and they don’t care about how to save the passwords! The above command will search for the word “pass” in the field “description” for each user in the domain.

```
PS C:\Users\yasser\Desktop> Find-UserField -SearchField Description -SearchTerm "pass"

samaccountname                                          description
--------------
Rashed                                                  his password is "Password123"
```

To make it more clear, here what it looks like in the description of the user “Rashed” from the Active Directory:



Get-NetComputer

Use this command to list all the computers in the current domain.

```
PS C:\Users\yasser\Desktop> Get-NetComputer
DC-01.Fanzy.com
Client-02.Fanzy.com
CLIENT-01.Fanzy.com
CDC-01.Fanzy.com
SQL-Server.Fanzy.com
```

Get-NetComputer –OperatingSystem "Windows 7 Ultimate"

Use this command to list all the operating systems “Windows 7 Ultimate”.

```
PS C:\Users\yasser\Desktop> Get-NetComputer -OperatingSystem "Windows 7 Ultimate"
CLIENT-01.Fanzy.com
```

Get-NetComputer -Ping

Use this command to get all the pingable computers (live hosts) in the current domain.

```
PS C:\Users\yasser\Desktop> Get-NetComputer -Ping
DC-01.Fanzy.com
Client-02.Fanzy.com
CLIENT-01.Fanzy.com
```

Get-NetGroup

Use this command to get all the groups in the current domain.

```
PS C:\Users\yasser\Desktop> Get-NetGroup
WinRMRemoteWMIUsers__
Administrators
Users
Guests
Print Operators
Backup Operators
Replicator
Remote Desktop Users
Network Configuration Operators
Performance Monitor Users
Performance Log Users
Distributed COM Users
IIS_IUSRS
Cryptographic Operators
Event Log Readers
Certificate Service DCOM Access
RDS Remote Access Servers
RDS Endpoint Servers
RDS Management Servers
Hyper-V Administrators
Access Control Assistance Operators
Remote Management Users
Domain Computers
Domain Controllers
Schema Admins
Enterprise Admins
Cert Publishers
Domain Admins
Domain Users
Domain Guests
Group Policy Creator Owners
RAS and IAS Servers
Server Operators
Account Operators
Pre-Windows 2000 Compatible Access
Incoming Forest Trust Builders
Windows Authorization Access Group
```

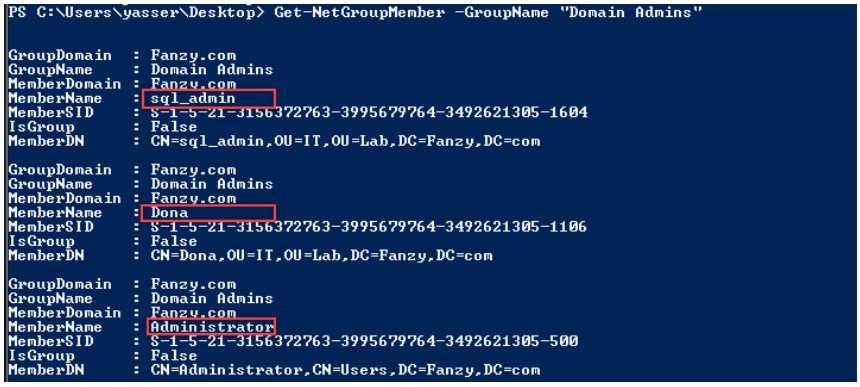
Get-NetGroup *admin*

Use this command to get all the groups that contain the word “admin” in the group name.

```
PS C:\Users\yasser\Desktop> Get-NetGroup *admin*
Administrators
Hyper-V Administrators
Schema Admins
Enterprise Admins
Domain Admins
DnsAdmins
DHCP Administrators
IT_Admins
```

Get-NetGroupMember -GroupName "Domain Admins"

Use this command to get the members of the group “Domain Admin”.

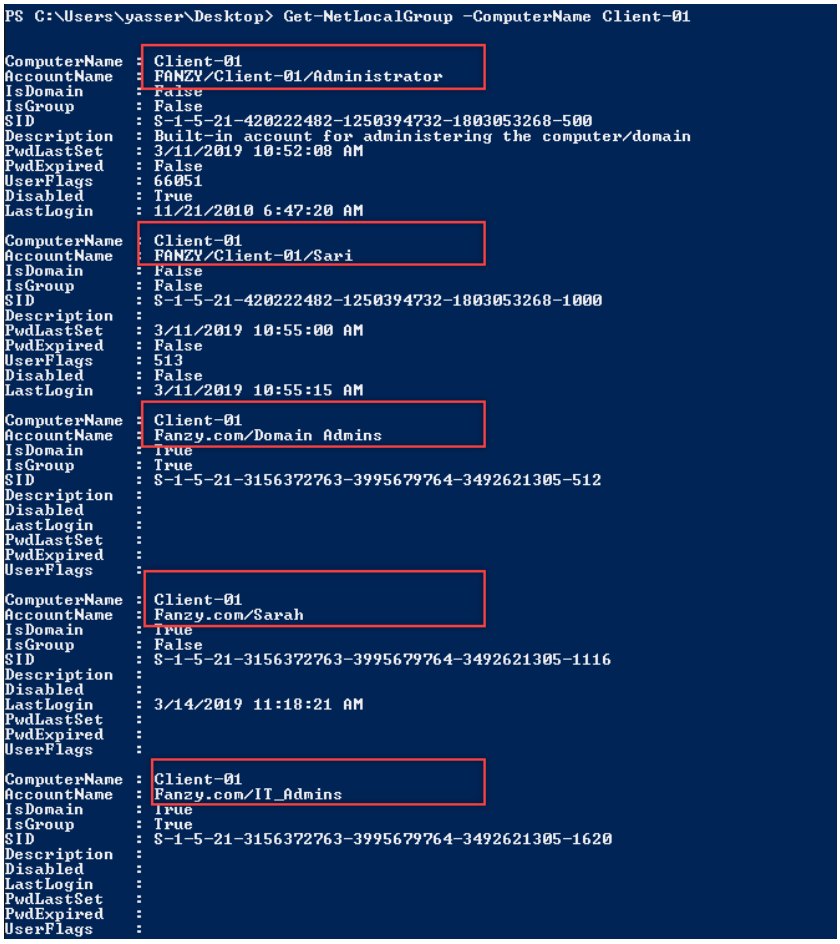


Get-NetGroup –UserName "khalid"

Use this command to get the group membership of the user “Khalid”

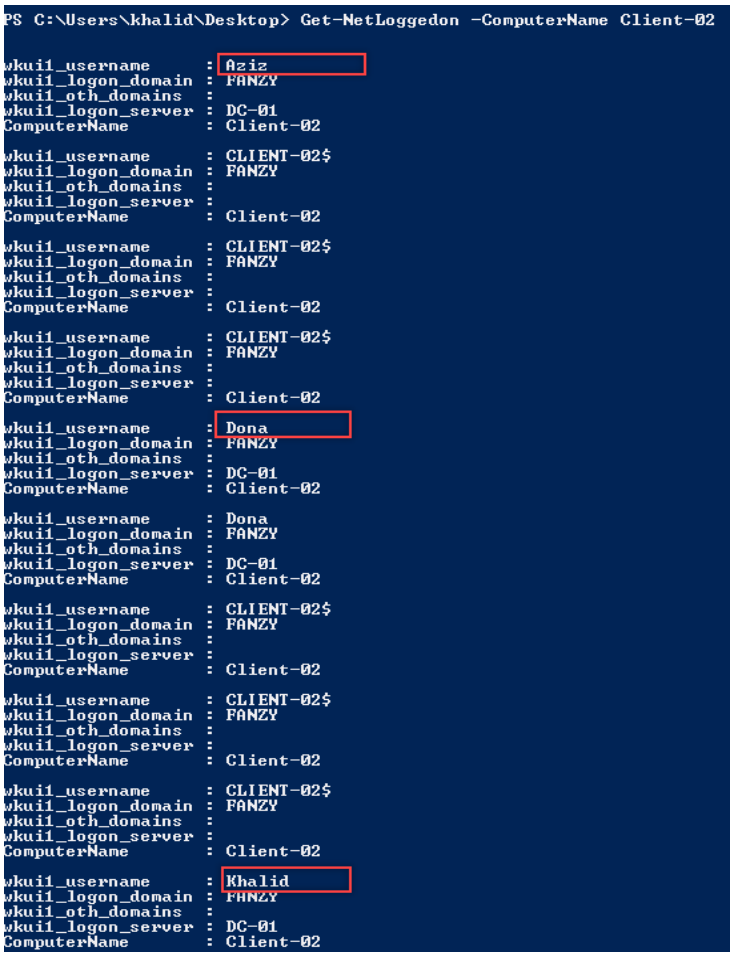
Get-NetLocalGroup –ComputerName Client-02

