

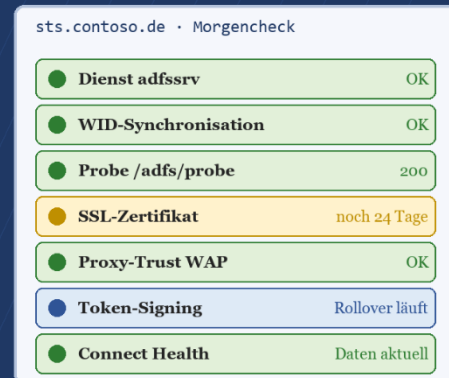
ADFS überwachen – Health Checks, Diagnose-Toolbox und Entra Connect Health

ADFS · Monitoring und Health Check

ADFS stirbt leise – sorg dafür, dass du es vorher hörst

Zertifikate, Dienste, Farm-Synchronisation, WAP-Trust, Probe-Endpoint – plus ADFS Toolbox und Entra Connect Health.

boddenberg.de



WISSEN

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

> [Active Directory Federation Services](#)

BERATUNG

Deine Farm läuft, aber keiner weiß so genau, wie lange noch? Wir richten mit dir ein Monitoring ein, das rechtzeitig Bescheid sagt.

> [Consulting zu ADFS](#)

SCHULUNG

Zertifikate ablaufen lassen, Sync abwürgen, Proxy-Trust zerlegen – im Labor, wo es niemandem wehtut.

> [ADFS-Schulungen](#)

ADFS hat eine Eigenschaft, die man bei Haustieren schätzt und bei Infrastruktur fürchtet: Es leidet still. Jahrelang stellt die Farm brav Tokens aus, niemand schaut hin, niemand bedankt sich. Und dann, an einem Samstag um 3:12 Uhr, läuft das SSL-Zertifikat ab, das vor zwei Jahren ein Kollege bestellt hat, der inzwischen bei der Konkurrenz Kaffee trinkt. Montagmorgen kommt niemand mehr in Microsoft 365, die Hotline hat ihr bestes Wochenende seit Jahren hinter sich und die schlechteste Woche vor sich. Im Nachgang stellt sich heraus: Das Ereignisprotokoll hatte wochenlang gewarnt. Es hat nur niemand gelesen.

Wer „adfs monitoring health check“ sucht, will genau das verhindern. Dieser Beitrag ist kein weiterer Überblick über ADFS – die Grundlagen stehen auf der Übersichtsseite [Active Directory Federation Services](#). Hier geht es um die konkrete Frage: Was musst du regelmäßig prüfen, wie oft und mit welchem Werkzeug? Dazu gehören Zertifikatslaufzeiten, Dienststatus, die Synchronisation der Farm, das Vertrauen zwischen Web Application Proxy und Farm sowie der Probe-Endpoint. Danach schauen wir uns an, was die ADFS Toolbox mit dem Diagnostics Analyzer wirklich kann, was Entra Connect Health for ADFS liefert – und am Ende bekommst du ein kleines PowerShell-Skript, das du morgen früh als geplante Aufgabe einrichten kannst.

FAKTEN – ADFS-Monitoring auf einen Blick

Der Endpunkt /adfs/probe antwortet über HTTP auf Port 80 mit 200 OK – lokal, ohne Abhängigkeit von Backend-Diensten. Er zeigt, dass der Knoten lebt, nicht, dass er Tokens ausstellen kann.

In einer WID-Farm fragt jeder Sekundärknoten den Primärknoten standardmäßig alle fünf Minuten nach Änderungen. Get-AdfsSyncProperties zeigt, wann das zuletzt geklappt hat.

Mit AutoCertificateRollover erzeugt ADFS das neue Token-Signing-Zertifikat standardmäßig 20 Tage vor Ablauf und macht es 5 Tage vor Ablauf zum primären Zertifikat. Das SSL-Zertifikat tauschst du immer von Hand.

Die ADFS Toolbox kommt als PowerShell-Modul ADFS-Toolbox aus der PowerShell Gallery. Die frühere Website AD FS Help ist eingestellt, der Diagnostics Analyzer gehört heute zu den AD FS Offline Tools.

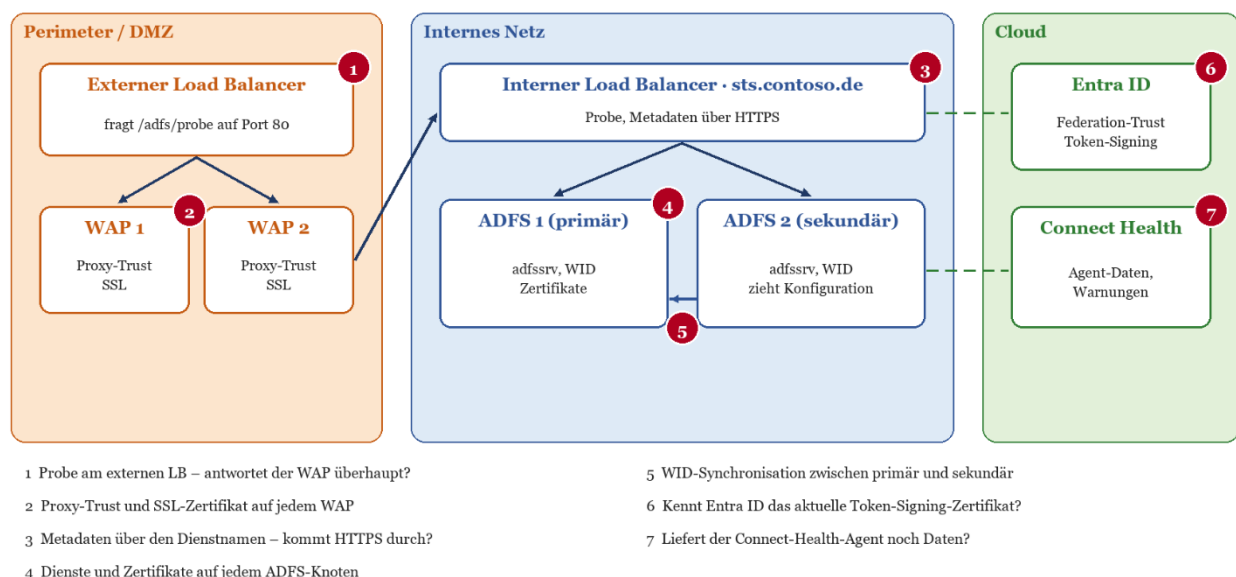
Entra Connect Health for ADFS ist eine Funktion von Microsoft Entra ID P1 oder P2 und braucht einen Agenten auf jedem ADFS- und jedem WAP-Server.

Was du regelmäßig prüfen solltest – die fünf Dauerbrenner

Fast jeder ADFS-Ausfall, den wir in den letzten Jahren auf den Tisch bekommen haben, fällt in eine von fünf Schubladen: ein abgelaufenes Zertifikat, ein gestoppter Dienst, eine Farm, deren Knoten sich nicht mehr einig sind, ein Web Application Proxy, dem die Farm nicht mehr vertraut, oder ein Load Balancer, der fröhlich Anfragen an einen Knoten schickt, der nur noch so tut als ob. Das Gute daran: Alle fünf lassen sich mit Bordmitteln prüfen. Das Schlechte: Von allein prüft sie niemand. Skizze 1 zeigt, wo die Prüfpunkte in einer typischen Umgebung sitzen.

Skizze 1: Wo du in einer ADFS-Umgebung hinschauen musst

Sieben Prüfpunkte von außen nach innen – jeder steht für eine typische Art, wie die Farm leise ausfällt



Skizze 1: Sieben Prüfpunkte von außen nach innen – vom Load Balancer in der DMZ bis zum Federation-Trust in Entra ID.

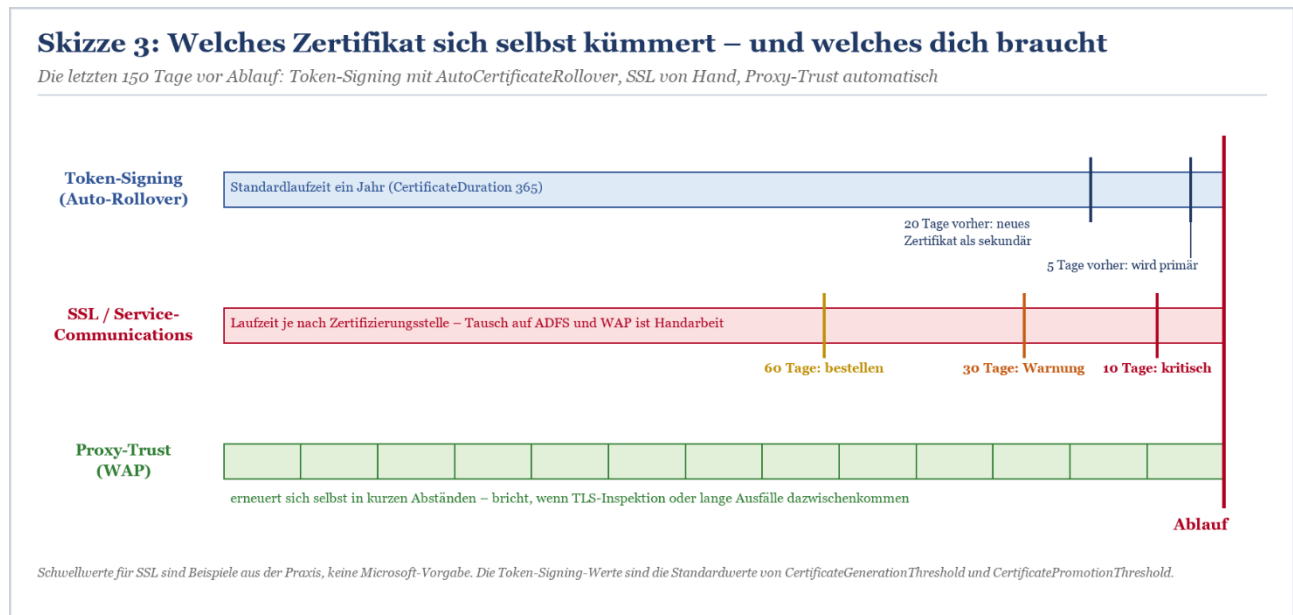
Zertifikatslaufzeiten: der Klassiker unter den Wochenendausfällen

In einer ADFS-Umgebung laufen mehr Zertifikate, als die meisten auf Anhieb aufzählen können. Das Token-Signing-Zertifikat signiert jedes ausgestellte Token, das Token-Decrypting-Zertifikat entschlüsselt verschlüsselte Tokens von Claims Providern, das SSL-Zertifikat (in der Konsole Service-Communications) sichert HTTPS auf ADFS und WAP. Dazu kommt das Proxy-Trust-Zertifikat, mit dem sich jeder WAP bei der Farm ausweist, und – falls du ADFS mit Entra-MFA gekoppelt hast – das Zertifikat des MFA-Adapters. Jedes davon hat ein Ablaufdatum, und keines davon interessiert sich für deinen Urlaubsplan.

Beim Token-Signing- und Token-Decrypting-Zertifikat hilft dir ADFS selbst, solange AutoCertificateRollover aktiv ist: Die Farm erzeugt neue selbstsignierte Zertifikate rechtzeitig und befördert sie automatisch. Das Problem sitzt auf der anderen Seite. Jede Relying Party, die das

Zertifikat fest hinterlegt hat und die Föderationsmetadaten nicht regelmäßig abrufen, lehnt nach dem Wechsel die Tokens ab. Bei Microsoft 365 ist das in der Regel unkritisch, weil Entra ID die Metadaten selbst überwacht – aber eben nur in der Regel. Was du dann tust, steht in [Microsoft-365-Federation-Trust nach Zertifikatswechsel aktualisieren](#). Die Details zum Rollover mit und ohne Automatik findest du in [ADFS Token-Signing-Zertifikat erneuern – mit und ohne AutoCertificateRollover](#).

Das SSL-Zertifikat dagegen ist immer Handarbeit. Es kommt von deiner Zertifizierungsstelle, muss auf allen Farmknoten und allen WAP-Servern getauscht werden und wird gern vergessen, weil es ja „letztes Mal problemlos lief“. Letztes Mal ist meist zwei Jahre her, und die Anleitung lag im Postfach eines Kollegen, der nicht mehr da ist. Wie der Tausch sauber gelingt, beschreibt [SSL-Zertifikat auf ADFS und Web Application Proxy tauschen](#).



Skizze 3: Token-Signing kümmert sich mit Auto-Rollover selbst, das SSL-Zertifikat braucht dich, der Proxy-Trust erneuert sich – solange nichts dazwischenfunkt.

WARNUNG — Auto-Rollover ist kein Freifahrtschein

AutoCertificateRollover verhindert, dass ADFS mit einem abgelaufenen Signaturzertifikat dasteht. Es verhindert nicht, dass eine SaaS-Anwendung mit fest hinterlegtem Zertifikat nach dem Wechsel jede Anmeldung ablehnt. Dein Monitoring sollte deshalb nicht nur das Ablaufdatum prüfen, sondern dich auch dann informieren, wenn ein sekundäres Token-Signing-Zertifikat auftaucht. Ab diesem Moment hast du noch die Zeit bis zur Beförderung, um die Anwendungsverantwortlichen zu warnen.

Dienststatus: läuft er, oder sieht er nur so aus?

Auf jedem Farmknoten läuft der Dienst adfssrv (Anzeigename Active Directory Federation Services). In einer WID-Farm kommt die Windows Internal Database hinzu, deren Dienst MSSQL\$MICROSOFT##WID heißt. Auf den WAP-Servern prüfst du den Zustand am einfachsten mit Get-WebApplicationProxyHealth: Das Cmdlet liefert je Komponente – AD FS Proxy und Web Application Proxy Core – einen HealthState. Und wenn du Connect Health nutzt, gehören dessen Agentendienste ebenfalls in die Liste, sonst merkst du erst nach zwei Stunden über die Cloud-Warnung, dass der Agent schweigt.

Ein laufender Dienst ist allerdings nur die halbe Wahrheit. Wir hatten schon Farmen, in denen adfssrv tadellos „Wird ausgeführt“ meldete, aber jede Anfrage mit einem Fehler beantwortete, weil

das Dienstkonto sein Kennwort nicht mehr kannte oder der private Schlüssel eines Zertifikats nicht mehr lesbar war. Deshalb kombinierst du den Dienststatus immer mit einem funktionalen Test, etwa dem Abruf der Föderationsmetadaten. Wer das Kennwortproblem gleich ganz loswerden will, stellt auf ein gruppenverwaltetes Dienstkonto um – wie das geht, steht in [gMSA als ADFS-Dienstkonto einrichten und umstellen](#).

Synchronisation der Farm: wenn sich die Knoten nicht mehr einig sind

In einer Farm mit Windows Internal Database gibt es genau einen Primärknoten, auf dem du Änderungen vornimmst. Die Sekundärknoten fragen ihn in festen Abständen nach Neuigkeiten – standardmäßig alle fünf Minuten. Bricht diese Synchronisation ab, merkt das zunächst niemand: Die Sekundärknoten stellen weiter Tokens aus, nur eben mit dem Stand von gestern. Das fällt erst auf, wenn eine neue Relying Party auf dem einen Knoten funktioniert und auf dem anderen nicht – und dein Load Balancer würfelt, wer welchen Knoten bekommt. Viel Spaß beim Nachstellen.

Get-AdfsSyncProperties zeigt dir auf einem Sekundärknoten den Primärknoten, die Rolle, das Abfrageintervall (PollDuration in Sekunden) sowie Zeitpunkt und Status der letzten Synchronisation. Ist die letzte erfolgreiche Synchronisation deutlich älter als ein paar Intervalle, stimmt etwas nicht – typischerweise Namensauflösung, Firewall oder ein Primärknoten, den jemand für „nur kurz“ heruntergefahren hat. Bei einer Farm mit SQL Server entfällt dieser Mechanismus, dafür überwachst du die Verfügbarkeit der Datenbank. Die Unterschiede beider Wege erklärt [WID oder SQL Server als ADFS-Konfigurationsdatenbank](#).

```
# Auf einem Sekundärknoten der WID-Farm
Get-AdfsSyncProperties |
    Select-Object Role, LastSyncFromPrimaryComputerName, LastSyncTime, LastSyncStatus,
    PollDuration
```

Synchronisationsstatus eines Sekundärknotens abfragen.

WAP-Trust: der stille Bruch zwischen DMZ und Farm

Jeder Web Application Proxy weist sich gegenüber der Farm mit einem Proxy-Trust-Zertifikat aus. Das Zertifikat wird bei der Einrichtung des WAP ausgestellt und danach automatisch in kurzen Abständen erneuert – unsichtbar, zuverlässig, solange die Verbindung zwischen WAP und Farm Ende zu Ende durchgeht. Steht dazwischen eine Firewall mit TLS-Inspektion oder war der WAP lange ausgeschaltet, klappt die Erneuerung nicht, das Vertrauen ist weg, und der Proxy kann seine Konfiguration nicht mehr von der Farm abholen. Von außen meldet sich niemand mehr an, innen funktioniert alles. Ein Fehlerbild, das gern mit „Microsoft 365 hat gerade Probleme“ verwechselt wird.

Überwachen lässt sich das auf zwei Wegen: auf dem WAP mit Get-WebApplicationProxyHealth, das den Zustand der Komponente AD FS Proxy meldet, und über einen Blick in den Zertifikatsspeicher des Computers, wo das Proxy-Trust-Zertifikat mit einem Betreff liegt, der mit „ADFS ProxyTrust“ beginnt. Ist es abgelaufen oder fehlt es, hilft nur, das Vertrauen neu aufzubauen. Wie das geht, steht in [WAP-Proxy-Vertrauen erneuern – wenn der Web Application Proxy plötzlich nicht mehr will](#). Hintergründe zur Rolle des Proxys und zu seiner Ablöse findest du in [Web Application Proxy: Architektur und Ablöse](#).

Probe-Endpunkt: ein Puls, kein Gesundheitszeugnis

Für Load Balancer stellt ADFS einen eigenen Endpunkt bereit: /adfs/probe, erreichbar über HTTP auf Port 80, sowohl auf den Farmknoten als auch auf den WAP-Servern. Er antwortet mit HTTP 200, solange der Dienst lauscht. Microsoft empfiehlt ausdrücklich, für Health Checks diesen HTTP-Endpunkt zu verwenden und nicht HTTPS – damit es keinen Ärger mit SNI gibt. Das ist

richtig und gut, hat aber einen Haken, den man kennen muss: Die Antwort entsteht lokal, ohne Abhängigkeit von Backend-Diensten.

Übersetzt heißt das: Die Probe meldet auch dann 200, wenn der Knoten seine Konfigurationsdatenbank nicht erreicht, das SSL-Zertifikat abgelaufen ist oder kein Domänencontroller antwortet. Für den Load Balancer ist das in Ordnung, er soll tote Knoten schnell aus dem Pool nehmen. Für dein Monitoring reicht es nicht. Dort brauchst du zusätzlich einen Abruf der Föderationsmetadaten über HTTPS und den Dienstnamen – also so, wie ihn ein Client sieht. Wie du die Probe im Load Balancer sauber konfigurierst, steht in [ADFS hinter dem Load Balancer – Health Probe, SNI und Persistenz](#).

Skizze 2: /adfs/probe ist ein Puls, kein Gesundheitszeugnis

Die Probe antwortet lokal mit HTTP 200 – ohne Rücksicht auf Datenbank, AD oder Zertifikate



Deshalb in zwei Ebenen überwachen

Ebene 1 – Load Balancer: /adfs/probe über HTTP auf Port 80, damit kein SNI-Ärger entsteht und tote Knoten schnell aus dem Pool fliegen.

Ebene 2 – Monitoring: Föderationsmetadaten über HTTPS mit dem Dienstnamen, Zertifikatslaufzeiten, Synchronisation, Dienste.

Skizze 2: Die Probe beantwortet nur die Frage, ob der Knoten lebt. Ob er Tokens ausstellen kann, prüft ein Abruf der Metadaten über HTTPS.

```

# Probe direkt am Knoten (HTTP, Port 80)
Invoke-WebRequest -Uri "http://adfs1.contoso.local/adfs/probe" -UseBasicParsing |
    Select-Object StatusCode

# Funktionstest über den Dienstnamen (HTTPS, wie ein Client)
Invoke-WebRequest -Uri "https://sts.contoso.de/FederationMetadata/2007-
06/FederationMetadata.xml" `
    -UseBasicParsing | Select-Object StatusCode, RawContentLength
    
```

Probe und Metadatenabruf – zwei Tests mit unterschiedlicher Aussagekraft.

Die Prüfliste: was, wie oft, womit

Damit aus guten Vorsätzen ein Betriebsprozess wird, hilft eine Tabelle, die jemand anderes als du lesen kann. Die Frequenzen sind Erfahrungswerte aus dem Betrieb, keine Microsoft-Vorgaben. Passe sie an deine Umgebung an – aber nicht nach unten, nur weil „da ja nie was ist“. Da ist nie was, bis da was ist.

Prüfpunkt	Frequenz	Werkzeug	Alarm, wenn ...
Probe /adfs/probe je Knoten	alle 10 bis 30 Sekunden	Load Balancer	kein HTTP 200 – Knoten raus aus dem Pool
Föderationsmetadaten über HTTPS	alle 5 Minuten	Monitoring-System oder Prüfskript	Abruf scheitert oder dauert auffällig lange
Dienst adfssrv und WID	alle 5 Minuten	Monitoring-System, Get-Service	Dienst nicht im Status Running
WAP-Zustand	alle 5 Minuten	Get-	HealthState ungleich OK

		WebApplicationProxyHealth	
WID-Synchronisation	stündlich	Get-AdfsSyncProperties	letzte Synchronisation älter als einige Intervalle
SSL-Zertifikat ADFS und WAP	täglich	Prüfskript, Get-AdfsSslCertificate	weniger als 30 Tage Restlaufzeit
Token-Signing und Token-Decrypting	täglich	Prüfskript, Get-AdfsCertificate	sekundäres Zertifikat neu aufgetaucht oder Restlaufzeit knapp
Proxy-Trust-Zertifikat	täglich	Prüfskript auf dem WAP	abgelaufen oder nicht vorhanden
Token-Signing in Entra ID	täglich	Microsoft Graph PowerShell	Zertifikat in Entra ID weicht vom primären ADFS-Zertifikat ab
Connect-Health-Agent	laufend	Entra Connect Health	Warnung „Health Service data is not up to date“
Fehlanmeldungen und Risky IP	täglich lesen	Entra Connect Health, SIEM	Ausreißer bei Benutzern oder Quelladressen
Gesamtcheck der Konfiguration	monatlich und nach Änderungen	ADFS Toolbox	Test-AdfsServerHealth meldet Fail

Tabelle 1: Prüfpunkte, Frequenzen und Werkzeuge für den ADFS-Betrieb – Erfahrungswerte, keine Herstellervorgabe.

TIPP – Warnschwellen mit Puffer für die Bürokratie

Plane die Warnschwelle für das SSL-Zertifikat nicht nach der technischen Dauer des Tauschs, sondern nach der Dauer des Bestellprozesses. Zwei Stunden Arbeit am Server stehen in vielen Häusern drei Wochen Freigabe, Bestellung und Change-Board gegenüber. 60 Tage für die Bestellung, 30 Tage für die Warnung und 10 Tage für den roten Alarm haben sich bewährt.

ADFS Toolbox und Diagnostics Analyzer – was sie wirklich können

Um die ADFS-Diagnosewerkzeuge von Microsoft ranken sich einige Legenden, nicht zuletzt, weil sich ihr Zuhause mehrfach geändert hat. Früher gab es die Website AD FS Help mit Online-Werkzeugen. Die ist eingestellt. Der Diagnostics Analyzer liegt heute auf GitHub und gehört zu den AD FS Offline Tools. Was geblieben ist, ist das PowerShell-Modul ADFSToolbox, und das ist für den Betrieb auch der eigentlich interessante Teil.

Installation: aus der Gallery oder per Kopie

Hat dein ADFS-Server Internetzugang, installierst du das Modul direkt aus der PowerShell Gallery. Ohne Internetzugang – und das ist auf einem ordentlich gehärteten Farmknoten die Regel – installierst du es auf einem Rechner mit Internetzugang und kopierst den Ordner ADFSToolbox aus dem Modulverzeichnis unter Programme\WindowsPowerShell\Modules an dieselbe Stelle auf dem ADFS- oder WAP-Server. Microsoft empfiehlt, das Modul auf dem primären ADFS-Server oder einem WAP einzurichten.

```
# Mit Internetzugang
Install-Module -Name ADFSToolbox -Force
Import-Module ADFSToolbox -Force

# Ohne Internetzugang: auf einem Rechner mit Internet installieren und den Ordner
# C:\Program Files\WindowsPowerShell\Modules\ADFSToolbox auf den ADFS-Server kopieren
Import-Module ADFSToolbox -Force
```

ADFSToolbox einrichten – online oder offline.

WICHTIG – Uralt-Farmen brauchen eine alte Toolbox

Läuft bei dir noch ADFS 2.1 oder älter, unterstützt die aktuelle ADFS-Toolbox diese Version nicht mehr. Microsoft nennt für diesen Fall die Version 1.0.13 des Moduls. Ehrlicherweise ist eine Farm in diesem Alter aber ein Fall für ein Upgrade und nicht für ein Monitoring-Projekt.

Die Cmdlets der Diagnose

Das Diagnosemodul der Toolbox bringt eine Handvoll Cmdlets mit. Für den Alltag sind zwei davon entscheidend. `Test-AdfsServerHealth` führt eine Reihe von Prüfungen für die häufigsten Konfigurationsprobleme aus, die Microsoft über die Jahre gesammelt hat. Jedes Ergebnis hat drei Eigenschaften: Name (die Prüfung), Result mit den Werten Pass, Fail oder NotRun und Detail mit der Erklärung, warum etwas fehlgeschlagen ist oder nicht laufen konnte. Zu den Prüfungen gehören zum Beispiel, ob der ADFS-Dienst und die WID laufen, ob die Föderationsmetadaten abrufbar sind, ob die SSL-Bindungen stimmen, ob die Überwachungsrichtlinie aktiv ist und ob das Token-Decrypting-Zertifikat im Speicher liegt, einen privaten Schlüssel hat oder bald abläuft.

`Export-AdfsDiagnosticsFile` führt die Diagnose über alle Server der Farm aus und nutzt dafür Remote-PowerShell-Sitzungen. Ab einer Farm auf Windows Server 2016 liest das Cmdlet die Serverliste aus der ADFS-Konfiguration, unter 2012 R2 prüft es nur den lokalen Rechner – es sei denn, du gibst die Server mit `-ServerNames` selbst vor. Das Ergebnis ist eine JSON-Datei im aktuellen Verzeichnis, benannt nach dem Muster `AdfsDiagnosticsFile-<Zeitstempel>.json`. Diese Datei wertet der Diagnostics Analyzer aus und sortiert die Ergebnisse in Failed, Warning, Passed, Not run und Not applicable – jeweils mit Beschreibung und Lösungsschritten.

```
# Nur die Auffälligkeiten des lokalen Servers anzeigen
Test-AdfsServerHealth |
  Where-Object Result -ne 'Pass' |
  Format-Table Name, Result, Detail -Wrap

# Diagnose über die ganze Farm als JSON-Datei
Export-AdfsDiagnosticsFile -ServerNames @"(\"ads1.contoso.local\", \"ads2.contoso.local\")"
```

Test-AdfsServerHealth und Export-AdfsDiagnosticsFile im Einsatz.

Cmdlet	Zweck	Wann einsetzen
Test-AdfsServerHealth	Prüfkatalog für häufige Konfigurationsfehler, Ergebnis Pass, Fail oder NotRun	monatlich, nach jeder Änderung, bei Störungen
Export-AdfsDiagnosticsFile	Diagnose über alle Farmknoten, Ausgabe als JSON-Datei für den Diagnostics Analyzer	bei Störungen, vor und nach Upgrades, für den Support-Fall
Test-AdfsServerToken	fordert ein Token an, um die Ausstellung zu testen	wenn du wissen willst, ob ADFS wirklich Tokens ausstellt
Get-AdfsSystemInformation	sammelt Systeminformationen des Servers	Bestandsaufnahme, Supportfall
Get-AdfsServerConfiguration	liest die ADFS-Konfiguration aus	Dokumentation, Vergleich zweier Knoten

Tabelle 2: Die Diagnose-Cmdlets der ADFS-Toolbox und ihr typischer Einsatz.

WARNUNG – Die JSON-Datei ist kein Urlaubsfoto

Die Diagnosedatei enthält Details zur Konfiguration deiner Farm – Server, Zertifikate, Einstellungen. Behandle sie so, wie du eine Konfigurationssicherung behandeln würdest: nicht per Mail quer durchs Haus schicken, nicht auf einem Fileshare mit Jedermann-Rechten ablegen und nach der Analyse löschen. Warum das wichtig ist, erklärt der Beitrag [ADFS-Sicherheit: Angriffe, Härtung, Golden SAML und MFA](#).

Und noch eine ehrliche Einordnung: Die Toolbox ist ein hervorragendes Diagnosewerkzeug, aber kein Monitoring. Sie liefert eine Momentaufnahme, wenn du sie startest. Sie ruft dich nicht an, wenn nachts der Proxy-Trust bricht. Dafür brauchst du etwas, das dauerhaft läuft – entweder dein eigenes Skript oder Connect Health, am besten beides.

Entra Connect Health for ADFS – der Blick aus der Cloud

Microsoft Entra Connect Health überwacht deine lokale Identitätsinfrastruktur aus der Cloud – neben der Synchronisation und den Domänencontrollern eben auch ADFS. Wer noch den alten Namen Azure AD Connect Health im Kopf hat: gleiches Produkt, neues Etikett. Das Werkzeug ist eine Funktion von Microsoft Entra ID P1 oder P2. Wenn du diese Lizenzen ohnehin für Conditional Access oder andere Funktionen hast, ist Connect Health praktisch geschenkt. Mehr zu Entra ID selbst findest du auf der Übersichtsseite [Microsoft Entra ID](#).

Voraussetzungen und Agenteninstallation

Der Agent muss auf jedem ADFS-Server und jedem WAP-Server installiert werden, sonst fehlen dir Daten. Microsoft unterstützt den Agenten für ADFS auf Windows Server 2016, 2019, 2022 und 2025 – nicht aber auf Server Core. Er braucht ausgehende Verbindungen zu den Connect-Health-Endpunkten auf Port 443; aktuelle Agentenversionen kommen ohne den früher zusätzlich nötigen Port 5671 aus. TLS-Inspektion auf dem ausgehenden Weg kann Registrierung und Datenübertragung scheitern lassen. Installiere den ADFS-Agenten nicht auf deinem Entra-Connect-Synchronisationsserver, und achte darauf, dass der Hostname des ADFS-Servers nicht mit dem Namen des Federation Service übereinstimmt.

Nach der Installation laufen auf dem Server zwei Dienste: Microsoft Entra Connect Agent Updater und Microsoft Entra Connect Health Agent. Neue Agentenversionen werden automatisch eingespielt. Kann ein Agent länger als zwei Stunden keine Daten senden, zeigt das Portal die Warnung „Health Service data is not up to date“. Ob der Agent die Cloud erreicht, prüfst du direkt auf dem Server.

```
# Verbindung des Connect-Health-Agenten zur Cloud testen (auf dem ADFS- oder WAP-Server)
Test-AzureADConnectHealthConnectivity -Role ADFS

# Laufen die Agentendienste?
Get-Service -DisplayName "Microsoft Entra Connect*" | Select-Object DisplayName, Status
```

Der Cmdlet-Name trägt noch den alten Produktamen – das ist kein Tippfehler.

Ohne Auditing keine Nutzungsdaten

Für Usage Analytics und die Sicherheitsberichte wertet Connect Health die ADFS-Überwachungsereignisse aus. Die sind standardmäßig nicht vollständig aktiv. Du brauchst drei Dinge: Das ADFS-Dienstkonto muss in der lokalen Sicherheitsrichtlinie das Recht „Generieren von Sicherheitsüberwachungen“ haben, die Unterkategorie Anwendungsgeneriert muss für Erfolg und Fehler überwacht werden, und auf dem primären ADFS-Server stellst du die Überwachungsstufe auf Verbose. Achtung: Eine Gruppenrichtlinie kann das Auditing wieder abschalten, und dann bleiben die Nutzungsdaten leer, ohne dass es laut wird.

```
# Überwachung der Unterkategorie "Application Generated" einschalten
# (auf deutschsprachigen Servern den lokalisierten Namen aus "auditpol /list /subcategory:*"
# verwenden)
auditpol.exe /set /subcategory:"Application Generated" /failure:enable /success:enable

# Auf dem primären ADFS-Server: ausführliche Überwachung
Set-AdfsProperties -AuditLevel Verbose
(Get-AdfsProperties).AuditLevel
```

Auditing für Connect Health vorbereiten.

Welche Ereignisse dabei entstehen und wie du sie liest, auch ohne Cloud, erklärt [ADFS-Ereignisprotokolle lesen – Admin-Log, Debug-Tracing und Auditing](#).

Was Connect Health dir zeigt

Im Portal findest du je ADFS-Dienst den Status der Server, Warnungen mit Lösungshinweisen, Diagramme und Sicherheitsberichte. Die Warnungen sind der eigentliche Mehrwert für den Betrieb: Connect Health kennt eine ganze Reihe typischer ADFS-Probleme und sagt dir nicht nur, dass etwas klemmt, sondern auch, wo du anfangen solltest. Benachrichtigungen per Mail lassen sich einrichten – tu das, sonst ist das schönste Dashboard so nützlich wie ein Rauchmelder im Keller eines Hauses, in dem niemand wohnt.

Bereich	Was du siehst	Wofür du es brauchst
Alerts	aktive und gelöste Warnungen mit betroffenen Servern und Lösungshinweisen	Störungen früh erkennen, auch die, an die dein Skript nicht gedacht hat
Usage Analytics	Anmeldungen je Anwendung, Top-Anwendungen nach Verkehr (setzt Auditing voraus)	Inventar der genutzten Relying Parties, Vorbereitung einer Migration
Performance Monitoring	Tokenanfragen pro Sekunde je Server neben den Anwendungsaufrufen	Lastverteilung prüfen, Kapazität planen, Ausreißer erkennen
Bad Password Attempts	Top 50 Benutzer mit Fehlanmeldungen über 30 Tage, Aktualisierung alle 12 Stunden	Passwort-Spray und Anwendungen mit veralteten Kennwörtern finden
Risky IP	Quelladressen mit auffällig vielen Fehlanmeldungen	Angriffe erkennen, Smart Lockout und Firewall-Regeln nachschärfen

Tabelle 3: Die wichtigsten Bereiche von Entra Connect Health for ADFS.

FAKTEN – Connect Health for ADFS – die harten Daten

Lizenz: Microsoft Entra ID P1 oder P2. In der China-Cloud ist Connect Health nicht verfügbar.
Agent auf jedem ADFS- und WAP-Server, unterstützt auf Windows Server 2016 bis 2025, nicht auf Server Core.
Ausgehend Port 443, ältere Agenten zusätzlich 5671. TLS-Inspektion kann die Registrierung verhindern.
Warnung im Portal, wenn ein Agent länger als zwei Stunden keine Daten liefert.
Der Bericht Bad Password Attempts zeigt 30 Tage und wird alle 12 Stunden aktualisiert – er ist kein Echtzeit-Alarm.

Gerade der Bericht zu Fehlanmeldungen ist in der Praxis Gold wert, weil er zwei Dinge zeigt, die sonst im Rauschen untergehen: Angreifer, die Kennwörter durchprobieren, und Anwendungen, die mit einem längst geänderten Kennwort im Minutentakt anklopfen. Wie du beides unterscheidest und was du dagegen tust, beschreiben [Passwort-Spray und Brute Force auf ADFS erkennen](#) und [Extranet Smart Lockout – ADFS gegen Passwort-Spray schützen](#). Und weil Usage Analytics zeigt, welche Anwendungen tatsächlich genutzt werden, ist Connect Health nebenbei eine gute Grundlage, wenn du den Weg Richtung Entra ID planst.

Skizze 4: Drei Werkzeuge, drei Blickwinkel

Kein Werkzeug sieht alles. Zusammen decken sie die typischen Ausfälle ab.



Skizze 4: Eigenes Skript, ADFS Toolbox und Connect Health ergänzen sich – jedes Werkzeug hat seinen blinden Fleck.

Werkzeug	Läuft	Kosten	Stärke	Grenze
Eigenes Prüfskript	lokal, geplante Aufgabe	Pflegeaufwand	prüft genau deine Risiken mit deinen Schwellen	kennt nur, was du hineinschreibst
ADFS Toolbox	lokal, bei Bedarf	keine	breiter Prüfkatalog von Microsoft	Momentaufnahme, kein Dauerbetrieb
Entra Connect Health	Agent plus Cloud	Entra ID P1 oder P2	Warnungen, Nutzung, Sicherheitsberichte	braucht Internetzugang der Server
Ereignisprotokolle und SIEM	lokal oder zentral	je nach Produkt	vollständige Spur jeder Anmeldung	nur so gut wie die Auswerteregeln

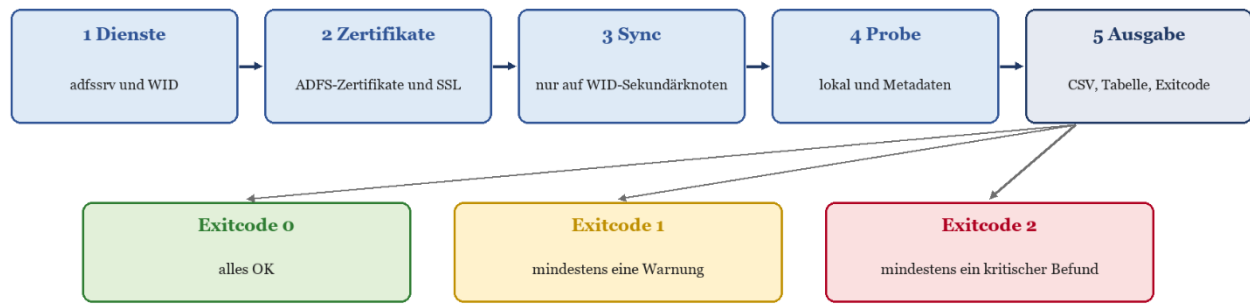
Tabelle 4: Werkzeuge für das ADFS-Monitoring im Vergleich.

Das eigene Prüfskript – klein, ehrlich, täglich

Kein Werkzeug von der Stange kennt deine Umgebung so gut wie du. Deshalb gehört zu einem ordentlichen ADFS-Monitoring ein eigenes kleines Skript, das genau die Fragen beantwortet, die bei dir wehtun. Das folgende Beispiel läuft auf jedem Farmknoten, prüft Dienste, Zertifikate, Synchronisation, Probe und Metadaten und liefert einen Exitcode, den jedes gängige Monitoring-System versteht. Es ist bewusst schlicht gehalten: lieber ein Skript, das jeden Tag läuft, als ein Framework, das nie fertig wird.

Skizze 5: So arbeitet das Beispielskript

Fünf Prüfböcke, ein Ergebnisobjekt, ein Exitcode – den Rest erledigt dein Monitoring-System



Der Exitcode ist die Schnittstelle zu PRTG, Checkmk, Nagios, SCOM oder was auch immer bei dir schon läuft.

Skizze 5: Fünf Prüfböcke, ein Ergebnisobjekt, ein Exitcode – die Schnittstelle zu deinem Monitoring-System.

Das Skript für die Farmknoten

```

# Test-AdfsKnoten.ps1 – täglicher Gesundheitscheck für einen ADFS-Farmknoten
# Ausführung als geplante Aufgabe mit lokalen Administratorrechten
param(
    [int]$WarnTage      = 30,
    [int]$KritischTage  = 10,
    [string]$DienstName = "sts.contoso.de",
    [string]$Protokollpfad = "D:\Monitoring"
)

$ergebnis = [System.Collections.Generic.List[object]]::new()

function Melde([string]$Punkt, [string]$Status, [string]$Detail) {
    $ergebnis.Add([pscustomobject]@{
        Zeit      = Get-Date -Format "yyyy-MM-dd HH:mm"
        Server    = $env:COMPUTERNAME
        Punkt     = $Punkt
        Status    = $Status
        Detail    = $Detail
    })
}

function Test-Laufzeit([string]$Name, [datetime]$NotAfter) {
    $rest = [int]($NotAfter - (Get-Date)).TotalDays
    $status = if ($rest -le $KritischTage) { "KRITISCH" }
    elseif ($rest -le $WarnTage) { "WARNUNG" }
    else { "OK" }
    Melde $Name $status ("läuft ab am {0:dd.MM.yyyy}, noch {1} Tage" -f $NotAfter, $rest)
}

# 1. Dienste
foreach ($dienst in 'adfssrv', 'MSSQL$MICROSOFT##WID') {
    $s = Get-Service -Name $dienst -ErrorAction SilentlyContinue
    if (-not $s) { Melde "Dienst $dienst" "INFO" "nicht vorhanden (SQL-Farm?)" ; continue }
    $status = if ($s.Status -eq 'Running') { "OK" } else { "KRITISCH" }
    Melde "Dienst $dienst" $status $s.Status
}

# 2. Zertifikate aus der ADFS-Konfiguration und SSL-Zertifikat
foreach ($c in Get-AdfsCertificate) {
    Test-Laufzeit ("{"0} (primär: {1})" -f $c.CertificateType, $c.IsPrimary)
    $c.Certificate.NotAfter
}
$neues = Get-AdfsCertificate -CertificateType Token-Signing | Where-Object { -not $_.IsPrimary }
if ($neues) { Melde "Token-Signing-Rollover" "WARNUNG" "sekundäres Zertifikat vorhanden -

```

```

Relying Parties informieren" }

$hash = (Get-AdfsSslCertificate | Select-Object -First 1).CertificateHash
$ssl = Get-Item -Path "Cert:\LocalMachine\My\$hash" -ErrorAction SilentlyContinue
if ($ssl) { Test-Laufzeit "SSL-Zertifikat" $ssl.NotAfter }
else { Melde "SSL-Zertifikat" "KRITISCH" "Zertifikat $hash nicht im Speicher gefunden" }

Melde "AutoCertificateRollover" "INFO" (Get-AdfsProperties).AutoCertificateRollover

# 3. Synchronisation – nur auf Sekundärknoten einer WID-Farm
$sync = Get-AdfsSyncProperties
if ($sync.Role -eq 'SecondaryComputer') {
    # