

Intro

As we saw on the introduction of this section we can enumerate the NTDS Database that the domain controller has, even without an elevated user privs on the domain, through the **LDAP protocol** we just need to use the credentials of a legitim domain user to enumerate remotely all that information.

We can use a lot of tools to enumerate NTDS Databases and in this section we are going to see how to use some tools to do that.

NetExec (formerly known as CrackMapExec) is a post-exploitation and enumeration tool that allows you to interact with common Windows networking services such as SMB, WinRM, RDP, and of course, **LDAP**.

When connected via **LDAP**, NetExec allows us to enumerate key information from the Active Directory domain (**NTDS**) in the **Domain Controller** database without using SMB or executing remote commands. Of course we will need credentials of a legitim domain user.

Among the things we can enumerate, specifying the DC IP address, are:

netexec ldap 10.10.10.5 -u Username -p Password --users

```
netexec ldap 192.168.20.5 -u employer1 -p Password01 --users
SMB 192.168.20.5 445 DC01 [*] Windows Server 2022 Build 20348 x64 (name:DC01) (domain:corp.local) (signing:True) (SMBv1:False)
LDAP 192.168.20.5 389 DC01 [*] corp.local\employer1:Password01
LDAP 192.168.20.5 389 DC01 [*] Enumerated 156 domain users: corp.local
LDAP 192.168.20.5 389 DC01 -Username- -Last PW Set- -BadPW- -Description-
LDAP 192.168.20.5 389 DC01 Administrator 2024-12-05 23:51:00 0 Built-in account for administering the computer/domain
LDAP 192.168.20.5 389 DC01 Guest <never> 0 Built-in account for guest access to the computer/domain
LDAP 192.168.20.5 389 DC01 krbtgt 2024-12-06 00:28:46 0 Key Distribution Center Service Account
LDAP 192.168.20.5 389 DC01 employer1 2024-12-07 01:46:55 0
LDAP 192.168.20.5 389 DC01 employer2 2024-12-07 01:49:57 0
LDAP 192.168.20.5 389 DC01 faustine.lars 2024-12-28 02:15:37 0
LDAP 192.168.20.5 389 DC01 ada.jammie 2024-12-28 02:15:38 0
LDAP 192.168.20.5 389 DC01 aryn.maurita 2024-12-28 02:15:38 0
LDAP 192.168.20.5 389 DC01 king.pammy 2024-12-28 02:15:38 0
LDAP 192.168.20.5 389 DC01 leda.raye 2024-12-28 02:15:38 0
LDAP 192.168.20.5 389 DC01 kendra.dorine <never> 2 New User - DefaultPassword
```


netexec ldap 10.10.10.5 -u Username -p Password --groups

```
netexec ldap 192.168.20.5 -u employer1 -p Password01 --groups
SMB 192.168.20.5 445 DC01 [*] Windows Server 2022 Build 20348 x64 (name:DC01) (domain:corp.local) (signing:True) (SMBv1:False)
LDAP 192.168.20.5 389 DC01 [*] corp.local\employer1:Password01
LDAP 192.168.20.5 389 DC01 Administrators
LDAP 192.168.20.5 389 DC01 Users
LDAP 192.168.20.5 389 DC01 Guests
LDAP 192.168.20.5 389 DC01 Print Operators
LDAP 192.168.20.5 389 DC01 Backup Operators
LDAP 192.168.20.5 389 DC01 Replicator
LDAP 192.168.20.5 389 DC01 Remote Desktop Users
LDAP 192.168.20.5 389 DC01 Network Configuration Operators
LDAP 192.168.20.5 389 DC01 Performance Monitor Users
LDAP 192.168.20.5 389 DC01 Performance Log Users
```

Something interesting is that we can enumerate that information using **SMB** protocol, in some cases with better results. For example if we enumerate **groups**, specifying the **SMB** protocol, not only do they show us the groups that are in the domain, but also how **many members belong to them**.

```

netexec smb 192.168.20.5 -u employer1 -p Password01 --groups
[*] Windows Server 2022 Build 20348 x64 (name:DC01) (domain:corp.local) (signing:True) (SMBv1:False)
[+] corp.local\employer1:Password01
[+] Enumerated domain group(s)
accounting membercount: 4
sales membercount: 3
marketing membercount: 6
Project management membercount: 7
Senior management membercount: 4
Executives membercount: 6

```

→ Groups Members

We can also list the domain's password policy with the **--pass-pol** flag.

```

netexec smb 192.168.20.5 -u employer1 -p Password01 --pass-pol
[*] Windows Server 2022 Build 20348 x64 (name:DC01) (domain:corp.local) (signing:True) (SMBv1:False)
[+] corp.local\employer1:Password01
[+] Dumping password info for domain: CORP
Minimum password length: 4
Password history length: 24
Maximum password age: 41 days 23 hours 53 minutes
Password Complexity Flags: 000000
Domain Refuse Password Change: 0
Domain Password Store Cleartext: 0
Domain Password Lockout Admins: 0
Domain Password No Clear Change: 0
Domain Password No Anon Change: 0
Domain Password Complex: 0
Minimum password age: 1 day 4 minutes
Reset Account Lockout Counter: 1 minute
Locked Account Duration: 1 minute
Account Lockout Threshold: None
Forced Log off Time: Not Set

```

With this information we can design more realistic brute force attacks that do not endanger the productivity of the environment we are auditing.

Pywerview

Intro

Pywerview is a tool writing in python for enumeration using LDAP protocol. This tool is very similar to Power View, in that it will allow us to enumerate the NTDS database with one big difference : that we can use **Pywerview** to enumerate from our attacker machine.

To use **Pywerview** we will need credentials from a domain user.

This tool comes pre install on Kali Linux 🐧 and Parrot 🦋 systems and it will be use it by terminal command line.

Pywerview on action

The command to use pywerview will consist of calling the tool “**pywerview**” then the command that we will use to enumerate, for example “**get-netuser**”, next we have to use the **username** with the parameter “**-u user**”. The next step is put the **DC** IP address “**--dc-ip 192.0.0.5**” and finally the password with “**-p password:D**”

There some examples of commands that we can use to enumerate NTDS database

pywerview get-netuser -u Username --dc-ip 192.0.0.5 -p Password

This command will dump information about all domain users

```
pywerview get-netuser -u employer1 --dc-ip 192.168.20.5 -p Password01
objectclass: top, person, organizationalPerson, user
cn: employer2
givenname: employer2
distinguishedname: CN=employer2,CN=Users,DC=corp,DC=local
instancetype: 4
whencreated: 2024-12-07 01:47:56+00:00
whenchanged: 2024-12-07 01:52:51+00:00
displayname: employer2
usncreated: 16413
usnchanged: 16426
name: employer2
objectguid: {f8d058a1-cc90-4fa4-97fc-83e12e228201}
useraccountcontrol: NORMAL_ACCOUNT, DONT_EXPIRE_PASSWORD
usnpwdcount: 0
codepage: 0
countrycode: 0
badpasswordtime: 2024-12-10 22:05:00.009581+00:00
lastlogoff: 1601-01-01 00:00:00+00:00
lastlogon: 2024-12-12 00:28:00.590655+00:00
pwdlastset: 2024-12-07 01:47:57.282837+00:00
primarygroupid: 513
accountexpires: 9999-12-31 23:59:59.999999+00:00
logoncount: 8
samaccountname: employer2
samaccounttype: 805306368
objectcategory: CN=Person,CN=Schema,CN=Configuration,DC=corp,DC=local
discoropagationdata: 2024-12-07 01:47:57+00:00, 1601-01-01 00:00:00+00:00
lastlogontimestamp: 2024-12-07 01:52:51.754313+00:00
objectclass: top, person, organizationalPerson, user
cn: employer1
givenname: employer1
```

pywerview get-netdomaincontroller -u Username --dc-ip 192.0.0.5 -p Password

This command allows us gather information about the domain controller

```
pywerview get-netdomaincontroller -u employer1 --dc-ip 192.168.20.5 -p Password01
objectclass: top, person, organizationalPerson, user, computer
cn: DC01
distinguishedname: CN=DC01,OU=Domain Controllers,DC=corp,DC=local
instancetype: 4
whencreated: 2024-12-06 00:28:45+00:00
whenchanged: 2024-12-17 21:39:57+00:00
usncreated: 12293
usnchanged: 28665
name: DC01
objectguid: {3f1f2fbd-c4be-4e5d-ac41-a12536e13af7}
useraccountcontrol: SERVER_TRUST_ACCOUNT, TRUSTED_FOR_DELEGATION
usnpwdcount: 0
codepage: 0
countrycode: 0
badpasswordtime: 1601-01-01 00:00:00+00:00
lastlogoff: 1601-01-01 00:00:00+00:00
lastlogon: 2024-12-23 22:17:23.005436+00:00
localpolicysflags: 0
pwdlastset: 2024-12-06 00:29:14.468855+00:00
primarygroupid: 516
objectsids: 5-1-5-21-3525594078-1931719227-2786532380-1000
accountexpires: 9999-12-31 23:59:59.999999+00:00
logoncount: 49
samaccountname: DC01$
samaccounttype: 805306369
operatingsystem: Windows Server 2022 Standard Evaluation
operatingsystemversion: 10.0 (20348)
serverreferencebl: CN=DC01,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=corp,DC=local
dshostname: DC01.corp.local
ridsetreferences: CN=RID Set,CN=DC01,OU=Domain Controllers,DC=corp,DC=local
serviceprincipalname: Dfsr-1f59a27c-8b97-4a87-93d4-03b6c558b4/DC01.corp.local, ldap/DC01.corp.local/forestDnsZones.corp.local, ldap/DC01.corp.local/DomainDnsZones.corp.local, Dfsr/DC01.corp.local, DC/DC01.corp.local/corp.local, RestrictedKrmHost/DC01.corp.local,
```

There are just two examples of what we can do with **pywerview**. Of course, it has a lot of functionality that we can use to enumerate a domain and that we can use according to what we want to enumerate in the domain

Conclusions

As we have seen, **pywerview** is a very interesting tool that we can use if we don't have access to a domain computer or if we just want to use an alternative tool.

If we want more info about all commands that we can use, we just have to use the the command **pywerview -h**

```
get-adoobject Takes a domain SID, samaccountname or name, and return the associated object
get-objectacl Takes a domain SID, samaccountname or name, and return the ACL of the associated object
get-netuser Queries information about a domain user
get-netgroup Get a list of all current domain groups, or a list of groups a domain user is member of
get-netcomputer Queries informations about domain computers
get-netdomaincontroller Get a list of domain controllers for the given domain
get-netfileserv Return a list of file servers, extracted from the domain users' homeDirectory, scriptPath, and profilePath fields
get-dfsshare Return a list of all fault tolerant distributed file systems for a given domain
get-netou Get a list of all current OUs in the domain
get-netsite Get a list of all current sites in the domain
get-netsubnet Get a list of all current subnets in the domain
get-netdomaintrust Returns a list of all the trusts of the specified domain
get-netgpo Get a list of all current GPOs in the domain
get-netgpo Get a list of all current GPOs in the domain
get-domainpolicy Returns the default domain or DC policy for the queried domain or DC
get-gpttmpl Helper to parse a Gpttmpl.inf policy file path into a custom object
get-netgroup Parse all GPOs in the domain that set "Restricted Group" or "Groups.xml"
find-gpocomputeradmin Takes a computer (or OU) and determine who has administrative access to it via GPO
find-gpolocation Return a list of members of a domain group
get-netgroupmember Return a list of members of a domain group
get-netsession Queries a host to return a list of active sessions on the host (you can use local credentials instead of domain credentials)
get-localdisks Queries a host to return a list of active disks on the host (you can use local credentials instead of domain credentials)
get-netdomain Queries a host for available domains
get-netshare Queries a host to return a list of available shares on the host (you can use local credentials instead of domain credentials)
```

Ldapsearch

Intro

Ldapsearch is a tool that allows us to interact with a database using the LDAP **protocol** and we use it to interact with the **NTDS Database** that DC has.

This tool comes pre installed on Kali Linux 🐧 and Parrot 🦋 systems and it will be used by terminal command line

Ldapsearch on action

Ok, let's see some commands that we can perform with **LDAP search**.

ldapsearch -x -H ldap://10.10.10.5 -D 'DOMAIN\Username' -w 'Password' -b "DC=domain,DC=local"

```
ldapsearch -x -H ldap://192.168.20.5 -D 'CORP\employer1' -w 'Password01' -b "DC=corp,DC=local"
# extended LDIF
#
# LDAPv3
# base <DC=corp,DC=local> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#
# corp.local
dn: DC=corp,DC=local
objectClass: top
objectClass: domain
objectClass: domainDNS
distinguishedName: DC=corp,DC=local
instanceType: 5
```

When we successfully perform this command we will see that we received an enormous output, because **ldapsearch** is dumping all the information that this NTDS database has, user, groups, gpos, computers... etc.

But if we just receive a specific information we just have to add the flag **CN=** on the **-b** part followed by that information that interests us. For example, if we are interested of dump all domain user will be something like:

-b "CN=UsersDC=domain,DC=local".

```
ldapsearch -x -H ldap://192.168.20.5 -D 'CORP\employer1' -w 'Password01' -b "CN=Users,DC=corp,DC=local"
# extended LDIF
#
# LDAPv3
# base <CN=Users,DC=corp,DC=local> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#
# Users, corp.local
dn: CN=Users,DC=corp,DC=local
objectClass: top
objectClass: container
cn: Users
description: Default container for upgraded user accounts
distinguishedName: CN=Users,DC=corp,DC=local
instanceType: 4
whenCreated: 20241206002752.0Z
whenChanged: 20241206002752.0Z
uSNCreated: 5660
```

And if we are interested on a particular user we can add that username adding
"(sAMAccountName=user.name)"

```
ldapsearch -x -H ldap://192.168.20.5 -D 'CORP\employer1' -w 'Password01' -b "DC=corp,DC=local" "(sAMAccountName=maridel.giulia)"
# extended LDIF
#
# LDAPv3
# base <DC=corp,DC=local> with scope subtree
# filter: (sAMAccountName=maridel.giulia)
# requesting: ALL
#
# Maridel Giulia, Users, corp.local
dn: CN=Maridel Giulia,CN=Users,DC=corp,DC=local
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: user
cn: Maridel Giulia
sn: Giulia
```

If we what enumerate de computes on the domain we can add **"CD=Computer"**:

```
ldapsearch -x -H ldap://192.168.20.5 -D 'CORP\employer1' -w 'Password01' -b "CN=Computers,DC=corp,DC=local"
# extended LDIF
#
# LDAPv3
# base <CN=Computers,DC=corp,DC=local> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#
# Computers, corp.local
dn: CN=Computers,DC=corp,DC=local
objectClass: top
objectClass: container
cn: Computers
description: Default container for ungraded computer accounts
distinguishedName: CN=Computers,DC=corp,DC=local
instanceType: 4
whenCreated: 20241206002752.0Z
whenChanged: 20241206002752.0Z
```

And if we what to enumerate a particular group and its members, we can do it adding the
flag **"CD=GroupName,CD=Users,"**

```
ldapsearch -x -H ldap://192.168.20.5 -D 'CORP\employer1' -w 'Password01' -b "CN=IT Admins,CN=Users,DC=corp,DC=local"
# extended LDIF
#
# LDAPv3
# base <CN=IT Admins,CN=Users,DC=corp,DC=local> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#
# IT Admins, Users, corp.local
dn: CN=IT Admins,CN=Users,DC=corp,DC=local
objectClass: top
objectClass: group
cn: IT Admins
member: CN=Leila Katherine,CN=Users,DC=corp,DC=local
member: CN=Kevin Kameko,CN=Users,DC=corp,DC=local
member: CN=Krisha Sarita,CN=Users,DC=corp,DC=local
member: CN=Estrella Letitia,CN=Users,DC=corp,DC=local
member: CN=Emmalee Layney,CN=Users,DC=corp,DC=local
member: CN=Daryl Wirella,CN=Users,DC=corp,DC=local
member: CN=Ileane Lynn,CN=Users,DC=corp,DC=local
member: CN=King Pammy,CN=Users,DC=corp,DC=local
distinguishedName: CN=IT Admins,CN=Users,DC=corp,DC=local
instanceType: 4
whenCreated: 20241206002752.0Z
whenChanged: 20241206002752.0Z
```

→ Group Membres

And we can more

Anonymous enumeration

ldapsearch -x -H ldap://10.10.10.5 -D '' -w '' -b "DC=domain,DC=local"

We can perform this command to see if the domain allows enumeration by an Anonymous user. An anonymous enumeration is unlikely to be carried out, but it never hurts to try.

Conclusion

There are some information tools that we can use to enumerate a domain via **NTDS database** through **LDAP** and other protocols from our attacker Machine.