Use this command to get all the local administrators on a machine. (Note that it needs administrative rights).



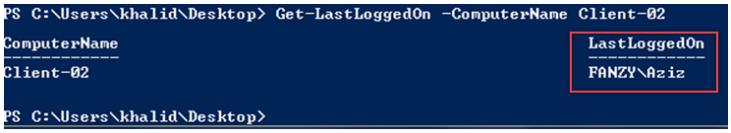
Get-NetLoggedon –ComputerName “Client-02”

Use this command to get actively logged users on a computer (Note that it needs administrative rights)



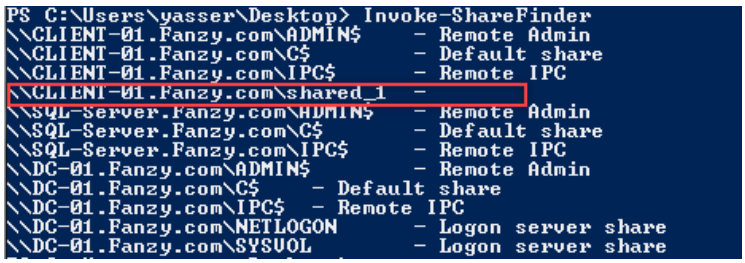
Get-LastLoggedOn –ComputerName Client-02

Use this command to get the last logged user on a computer (Note that it needs administrative rights)



Invoke-ShareFinder

Use this command to find shares on the hosts in the current domain.



GROUP POLICY (GPO) ENUMERATION

In an Active Directory environment, Group Policy is an easy way to configure computer and user settings on computers that are part of the domain. Group Policy allows you to centralize the management of computers on your network without having to physically go to and configure each computer individually.

So let's going to enumerate the GPO on the domain environment.

Get-NetGPO -ComputerName client-02.fanzy.com

Use this command to get a list of the GPO in the computer (**Client-02**).

```
PS C:\Users\khalid\Desktop> Get-NetGPO -ComputerName client-02.fanzy.com
gpcmachineextensionnames : [{35378EAC-683F-11D2-A89A-00C04FBBCFA2}{B05566AC-FE9C-4368-BE01-7A4CBB6CBA11}]
gpcfunctionalityversion  : 2
instancetype             : 4
whenchanged              : 3/18/2019 10:38:21 AM
name                     : {53335003-0171-467F-97BA-5C59B60DC93C}
gpcfilesyspath           : \\Fanzy.com\SysVol\Fanzy.com\Policies\{53335003-0171-467F-97BA-5C59B60DC93C}
distinguishedname        : CN={53335003-0171-467F-97BA-5C59B60DC93C},CN=Policies,CN=System,DC=Fanzy,DC=com
showinadvancedviewonly   : True
usncreated               : 119217
dscorepropagationdata    : 1/1/1601 12:00:00 AM
versionnumber            : 12
cn                       : {53335003-0171-467F-97BA-5C59B60DC93C}
objectguid               : b8ab5121-cc1d-4138-b6b8-ab974c4d9c0b
displayname              : Firewall OFF
whencreated              : 3/18/2019 10:37:48 AM
objectcategory           : CN=Group-Policy-Container,CN=Schema,CN=Configuration,DC=Fanzy,DC=com
adspath                  : LDAP://CN={53335003-0171-467F-97BA-5C59B60DC93C},CN=Policies,CN=System,DC=Fanzy,DC=com
usnchanged               : 119235
flags                    : 0
objectclass              : {top, container, groupPolicyContainer}
ComputerName             : client-02.fanzy.com
```

We can see that there is a group policy name (**Firewall OFF**) which it's clearly that it turns off the firewall on all the computers on the current domain.

