Administrate your local accounts with LAPS

azureblog.pl/2020/02/05/laps-deployment

Robert Przybylski

Hi,

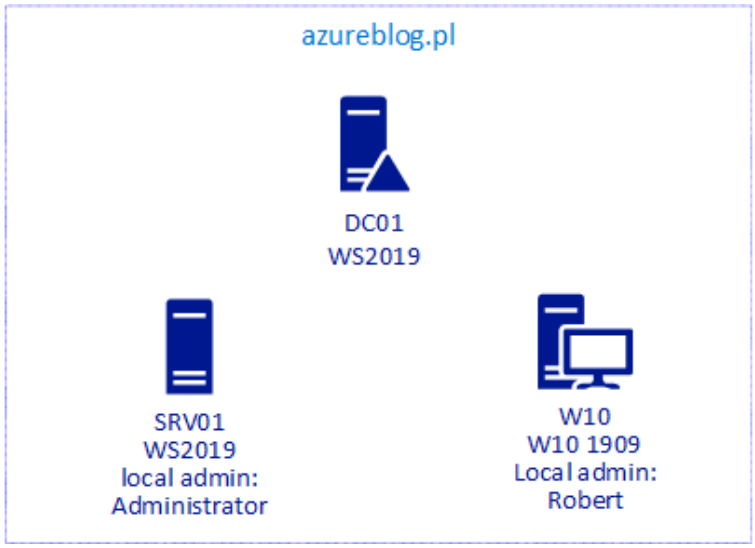
Have you ever been asked how to manage local admin passwords from one place without any additional costs?

If the answer is "**yes**" you are in the right place.

Let me show you the idea of Local Administrator Password Solution (LAPS).
This is a small application (available to download from *https://aka.ms/LAPS*) that provides management of local account passwords of domain-joined computers.

I guess your next question could be "**OK, but what about security?**"
When you configure LAPS you are able to configure ACL on the specific OU's to allow a specific group of users to read and change passwords.
So let's do a little brief about the environment before we will start the implementation part.

Below you can find LAB configuration



LAb HLD

As you can see under *azureblog.pl* domain I have 3 entities:

- DC01 – Domain controller for azureblog.pl domain (Windows Server 2019)
- SRV01 – Member server in *azureblog.pl* domain (Windows Server 2019)
- W10 – End-user workstation in *azureblog.pl* domain (Windows 10 1909)

For LAPS purpose I have also created the following groups and accounts:

- Groups
  - t0-admins (Security Group) – a member of “Domain Admins” and “Schema Admins” groups
  - t1-admins (Security Group) – members of this group are able to connect to SRV01
  - t2-admins (Security Group)
- Users
  - Domop – a member of t0-admins
  - t1-admin – a member of t1-admins
  - t2-admin – a member of t2-admins

As every software installation, first of all, we need to download a zip file (LAPS is available under https://*aka.ms/LAPS*).
When you will have a zip file, copy it onto one of your domain controllers and yes, I know it is not the best approach but it’s only for demonstration purposes.

When the files will be stored on the domain controller you can extract them and copy to the following path :
\\DOMAIN_NAME\SysVol\DOMAIN_NAME\Scripts\LAPS
(default local path to SysVol on a domain controller is C:\Windows\SysVol)

Another possible question ” **Why we are using SysVol?”**
We will use it to provide installation files later during GPO configuration.

Let’s start configuring something
Our first step will be LAPS installation. In order to do that we need to run the following PowerShell code:

```
$domain = $env:USERDNSDOMAIN
$lapsPath = "\\$Domain\SysVol\$Domain\Scripts\laps\LAPS.x64.msi"
$expression = "C:\Windows\System32\msiexec.exe /i $LapsPath
ADDLOCAL=CSE,Management,Management.UI,Management.PS,Management.ADMX /quiet"
Invoke-Expression $expression
```

But wait. what are the options that we are going to install:

- CSE – Client Side Extension – simply the application and related dll files,
- Management – management tools including UI, PS, and AMDX,
- Management.UI – “fat client” with UI,
- Management.PS – PowerShell module,
- Management.ADMX – GPO editor templates

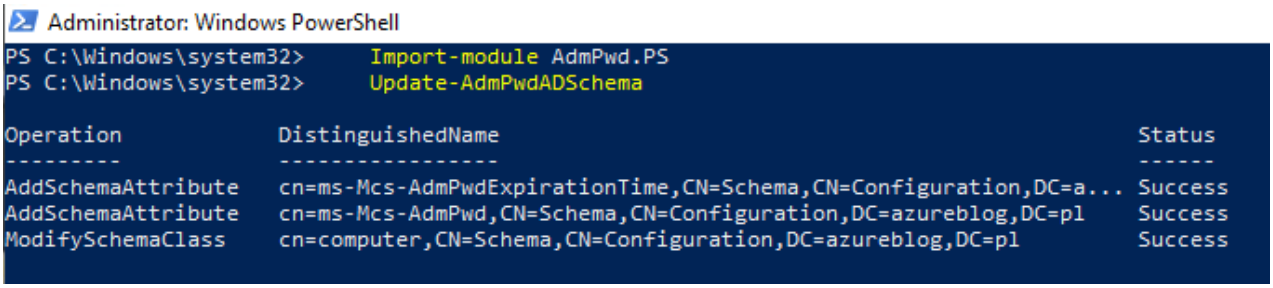
Below you can see the code execution in Powershell

LAPS installation

When we have LAPS installed we can start the next step which is AD schema update. In order to do this, you need please remember to use the account that is a member of “Schema Admins” group in your domain, and run the following code:

```
Import-module AdmPwd.PS
Update-AdmPwdADSchema
```

After successful code execution, you will have 2 new attributes in your Active Directory schema: **ms-Mcs-AdmPwd** (attribute where the password will be stored) and **ms-Mcs-AdmPwdExpirationTime** (attribute where password expiration date will be stored) and you should receive similar output



LAPS schema update

So half of the configuration done 🙂
Let's configure ACL to make it more secure
In order to allow assign computer self permissions, we need to run the following code

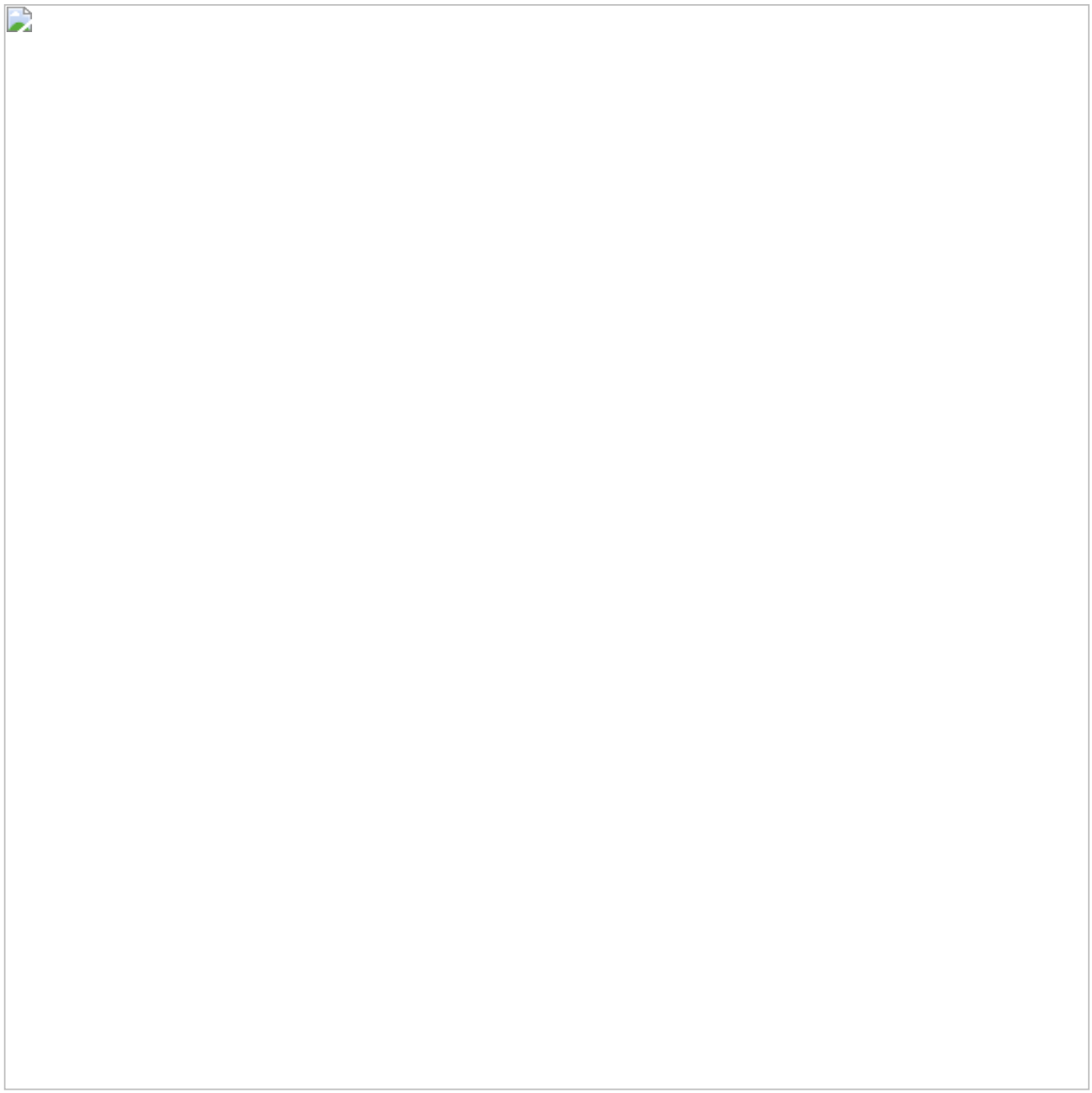
```
$dsnAME = (Get-ADDomain).DistinguishedName
Import-module AdmPwd.PS
Set-AdmPwdComputerSelfPermission -Identity "OU=PAW,OU=Tier0,OU=Admin,$dsname"
Set-AdmPwdComputerSelfPermission -Identity
"OU=Servers,OU=Tier0,OU=Admin,$dsname"
Set-AdmPwdComputerSelfPermission -Identity "OU=PAW,OU=Tier1,OU=Admin,$dsname"
Set-AdmPwdComputerSelfPermission -Identity
"OU=Servers,OU=Tier1,OU=Admin,$dsname"
Set-AdmPwdComputerSelfPermission -Identity
"OU=Computers,OU=AzureBlog,$dsname"
Set-AdmPwdComputerSelfPermission -Identity "OU=Quarantine,$dsname"
```

LAPS Computers configuration

The previous code allows computer objects from provided OU's to update ms-Mcs-AdmPwd and ms-Mcs-AdmPwdExpirationTime attributes in Active Directory.

The next step is to Allow users to read passwords from ms-Mcs-AdmPwd and ms-Mcs-AdmPwdExpirationTime attributes.

```
$dsnAME = (Get-ADDomain).DistinguishedName
Import-module AdmPwd.PS
Set-AdmPwdReadPasswordPermission -Identity "OU=PAW,OU=Tier0,OU=Admin,$dsname"
-AllowedPrincipals "t0-admins"
Set-AdmPwdReadPasswordPermission -Identity
"OU=Servers,OU=Tier0,OU=Admin,$dsname" -AllowedPrincipals "t0-admins"
Set-AdmPwdReadPasswordPermission -Identity "OU=PAW,OU=Tier1,OU=Admin,$dsname"
-AllowedPrincipals "t0-admins","t1-admins"
Set-AdmPwdReadPasswordPermission -Identity
"OU=Servers,OU=Tier1,OU=Admin,$dsname" -AllowedPrincipals "t0-admins","t1-
admins"
Set-AdmPwdReadPasswordPermission -Identity
"OU=Computers,OU=AzureBlog,$dsname" -AllowedPrincipals "t0-admins","t1-
admins"
Set-AdmPwdReadPasswordPermission -Identity "OU=Quarantine,$dsname" -
AllowedPrincipals "t0-admins","t2-admins"
```

