

# ADFS mit Entra-MFA koppeln – Adapter, Zertifikat und Fallstricke

## ADFS · Entra-MFA-Adapter

### Zwei Faktoren, ein Zertifikat – und ein Ablaufdatum, das keiner im Kalender hat

*Adapter einrichten, Mandant zuordnen, Zertifikat rechtzeitig erneuern – damit die zweite Abfrage nicht zur letzten wird.*

boddenberg.de



#### WISSEN

Grundlagen, Architektur und alle Praxisbeiträge rund um ADFS an einem Ort.

› [Active Directory Federation Services](#)

#### BERATUNG

MFA an ADFS soll morgen live gehen, und heute weiß noch keiner, wo das Zertifikat liegt? Wir helfen.

› [Consulting zu ADFS](#)

#### SCHULUNG

Adapter einrichten, Zertifikat ablaufen lassen, wiederbeleben – im Labor statt im Echtbetrieb.

› [ADFS-Schulungen](#)

Es gibt diese eine Sorte Störung, die immer gleich anfängt: Montag, 7:40 Uhr, die ersten Anrufe. „Die Anmeldung fragt nach dem Handy, und dann passiert nichts.“ Oder schlimmer: Jeder zweite Versuch klappt, jeder zweite nicht. Irgendwer erinnert sich dunkel, dass vor zwei Jahren ein Dienstleister „das mit der MFA an ADFS“ eingerichtet hat. Dokumentation? Ein Screenshot in einem Ticket, das inzwischen archiviert ist. Willkommen beim integrierten MFA-Adapter für Entra ID – einer Funktion, die ausgesprochen zuverlässig läuft, bis genau zu dem Tag, an dem ihr Zertifikat abläuft.

Dieser Beitrag löst ein konkretes Problem: Du willst deine ADFS-Farm an die Multifaktor-Authentifizierung von Microsoft Entra ID koppeln – früher „Azure MFA“ genannt –, sauber, auf allen Servern, mit einem Plan für die Zertifikatserneuerung. Du erfährst, wie du das Adapter-Zertifikat erzeugst und im Mandanten hinterlegst, wann der Adapter als zusätzliche und wann als primäre Methode taugt, wie das Zusammenspiel mit Conditional Access funktioniert und an welchen Stellen die Sache in der Praxis regelmäßig umkippt. Die Grundlagen zu ADFS setzen wir voraus – Sicherheitsfragen rund um MFA, Golden SAML und Härtung behandelt ausführlich der Beitrag [ADFS-Sicherheit: Angriffe, Härtung, Golden SAML und MFA](#).

#### FAKTEN – Der Entra-MFA-Adapter in Kürze

Eingebaut in ADFS ab Windows Server 2016 – keine separate Installation, kein Agent, kein MFA-Server vor Ort.

Interner Name des Anbieters: AzureMfaAuthentication. In der Verwaltungskonsole heißt er je nach Version „Azure MFA“ oder „Microsoft Entra multifactor authentication“.

Authentifiziert sich bei Entra ID mit einem selbstsignierten Zertifikat pro ADFS-Server, das standardmäßig zwei Jahre gültig ist.

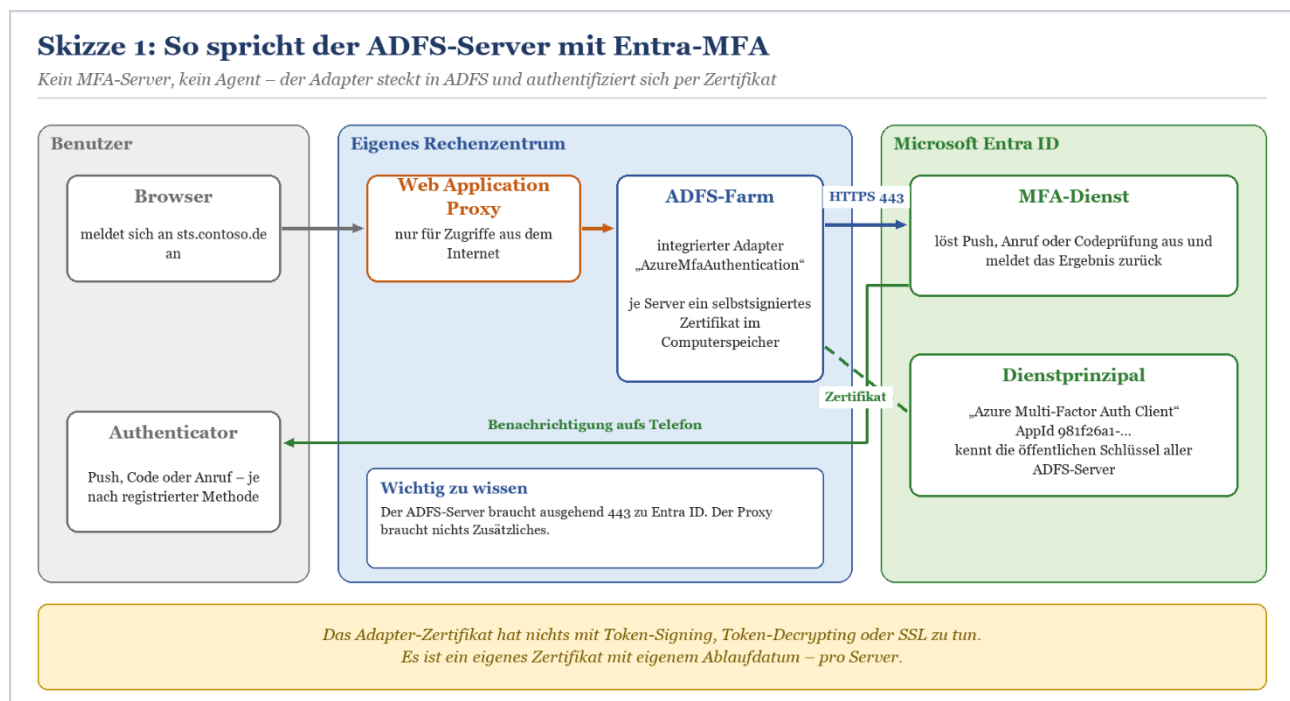
Gegenstelle im Mandanten ist der Dienstprinzipal „Azure Multi-Factor Auth Client“ mit der AppId 981f26a1-7f43-403b-a875-f8b09b8cd720.

Voraussetzung: Die Benutzer haben ihre MFA-Methoden in Entra ID bereits registriert. ADFS kann

diese Registrierung nicht selbst durchführen.