Find-GPOComputerAdmin –Computername client-02.fanzy.com

Use this command to find users who have local admin rights over the machine **Client-02** through GPO.

```
PS C:\Users\khalid\Desktop> Find-GPOComputerAdmin -Computername client-02.fanzy.com

ComputerName    : client-02.fanzy.com
ObjectName      : IT_Admins                                                    Group Name
ObjectDN        : CN=IT_Admins,CN=Users,DC=Fanzy,DC=com
ObjectSID       : S-1-5-21-3156372763-3995679764-3492621305-1620
IsGroup         : True
GPODisplayName  : Local Administrators                                         GPO Name
GPOGuid         : {6A71FCCA-EF5A-43A2-AA37-A8BFABE5837A}
GPOPath         : \\Fanzy.com\SysVol\Fanzy.com\Policies\{6A71FCCA-EF5A-43A2-AA37-A8BFABE5837A}
GPOType         : RestrictedGroups
```

Find-GPOLocation -UserName Aziz

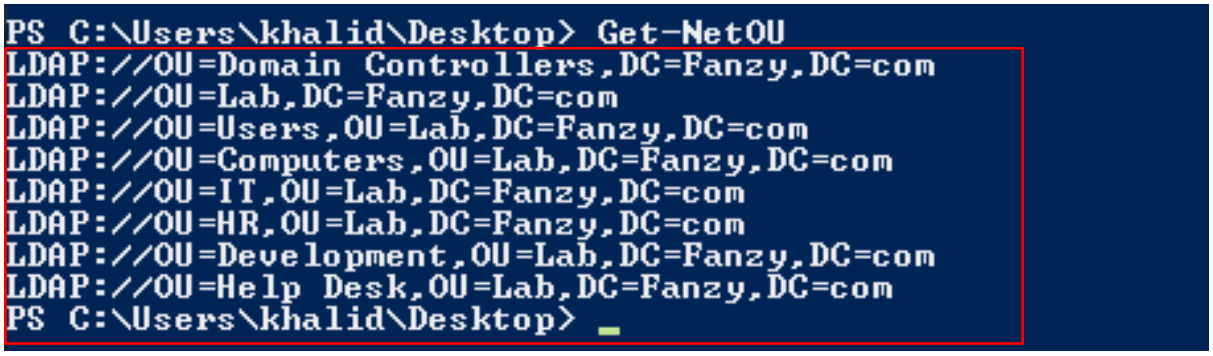
Use this command to find all computers that "Aziz" has local administrator rights in the current domain through the applied GPO.

```
PS C:\Users\khalid\Desktop> Find-GPOLocation -UserName Aziz

ObjectName      : Aziz
ObjectDN        : CN=Aziz,OU=IT,OU=Lab,DC=Fanzy,DC=com
ObjectSID       : S-1-5-21-3156372763-3995679764-3492621305-1619
Domain          :
IsGroup         : False
GPODisplayName  : Local Administrators
GPOGuid         : {6A71FCCA-EF5A-43A2-AA37-A8BFABE5837A}
GPOPath         : \\Fanzy.com\SysVol\Fanzy.com\Policies\{6A71FCCA-EF5A-43A2-AA37-A8BFABE5837A}
GPOType         : RestrictedGroups
ContainerName   : OU=Lab,DC=Fanzy,DC=com
ComputerName    : {Client-02.Fanzy.com, CLIENT-01.Fanzy.com, CDC-01.Fanzy.com, SQL-Server.Fanzy.com}
```

Get-NetOU

Use this command to get all the OUs (Organization Units) in the current domain.



DOMAIN TRUSTS ENUMERATION

In an AD environment, trust is a relationship between two domains or forests which allows users of one domain or forest to access resources in the other domain or forest. For example, a user in domain A can request or access resources in domain B (like query the computers in the domain B).