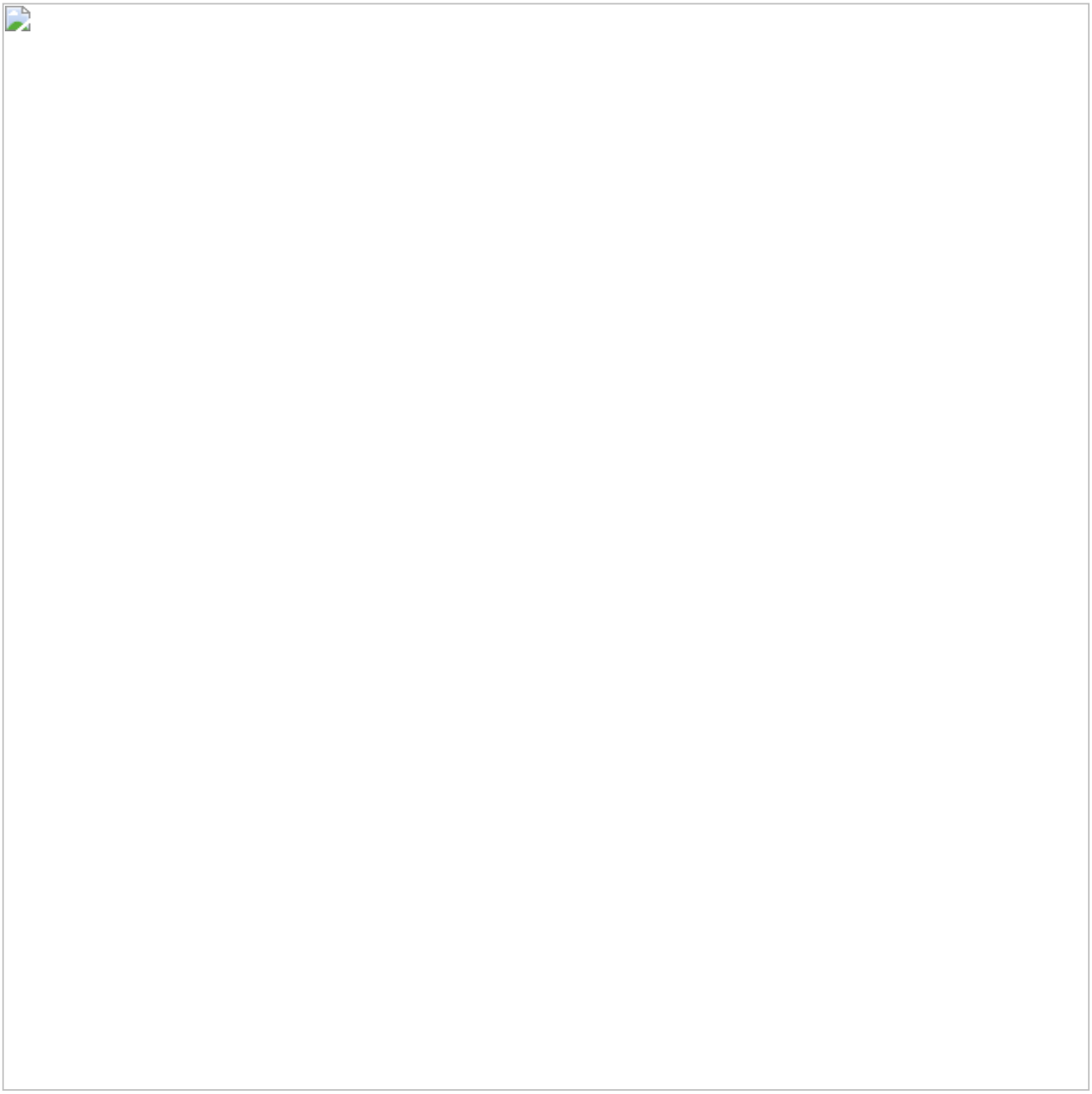


LAPS read permissions configuration

As you can see that under the **Identity** parameter we are providing the distinguished name to the OU and unred **AllowedPrincipals** we are providing groups that should have permissions to read attributes.