## Wie der Adapter funktioniert – und was er nicht ist

Bis Windows Server 2012 R2 gab es für MFA an ADFS im Microsoft-Umfeld im Wesentlichen einen Weg: den lokalen Azure Multi-Factor Authentication Server mit seinem eigenen ADFS-Adapter. Seit ADFS in Windows Server 2016 ist das anders. Der Adapter für Entra-MFA steckt direkt im Produkt und spricht ohne Umweg mit dem MFA-Dienst in Entra ID. Der ADFS-Server meldet sich dort mit einem Zertifikat an, Entra ID schickt die Benachrichtigung an die Authenticator-App oder löst einen Anruf aus, und das Ergebnis landet wieder bei ADFS. Den Rest – Token ausstellen, Claims setzen, Relying Party bedienen – erledigt ADFS wie gewohnt.



Skizze 1: Der ADFS-Server spricht direkt mit dem MFA-Dienst in Entra ID. Der Schlüssel dazu ist ein Zertifikat pro Server, dessen öffentlicher Teil am Dienstprinzipal hinterlegt ist.

Daraus folgt eine Reihe von Eigenschaften, die du kennen solltest, bevor du loslegst. Erstens: Die Benutzerkonten müssen in Entra ID existieren und synchronisiert sein, und die Benutzer müssen dort ihre Sicherheitsinformationen registriert haben. Zweitens: Jeder ADFS-Server braucht ausgehend HTTPS auf Port 443 zu den Endpunkten des MFA-Dienstes und zum Anmeldeendpunkt von Entra ID. Die genaue Liste führt Microsoft in der Anleitung zur Adapter-Konfiguration. Drittens: Der Web Application Proxy braucht nichts Zusätzliches, er reicht die Anmeldeseite einfach durch. Und viertens: Deine Farm muss mit Entra ID föderiert sein – der Adapter ist für genau diese Hybridwelt gebaut.

## Abgrenzung zum MFA-Server

Wenn du in alten Anleitungen auf den „Azure MFA Server“ stößt: Der ist Geschichte, und zwar belegt. Microsoft hat die Abkündigung im September 2022 angekündigt. Neue Aktivierungen gab es schon seit dem 1. Juli 2019 nur noch für Bestandskunden, die vorher aktiviert hatten. Seit dem 30. September 2024 bearbeiten MFA-Server-Installationen nach Aussage von Microsoft keine MFA-Anfragen mehr – Anmeldungen, die noch daran hängen, scheitern. Für die Migration stellte Microsoft ein Migrationswerkzeug bereit, das die Benutzerdaten in den Clouddienst überträgt.

### WARNUNG – Läuft bei dir noch ein MFA-Server-Adapter in ADFS?

Dann ist das kein Thema für die nächste Planungsrunde, sondern für heute. Prüfe in der ADFS-Verwaltung unter den Authentifizierungsmethoden, welche zusätzlichen Anbieter registriert sind. Taucht dort noch der Adapter des MFA-Servers auf und ist er in Access Control Policies aktiv, verlangst du von deinen Benutzern eine Anmeldung über ein System, das seine Arbeit eingestellt hat. Der integrierte Entra-MFA-Adapter ist der direkte Nachfolger – und er ist in deiner Farm bereits vorhanden.

| Merkmal                        | MFA-Server (abgekündigt)                           | Integrierter Entra-MFA-Adapter                         |
|--------------------------------|----------------------------------------------------|--------------------------------------------------------|
| <b>Installation</b>            | eigene Server vor Ort, eigener ADFS-Adapter        | in ADFS ab Windows Server 2016 enthalten               |
| <b>Benutzerdaten</b>           | lokal in der Datenbank des MFA-Servers             | in Entra ID, Registrierung über das Benutzerkonto      |
| <b>Verbindung zu Microsoft</b> | MFA-Server zum Clouddienst                         | jeder ADFS-Server direkt per Zertifikat                |
| <b>Status</b>                  | bearbeitet seit 30.09.2024 keine MFA-Anfragen mehr | aktueller Weg für MFA an ADFS                          |
| <b>RADIUS, LDAP</b>            | konnte als Proxy dienen                            | nein – dafür gibt es die NPS-Erweiterung für Entra-MFA |

Tabelle 1: MFA-Server und integrierter Adapter im Vergleich

### Zusätzliche oder primäre Methode?

Der Adapter kann zwei Rollen übernehmen. Als zusätzliche Methode kommt er nach der Primäranmeldung – also nach Kennwort oder Windows-integrierter Anmeldung – und liefert den zweiten Faktor. Das ist der Normalfall und der, den du zuerst einrichten solltest. Als primäre Methode ersetzt er die Kennwortabfrage: Der Benutzer gibt seinen Namen ein, bestätigt auf dem Telefon und ist drin. Klingt schick, hat aber einen Haken, den Microsoft selbst klar benennt: Entra-MFA als primäre Methode zählt als ein einziger Faktor. Wer echte MFA will, braucht danach noch einen zweiten Schritt.

| Kriterium                     | Als zusätzliche Methode                                                                | Als primäre Methode                                                         |
|-------------------------------|----------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|
| <b>Ablauf</b>                 | Kennwort oder Windows-Anmeldung, danach Bestätigung per Authenticator, Code oder Anruf | Benutzername, danach Bestätigung – kein Kennwort                            |
| <b>Zählt als</b>              | zweiter Faktor, also echte MFA                                                         | ein Faktor                                                                  |
| <b>Aktivieren</b>             | Reiter für zusätzliche Authentifizierung (in ADFS 2019: „Additional“)                  | Reiter für primäre Authentifizierung, Extranet und/oder Intranet            |
| <b>Besonderheit ADFS 2019</b> | keine                                                                                  | Anker-Claim des Claims Provider Trusts „Active Directory“ auf UPN umstellen |
| <b>Typischer Einsatz</b>      | Extranet-Zugriffe, Microsoft 365, sensible Anwendungen                                 | Kennwort-Spray ins Leere laufen lassen, Vorstufe zu kennwortloser Anmeldung |

Tabelle 2: Die beiden Rollen des Adapters

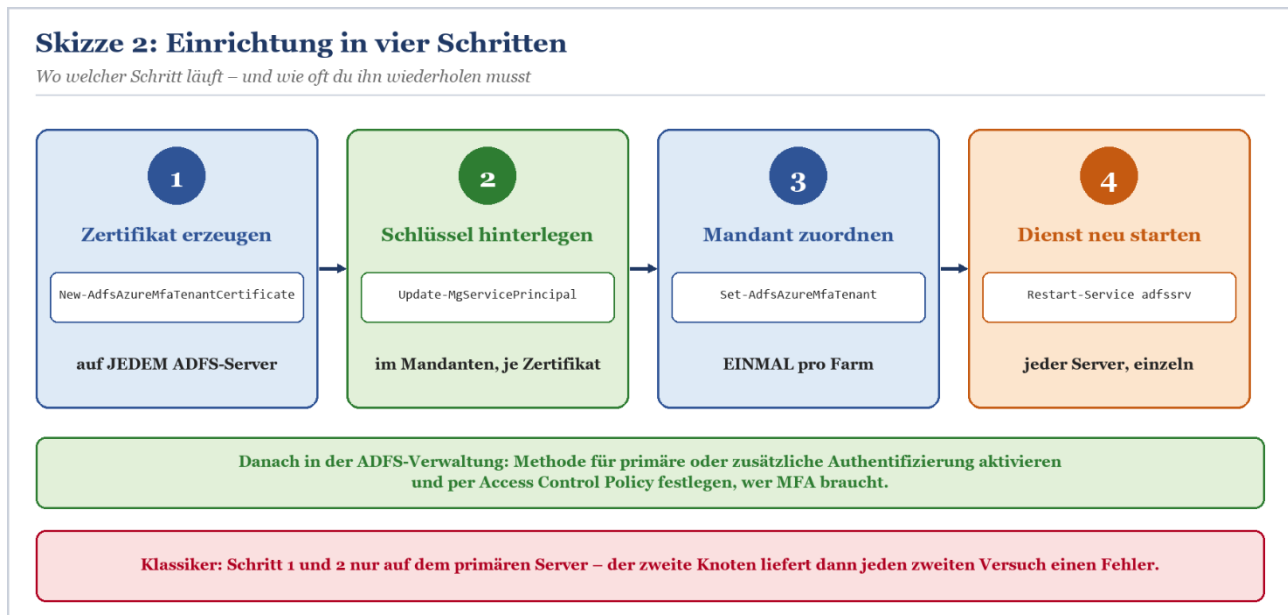
### TIPP – Fang mit der zusätzlichen Methode an

Richte den Adapter zuerst als zusätzliche Methode für eine Testgruppe ein. Erst wenn das stabil läuft, denkst du über die primäre Rolle nach. Andersherum testest du zwei Baustellen gleichzeitig – und weißt bei einer Störung nicht, welche es war.

## Einrichtung: Zertifikat, Mandant, Methode

Die Einrichtung besteht aus vier Schritten, von denen zwei auf jedem einzelnen ADFS-Server laufen müssen. Genau hier entstehen die meisten späteren Störungen: Der erste Server wird

sorgfältig eingerichtet, der zweite „macht ja die Farm-Replikation“. Macht sie nicht. Das Zertifikat liegt im Computerspeicher des jeweiligen Servers, und jeder Server braucht seinen eigenen Schlüssel im Mandanten.



Skizze 2: Zertifikat und Schlüssel gibt es pro Server, die Mandanten-Zuordnung einmal pro Farm.

### WICHTIG – Bevor du anfängst

Du brauchst lokal Administratorrechte auf allen ADFS-Servern und im Mandanten ein Konto mit der Rolle „Anwendungsadministrator“ (Application Administrator), um den Dienstprinzipal zu ändern. Auf dem Rechner, von dem aus du den Mandanten bearbeitest, muss das Modul Microsoft Graph PowerShell installiert sein. Die Mandanten-ID (eine GUID) findest du in der Übersicht des Entra Admin Centers.

Plane für den Neustart des ADFS-Dienstes ein kleines Wartungsfenster ein und nimm jeden Server dafür einzeln aus dem Load Balancer.

## Schritt 1: Adapter-Zertifikat auf jedem Server erzeugen

Auf jedem ADFS-Server erzeugst du mit `New-AdfsAzureMfaTenantCertificate` ein selbstsigniertes Zertifikat. Das Cmdlet legt es im Zertifikatspeicher des lokalen Computers ab und gibt den öffentlichen Teil als Base64-Zeichenfolge zurück. Diese Zeichenfolge brauchst du im nächsten Schritt – also nicht einfach in die Konsole laufen lassen, sondern in einer Variable oder Datei festhalten.

```

# Auf JEDEM ADFS-Server ausführen (Beispiel: adfs01.contoso.local)
$tenantId = "00000000-0000-0000-0000-000000000000" # deine Mandanten-ID
$certBase64 = New-AdfsAzureMfaTenantCertificate -TenantID $tenantId

# Öffentlichen Teil sichern, damit Schritt 2 auch von einem anderen Rechner aus geht
$certBase64 | Out-File "C:\Temp\mfa-$env:COMPUTERNAME.txt" -Encoding ascii

# Kontrolle: Das Zertifikat trägt die Mandanten-ID im Antragsteller
Get-ChildItem Cert:\LocalMachine\My |
    Where-Object { $_.Subject -like "*$tenantId*" } |
    Select-Object Subject, NotBefore, NotAfter, Thumbprint
    
```

Listing 1: Adapter-Zertifikat erzeugen und prüfen

## Schritt 2: Schlüssel am Dienstprinzipal hinterlegen

Jetzt bekommt Entra ID den öffentlichen Schlüssel jedes Servers. Er wird als zusätzliche Anmeldeinformation an den Dienstprinzipal „Azure Multi-Factor Auth Client“ gehängt. Wichtig ist das Wort zusätzlich: Die vorhandenen Schlüssel bleiben stehen, der neue kommt dazu. Wer hier versehentlich die Liste überschreibt, sperrt alle anderen ADFS-Server aus – und vielleicht auch eine zweite Farm, die denselben Mandanten nutzt.

```
Connect-MgGraph -Scopes "Application.ReadWrite.All"

$appId = "981f26a1-7f43-403b-a875-f8b09b8cd720" # Azure Multi-Factor Auth Client
$sp = Get-MgServicePrincipal -Filter "appid eq '$appId'"

$certBase64 = Get-Content "C:\Temp\mfa-ADFS01.txt" -Raw
$certX509 = [System.Security.Cryptography.X509Certificates.X509Certificate2]([System.Convert]::FromBase64String($certBase64))

$newKey = @( @{
    CustomKeyIdentifier = $null
    DisplayName         = $certX509.Subject
    EndDateTime         = $null
    Key                 = $certX509.GetRawCertData()
    KeyId               = [guid]::NewGuid()
    StartDateTime       = $null
    Type                = "AsymmetricX509Cert"
    Usage               = "Verify"
    AdditionalProperties = $null
} )

# Vorhandene Schlüssel behalten, neuen anhängen
$keys = $sp.KeyCredentials + $newKey
Update-MgServicePrincipal -ServicePrincipalId $sp.Id -KeyCredentials $keys
```

*Listing 2: Öffentlichen Schlüssel eines Servers per Microsoft Graph PowerShell hinterlegen – für jeden Server wiederholen*

### FAKTEN — So stand es früher in jeder Anleitung

Bis vor wenigen Jahren erledigte man Schritt 2 mit dem Modul MSOnline: New-MsolServicePrincipalCredential mit der AppId des MFA-Clients, Typ „asymmetric“ und Verwendung „verify“. Die Module MSOnline und AzureAD hat Microsoft ausgemustert. Wenn dein Betriebshandbuch noch so aussieht, ist es Zeit für ein Update – das Skript läuft schlicht nicht mehr.

## Schritt 3: Mandant zuordnen und Dienst neu starten

Die Zuordnung zwischen Farm und Mandant erfolgt einmal pro Farm mit Set-AdfsAzureMfaTenant. Das Cmdlet schreibt die Einstellung in die Farmkonfiguration und weist dich darauf hin, dass der ADFS-Dienst auf jedem Server neu gestartet werden muss. Mach das Server für Server, jeweils außerhalb des Load-Balancer-Pools, und warte, bis die laufenden Verbindungen abgearbeitet sind. Für Mandanten in der US-Government-Cloud gibt es zusätzlich den Parameter -Environment – für die allermeisten Leser hier spielt er keine Rolle.

```
# Einmal pro Farm, auf dem primären Server (bei WID) bzw. einem beliebigen Knoten (bei SQL)
Set-AdfsAzureMfaTenant -TenantId $tenantId -ClientId 981f26a1-7f43-403b-a875-f8b09b8cd720

# Danach auf JEDEM Server, nacheinander
Restart-Service adfssrv

# Kontrolle: Ist der Anbieter registriert?
Get-AdfsAuthenticationProvider -Name AzureMfaAuthentication
```

*Listing 3: Mandanten-Zuordnung und Neustart*

## Schritt 4: Methode aktivieren und Zugriff regeln

Jetzt ist der Adapter verfügbar, aber noch nicht im Einsatz. In der ADFS-Verwaltung aktivierst du ihn unter den Authentifizierungsmethoden – für die zusätzliche Authentifizierung auf dem entsprechenden Reiter, für die primäre Rolle getrennt für Extranet und Intranet. Per PowerShell geht das über die globale Authentifizierungsrichtlinie. Achtung: Die Parameter erwarten die vollständige Liste der Anbieter. Wer nur AzureMfaAuthentication übergibt, wirft gegebenenfalls andere zusätzliche Anbieter hinaus.

```
# Aktuelle Einstellungen ansehen
Get-AdfsGlobalAuthenticationPolicy

# Entra-MFA als zusätzliche Methode freischalten (bestehende Anbieter mitnehmen!)
$pol = Get-AdfsGlobalAuthenticationPolicy
$add = @($pol.AdditionalAuthenticationProvider) + "AzureMfaAuthentication" | Select-Object -
Unique
Set-AdfsGlobalAuthenticationPolicy -AdditionalAuthenticationProvider $add

# Optional: als primäre Methode im Extranet zulassen
Set-AdfsGlobalAuthenticationPolicy -AllowAdditionalAuthenticationAsPrimary $true
Set-AdfsGlobalAuthenticationPolicy -PrimaryExtranetAuthenticationProvider
@("FormsAuthentication", "AzureMfaAuthentication")
```

*Listing 4: Methode in der globalen Authentifizierungsrichtlinie aktivieren*

Wer MFA wann leisten muss, regelst du anschließend pro Relying Party über Access Control Policies – etwa mit der mitgelieferten Richtlinie, die allen Zugriff erlaubt und MFA verlangt, oder mit einer eigenen, die MFA nur für Zugriffe aus dem Extranet oder für bestimmte Gruppen fordert. Wie du solche Richtlinien baust, ohne in Claim-Rule-Syntax zu versinken, zeigt der Beitrag [Access Control Policies in ADFS – Zugriffsregeln ohne Claim-Rule-Akrobatik](#).

### WARNUNG – ADFS 2019 und die primäre Rolle

Willst du Entra-MFA in ADFS 2019 als primäre Methode nutzen, verlangt Microsoft eine Änderung am Claims Provider Trust „Active Directory“: Der Anker-Claim wird von windowsaccountname auf UPN umgestellt (Set-AdfsClaimsProviderTrust mit -AnchorClaimType und -TargetName "Active Directory"). Laut Microsoft hat das keine Auswirkung auf die interne Funktion der Farm, einzelne Benutzer müssen sich aber eventuell einmal neu anmelden. Mach es trotzdem nicht freitags um 16 Uhr.

## Das Zertifikat erneuern – bevor es dich erneuert

Hier kommt der Teil, der in den meisten Umgebungen fehlt. Das Adapter-Zertifikat ist standardmäßig zwei Jahre gültig. Es erneuert sich nicht selbst, es meldet sich nicht im Kalender, und es taucht in keiner der üblichen Zertifikatslisten auf, die du für Token-Signing oder SSL pflegst. Zwei Jahre sind genau lang genug, damit alle Beteiligten vergessen haben, dass es existiert – und genau kurz genug, damit es noch in der Amtszeit derselben IT-Leitung zuschlägt.

| Zertifikat                              | Wer stellt aus?                          | Erneuerung                                            | Beitrag                                                                                       |
|-----------------------------------------|------------------------------------------|-------------------------------------------------------|-----------------------------------------------------------------------------------------------|
| <b>Entra-MFA-Adapter</b>                | ADFS selbst, selbstsigniert, pro Server  | manuell mit -Renew, Schlüssel im Mandanten nachtragen | dieser Beitrag                                                                                |
| <b>Token-Signing / Token-Decrypting</b> | ADFS selbst oder deine PKI               | AutoCertificateRollover oder manuell                  | <a href="#">ADFS Token-Signing-Zertifikat erneuern – mit und ohne AutoCertificateRollover</a> |
| <b>SSL / Service Communications</b>     | deine PKI oder ein öffentlicher Anbieter | manuell, auf ADFS und WAP                             | <a href="#">SSL-Zertifikat auf ADFS und Web Application Proxy tauschen</a>                    |

|                            |                    |                                    |                                                                                                         |
|----------------------------|--------------------|------------------------------------|---------------------------------------------------------------------------------------------------------|
| <b>Proxy-Trust des WAP</b> | ADFS für den Proxy | automatisch, solange der WAP läuft | <a href="#">WAP-Proxy-Vertrauen erneuern – wenn der Web Application Proxy plötzlich nicht mehr will</a> |
|----------------------------|--------------------|------------------------------------|---------------------------------------------------------------------------------------------------------|

Tabelle 3: Vier Zertifikate, vier Ablaufdaten – das Adapter-Zertifikat ist das, das niemand auf dem Zettel hat

## Ablaufdatum prüfen

Das Adapter-Zertifikat findest du im Speicher „Eigene Zertifikate“ des lokalen Computers. Es ist selbstsigniert und trägt die Mandanten-ID im Antragsteller. Prüfe es auf allen Servern in einem Rutsch und sieh dir gleich an, welche Schlüssel am Dienstprinzipal hinterlegt sind und wann sie ablaufen.

```
# Ablaufdaten auf allen Farm-Servern
Invoke-Command -ComputerName adfs01.contoso.local, adfs02.contoso.local -ScriptBlock {
    param($t)
    Get-ChildItem Cert:\LocalMachine\My |
        Where-Object { $_.Subject -like "*$t*" } |
        Select-Object @{n="Server";e={$env:COMPUTERNAME}}, NotBefore, NotAfter, Thumbprint
} -ArgumentList $tenantId

# Gegenprobe im Mandanten
(Get-MgServicePrincipal -Filter "appid eq '981f26a1-7f43-403b-a875-f8b09b8cd720'" ).KeyCredentials |
    Select-Object DisplayName, StartDateTime, EndDateTime, KeyId | Sort-Object EndDateTime
```

Listing 5: Ablaufdaten der Adapter-Zertifikate und der hinterlegten Schlüssel

### TIPP – Ins Monitoring damit

Ein Zertifikat, das alle zwei Jahre abläuft, gehört nicht in den Kopf eines Admins, sondern in eine Überwachung, die 60 Tage vorher Alarm schlägt. Listing 5 lässt sich als geplante Aufgabe oder Prüfskript in jedes Monitoring einbauen. Wie du ADFS insgesamt überwachst, steht im Beitrag [https://www.boddenberg.de/adfs-monitoring-health-check/ADFS überwachen – Health Checks, Diagnose-Toolbox und Entra Connect Health](https://www.boddenberg.de/adfs-monitoring-health-check/ADFS%20überwachen%20–%20Health%20Checks,%20Diagnose-Toolbox%20und%20Entra%20Connect%20Health)

## Rechtzeitig erneuern: der entspannte Weg

Solange das alte Zertifikat noch gültig ist, verläuft die Erneuerung ohne Ausfall. Auf jedem Server rufst du New-AdfsAzureMfaTenantCertificate mit dem Parameter -Renew \$true auf. Dabei entsteht ein zweites Zertifikat, das alte bleibt liegen. Der Clou: Laut Microsoft ist das neue Zertifikat erst ab zwei Tagen nach dem Erstellungsdatum gültig, und zwar für zwei Jahre plus zwei Tage. Diese zwei Tage sind deine Frist, um den neuen Schlüssel am Dienstprinzipal nachzutragen – exakt wie in Listing 2.

```
# Auf JEDEM Server, solange das alte Zertifikat noch gültig ist
$newCert = New-AdfsAzureMfaTenantCertificate -TenantId $tenantId -Renew $true
$newCert | Out-File "C:\Temp\mfa-$env:COMPUTERNAME-neu.txt" -Encoding ascii

# Danach: Schlüssel wie in Listing 2 am Dienstprinzipal anhängen. Kein Neustart nötig.
```

Listing 6: Erneuerung vor Ablauf

Sobald das neue Zertifikat gültig wird, wechselt ADFS von selbst – Microsoft spricht von einigen Stunden bis zu einem Tag. Den Wechsel dokumentiert jeder Server im Admin-Protokoll von ADFS mit dem Ereignis 547: „The tenant certificate for Azure MFA has been renewed“, inklusive alter und neuer Fingerabdrücke und Ablaufdaten. Dieses Ereignis ist dein Beleg, dass alles geklappt hat. Wie du das Protokoll gezielt auswertest, beschreibt [ADFS-Ereignisprotokolle lesen – Admin-Log, Debug-Tracing und Auditing](#).

### Skizze 3: Zertifikat erneuern, bevor es abläuft

Mit `-Renew $true` entsteht ein Nachfolger, der erst zwei Tage später gültig wird – Zeit genug für Schritt 2



#### Rechtzeitig erneuert

Kein Neustart, kein Ausfall. Das alte Zertifikat bleibt liegen, bis der Nachfolger übernimmt. Auf jedem Server einzeln – und für jeden Server einen eigenen Schlüssel im Mandanten.

#### Schon abgelaufen

OHNE `-Renew` ausführen: Das abgelaufene Zertifikat wird ersetzt. Schlüssel im Mandanten eintragen, danach ADFS-Dienst neu starten. Bis dahin scheitert jede MFA über diesen Server.

Skizze 3: Die zwei Tage Vorlauf sind Absicht – genug Zeit, den neuen Schlüssel im Mandanten einzutragen.

## Schon abgelaufen: der hektische Weg

Ist das Zertifikat bereits abgelaufen, scheitert jede MFA-Anfrage über den betroffenen Server. Jetzt verwendest du `New-AdfsAzureMfaTenantCertificate` ausdrücklich ohne `-Renew`. In diesem Fall ersetzt das Cmdlet das abgelaufene Zertifikat durch ein neues, das sofort gilt. Den Schlüssel trägst du wie gewohnt im Mandanten nach, und anschließend startest du den ADFS-Dienst neu, damit er das neue Zertifikat aufnimmt. Danach geht es weiter – mit einem gewissen Erklärungsbedarf in der nächsten Teamrunde.

### WARNUNG – `-Renew` bei abgelaufenem Zertifikat ist ein Eigentor

Microsoft warnt ausdrücklich: Bei einem schon abgelaufenen Zertifikat nicht `-Renew $true` verwenden. Sonst bleibt das abgelaufene Zertifikat aktiv, und das neue wird erst in zwei Tagen gültig. Zwei Tage ohne MFA über diesen Server sind keine Option, die jemand gern im Störungsbericht stehen hat.