Trusts Direction:

- **Two-way trust (Bi-directional):** Users from Domain A can access resources in Domain B and vice versa.
- **One-way trust (Unidirectional):** Users in the trusted domain can access resources in the trusting domain but the reverse is not true

Trusts Transitivity:

- **Parent-child trust:** It is created automatically between the new domain and the domain that precedes it in the namespace hierarchy, whenever a new domain is added in a tree. For example, usa.fanzy.com is a child of fanzy.com). This trust is always two-way transitive.
- **Tree-root trust:** It is created automatically between whenever a new domain tree is added to a forest root. This trust is always two-way transitive.

External Trusts: Between two domains in different forests when forests do not have a trust relationship. It can be one-way or two-way and is nontransitive.

As read teamers, it's important to enumerate the domain trusts in order to expand the attack surface.

Get-NetDomainTrust

Use this command to get a list of all domain trusts for the current domain to map the domain trust.

```
PS C:\Users\khalid\Desktop> Get-NetDomainTrust

SourceName                     TargetName                     TrustType                      TrustDirection
----------                     ----------                     ---------                      --------------
Fanzy.com                      USA.Fanzy.com                  ParentChild                    Bidirectional
```

Get-NetForest

Use this command to get details about the current forest.

```
PS C:\Users\khalid\Desktop> Get-NetForest


RootDomainSid          : S-1-5-21-3156372763-3995679764-3492621305
Name                   : Fanzy.com
Sites                  : {France, USA}
Domains                : {Fanzy.com, USA.Fanzy.com}
GlobalCatalogs         : {DC-01.Fanzy.com, CDC-01.USA.Fanzy.com}
ApplicationPartitions  : {DC=ForestDnsZones,DC=Fanzy,DC=com, DC=DomainDnsZones,DC=Fanzy,DC=com, DC=DomainDnsZones,DC=USA
                         ,DC=Fanzy,DC=com}
ForestMode             : 6
RootDomain             : Fanzy.com
Schema                 : CN=Schema,CN=Configuration,DC=Fanzy,DC=com
SchemaRoleOwner        : DC-01.Fanzy.com
NamingRoleOwner        : DC-01.Fanzy.com
```

Get-NetForest -Forest dampy.com

Use this command to get details about another forest.

```
PS C:\Users\khalid\Desktop> Get-NetForest -Forest dampy.com


RootDomainSid          : S-1-5-21-55717269-2011424420-857468137
Name                   : Dampy.com
Sites                  : {Default-First-Site-Name}
Domains                : {Dampy.com}
GlobalCatalogs         : {DC-02.Dampy.com}
ApplicationPartitions  : {DC=ForestDnsZones,DC=Dampy,DC=com, DC=DomainDnsZones,DC=Dampy,DC=com}
ForestMode             : 6
RootDomain             : Dampy.com
Schema                 : CN=Schema,CN=Configuration,DC=Dampy,DC=com
SchemaRoleOwner        : DC-02.Dampy.com
NamingRoleOwner        : DC-02.Dampy.com
```

Get-NetForestDomain

Use this command to get all the domains in the current forest.

```
PS C:\Users\khalid\Desktop> Get-NetForestDomain


Forest                 : Fanzy.com
DomainControllers      : {DC-01.Fanzy.com}
Children               : {USA.Fanzy.com}
DomainMode             :
Parent                 :
PdcRoleOwner           : DC-01.Fanzy.com
RidRoleOwner           : DC-01.Fanzy.com
InfrastructureRoleOwner : DC-01.Fanzy.com
Name                   : Fanzy.com

Forest                 :
DomainControllers      :
Children               :
DomainMode             :
Parent                 :
PdcRoleOwner           :
RidRoleOwner           :
InfrastructureRoleOwner :
Name                   : USA.Fanzy.com
```

Get-NetForestCatalog

Use this command to get all global catalogs for the current forest.

```
PS C:\Users\khalid\Desktop> Get-NetForestCatalog

Forest                      : Fanzy.com
CurrentTime                 : 3/24/2019 4:32:58 PM
HighestCommittedUsn         : 121641
OSVersion                   : Windows Server 2012 R2 Standard Evaluation
Roles                       : {SchemaRole, NamingRole, PdcRole, RidRole...}
Domain                      : Fanzy.com
IPAddress                   : 10.10.10.10
SiteName                    : France
SyncFromAllServersCallback  :
InboundConnections          : {88f85e80-92bb-40f1-8c32-cdbffe0cda92}
OutboundConnections         : {}
Name                        : DC-01.Fanzy.com
Partitions                  : {DC=Fanzy,DC=com, CN=Configuration,DC=Fanzy,DC=com, CN=Schema,CN=Configuration,DC=Fanzy,DC
                              =com, DC=DomainDnsZones,DC=Fanzy,DC=com...}

Forest                      :
CurrentTime                 :
HighestCommittedUsn         :
OSVersion                   :
Roles                       :
Domain                      :
IPAddress                   :
SiteName                    :
SyncFromAllServersCallback  :
InboundConnections          :
OutboundConnections         :
Name                        : CDC-01.USA.Fanzy.com
Partitions                  :
```

Get-NetForestTrust

Use this command to map the trusts of a forest.

```
PS C:\Users\khalid\Desktop> Get-NetForestTrust


TopLevelNames               : {Dampy.com}
ExcludedTopLevelNames       : {}
TrustedDomainInformation    : {Dampy.com}
SourceName                  : Fanzy.com
TargetName                  : Dampy.com
TrustType                   : Forest
TrustDirection              : Bidirectional

TopLevelNames               : {Lenda.com}
ExcludedTopLevelNames       : {}
TrustedDomainInformation    : {Lenda.com}
SourceName                  : Fanzy.com
TargetName                  : Lenda.com
TrustType                   : Forest
TrustDirection              : Bidirectional
```

From the above result, we can see that the domain (**Fanzy.com**) has a two-way trust (Bidirectional) with the domain (**Dampy.com**) as well as with the domain (**Lenda.com**).

So, from the domain trust we can for example query the computers name of another domain (**Dampy.com**) as shown below.

```
PS C:\Windows\system32> Get-NetComputer -Domain Dampy.com
DC-02.Dampy.com
Client-003.Dampy.com
CLIENT-04.Dampy.com
```

User Hunting

When we got a foothold on a machine in the AD environment, it's important to look for the privileged users such as the Local Administrators or the Domain Admins. In this section, we are going to hunt those users in the AD environment in order to gain Domain Admin rights from a domain user (normal user).

Find-LocalAdminAccess

Use this command to find all machines on the current domain where the current user has local admin access.

```
PS C:\Users\khalid\Desktop> Find-LocalAdminAccess
SQL-Server.Fanzy.com
Client-02.Fanzy.com
```

Invoke-EnumerateLocalAdmin

Use this command to find local admins on all machines of the domain (needs administrator privs on non-dc machines).

```
ComputerName : CLIENT-01.Fanzy.com
AccountName  : Fanzy.com/Sarah
IsDomain     : True
IsGroup      : False
SID          : S-1-5-21-3156372763-3995679764-3492621305-1116
Description  :
Disabled     :
LastLogin    : 3/14/2019 11:18:21 AM
PwdLastSet   :
PwdExpired   :
UserFlags    :

ComputerName : CLIENT-01.Fanzy.com
AccountName  : Fanzy.com/IT_Admins
IsDomain     : True
IsGroup      : True
SID          : S-1-5-21-3156372763-3995679764-3492621305-1620
Description  :
Disabled     :
LastLogin    :
PwdLastSet   :
PwdExpired   :
UserFlags    :
```

Invoke-UserHunter

Use this command to find computers where a domain has logged in.

```
PS C:\Windows\system32> Invoke-UserHunter


UserDomain       : FANZY
UserName         : Dona
ComputerName     : CLIENT-01.Fanzy.com
IPAddress        : 10.10.10.20
```

From the above output, we can see that the Domain Admin (**Dona**) in logged in the machine (**Client-01**) with its IP (**10.10.10.20**).

Invoke-UserHunter -UserName “Aziz"

Use this command to find computers where a specific user has sessions.

```
PS C:\Users\khalid\Desktop> Invoke-UserHunter -UserName "Aziz"

UserDomain      : FANZY
UserName        : Aziz
ComputerName    : Client-02.Fanzy.com
IPAddress       : 10.10.10.30
```

Invoke-UserHunter -CheckAccess

Use this command to find computers where a domain admin is logged in and current user has access.

```
UserDomain      : FANZY
UserName        : Dona
ComputerName    : Client-02.Fanzy.com
IPAddress       : 10.10.10.30
SessionFrom     :
SessionFromName :
LocalAdmin      : True

UserDomain      : FANZY
UserName        : Dona
ComputerName    : CLIENT-01.Fanzy.com
IPAddress       : 10.10.10.20
SessionFrom     :
SessionFromName :
LocalAdmin      : True
```

ACCESS CONTROL LISTS (ACL) ENUMERATION

An Access Control List (ACL) is a list of access control entries (ACE). Each ACE in an ACL identifies a trustee and specifies the access rights allowed, denied, or audited for that trustee. The security descriptor for a securable object can contain two types of ACLs: a DACL and a SACL.

DACL: Defines the permissions trustees (a user or group) have on an object.

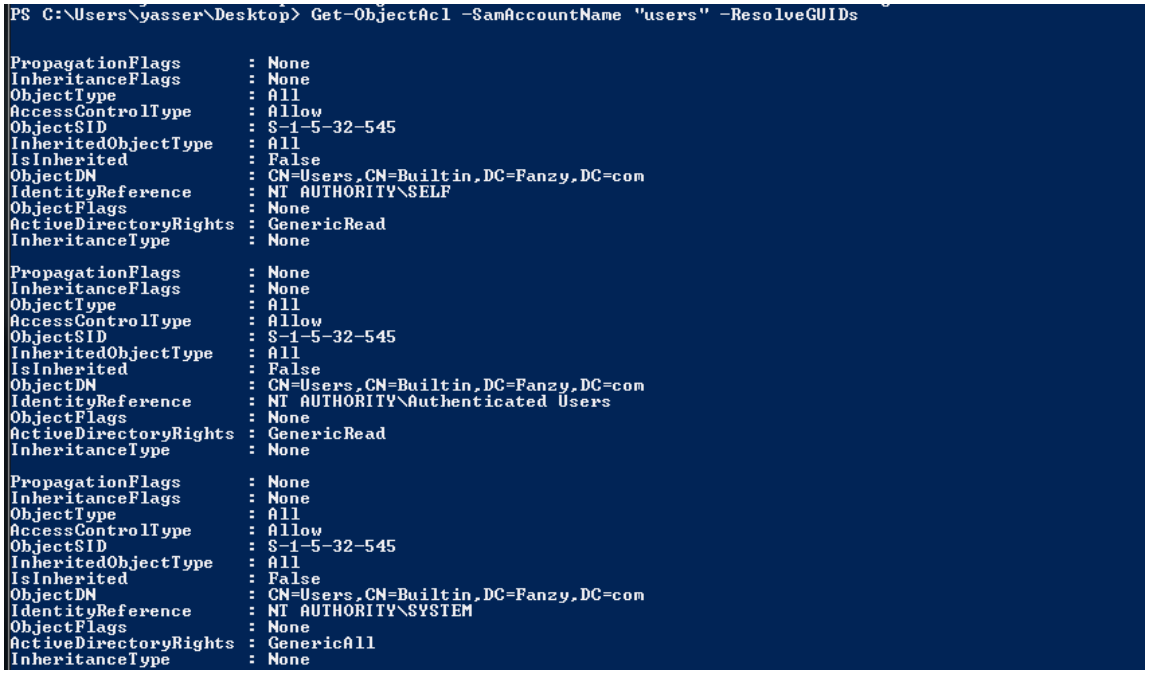
SACL: Logs success and failure audit messages when an object is accessed.

In other words, the ACL is like asking: who has permission and what can be done on an object?

Most of the system administrators are wrongly configuring the ACL (such as granting a normal user to important permissions). So as attackers, we are interested in enumerating the ACL in order to find interesting ACLs!

Get-ObjectAcl -SamAccountName "users" -ResolveGUIDs

Use this command to enumerate the ACLs for the users group.



Get-NetGPO | %{Get-ObjectAcl -ResolveGUIDs -Name $_.Name}

Use this command to see if there is any user has a modification rights to a GPO.

```
PS C:\Users\yasser\Desktop> Get-NetGPO | %{Get-ObjectAcl -ResolveGUIDs -Name $_.Name}


PropagationFlags        : InheritOnly
InheritanceFlags        : ContainerInherit
ObjectType              : All
AccessControlType       : Allow
ObjectSID               :
InheritedObjectType     : All
IsInherited             : False
ObjectDN                : CN={31B2F340-016D-11D2-945F-00C04FB984F9},CN=Policies,CN=System,DC=Fanzy,DC=com
IdentityReference       : CREATOR OWNER
ObjectFlags             : None
ActiveDirectoryRights   : CreateChild, DeleteChild, Self, WriteProperty, DeleteTree, Delete, GenericRead, WriteDac
                          eOwner
InheritanceType         : Descendents

PropagationFlags        : None
InheritanceFlags        : ContainerInherit
ObjectType              : All
AccessControlType       : Allow
ObjectSID               :
InheritedObjectType     : All
IsInherited             : False
ObjectDN                : CN={31B2F340-016D-11D2-945F-00C04FB984F9},CN=Policies,CN=System,DC=Fanzy,DC=com
IdentityReference       : NT AUTHORITY\ENTERPRISE DOMAIN CONTROLLERS
ObjectFlags             : None
ActiveDirectoryRights   : GenericRead
InheritanceType         : All

PropagationFlags        : None
InheritanceFlags        : ContainerInherit
ObjectType              : All
AccessControlType       : Allow
ObjectSID               :
InheritedObjectType     : All
IsInherited             : False
ObjectDN                : CN={31B2F340-016D-11D2-945F-00C04FB984F9},CN=Policies,CN=System,DC=Fanzy,DC=com
```

Get-ObjectAcl -SamAccountName labuser -ResolveGUIDs -RightsFilter "ResetPassword"

Use this command to check if the user “Sarah” has the permission (Reset Password).

```
PS C:\Users\yasser\Desktop> Invoke-ACLScanner -ResolveGUIDs | ?{$_.IdentityReference -match "Sarah"}


PropagationFlags        : InheritOnly
InheritanceFlags        : ContainerInherit
ObjectType              : Pwd-Last-Set
AccessControlType       : Allow
ObjectSID               :
InheritedObjectType     : User
IsInherited             : False
ObjectDN                : OU=Lab,DC=Fanzy,DC=com
IdentityReference       : FANZY\Sarah
ObjectFlags             : ObjectAceTypePresent, InheritedObjectAceTypePresent
ActiveDirectoryRights   : ReadProperty, WriteProperty
InheritanceType         : Descendents
IdentitySID             : S-1-5-21-3156372763-3995679764-3492621305-1116

PropagationFlags        : InheritOnly
InheritanceFlags        : ContainerInherit
ObjectType              : User-Force-Change-Password
AccessControlType       : Allow
ObjectSID               :
InheritedObjectType     : User
IsInherited             : False
ObjectDN                : OU=Lab,DC=Fanzy,DC=com
IdentityReference       : FANZY\Sarah
ObjectFlags             : ObjectAceTypePresent, InheritedObjectAceTypePresent
ActiveDirectoryRights   : ExtendedRight
InheritanceType         : Descendents
IdentitySID             : S-1-5-21-3156372763-3995679764-3492621305-1116
```

We can see that the user “Sarah” has this interesting permission (Reset Password) which she can reset the password of any user in the domain even though that “Sarah” is a normal user!

CONCLUSION

As red teamers, it's extremely important to enumerate the Active Directory environment whenever we have a foothold on a machine in the AD. Without a proper enumeration, we may don't achieve our goals as enumerating the AD will help us to gain Domain Admin rights and reach the DC. We can enumerate the domain and the trusts of the domains and the forests, the group policy, the access control list (ACL) and hunting the users in order to reach our goals on an engagement.

REFERENCES

- https://www.harmj0y.net/blog/tag/powerview/
- https://medium.com/tech-jobs-academy/trust-relationships-within-active-directory-directory-services-9f18b3a9e7da
- https://www.labofapenetrationtester.com/p/active-directory-attacks-for-red-and.html