Read password permissions configuration is not the end of, we need to configure reset password permissions

```
$dsnAME = (Get-ADDomain).DistinguishedName
Import-module AdmPwd.PS
Set-AdmPwdResetPasswordPermission -Identity
"OU=PAW,OU=Tier0,OU=Admin,$dsname" -AllowedPrincipals "t0-admins"
Set-AdmPwdResetPasswordPermission -Identity
"OU=Servers,OU=Tier0,OU=Admin,$dsname" -AllowedPrincipals "t0-admins"
Set-AdmPwdResetPasswordPermission -Identity
"OU=PAW,OU=Tier1,OU=Admin,$dsname" -AllowedPrincipals "t0-admins","t1-admins"
Set-AdmPwdResetPasswordPermission -Identity
"OU=Servers,OU=Tier1,OU=Admin,$dsname" -AllowedPrincipals "t0-admins","t1-
admins"
Set-AdmPwdResetPasswordPermission -Identity
"OU=Computers,OU=AzureBlog,$dsname" -AllowedPrincipals "t0-admins","t1-
admins"
Set-AdmPwdResetPasswordPermission -Identity "OU=Quarantine,$dsname" -
AllowedPrincipals "t0-admins","t2-admins"
```



LAPS reset permissions configuration

Like before you can see that under the **Identity** parameter we are providing the distinguished name to the OU and unred **AllowedPrincipals** we are providing groups that should have permissions to read attributes.

So PowerShell part is done, let's switch to the Group Policy Management console. We need to create 2 GPO's :

- LAPSInstallation-v1.0 – we will use msi files from SysVol to install it on member servers and workstations
- LAPSConfiguration-v1.0 – we will configure password settings

Below you can find the configuration of GPO's

Name: **LAPSInstallaton-v.1.0**
GPO Status: User configuration settings disabled
Category: *Computer Configuration\Policies\Software Installation*
Package placement: *\\DOMAIN_NAME\SysVol\Domain_NAME\Scripts\LAPS\LAPS.x64.msi*
Deployment State: *Assigned*
Package placement: *\\DOMAIN_NAME\SysVol\Domain_NAME\Scripts\LAPS\LAPS.x86.msi*
Deployment State: *Assigned*
Advanced deployment Options: *Uncheck make this 32-bit x86 application available to Win64 machines.*

Name: **LAPSConfiguration-v.1.0**
GPO Status: User configuration settings disabled
Category: *Computer Configuration\Policies\Administrative Templates\LAPS*
Policy: *Enable local admin password management*
Status: *Enabled*
Policy: *Password Settings*
Password Complexity: *Large Letters + small letters + numbers + specials*
Password Lenght: *14*
PAssword Age (Days): *30*

If at this moment you are about to ask **"OK, but what if I renamed my local administrator account, or I want to use it for different account?"**
Let's split the question into 2 questions:

1. "**OK, but what if I renamed my local administrator account**." – the answer is simple if you rename a built-in administrator account to any other like WKSADM etc. it will still have well-known SID and it will be detected automatically.

2." **I want to use it for different account?** "
For that purpose, you should configure an additional policy in LAPSConfiguration-v.1.0 called:
Name of an administrator account to manage

We are almost at the end of the configuration part. The last thing to do is to link our GPO's to the proper place

In my LAB I linked the GPO's like in the picture below:

Basically, LAPSConfiguration-v.1.0 GPO is linked to the root level of azureblog.pl domain and LAPSInstallation-v.1.0 was linked to each OU where I have computer objects.

LAPS GPO's assignment

After quick **gpupdate /force** command I can see that LAPS was installed on my member server and member workstation.
After a couple of minutes I should be able to read the password from ms-Mcs-AdmPwd attribute using one of the below techniques:

- LAPS UI
- Active Directory Users and Computers
- Powershell

LAPS UI password check

ADUC read LAPS password

Powershell read the password from LAPS

So this is the end.. We have successfully deployed and configured LAPS in a LAB environment.

I hope that this article will help you somehow.