HYBRID IDENTITY PROTECTION conf25

CHARLESTON

SCAN ME

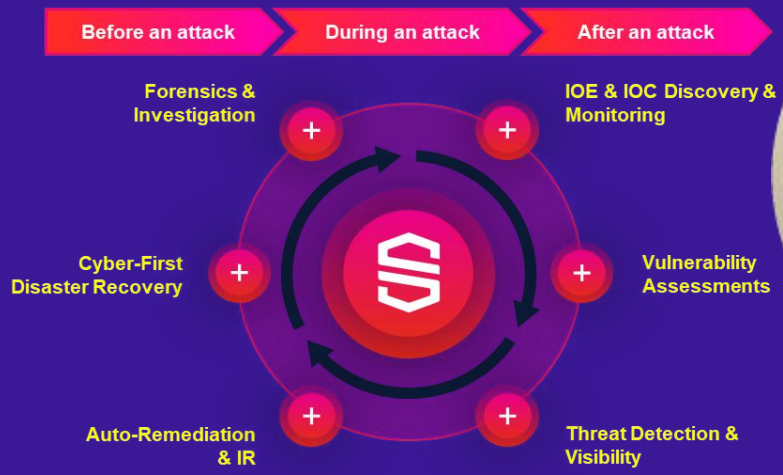
Demystifying Managed Service Accounts: Best Practices & Security Measures to Reduce Risk and Impact

Jorge de Almeida Pinto

Senior Incident Response Lead, SEMPERIS

jorged@semperis.com

Introducing Me, Myself & I!
...and Semperis



Microsoft Most Valuable Professional MVP SINCE 2006

Jorge de Almeida Pinto
Senior Incident Response Lead

LinkedIn	http://tiny.cc/JorgeLinkedIn
Blog	http://tiny.cc/JQFKblog
Twitter	http://tiny.cc/JQFKtwitter
Website	https://www.semperis.com/
Blog	https://www.semperis.com/blog/
Podcast	https://hipconf.com/
Contact	jorged@semperis.com

- *Technology Focus*: Identity, Security And Recovery
- *Product Focus*: AD, ADFS, Entra Connect/Cloud Sync, FIM/MIM, Entra (ID) Technologies.
- *Work*: Architecting, designing, implementing and maintaining secure identity solutions… and recovery
- *Writer Of*: "KRBTGT Pwd Reset", "AD Convergence", "SYSVOL Convergence" Scripts (Feedback WELCOME!)

SCAN ME

Evolution of Service Accounts

LEGACY

NT3.5
NT4

2000

2003
(R2)

2008

2008
R2

2012

2025

WINDOWS SERVER 2012:
Group Managed Service Accounts
(msDS-GroupManagedServiceAccount)
(gMSA)

WINDOWS SERVER 2008R2:
Stand-Alone Managed Service Accounts
(msDS-ManagedServiceAccount)
(sMSA)

WINDOWS SERVER 2025:
Delegated Managed Service Accounts
(msDS-DelegatedManagedServiceAccount)
(dMSA)

PS: Throughout presentation you may see “xMSA”, which means “sMSA, gMSA and/or dMSA”

(Legacy) Service Accounts

Common Good, Bad and Ugly Reality

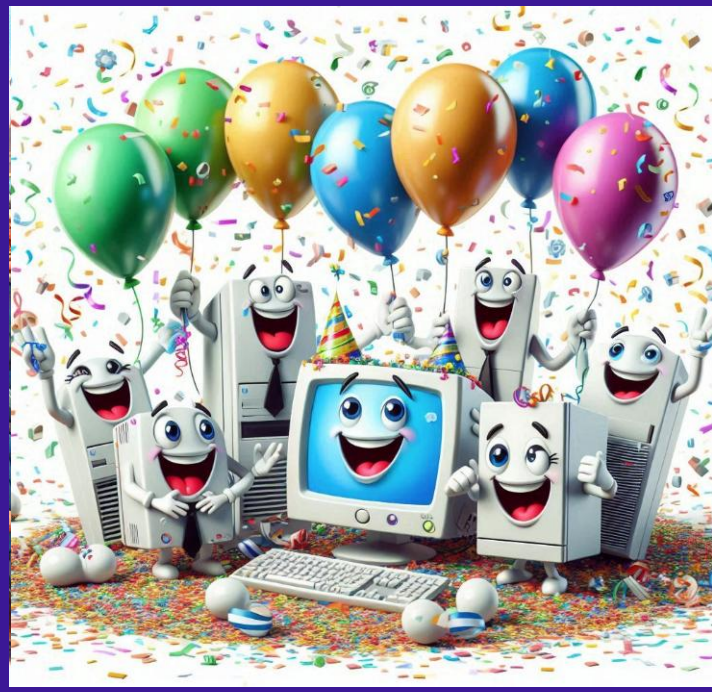
- Used for svcs, apps, IIS, scheduled tasks, keytabs, etc. (i.e., all over the place)
- Based on USER objectClass + "Password Never Expires“
- Configured with SPN(s) + RC4 support + Overprivileged
- Application owners with multiple svc accounts sharing same password
    - Very likely crappy/reused password, incl. bad account hygiene
- In many occasions no clear/unique/consistent naming convention + reusing
- No ownership/periodic recertification → hard to discover, secure and audit in AD
- Prime targets for attackers using the “*Kerberoasting Attack*” because…

Service Accounts vs xMSAs

Main benefits of sMSAs/gMSAs/dMSAs over (legacy) service accounts?

- Automatic, better & stronger credential management

RETRY…
What happens when you ask AI: “Kerberoasting Being History!”



The following still applies for sMSAs/gMSAs/dMSAs

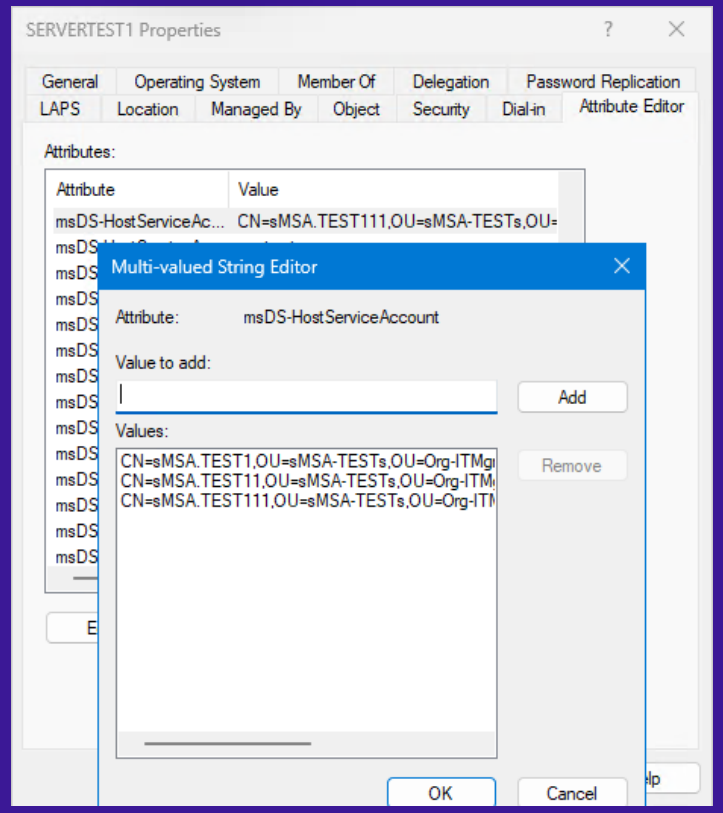
- Clear and unique naming convention
- Ownership and recertification
- Least privilege
- Protecting access to, usage of account and its credentials (incl. server it runs on)

Managed Service Accounts

Stand-Alone (sMSA)

2008 R2

- sMSAs → objectClass = msDS-ManagedServiceAccount
- sMSA is linked to 1 specific computer
  - Forward link "*msDS-HostServiceAccount*" on computer
  - Back link "*msDS-HostServiceAccountBL*" on sMSA
- sMSA can be transferred to another computer (relink + reinstall)



Managed Service Accounts

Stand-Alone (sMSA)

- Auto password/SPN management by computer (No KDS Root Key Required)
  - Initial password generated and set when installing the sMSA on computer (Possible to reset password: *Reset-ADServiceAccountPassword -Identity* <*sMSA*>)
  - sMSA uses the exact same logic/behavior and password update interval as the computer it is being used on
- Like for computers, following policy settings also impact management of sMSAs
  - Security Option "*Domain member: Disable machine account password changes*“ (Not Configured = Default = DO Change Password)
  - Security Option "*Domain member: Maximum machine account password age*“ (Not Configured = Default = 30 Days)
  - Security Option "*Domain controller: Refuse machine account password changes*“ (Not Configured = Default = DO NOT Refuse Password Changes)

Managed Service Accounts

Stand-Alone (sMSA)

- Get relevant data from all sMSAs (Stand-Alone Managed Service Accounts) in AD Domain (https://gist.github.com/zjorz/1d454aaa7c8fb7f0a696092b332af49b)
  - Password Change Interval: Very likely the default of 30 days…. But…

sMSAs In The AD Domain 'ADTEC.NET' (SID: S-1-5-21-274783270-2712129839-3354909249) (2025-05-27 20:56:09)

Filter

Add criteria

DistinguishedName	SamAccountName	RID	Type	description	Enabled	KerbEncryptType	WhenCreated	WhenChanged	PasswordLastSetSmsa	PasswordLastSetHost	msDS-HostServiceA...	MemberOf
CN=sMSA.TEST2,OU=sMSA-TESTs,...	sMSA.TEST2$	12804	sMSA	sMSA For TEST SERVER 2	True	RC4, AES128, AES256	2025-05-02 23:23:35	2025-05-17 14:16:29	2025-05-16 20:50:18	2025-05-16 06:50:16	{CN=SERVERTEST2,...	{CN=Domain Admins,CN=Users,DC=ADTEC,DC=NET}
CN=sMSA.TEST3,OU=sMSA-TESTs,...	sMSA.TEST3$	12868	sMSA	sMSA For TEST SERVER 3	True	RC4, AES128, AES256	2025-05-03 23:27:31	2025-05-17 14:16:29	2025-05-15 06:54:00	2025-05-14 20:28:58	{CN=SERVERTEST3,...	{CN=Domain Admins,CN=Users,DC=ADTEC,DC=NET}
CN=sMSA.TEST1,OU=sMSA-TESTs,...	sMSA.TEST1$	12803	sMSA	sMSA For TEST SERVER 1	True	RC4, AES128, AES256	2025-05-02 23:23:34	2025-05-17 14:16:29	2025-05-16 00:35:16	2025-05-16 00:35:16	{CN=SERVERTEST1,...	{CN=Domain Admins,CN=Users,DC=ADTEC,DC=NET}
CN=sMSA.TEST9,OU=sMSA-TESTs,...	sMSA.TEST9$	12817	sMSA	sMSA For TEST SERVER 9	True	RC4, AES128, AES256	2025-05-02 23:23:39	2025-05-17 14:16:29	2025-05-05 01:43:30	2025-05-02 23:23:39	{CN=SERVERTEST9,...	{}
CN=sMSA.RODC,OU=sMSA-TESTs,...	sMSA.RODC$	12884	sMSA	sMSA For TEST SERVER 3	True	RC4, AES128, AES256	2025-05-09 12:53:18	2025-05-17 14:16:29	2025-05-15 14:06:56	2025-05-14 20:28:58	{CN=SERVERTEST3,...	{CN=GRP_R0_ALLOWCache-R0FSRODC1,OU=Grou...
CN=sMSA.TEST4,OU=sMSA-TESTs,...	sMSA.TEST4$	12807	sMSA	sMSA For TEST SERVER 4	True	RC4, AES128, AES256	2025-05-02 23:23:36	2025-05-17 14:16:29	2025-05-02 23:23:36	2025-05-02 23:23:36	{CN=SERVERTEST4,...	{}
CN=sMSA.TEST5,OU=sMSA-TESTs,...	sMSA.TEST5$	12809	sMSA	sMSA For TEST SERVER 5	True	RC4, AES128, AES256	2025-05-02 23:23:37	2025-05-17 14:16:29	2025-05-02 23:23:37	2025-05-02 23:23:36	{CN=SERVERTEST5,...	{}
CN=sMSA.TEST6,OU=sMSA-TESTs,...	sMSA.TEST6$	12811	sMSA	sMSA For TEST SERVER 6	True	RC4, AES128, AES256	2025-05-02 23:23:37	2025-05-17 14:16:29	2025-05-02 23:23:37	2025-05-02 23:23:37	{CN=SERVERTEST6,...	{}
CN=sMSA.TEST7,OU=sMSA-TESTs,...	sMSA.TEST7$	12813	sMSA	sMSA For TEST SERVER 7	True	RC4, AES128, AES256	2025-05-02 23:23:38	2025-05-17 14:16:29	2025-05-02 23:23:38	2025-05-02 23:23:38	{CN=SERVERTEST7,...	{}
CN=sMSA.TEST8,OU=sMSA-TESTs,...	sMSA.TEST8$	12815	sMSA	sMSA For TEST SERVER 8	True	RC4, AES128, AES256	2025-05-02 23:23:39	2025-05-17 14:16:29	2025-05-02 23:23:39	2025-05-02 23:23:38	{CN=SERVERTEST8,...	{}
CN=sMSA.TEST22,OU=sMSA-TESTs...	sMSA.TEST22$	12873	sMSA	sMSA For TEST SERVER 2	True	RC4, AES128, AES256	2025-05-04 22:41:20	2025-05-17 14:16:29	2025-05-04 22:41:20	2025-05-16 06:50:16	{CN=SERVERTEST2,...	{}
CN=sMSA.TEST11,OU=sMSA-TESTs...	sMSA.TEST11$	12872	sMSA	sMSA For TEST SERVER 1	True	RC4, AES128, AES256	2025-05-04 22:41:19	2025-05-17 14:16:29	2025-05-15 13:37:48	2025-05-16 00:35:16	{CN=SERVERTEST1,...	{}
CN=sMSA.TEST111,OU=sMSA-TES...	sMSA.TEST111$	12883	sMSA	sMSA For TEST SERVER 1	True	RC4, AES128, AES256	2025-05-08 18:01:07	2025-05-17 14:16:29	2025-05-16 00:20:16	2025-05-16 00:35:16	{CN=SERVERTEST1,...	{}
CN=sMSA.TEST33,OU=sMSA-TESTs...	sMSA.TEST33$	12871	sMSA	sMSA For TEST SERVER 3	True	RC4, AES128, AES256	2025-05-04 21:51:52	2025-05-17 14:16:29	2025-05-16 04:40:05	2025-05-14 20:28:58	{CN=SERVERTEST3,...	{}

Managed Service Accounts

Group (gMSA)

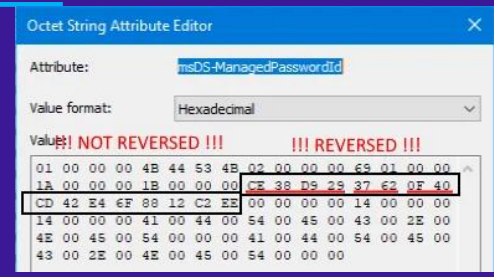
2012

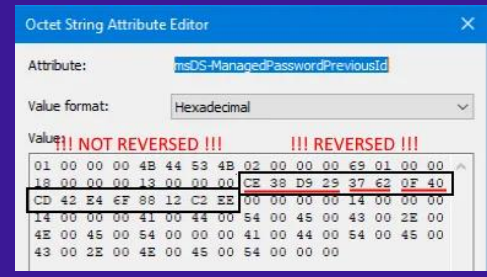
- gMSAs → objectClass = msDS-GroupManagedServiceAccount
- Centralized password management → KDS Root Key (at least 1) in AD Forest
  - KDS Root Keys are stored in AD in container: "*CN=Master Root Keys,CN=Group Key Distribution Service,CN=Services,CN=Configuration,DC=<forest name>*"
  - *name* (KeyId) → identifier of KDS Root Key object referenced by gMSAs in attributes "*msDS-ManagedPasswordId*" and "*msDS-ManagedPasswordPreviousId*"
  - "*msKds-CreateTime*" (CreationTime) → time KDS Root Key object was created in AD
  - "*msKds-UseStartTime*" (EffectiveTime) → time any RWDC can start using KDS Root Key Object for gMSAs
    - For subsequent KDS Root Keys: Create KDS Root Key + Force AD Repl + Restart KDSSVC
- gMSA can be shared by multiple computers or be restricted to just 1 (flexibility!)