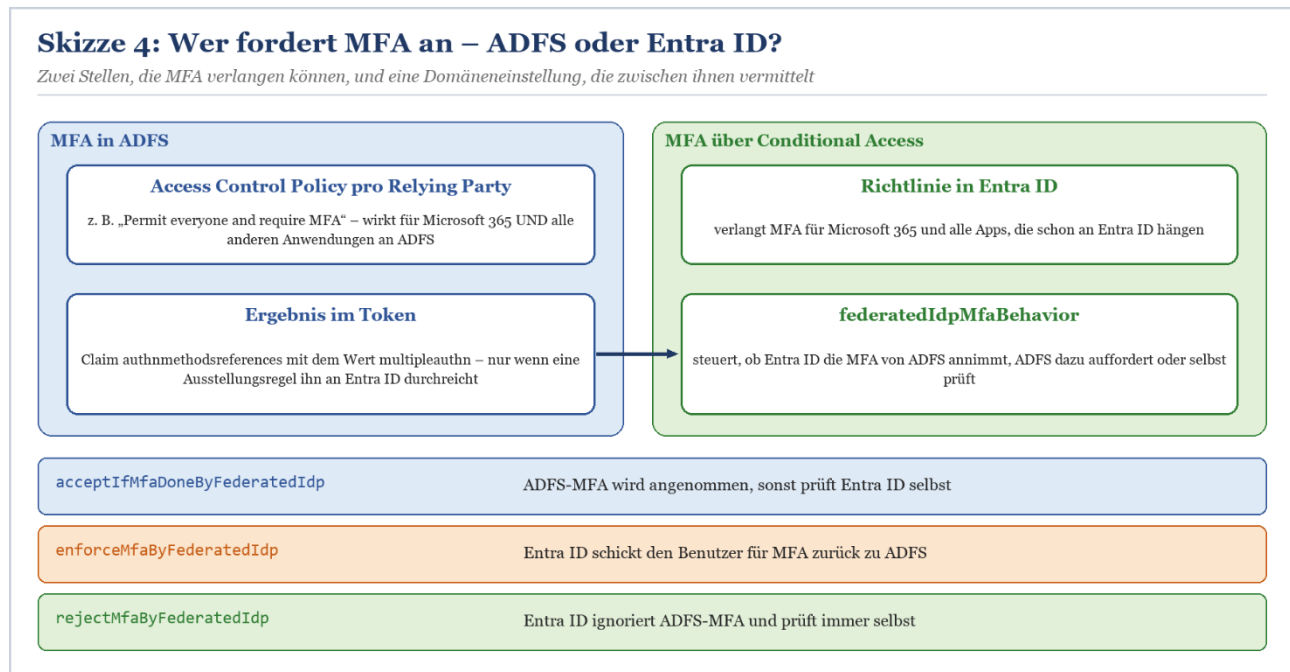
| Situation                    | Befehl                                                                  | Schlüssel im Mandanten       | Neustart ADFS            |
|------------------------------|-------------------------------------------------------------------------|------------------------------|--------------------------|
| Altes Zertifikat noch gültig | <code>New-AdfsAzureMfaTenantCertificate -Renew \$true</code>            | ja, innerhalb von zwei Tagen | nein                     |
| Altes Zertifikat abgelaufen  | <code>New-AdfsAzureMfaTenantCertificate</code> ohne <code>-Renew</code> | ja, sofort                   | ja                       |
| Neuer Server in der Farm     | <code>New-AdfsAzureMfaTenantCertificate</code>                          | ja, für diesen Server        | ja, auf dem neuen Server |

Tabelle 4: Welcher Befehl in welcher Lage

Den letzten Fall unterschätzen viele: Wer die Farm um einen Knoten erweitert oder im Zuge eines Upgrades neue Server mit neuerer Windows-Version hinzunimmt, muss Schritt 1 und 2 auf jedem neuen Server wiederholen. Die Farmkonfiguration kennt zwar den Mandanten, aber das Zertifikat liegt lokal. Gehört deshalb auf die Checkliste in [ADFS-Farm upgraden – Farm Behavior Level anheben ohne Ausfall](#) und in [ADFS-Farm installieren – Schritt für Schritt auf Windows Server 2022](#).

# MFA in ADFS oder Conditional Access in Entra ID?

Spätestens bei Microsoft 365 stellt sich die Frage, wer eigentlich MFA verlangt. Es gibt zwei Stellen, die das können: ADFS über Access Control Policies und Entra ID über Conditional Access. Beides gleichzeitig ohne Abstimmung führt zu einem bekannten Bild – der Benutzer bestätigt auf dem Telefon, und kurz danach bestätigt er noch einmal. Oder Entra ID verlangt MFA, ADFS liefert sie auch, aber Entra ID erfährt nichts davon und fragt trotzdem nach.



Skizze 4: ADFS und Entra ID können beide MFA verlangen. Die Domäneneinstellung *federatedIdpMfaBehavior* entscheidet, ob Entra ID die MFA von ADFS anerkennt.

## Der direkte Vergleich

| Kriterium              | MFA in ADFS (Adapter + Access Control Policy)                                               | Conditional Access in Entra ID                                                                                |
|------------------------|---------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|
| Wo konfiguriert        | ADFS-Verwaltung bzw. ADFS-PowerShell, pro Relying Party                                     | Entra Admin Center, Richtlinien für Benutzer, Apps, Bedingungen                                               |
| Reichweite             | alle Anwendungen, die an ADFS hängen – auch lokale SAML- und WS-Fed-Apps                    | alle Anwendungen, die an Entra ID hängen – Microsoft 365 und dort angebundene Apps                            |
| Bedingungen            | Gruppen, Intranet oder Extranet, Geräteinformationen, Claims                                | Benutzer und Gruppen, Standorte, Gerätezustand, Anmelderisiko, Authentifizierungsstärke (abhängig von Lizenz) |
| Methoden               | die vom Benutzer registrierten Methoden wie Authenticator-Benachrichtigung, Code oder Anruf | alle in der Methodenrichtlinie erlaubten Methoden, auch phishingresistente wie FIDO2                          |
| Protokolle             | ADFS-Admin-Log, Auditing auf den ADFS-Servern                                               | Anmeldeprotokolle in Entra ID                                                                                 |
| Abhängigkeiten         | Adapter-Zertifikat pro Server, ausgehende Verbindung, Registrierung in Entra ID             | Lizenz für Conditional Access (Entra ID P1 oder höher)                                                        |
| Beim Ausstieg aus ADFS | verschwindet mit der Farm                                                                   | bleibt bestehen und übernimmt                                                                                 |

Tabelle 5: MFA in ADFS und Conditional Access in Entra ID im Vergleich

Die pragmatische Faustregel aus vielen Projekten: Für Microsoft 365 und alles, was ohnehin an Entra ID hängt, ist Conditional Access die bessere Wahl – mehr Signale, bessere Protokolle, und es

bleibt, wenn ADFS irgendwann geht. Den ADFS-Adapter brauchst du für die Anwendungen, die (noch) direkt an ADFS hängen und MFA verlangen sollen. Für beide Welten gilt: Eine Stelle verlangt, die andere erkennt an. Wie du bestehende Claim Rules und Richtlinien in Conditional Access überführst, beschreibt [Claim Rules nach Conditional Access übersetzen](#); mehr zu Entra ID selbst findest du auf der Seite [Microsoft Entra ID](#).

## Entra ID die ADFS-MFA anerkennen lassen

Damit Entra ID weiß, dass ADFS bereits MFA geleistet hat, muss das Token für Microsoft 365 einen Nachweis enthalten: den Claim `authnmethodsreferences` mit dem Wert `multipleauthn`. ADFS setzt ihn nach erfolgreicher MFA, gibt ihn aber nur weiter, wenn eine Ausstellungsregel am Relying Party Trust ihn durchreicht. Prüfe also zuerst, ob die Regel schon existiert – in vielen über Entra Connect eingerichteten Umgebungen ist das der Fall. Mehr zur Regelsprache steht im [ADFS Claim Rules Leitaden](#).

```
$rp = Get-AdfsRelyingPartyTrust -Name "Microsoft Office 365 Identity Platform"
$rp.IssuanceTransformRules | Select-String "authnmethodsreferences"

# Nur falls nicht vorhanden: Regel anhängen
$rule = @'
@RuleName = "MFA-Nachweis an Entra ID"
c:[Type == "http://schemas.microsoft.com/claims/authnmethodsreferences"]
=> issue(claim = c);
'@
Set-AdfsRelyingPartyTrust -TargetName $rp.Name -IssuanceTransformRules
($rp.IssuanceTransformRules + $rule)
```

Listing 7: Claim `authnmethodsreferences` an Entra ID weiterreichen

Die zweite Hälfte ist die Domäneneinstellung `federatedIdpMfaBehavior` in Entra ID. Sie legt fest, was Entra ID mit der MFA des föderierten Identitätsanbieters anfängt. Früher übernahm das die Einstellung `SupportsMfa`, die man mit MSOnline setzte – heute ist `federatedIdpMfaBehavior` der Weg, und du bearbeitest sie mit Microsoft Graph PowerShell.

| Wert                                 | Verhalten                                                                    | Passt, wenn ...                                                                             |
|--------------------------------------|------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|
| <b>acceptIfMfaDoneByFederatedIdp</b> | Entra ID akzeptiert MFA von ADFS. Fehlt der Nachweis, prüft Entra ID selbst. | du MFA in ADFS und Conditional Access parallel nutzt und doppelte Abfragen vermeiden willst |
| <b>enforceMfaByFederatedIdp</b>      | Entra ID schickt den Benutzer zur MFA zurück an ADFS.                        | MFA bewusst zentral in ADFS stattfinden soll                                                |
| <b>rejectMfaByFederatedIdp</b>       | Entra ID ignoriert die MFA von ADFS und führt MFA immer selbst durch.        | du MFA vollständig nach Entra ID verlagerst, etwa als Vorbereitung auf den Ausstieg         |

Tabelle 6: Die drei Werte von `federatedIdpMfaBehavior`

```
Connect-MgGraph -Scopes "Domain.ReadWrite.All"

$fed = Get-MgDomainFederationConfiguration -DomainId "contoso.de"
$fed | Select-Object Id, IssuerUri, FederatedIdpMfaBehavior

Update-MgDomainFederationConfiguration -DomainId "contoso.de" `
  -InternalDomainFederationId $fed.Id `
  -FederatedIdpMfaBehavior "acceptIfMfaDoneByFederatedIdp"
```

Listing 8: `federatedIdpMfaBehavior` anzeigen und setzen

### WICHTIG – Änderungen brauchen Zeit

Microsoft weist darauf hin, dass eine Änderung der Föderationseinstellungen der Domäne bis zu 24 Stunden brauchen kann, bis sie überall wirkt. Teste also nicht fünf Minuten nach der Änderung und baue dann alles wieder zurück.

Und wenn die Reise ohnehin Richtung Entra ID geht: Mit Staged Rollout kannst du einzelne Gruppen schon auf Cloud-Authentifizierung umstellen, während die Föderation für alle anderen weiterläuft. Wie das geht, steht in [Microsoft 365 von Federated auf Managed umstellen – mit Staged Rollout](#).

## Die Fallstricke aus der Praxis

Die folgenden Fehlerbilder begegnen uns in Projekten immer wieder. Sie sind selten exotisch, aber zuverlässig dann akut, wenn gerade niemand Zeit hat.

| Fehlerbild                                                             | Wahrscheinliche Ursache                                                            | Lösung                                                                   |
|------------------------------------------------------------------------|------------------------------------------------------------------------------------|--------------------------------------------------------------------------|
| <b>Jeder zweite MFA-Versuch scheitert</b>                              | Zertifikat oder Schlüssel nur für einen Teil der Server eingerichtet               | Schritt 1 und 2 auf jedem Server prüfen (Listing 5)                      |
| <b>Plötzlich scheitert MFA auf allen Servern</b>                       | Adapter-Zertifikate abgelaufen – oft alle gleichzeitig, weil am selben Tag erzeugt | Neu erzeugen ohne -Renew, Schlüssel hinterlegen, Dienst neu starten      |
| <b>„No valid strong authentication method found“</b>                   | Methode nicht auf dem Reiter für zusätzliche Authentifizierung aktiviert           | Methode aktivieren (Listing 4)                                           |
| <b>Fehlermeldung, die Methode sei für den Benutzer nicht verfügbar</b> | Benutzer hat keine MFA-Methode in Entra ID registriert                             | Registrierung anstoßen, Fehlerseite anpassen                             |
| <b>Benutzer muss zweimal bestätigen</b>                                | ADFS und Conditional Access verlangen beide MFA, Nachweis kommt nicht an           | Claim authnmethodsreferences weitergeben, federatedIdpMfaBehavior prüfen |
| <b>Adapter reagiert gar nicht, Zeitüberschreitung</b>                  | ausgehende Verbindung vom ADFS-Server blockiert, Proxy ohne Ausnahme               | Firewall und Proxy-Konfiguration für die Microsoft-Endpunkte prüfen      |
| <b>Altes Betriebssystem bricht ab</b>                                  | Skript nutzt MSOnline oder AzureAD                                                 | auf Microsoft Graph PowerShell umstellen (Listing 2)                     |

Tabelle 7: Typische Fehlerbilder und ihre Lösung

## Unregistrierte Benutzer und die Fehlerseite

ADFS kann die Registrierung von MFA-Methoden nicht selbst anbieten. Ein Benutzer, der in Entra ID noch keine Methode hinterlegt hat, landet bei ADFS auf einer nüchternen Fehlerseite. Microsoft schlägt vor, die Seite über die Datei onload.js eines eigenen Web-Themes so anzupassen, dass sie den Fehlertext erkennt und den Benutzer zur Registrierungsseite für Sicherheitsinformationen weiterleitet. Das funktioniert, ist aber nur die zweitbeste Lösung. Die beste ist eine Einführungskampagne, bei der alle Benutzer ihre Methoden registriert haben, bevor die erste Access Control Policy MFA verlangt. Wie du die Anmeldeseite anpasst, zeigt [ADFS-Anmeldeseite anpassen – Logo, Texte, Hilfelinks](#).

## Alle Zertifikate am selben Tag erzeugt

Klingt harmlos, ist aber die Wurzel des Totalausfalls: Wer die Einrichtung an einem Nachmittag auf allen Servern durchzieht, bekommt Zertifikate, die alle am selben Nachmittag zwei Jahre später ablaufen. Redundanz über mehrere Server hilft dann nicht. Das ist kein Grund, die Einrichtung künstlich zu strecken – aber ein sehr guter Grund, die Erneuerung fest einzuplanen und zu überwachen.

## Primäre MFA und Kennwort-Spray

Entra-MFA als primäre Methode im Extranet nimmt Angreifern, die Kennwörter durchprobieren, die Grundlage – ohne Kennwortabfrage gibt es nichts zu raten. Die Kehrseite: Jeder Anmeldeversuch löst eine Benachrichtigung auf dem Telefon des Opfers aus, und ermüdete Benutzer neigen dazu, irgendwann zu bestätigen. Halte deine Farm deshalb aktuell, schule die Benutzer und kombiniere die primäre Rolle mit einem zweiten Faktor. Für den klassischen Kennwortweg bleibt der Schutz durch [Extranet Smart Lockout – ADFS gegen Passwort-Spray schützen](#) Pflicht.

### TIPP – Fehlersuche mit System

Fang beim Admin-Log von ADFS auf dem Server an, der die Anfrage bearbeitet hat – bei mehreren Knoten hinter dem Load Balancer also auf allen. Danach prüfst du Zertifikat, Schlüssel im Mandanten und ausgehende Verbindung, erst dann die Benutzerregistrierung. Die allgemeine Checkliste für gescheiterte Anmeldungen findest du in [https://www.boddenberg.de/adfs-anmeldung-fehlgeschlagen/ADFS-Anmeldung schlägt fehl](https://www.boddenberg.de/adfs-anmeldung-fehlgeschlagen/ADFS-Anmeldung-schlägt-fehl) – die häufigsten Ursachen und ihre Lösung

### WEITER – Passende Beiträge rund um MFA und ADFS

- › [Access Control Policies in ADFS – Zugriffsregeln ohne Claim-Rule-Akrobatik](#) – wer wann MFA leisten muss
- › [Claim Rules nach Conditional Access übersetzen](#) – der Weg von ADFS-Regeln zu Conditional Access
- › [ADFS-Sicherheit: Angriffe, Härtung, Golden SAML und MFA](#) – MFA im Gesamtbild der ADFS-Härtung

## Fazit: Ein Adapter, der hält – wenn du ihn pflegst

Der integrierte Entra-MFA-Adapter ist eine der unkompliziertesten Erweiterungen, die ADFS zu bieten hat. Kein zusätzlicher Server, kein Agent, keine eigene Datenbank – ein Zertifikat pro Server, ein Schlüssel im Mandanten, eine Zuordnung pro Farm. Der abgekündigte MFA-Server ist damit endgültig überflüssig, und wer ihn noch irgendwo in einer Access Control Policy stehen hat, sollte das heute ändern.

Die eigentliche Arbeit ist nicht die Einrichtung, sondern das, was danach kommt: die Erneuerung alle zwei Jahre auf jedem Server, die Abstimmung mit Conditional Access, damit Benutzer nicht doppelt bestätigen, und die Klarheit, welche Anwendung ihre MFA wo bekommt. ADFS läuft in vielen Unternehmen seit über zehn Jahren zuverlässig, und mit dieser Pflege bleibt das so – bis du den Wechsel zu Entra ID in Ruhe geplant hast. Wenn du dabei Unterstützung möchtest, findest du sie im [Consulting zu ADFS \(Active Directory Federation Services\)](#). Den Adapter, Access Control Policies und die Zertifikatspflege vertieft außerdem das Buch – mehr dazu unter [ADFS in der Praxis – das Buch im Detail](#).

## FAQ: ADFS mit dem Entra-MFA-Adapter

### Muss ich für den Azure-MFA-Adapter in ADFS etwas installieren?

Nein. Der Adapter ist ab Windows Server 2016 in ADFS enthalten. Du erzeugst nur das Zertifikat, hinterlegst den Schlüssel im Mandanten und ordnest die Farm dem Mandanten zu.

## **Auf welchen Servern muss ich New-AdfsAzureMfaTenantCertificate ausführen?**

Auf jedem ADFS-Server der Farm. Jeder Server bekommt sein eigenes Zertifikat, und für jeden Server muss der öffentliche Schlüssel am Dienstprinzipal hinterlegt sein. Set-AdfsAzureMfaTenant läuft dagegen nur einmal pro Farm.

## **Wie lange ist das Adapter-Zertifikat gültig?**

Standardmäßig zwei Jahre. Bei einer Erneuerung mit -Renew \$true wird das neue Zertifikat erst zwei Tage nach dem Erstellen gültig und läuft dann zwei Jahre plus zwei Tage.

## **Muss ich ADFS nach der Zertifikatserneuerung neu starten?**

Nur wenn das alte Zertifikat schon abgelaufen war. Bei rechtzeitiger Erneuerung wechselt ADFS von selbst und protokolliert das mit Ereignis 547 im Admin-Log.

## **Was passiert, wenn ich -Renew bei einem abgelaufenen Zertifikat verwende?**

Dann bleibt das abgelaufene Zertifikat in Gebrauch, und das neue wird erst nach zwei Tagen gültig. Bei abgelaufenem Zertifikat führst du das Cmdlet deshalb ohne -Renew aus.

## **Kann ich den Schlüssel noch mit MSOnline hinterlegen?**

Nein, MSOnline und AzureAD sind ausgemustert. Verwende Microsoft Graph PowerShell mit Get-MgServicePrincipal und Update-MgServicePrincipal.

## **Funktioniert der Adapter noch mit dem alten MFA-Server?**

Der Adapter braucht keinen MFA-Server. Der MFA-Server selbst bearbeitet laut Microsoft seit dem 30. September 2024 keine MFA-Anfragen mehr.

## **Warum muss der Benutzer bei Microsoft 365 zweimal MFA bestätigen?**

Weil Entra ID den MFA-Nachweis von ADFS nicht erhält oder nicht anerkennt. Gib den Claim authnmethodsreferences weiter und setze federatedIdpMfaBehavior passend, etwa auf acceptIfMfaDoneByFederatedIdp.

## **Kann ich Entra-MFA in ADFS als primäre Methode ohne Kennwort nutzen?**

Ja, ADFS kann den Adapter als primäre Methode für Extranet und Intranet verwenden. Microsoft wertet das aber als einen Faktor. In ADFS 2019 musst du dafür zusätzlich den Anker-Claim auf UPN umstellen.

## **Was sieht ein Benutzer ohne registrierte MFA-Methode?**

Eine ADFS-Fehlerseite. ADFS kann die Registrierung nicht selbst anbieten. Du kannst die Fehlerseite über onload.js so anpassen, dass sie zur Registrierung weiterleitet.