Managed Service Accounts

Group (gMSA)

- The inner guts of a gMSA
  - *"msDS-ManagedPasswordInterval"*: password rotation interval in days, set at creation ONLY. Default of 30 days = too long. Thoughts/suggestion: set it to 3-5 days. Depends on AD Replication Convergence!. Measure AD Replication Convergence for Configuration NC through → https://github.com/zjorz/Public-AD-Scripts/blob/master/Check-AD-Replication-Latency-Convergence.md
  - *"msDS-ManagedPasswordId"*: references keyID of KDS Root Key currently being used (N).
  - *"ms-DS-ManagedPasswordPreviousId"*: references keyID of KDS Root Key previously being used (N-1).

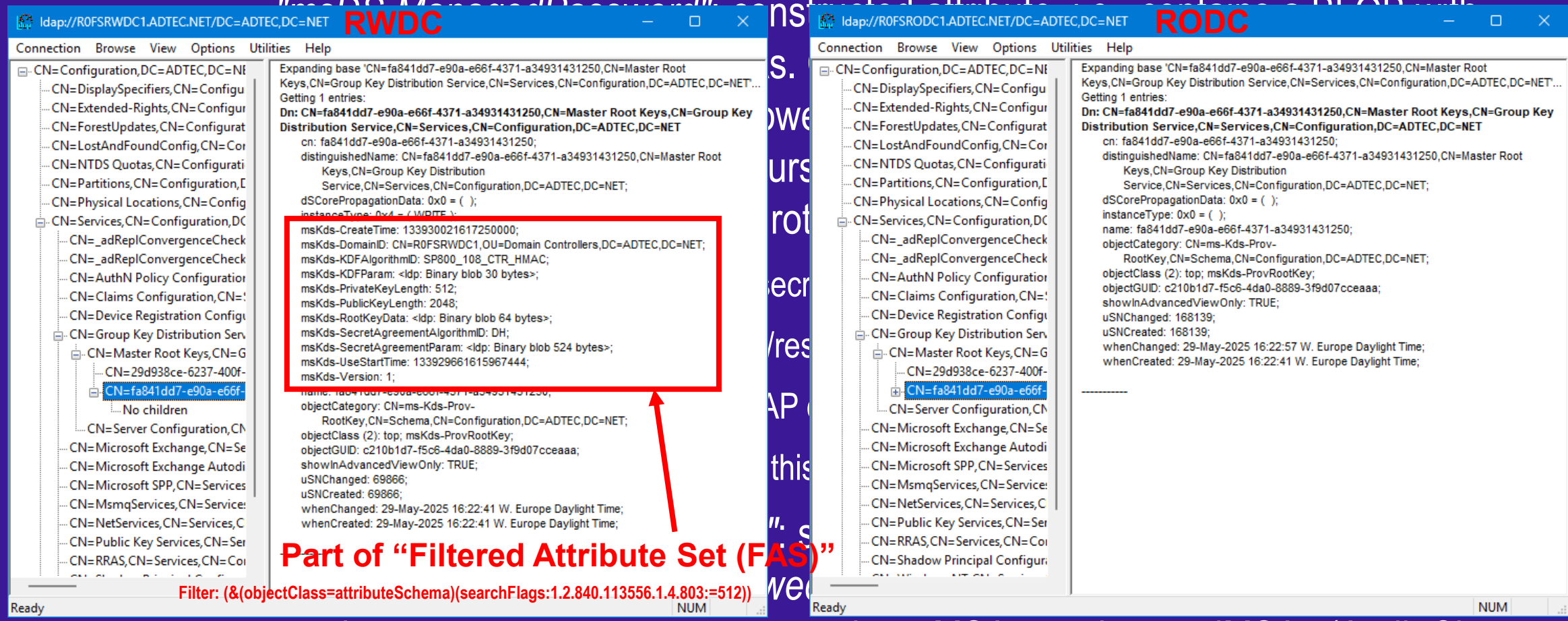




Managed Service Accounts

Group (gMSA)

- The inner guts of a gMSA (Continued…)



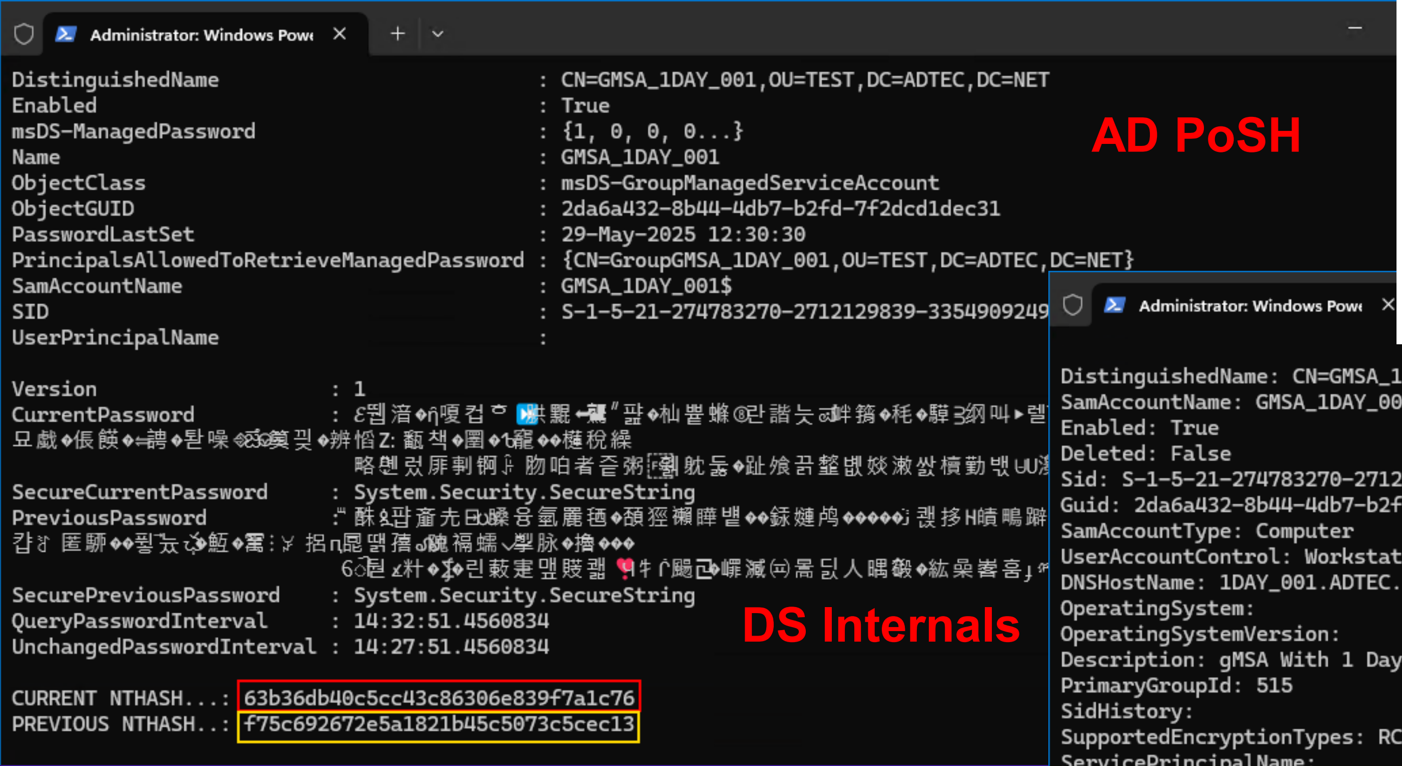
can be **groups**, computers, users, other gMSAs, and even dMSAs (Audit Changes!)

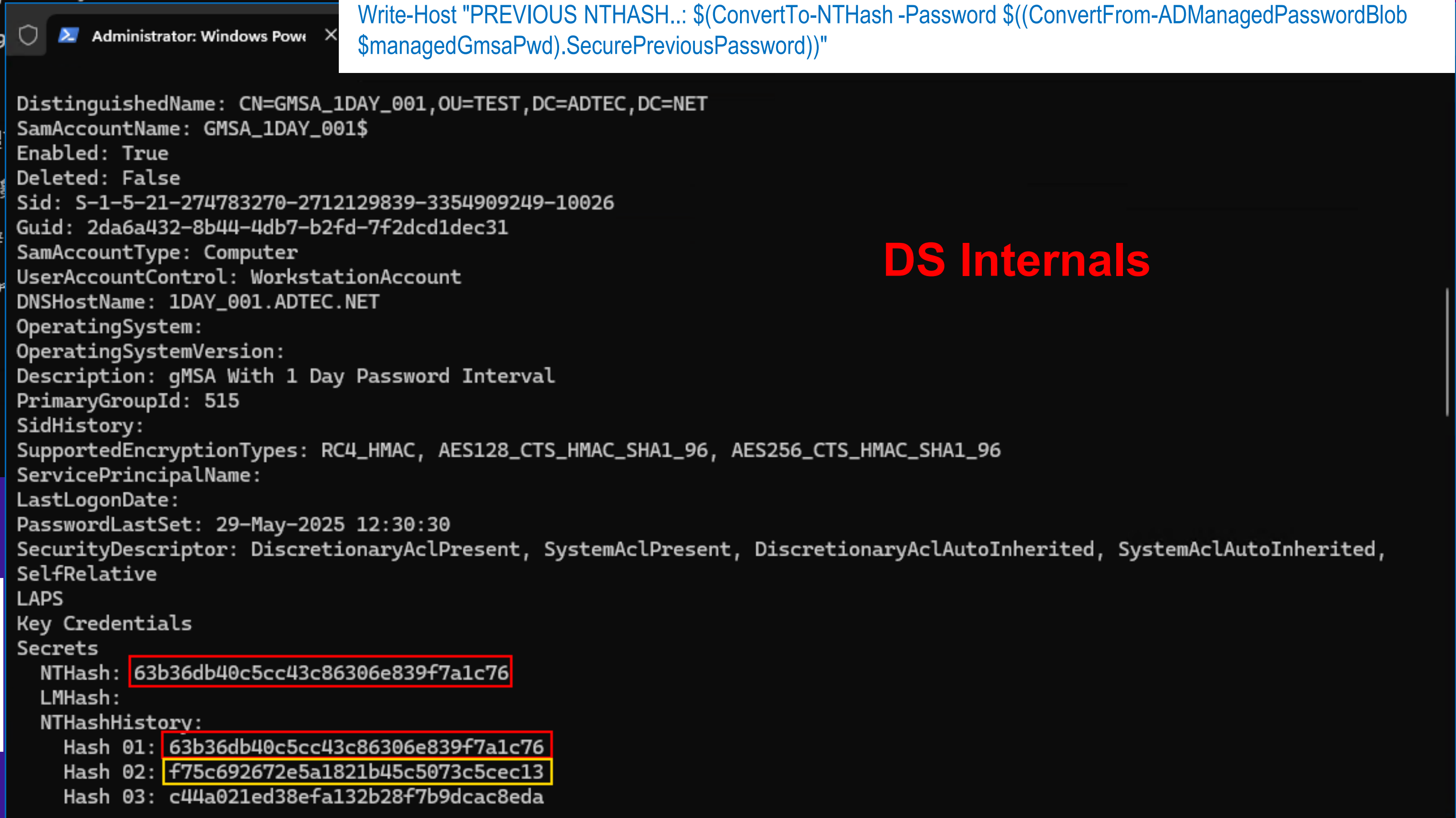
Managed Service Accounts

Group (gMSA)

Retrieving 'msDS-ManagedPassword' Using LDAP Query When Allowed ONLY Works With gMSA, As For dMSA A TGS Request Is Needed

```
$gMSASamAccountName = 'GMSA_1DAY_001$'
$gMSA = Get-ADServiceAccount -Identity $gMSASamAccountName -Properties 'msDS-ManagedPassword',PasswordLastSet,PrincipalsAllowedToRetrieveManagedPassword -Server $((Get-ADDomain -Current LocalComputer).PDCEmulator)
$gMSA
$managedGmsaPwd = $gMSA.'msDS-ManagedPassword'
ConvertFrom-ADManagedPasswordBlob $managedGmsaPwd
Write-Host "CURRENT NTHASH...: $(ConvertTo-NTHash -Password $((ConvertFrom-ADManagedPasswordBlob $managedGmsaPwd).SecureCurrentPassword))"
Write-Host "PREVIOUS NTHASH..: $(ConvertTo-NTHash -Password $((ConvertFrom-ADManagedPasswordBlob $managedGmsaPwd).SecurePreviousPassword))"
```


Administrator: Windows Powe
DistinguishedName : CN=GMSA_1DAY_001,OU=TEST,DC=ADTEC,DC=NET
Enabled : True
msDS-ManagedPassword : {1, 0, 0, 0...}
Name : GMSA_1DAY_001
ObjectClass : msDS-GroupManagedServiceAccount
ObjectGUID : 2da6a432-8b44-4db7-b2fd-7f2dcd1dec31
PasswordLastSet : 29-May-2025 12:30:30
PrincipalsAllowedToRetrieveManagedPassword : {CN=GroupGMSA_1DAY_001,OU=TEST,DC=ADTEC,DC=NET}
SamAccountName : GMSA_1DAY_001$
SID : S-1-5-21-274783270-2712129839-3354909249
UserPrincipalName :
AD PoSH
Version : 1
CurrentPassword :
SecureCurrentPassword : System.Security.SecureString
PreviousPassword :
SecurePreviousPassword : System.Security.SecureString
QueryPasswordInterval : 14:32:51.4560834
UnchangedPasswordInterval : 14:27:51.4560834
DS Internals
CURRENT NTHASH...: 63b36db40c5cc43c86306e839f7a1c76
PREVIOUS NTHASH..: f75c692672e5a1821b45c5073c5cec13


Administrator: Windows Powe
DistinguishedName: CN=GMSA_1DAY_001,OU=TEST,DC=ADTEC,DC=NET
SamAccountName: GMSA_1DAY_001$
Enabled: True
Deleted: False
Sid: S-1-5-21-274783270-2712129839-3354909249-10026
Guid: 2da6a432-8b44-4db7-b2fd-7f2dcd1dec31
SamAccountType: Computer
UserAccountControl: WorkstationAccount
DNSHostName: 1DAY_001.ADTEC.NET
OperatingSystem:
OperatingSystemVersion:
Description: gMSA With 1 Day Password Interval
PrimaryGroupId: 515
SidHistory:
SupportedEncryptionTypes: RC4_HMAC, AES128_CTS_HMAC_SHA1_96, AES256_CTS_HMAC_SHA1_96
ServicePrincipalName:
LastLogonDate:
PasswordLastSet: 29-May-2025 12:30:30
SecurityDescriptor: DiscretionaryAclPresent, SystemAclPresent, DiscretionaryAclAutoInherited, SystemAclAutoInherited, SelfRelative
LAPS
Key Credentials
Secrets
NTHash: 63b36db40c5cc43c86306e839f7a1c76
LMHash:
NTHashHistory:
Hash 01: 63b36db40c5cc43c86306e839f7a1c76
Hash 02: f75c692672e5a1821b45c5073c5cec13
Hash 03: c44a021ed38efa132b28f7b9dcac8eda
DS Internals

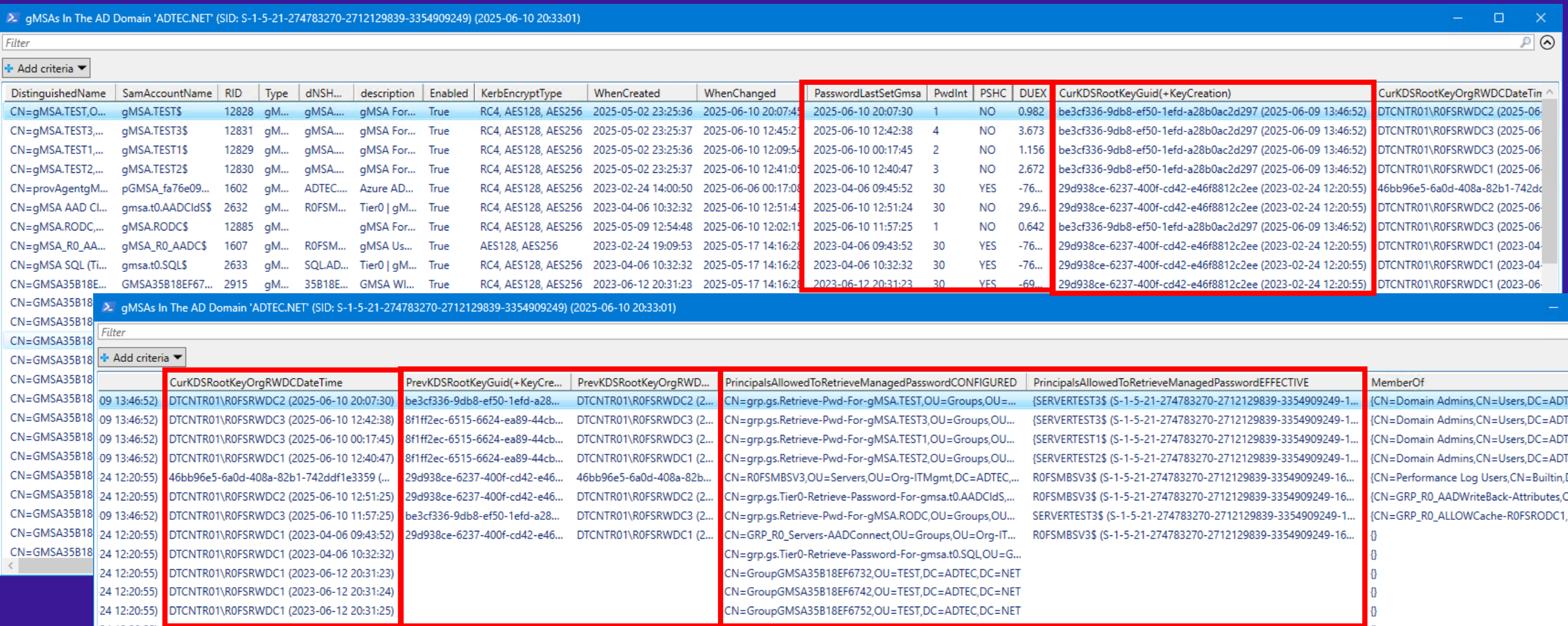
Works With BOTH gMSA And dMSA

```
$gMSASamAccountName = 'GMSA_1DAY_001$'
$adAccount = Get-ADReplAccount -SamAccountName $gMSASamAccountName -Server $((Get-ADDomain -Current LocalComputer).PDCEmulator)
$adAccount
```

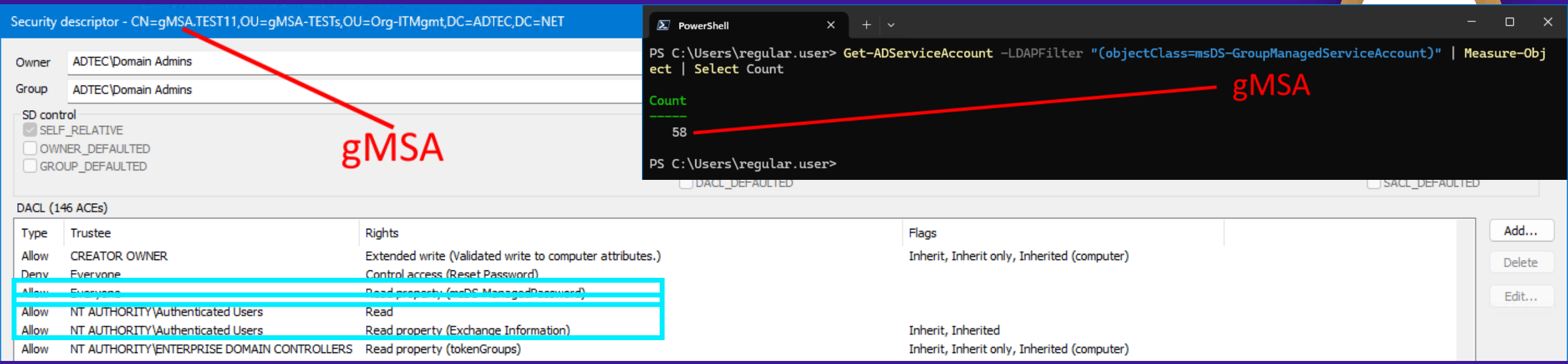
Managed Service Accounts

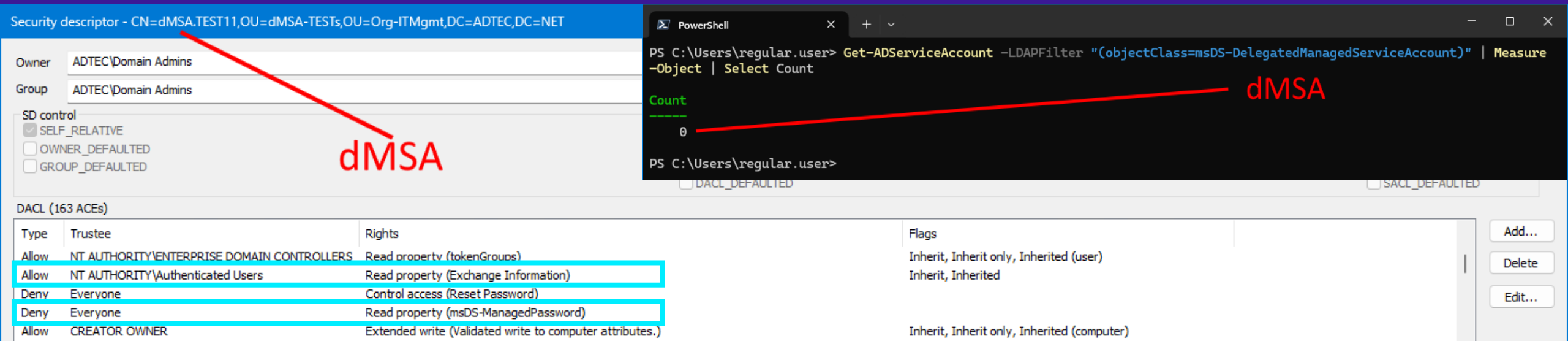
Group (gMSA)

- Get relevant data from all gMSAs (Group Managed Service Accounts) in AD Domain (https://gist.github.com/zjorz/d1906ac04964a29d87bd377e0298eb21)



Managed Service Accounts
Delegated (dMSA)

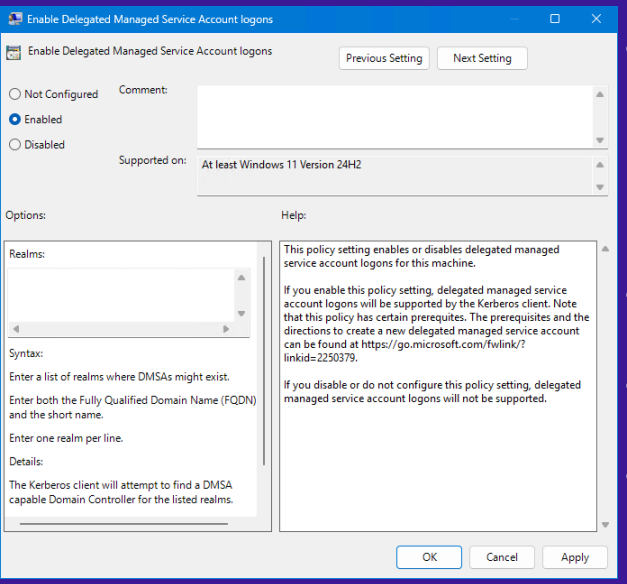




Managed Service Accounts

Delegated (dMSA)

- dMSA = gMSA with more steroids and more requirements! Key differences are:
  - Unlike gMSA, with regards to a dMSA:



- dMSA support NOT enabled by default
  - Not enabling support? → dMSA authN fails with username/password incorrect, logon failure, etc
  - Enable support through registry or GPO + realms (=optimize lookup of domains with W2K25 RWDCs)
- It supports native use and migration from legacy service account (last is main use case!)
- NATIVE dMSA use supports services, IIS App Pools but not Scheduled Tasks
- dMSA creation and management is to be considered as Tier 0!

Managed Service Accounts

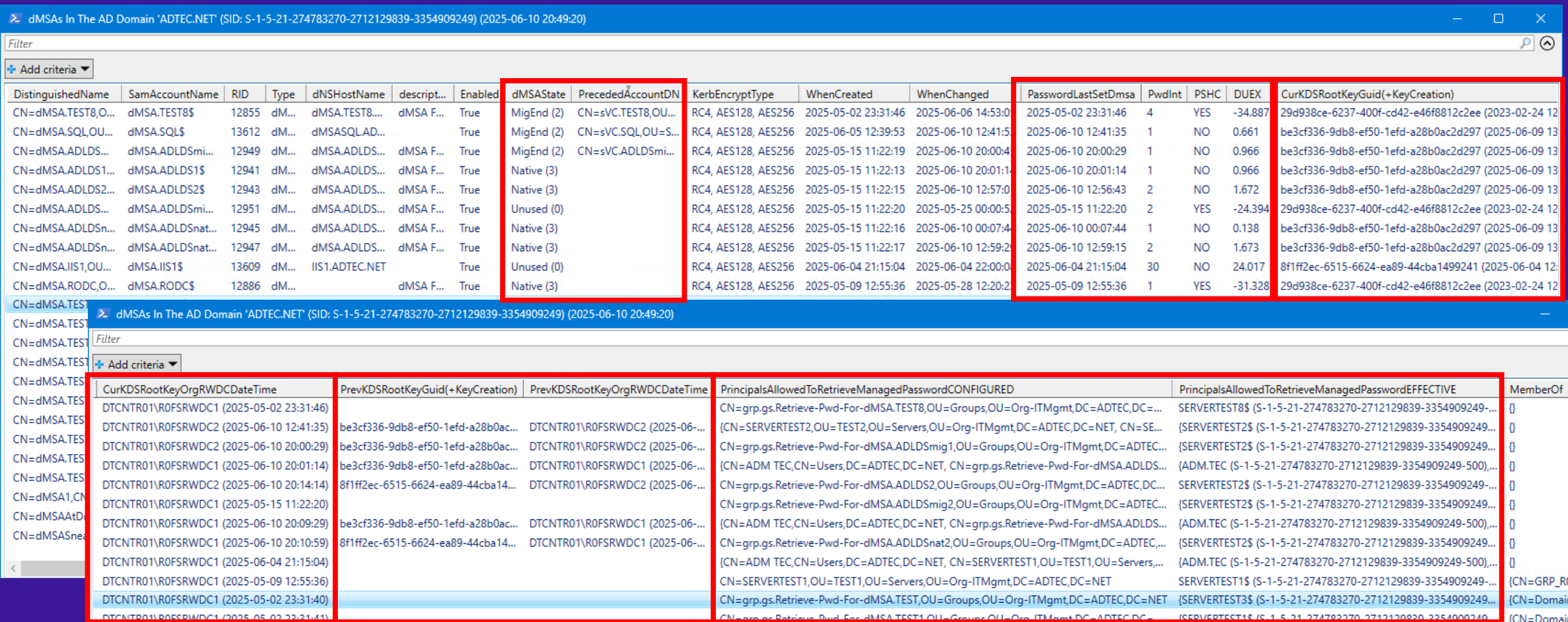
Delegated (dMSA)

- The inner guts of a dMSA / Superseded Account
  - *"msDS-DelegatedMSAState"*: state of dMSA and how it is being used if applicable
    - 0 = Unused (Default)
    - 1 = Migration Start | 2 = Migration End (Migration of legacy service account to dMSA!)
    - 3 = Native Use
  - *"msDS-SupersededServiceAccountState"*: state of superseded account
    - Empty = Not superseded (Default)
    - 1 = Migration Start | 2 = Migration End (Migration of legacy service account to dMSA!)
  - *"msDS-ManagedAccountPrecededByLink":* DN of legacy service account (a.k.a. account being superseded) (AT LEAST: how it is intended! ;-))
  - *"msDS-SupersededManagedAccountLink"*: DN of dMSA (a.k.a. account superseding)
  - adminSDHolder DOES NOT follow link!

Managed Service Accounts

Delegated (dMSA)

- Get relevant data from all dMSAs (Deleg. Managed Service Accounts) in AD Domain (https://gist.github.com/zjorz/62de8c4b5c8d10f7b3c1934c4332dfb8)



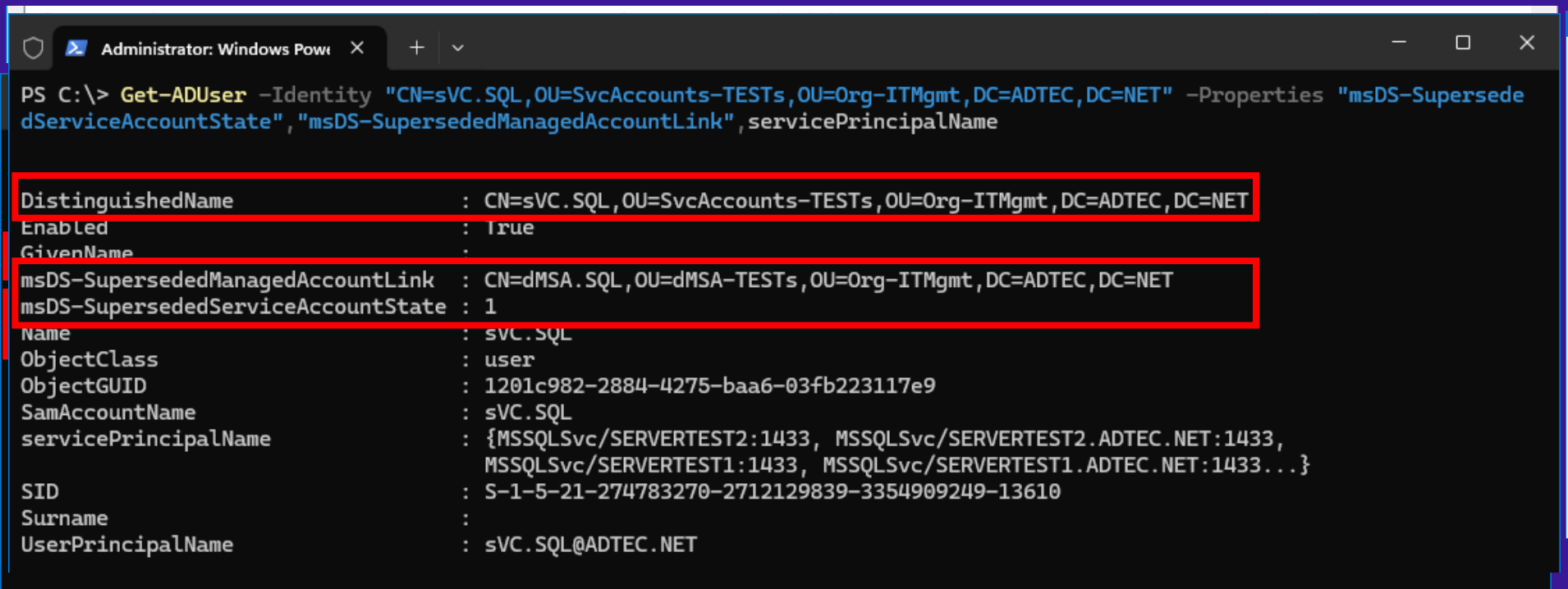
Migrating Service Accounts

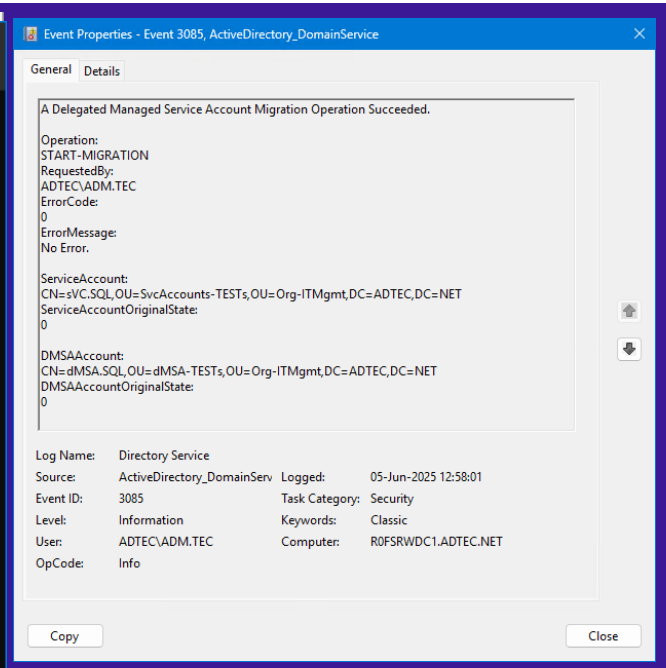
Legacy → dMSA (Good Successor!)

INITIATING MIGRATION through PoSH CMDlet (Domain Admin only!):

```
# Starting Migration Of Svc Account To dMSA
Start-ADServiceAccountMigration -Identity "<dMSA>" -SupersededAccount "<DN of Legacy Svc Account>"
```

```
# Starting Migration Of Svc Account To dMSA (Under The Hood)
$rootDSE = [ADSI]"LDAP://<RWDC FQDN>/RootDSE“
$rootDSE.Put("migrateADServiceAccount", "<DN of dMSA>:<DN of Legacy Svc Account>:1")
$rootDSE.SetInfo()
```

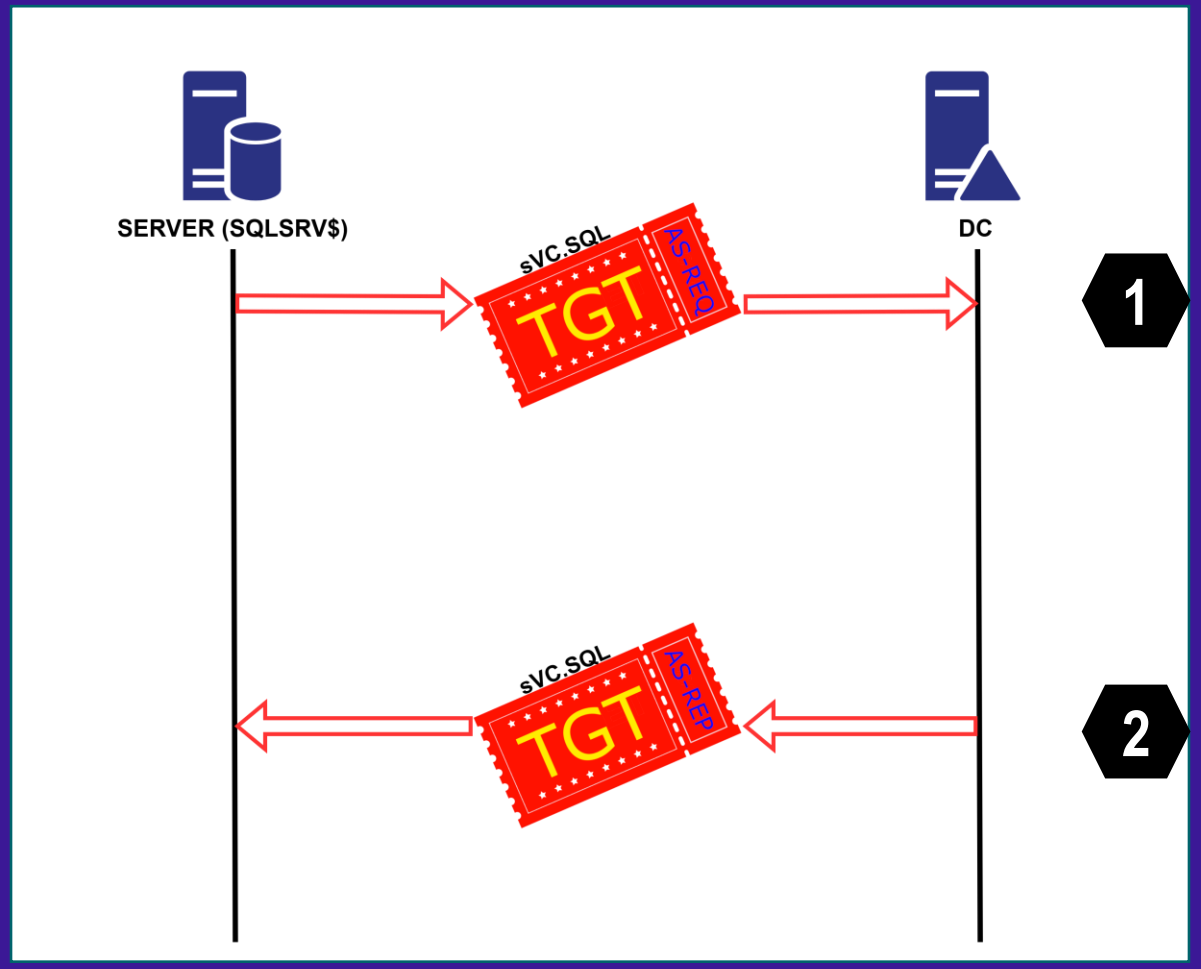




Migrating Service Accounts

Legacy → dMSA (Good Successor!)

Authentication BEFORE Migration State

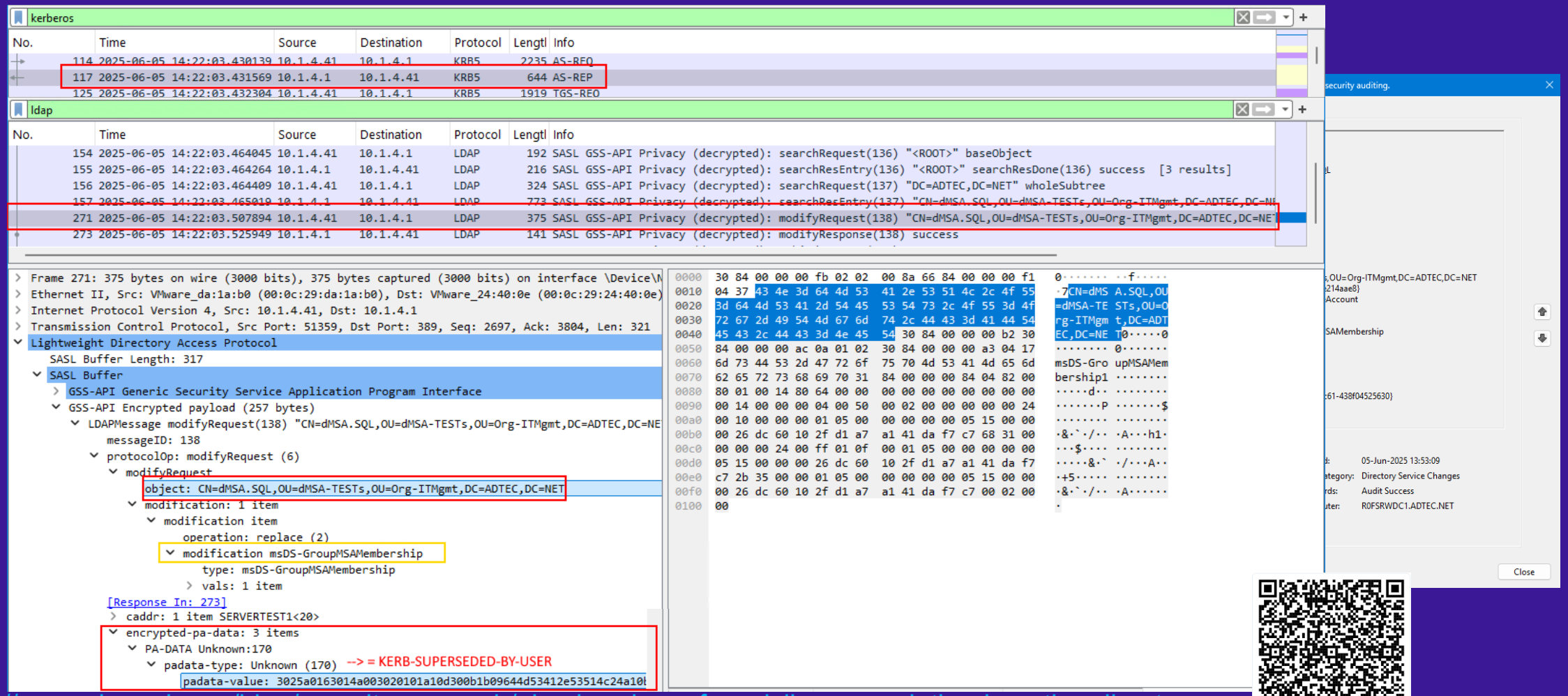


SOURCE: https://www.akamai.com/blog/security-research/abusing-dmsa-for-privilege-escalation-in-active-directory

Migrating Service Accounts

Legacy → dMSA (Good Successor!)

Authentication DURING Migration State (either force or take enough time!)



SOURCE: https://www.akamai.com/blog/security-research/abusing-dmsa-for-privilege-escalation-in-active-directory

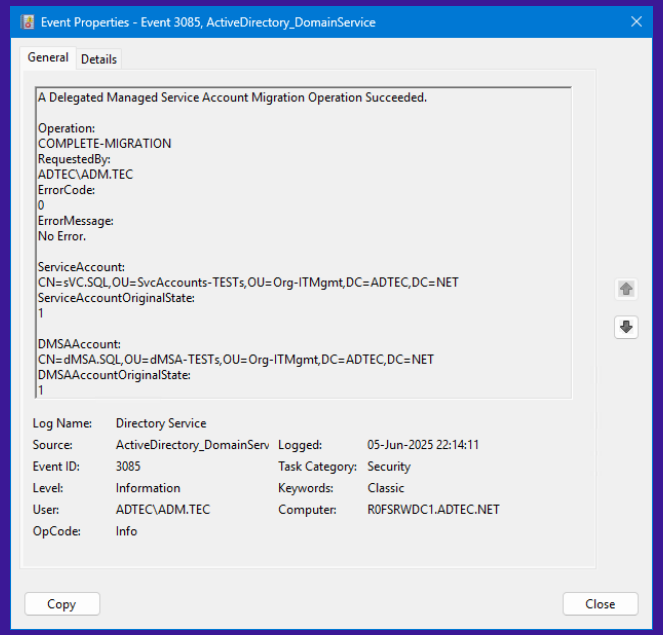
Migrating Service Accounts

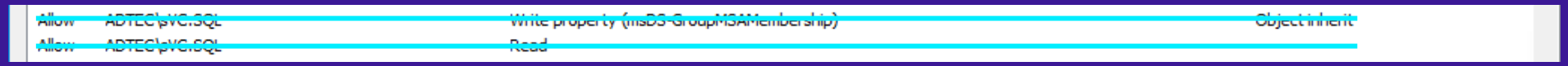
Legacy → dMSA (Good Successor!)

COMPLETING MIGRATION through PoSH CMDlet (Domain Admin only!):

```
# Completing Migration Of Svc Account To dMSA
Complete-ADServiceAccountMigration -Identity "<dMSA>" -SupersededAccount "<DN of Legacy Svc Account>"
```

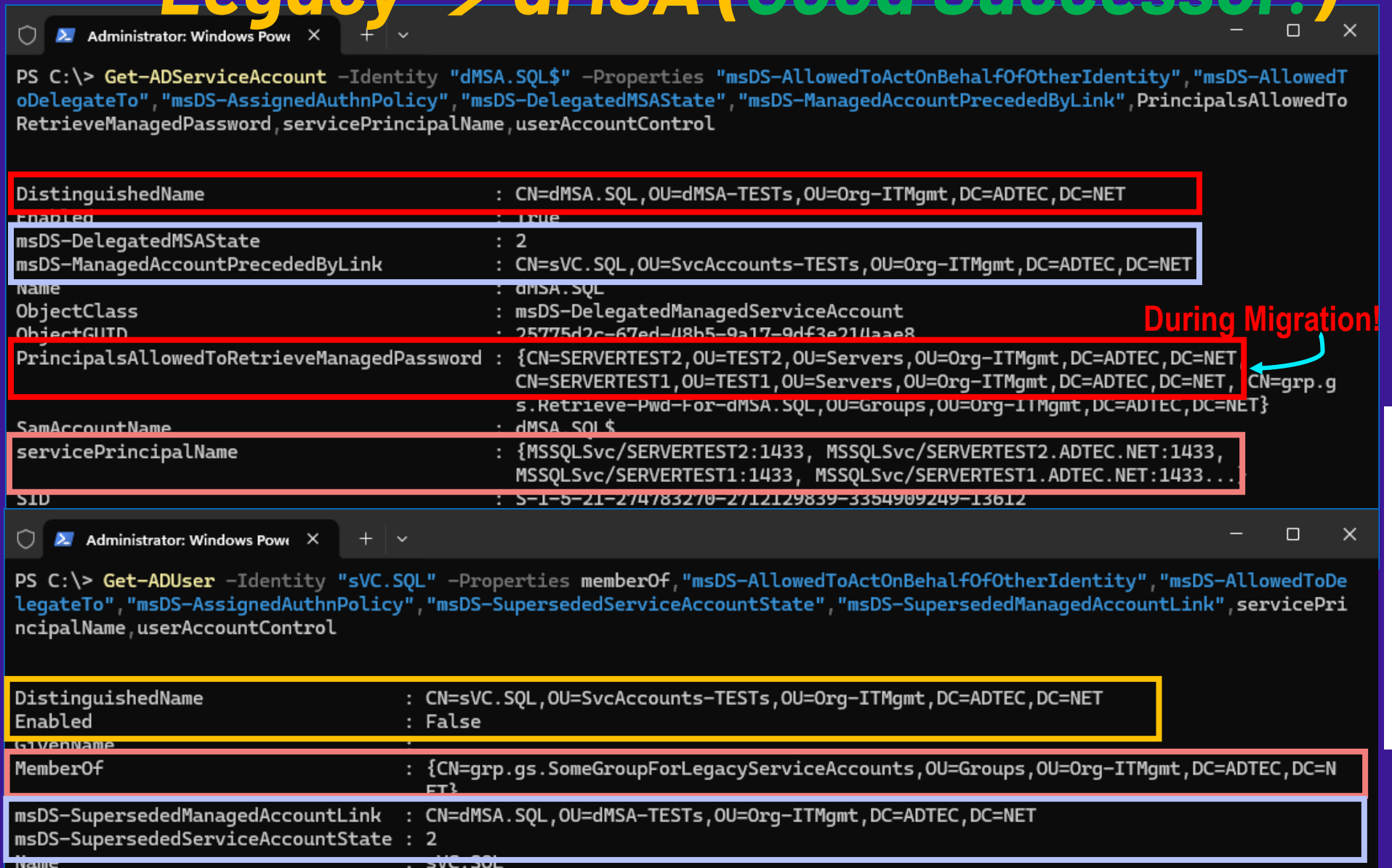
```
# Completing Migration Of Svc Account To dMSA (Under The Hood)
$rootDSE = [ADSI]"LDAP://<RWDC FQDN>/RootDSE“
$rootDSE.Put("migrateADServiceAccount", "<DN of dMSA>:<DN of Legacy Svc Account>:2")
$rootDSE.SetInfo()
```





Migrating Service Accounts

Legacy → dMSA (Good Successor!)



COMPLETING MIGRATION through PoSH CMDlet (Domain Admin only!):

Config Migration Legacy Svc Acc 2 dMSA

- Service Principal Names (SPNs)
- Allowed To Delegate To List
- Resource Based Constrained Delegation
- Assigned Authentication Policy
- Assigned Authentication Silo
- Trusted AuthN For Delegation UAC Bit

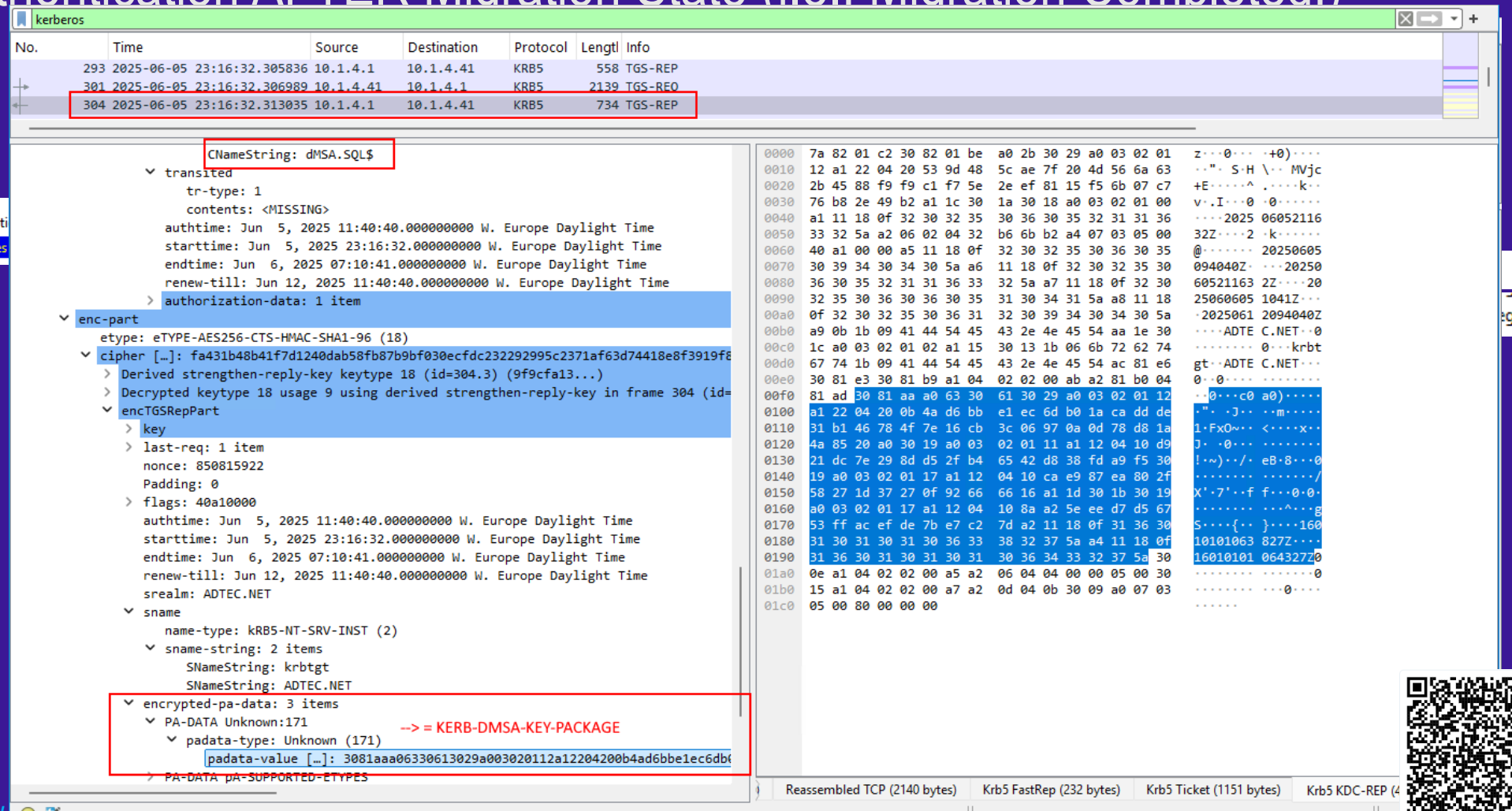
REQUIRES Attention if applicable!:

- Allow/Denied To Cache" List of RODC(s)

Migrating Service Accounts

Legacy → dMSA (Good Successor!)

Authentication AFTER Migration State (i.e., Migration Completed!)



SOURCE: https://www.akamai.com/blog/security-research/abusing-dmsa-for-privilege-escalation-in-active-directory

Migrating Service Accounts

Legacy → dMSA (Good Successor!)

WARNING: Some applications/services may still require attention!!! → e.g., ADFS

Administrator: Windows Powe

```
AllowedOnBehalfOfCallers SID
S-1-5-21-274783270-2712129839-3354909249-13636
```

BEFORE Start

```
AuthorizationPolicy
@RuleName = "Permit Service Account"
exists([Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/
29839-3354909249-13636"])
 => issue(Type = "http://schemas.microsoft.com/authorization/claims/perm

@RuleName = "Permit Local Administrators"
exists([Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/
 => issue(Type = "http://schemas.microsoft.com/authorization/claims/perm
```

```
AuthorizationPolicyReadOnly
@RuleName = "Permit Service Account"
exists([Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/
29839-3354909249-13636"])
 => issue(Type = "http://schemas.microsoft.com/authorization/claims/perm

@RuleName = "Permit Local Administrators"
exists([Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/
 => issue(Type = "http://schemas.microsoft.com/authorization/claims/perm
```

```
PS C:\>
```

Microsoft.IdentityServer.PolicyModel.Client.StorageAuthorizationException: ADMIN0120: The client is not authorized to access the endpoint net.tcp://localhost:1500/policy. The client process must be run with service administrative privileges.

at Microsoft.IdentityServer.PolicyModel.Client.PolicyStoreClientManager.SearchWorker(Filter filter, Int32 maxObjects, String[] propertyNames, Boolean firstTry, PropertyFactoryBase propertyFactory)

at Microsoft.IdentityServer.PolicyModel.Client.PolicyManagerBase.Search[T](Filter filter, Int32

Log Name:	AD FS/Admin		
Source:	AD FS	Logged:	01-Jul
Event ID:	102	Task Category:	None
Level:	Error	Keywords:	AD FS
User:	ADTEC\dMSA.ADFS$	Computer:	SERVE
OpCode:	Info		

Copy

Close

Administrator: Windows Powe

```
AllowedOnBehalfOfCallers SID
S-1-5-21-274783270-2712129839-3354909249-13636
```

AFTER Starting Mig
BEFORE Completing

```
AuthorizationPolicy
@RuleName = "Permit Service Account"
exists([Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/primarysi
29839-3354909249-13636"])
 => issue(Type = "http://schemas.microsoft.com/authorization/claims/permit", Value

@RuleName = "Permit Local Administrators"
exists([Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/groupsid"
 => issue(Type = "http://schemas.microsoft.com/authorization/claims/permit", Value
@RuleName = "Permit Service Account"
exists([Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/primarysi
29839-3354909249-13637"])
=> issue(Type = "http://schemas.microsoft.com/authorization/claims/permit", value
```

```
AuthorizationPolicyReadOnly
@RuleName = "Permit Service Account"
exists([Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/primarysi
29839-3354909249-13636"])
 => issue(Type = "http://schemas.microsoft.com/authorization/claims/permit", Value

@RuleName = "Permit Local Administrators"
exists([Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/groupsid"
 => issue(Type = "http://schemas.microsoft.com/authorization/claims/permit", Value
@RuleName = "Permit Service Account"
exists([Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/primarysid",
29839-3354909249-13637"])
=> issue(Type = "http://schemas.microsoft.com/authorization/claims/permit", value = "
```

Copy

Close

Event Properties - Event 143, AD FS

General Details

Administrator: Windows Powe

```
AllowedOnBehalfOfCallers SID
S-1-5-21-274783270-2712129839-3354909249-13636
```

AFTER Completing Migration

```
AuthorizationPolicy
@RuleName = "Permit Local Administrators"
exists([Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/groupsid", Value == "S-1-5-32-544"])
 => issue(Type = "http://schemas.microsoft.com/authorization/claims/permit", Value = "true");

@RuleName = "Permit Service Account"
exists([Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/primarysid", Value == "S-1-5-21-274783270-27121
29839-3354909249-13637"])
=> issue(Type = "http://schemas.microsoft.com/authorization/claims/permit", value = "true");
```

New Service Account

```
AuthorizationPolicyReadOnly
@RuleName = "Permit Local Administrators"
exists([Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/groupsid", Value == "S-1-5-32-544"])
 => issue(Type = "http://schemas.microsoft.com/authorization/claims/permit", Value = "true");

@RuleName = "Permit Service Account"
exists([Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/primarysid", Value == "S-1-5-21-274783270-27121
29839-3354909249-13637"])
=> issue(Type = "http://schemas.microsoft.com/authorization/claims/permit", value = "true");
```

New Service Account

```
PS C:\>
```

User:	ADTEC\dMSA.ADFS$	Computer:	SERVERTEST1.ADTEC.NET
OpCode:	Info		

Copy

Close

 # Adding NEW Service Account To AuthZ Rules

https://github.com/microsoft/adfsToolbox/tree/master/s

Add-AdfsServiceAccountRule -ServiceAccount "<New dMSA Account>" [-SecondaryServers "<List Of Secondary Servers>"]

 # Removing OLD Service Account From AuthZ Rules

https://github.com/microsoft/adfsToolbox/tree/master/serviceAccountModule

Remove-AdfsServiceAccountRule -ServiceAccount "<Old Legacy Account>" [-SecondaryServers "<List Of Secondary Servers>"]

Migrating Service Accounts
Account → dMSA (Bad Successor!)

- Bad Successor = **WITHOUT PATCH** Aug 12, 2025 - KB5063878 - W2K25 DCs
  - Merging PAC (= Privilege Attribute Certificate) and getting hashes/keys of supported encryption types from targeted account, under the following minimal conditions
    - dMSA attribute "*msDS-DelegatedMSAState*" = 2
    - dMSA attribute "*msDS-ManagedAccountPrecededByLink*" = "<*DN of some account, user/computer/sMSA/gMSA/dMSA*>" (anything that can authenticate!)
      Major Problem → Accounts with well-known DNs (default domain admin + KRBTGT)
  - Listed attributes ARE NOT protected from regular LDAP writes!
- Wow! That's a LOT of EASY power! What could go wrong? 🤔

Credits: Yuval Gordon, Akamai Security Researcher, 2025-05
SOURCE: https://www.akamai.com/blog/security-research/abusing-dmsa-for-privilege-escalation-in-active-directory

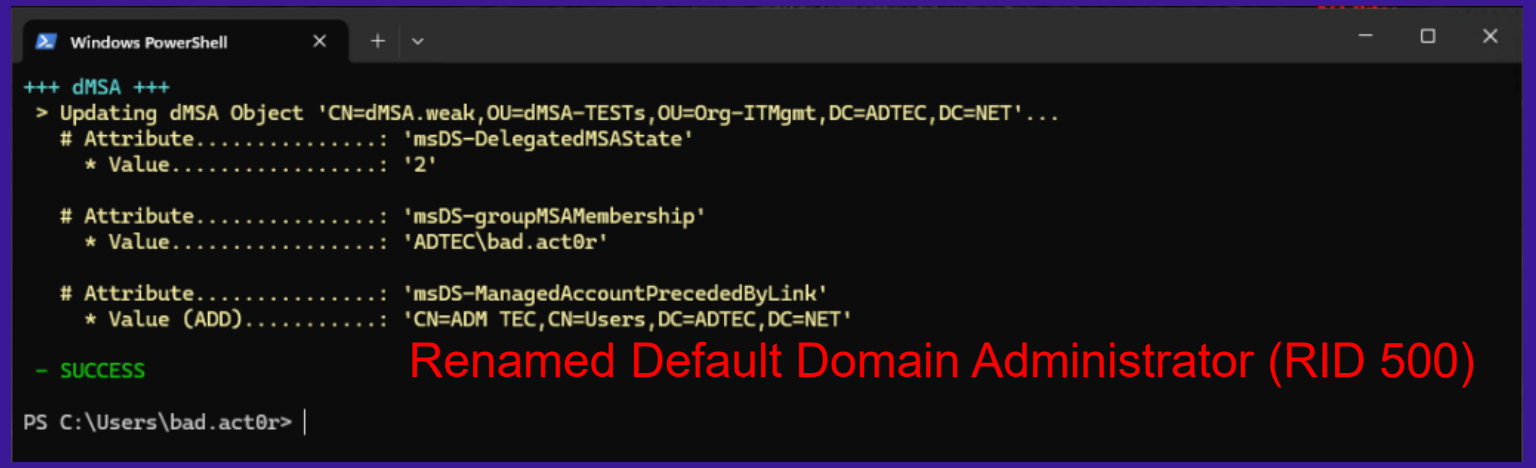
Migrating Service Accounts

Account → dMSA (Bad Successor!)

- Therefore, anyone controlling ANY dMSA through…..:
  - Create Child (Specific to dMSA or generic)
  - Full Control (e.g., Account Operators)
  - Write DACL
  - Write Owner
  - Write Property

Bad-Successor-WRITE-DATA-v2.ps1
Bad-Successor-VIEW-DATA-v2.ps1

```
# WRITE DATA INTO ATTRIBUTES "msDS-groupMSAMembership", "msDS-DelegatedMSAState", "msDS-ManagedAccountPrecededByLink" OF THE dMSA
$dMSADN = "CN=dMSA.weak,OU=dMSA-TESTs,OU=Org-ITMgmt,DC=ADTEC,DC=NET“
$objectState = 2
$accDN = "CN=ADM TEC,CN=Users,DC=ADTEC,DC=NET“ # = Renamed Default Domain Admin
$accAllowGetPwd = "ADTEC\bad.act0r“
$badSuccessorPatchInstalled = $false
```



Migrating Service Accounts

Account → dMSA (Bad Successor!)

Migrating Service Accounts

Accnt → dMSA (Patched Successor!)

- Patched Successor = **WITH PATCH** Aug12, 2025 - KB5063878 - W2K25 DCs
  - Merging PAC (= Privilege Attribute Certificate) and getting hashes/keys of supported encryption types from targeted account, under the following minimal conditions
    - dMSA attribute "*msDS-DelegatedMSAState*" = 2
    - dMSA attribute "*msDS-ManagedAccountPrecededByLink*" = "<*DN of some account, user/computer/sMSA/gMSA~~/dMSA~~*>" (anything that can authenticate, except dMSAs!)
    - (legacy service) account attribute "*msDS-SupersededServiceAccountState*" = 2
    - (legacy service) account attribute "*msDS-SupersededManagedAccountLink*" = <DN of dMSA referencing the (legacy service) account>
  - Listed attributes ARE STILL NOT protected from regular LDAP writes! 🤦
  - In addition to controlling dMSA, control is also needed on target/referenced account!
  - Protected Accounts (secured by adminSDHolder) are now exempt from this attack. But... what about over-permissioned non-protected accounts as the backdoor?

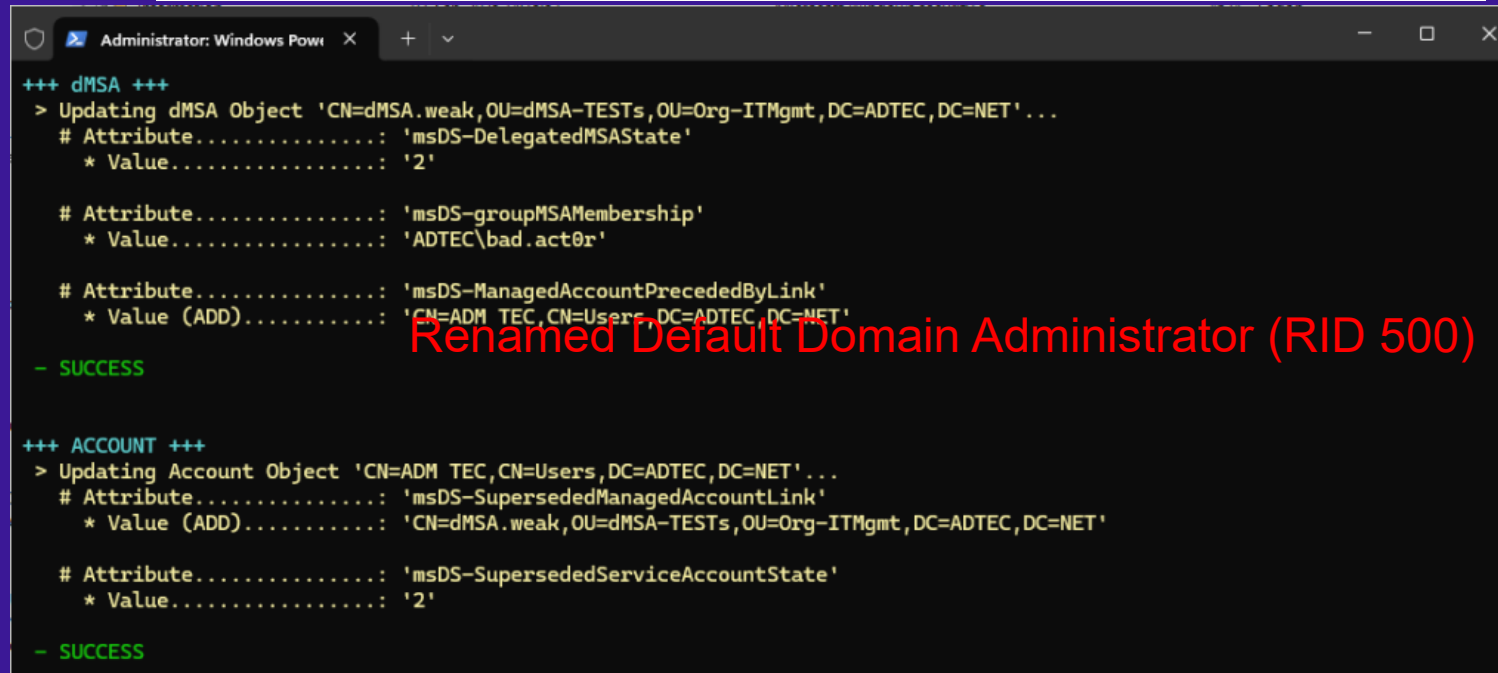
Migrating Service Accounts

Accnt → dMSA (Patched Successor!)

- Therefore, anyone controlling ANY Account + dMSA through:
  - Create Child (Specific to dMSA or generic)
  - Full Control (e.g., Account Operators)
  - Write DACL
  - Write Owner
  - Write Property
- Attack technique still valid!

[Bad-Successor-WRITE-DATA-v2.ps1]
[Bad-Successor-VIEW-DATA-v2.ps1]

```
# WRITE DATA INTO ATTRIBUTES "msDS-groupMSAMembership", "msDS-DelegatedMSAState", "msDS-ManagedAccountPrecededByLink" OF THE dMSA
# WRITE DATA INTO ATTRIBUTES "msDS-SupersededServiceAccountState", "msDS-SupersededManagedAccountLink" OF THE (LEGACY SERVICE) ACCOUNT
$dMSADN = "CN=dMSA.weak,OU=dMSA-TESTs,OU=Org-ITMgmt,DC=ADTEC,DC=NET“
$objectState = 2
$accDN = "CN=ADM TEC,CN=Users,DC=ADTEC,DC=NET“ # = Renamed Default Domain Admin
$accAllowGetPwd = "ADTEC\bad.act0r“
$badSuccessorPatchInstalled = $true
```

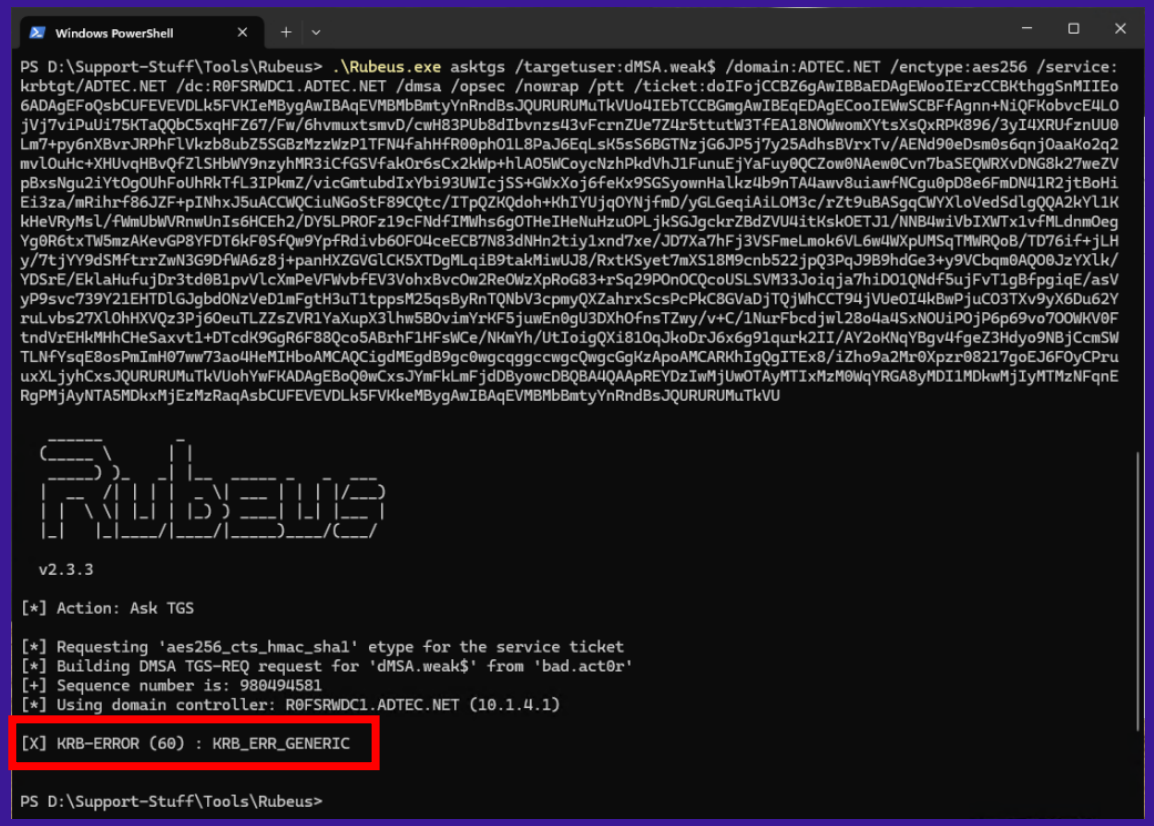


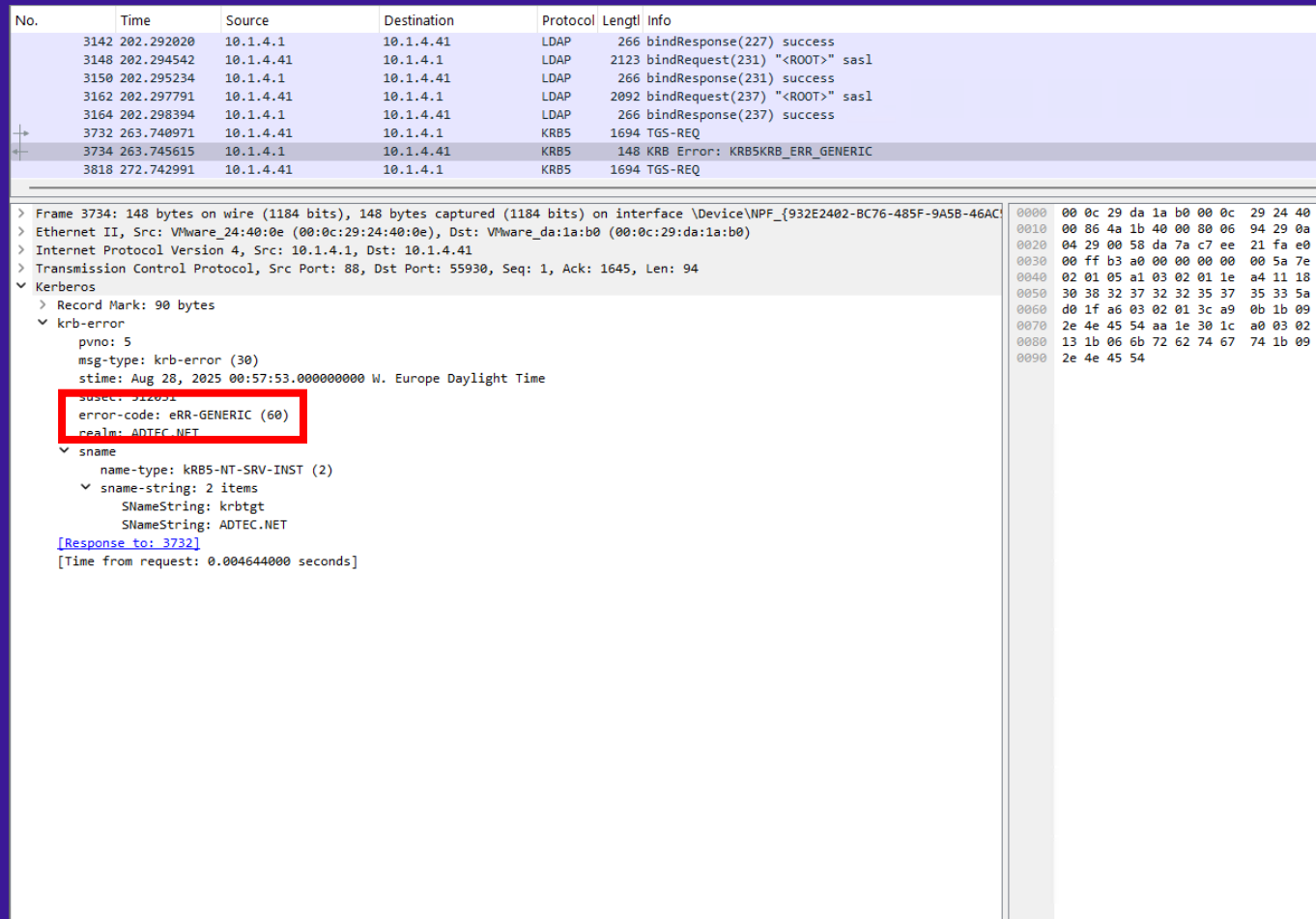
Renamed Default Domain Administrator (RID 500)

Migrating Service Accounts

Accnt → dMSA (Patched Successor!)

- Following only the conditions for Bad Successor while patch is installed, results in Kerberos Error when requesting the TGS for the dMSA!





Migrating Service Accounts

Legacy → dMSA (Protections)

- **REMEMBER**: dMSA creation and management is Tier0!
- Auditing
  - Event ID 5137 - dMSA creation
  - Event ID 5136 - writes to "*msDS-groupMSAMembership*" on dMSA object
  - Applicable For BadSuccessor → Event ID 5136 - writes to "*msDS-DelegatedMSAState*" with value of "2“ (not coming from 1) in combination with writes to "*msDS-ManagedAccountPrecededByLink*" on dMSA object (+not being DN of disabled user object), while also NOT writing anything to "*msDS-SupersededServiceAccountState*" and to "*msDS-SupersededManagedServiceAccountLink*" on the referenced account

Migrating Service Accounts

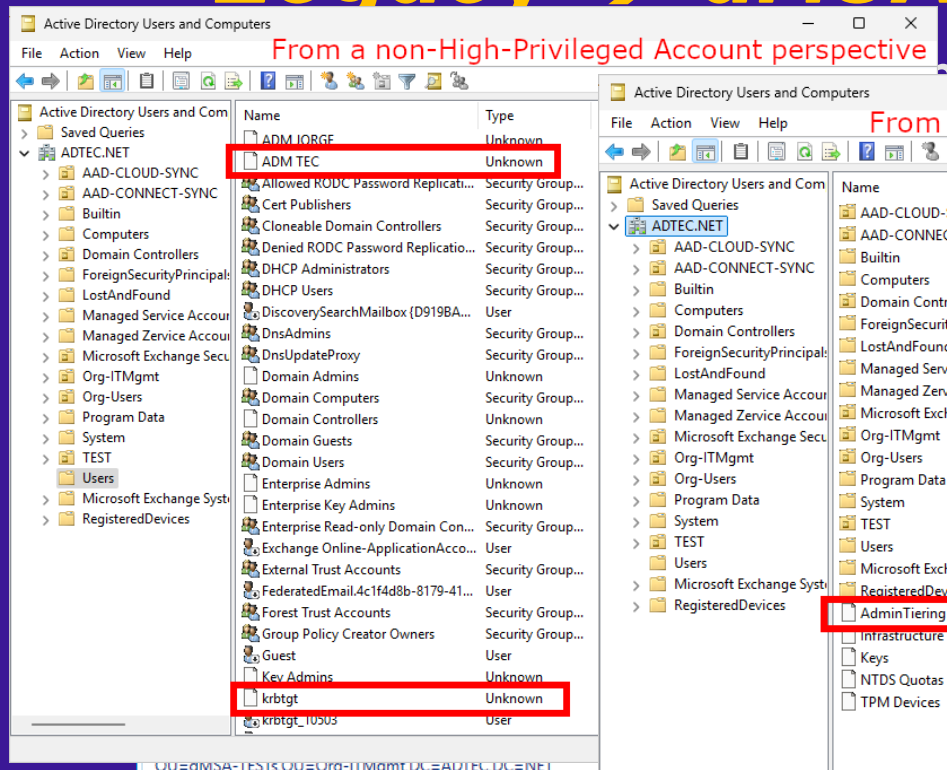
Legacy → dMSA (Protections)

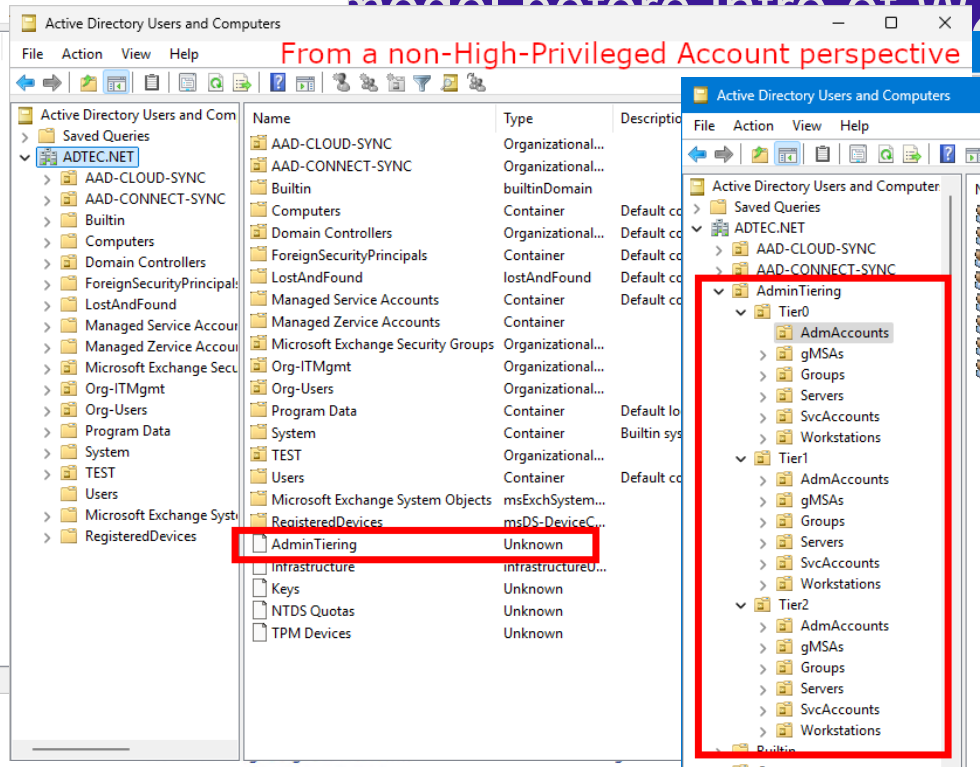
- Auditing (Continued)
  - Applicable For PatchedSuccessor → Event ID 5136 - writes to "*msDS-DelegatedMSAState*" with value of "2“ (not coming from 1) in combination with writes to "*msDS-ManagedAccountPrecededByLink*" on dMSA object (+not being DN of disabled user object), while also writing “2” (not coming from 1) to "*msDS-SupersededServiceAccountState*" and to "*msDS-SupersededManagedServiceAccountLink*" on the referenced account
  - Event ID 2946 - Audit fetching passwords of dMSAs (unusual)
  - Event ID 4768 - Audit Success Of Kerberos Tickets Operations (*TGT Request by Account*) followed by Event ID 4769 - Audit Failure Of Kerberos Tickets Operations (*TGS Request by same Account*)
    (When patched IS installed and BadSuccessor method, 2 attributes only, is used)

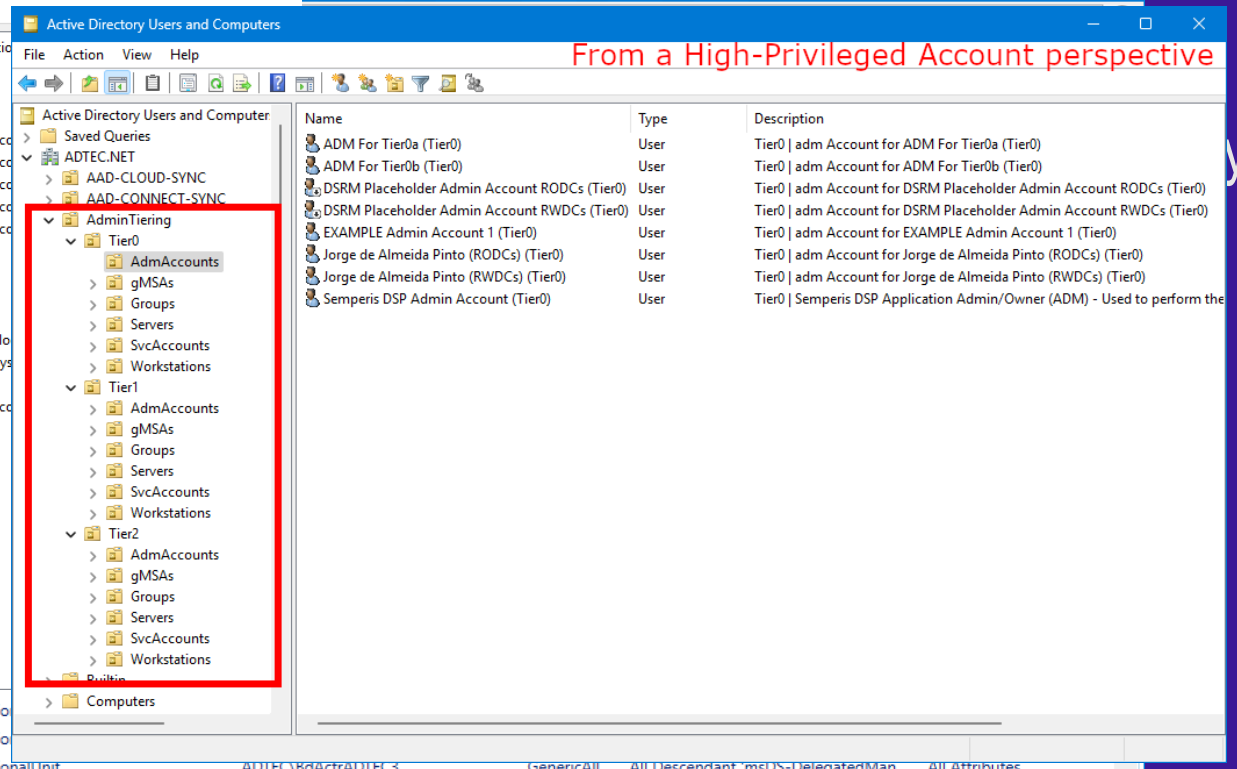
Migrating Service Accounts

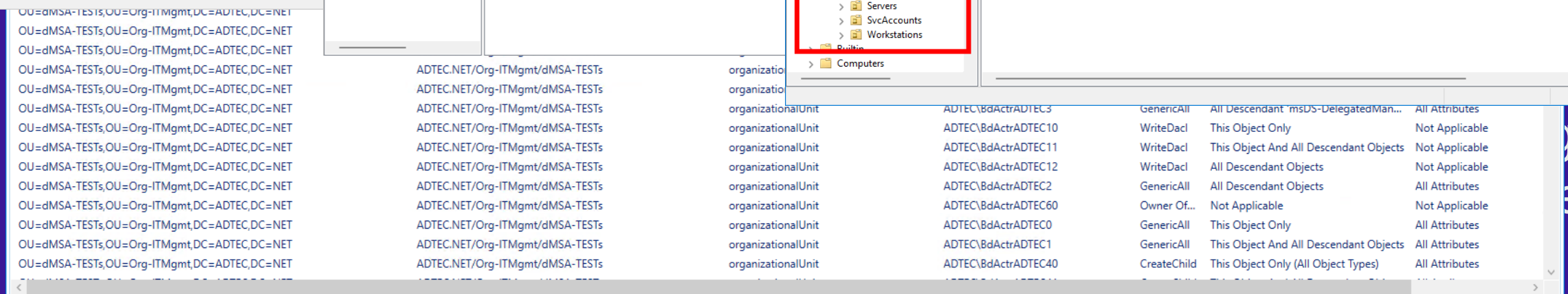
Legacy → dMSA (Protections)

model before intro of W2K25 DCs and enhance security









Migrating Service Accounts

Legacy → dMSA (The Ultimate Block)

- Block writing values into
  - *"msDS-ManagedAccountPrecededByLink"*
  - *"msDS-SupersededManagedServiceAccountLink"*
- Impacts regular LDAP writes & unfortunately also writes through PoSH CMDlets

Link Attribute on a dMSA

Link Attribute on any account

Migrating Service Accounts

Legacy → dMSA (More Information)

- Further Reading
  - BadSuccessor: Abusing dMSA to Escalate Privileges in Active Directory
  - BadSuccessor: How to Detect and Mitigate dMSA Privilege Escalation
  - (2025-05-25) Reviewing Your Delegation Model Before Introducing W2K25 DCs And Enhancing Security (Due To “BadSuccessor”)
  - Understanding & Mitigating BadSuccessor
  - (2025-07-11) How to Block BadSuccessor: The Good, Bad, and Ugly of dMSA Migration
  - (2025-09-02) From BadSuccessor To PatchedSuccessor



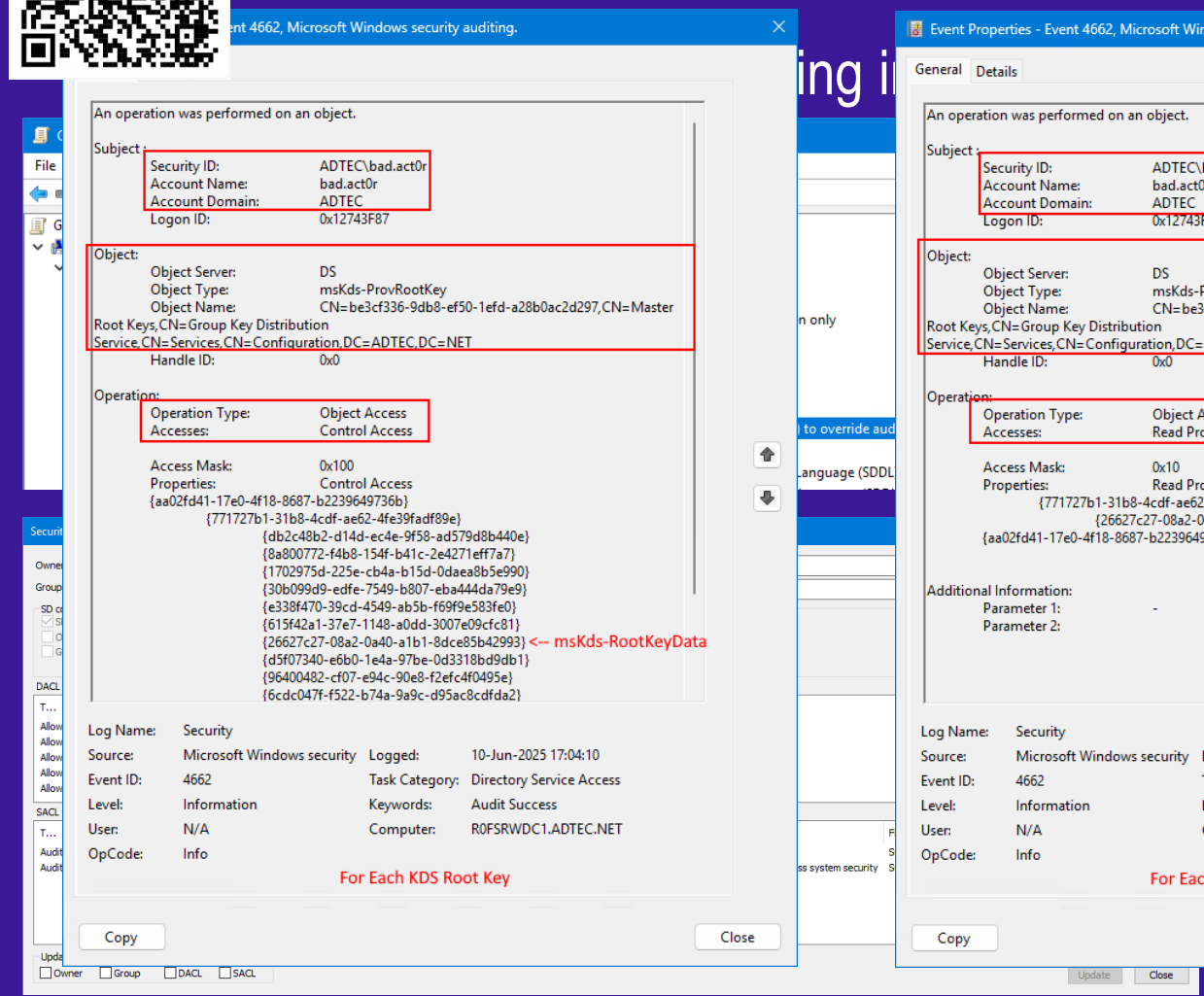


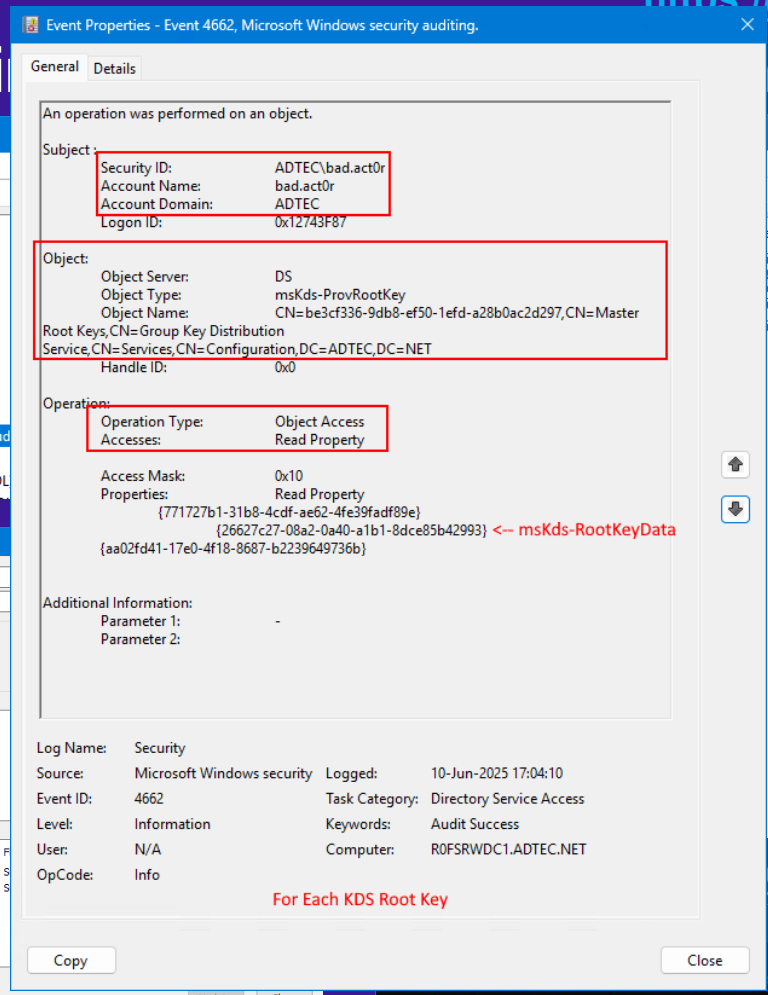

Auditing KDS Root Keys Access

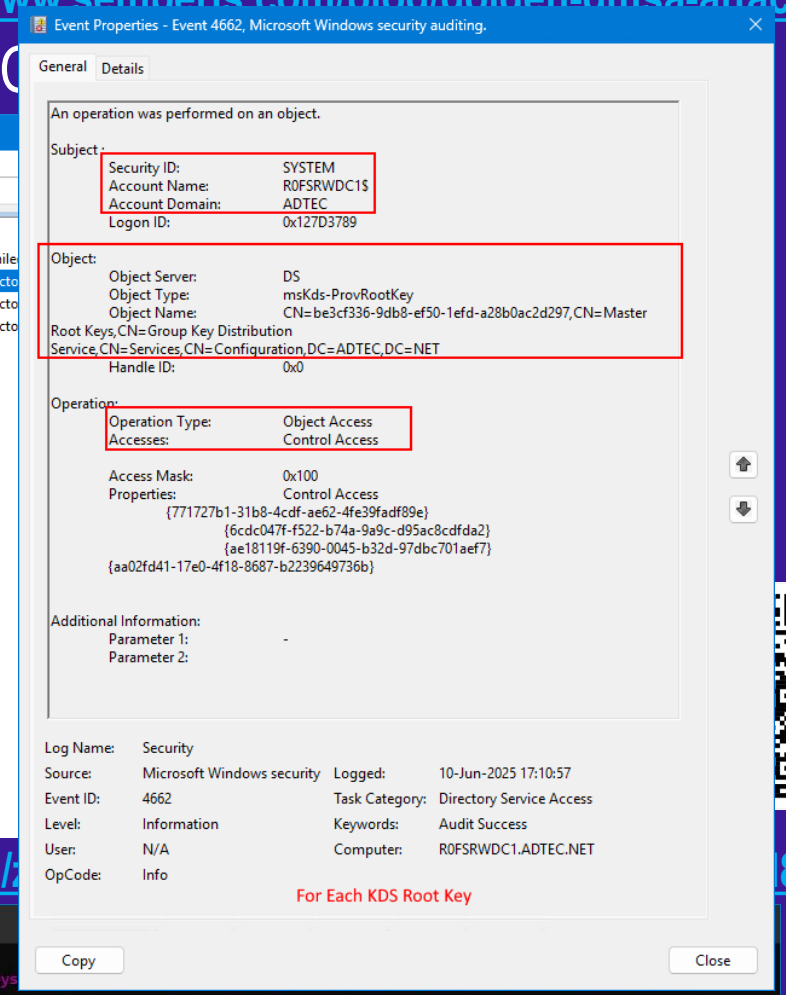
Detecting Golden gMSA/dMSA Attacks

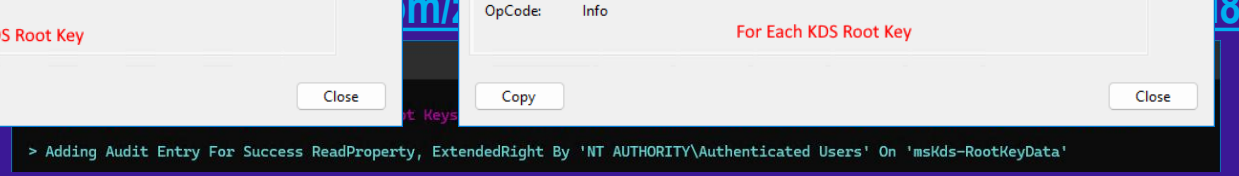
Access of any KDS Root Key by anyone is NOT audited in any way by default!

https://www.semperis.com/blog/golden-gmsa-attack/







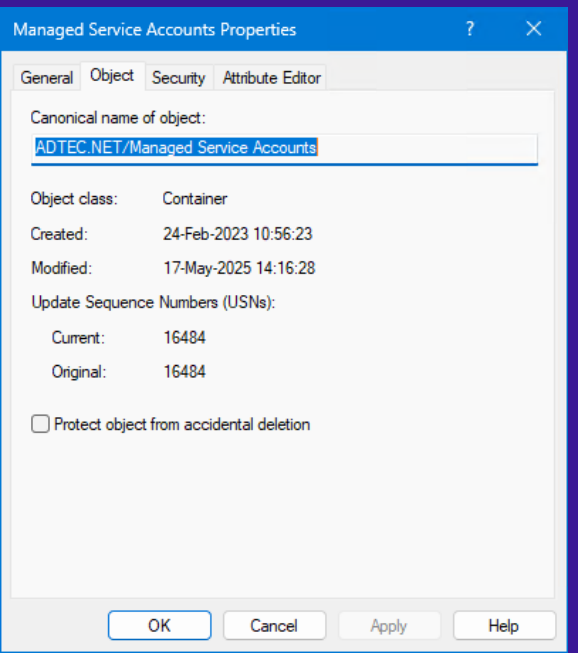


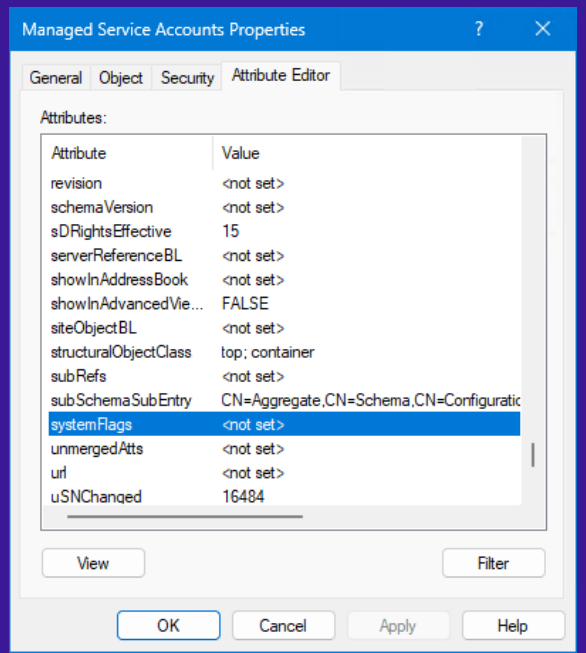
Managed Service Accounts

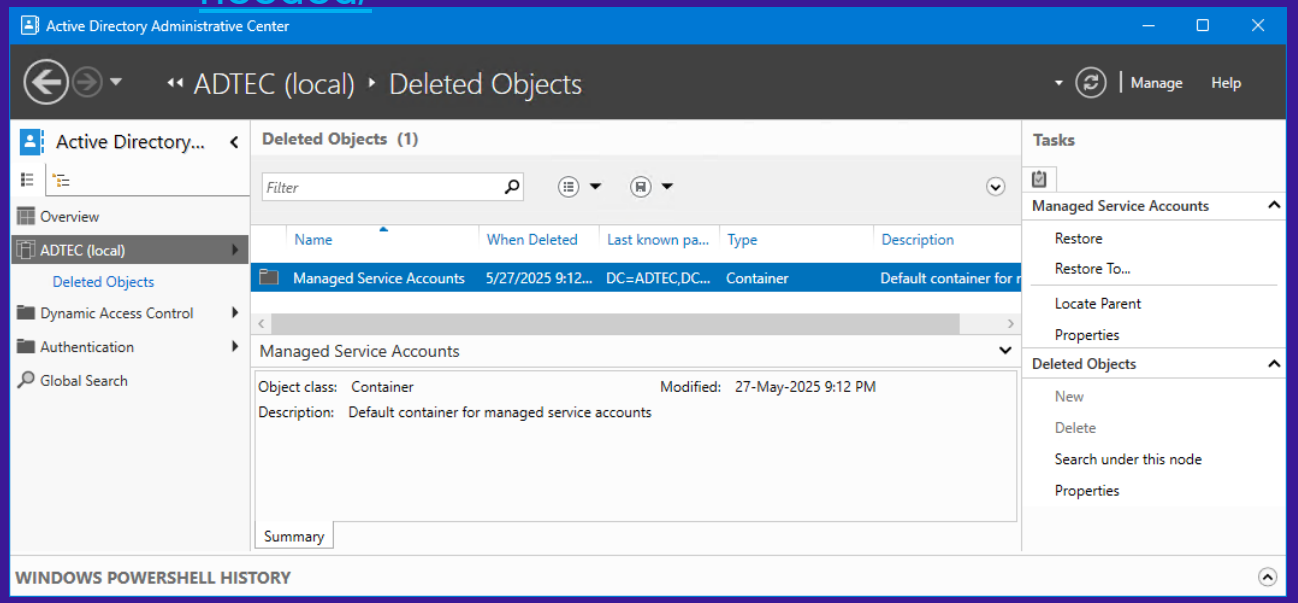
Container For sMSA/gMSA/dMSA

- Default Container in AD for sMSAs/gMSAs/dMSAs:
  "*CN=Managed Service Accounts,DC=<DOMAIN>,DC=<TLD>*"
  (sMSAs/gMSAs/dMSAs can live in ANY other container or OU!)
  - NOT protected, can be deleted!
  - It can be protected from deletion!

https://jorgequestforknowledge.wordpress.com/2025/06/27/well-known-containers-in-an-ad-domain-how-to-restore-and-or-repair-as-needed/







HYBRID IDENTITY PROTECTION conf25

CHARLESTON

ANY QUESTIONS?

THANK YOU!

SCAN ME

Jorge de Almeida Pinto
Senior Incident Response Lead

LinkedIn	http://tiny.cc/JorgeLinkedIn
Blog	http://tiny.cc/JQFKblog
Twitter	http://tiny.cc/JQFKtwitter
Website	https://www.semperis.com/
Blog	https://www.semperis.com/blog/
Podcast	https://hipconf.com/
Contact	jorged@semperis.com


CHARLESTON
HYBRID
IDENTITY
PROTECTION
conf25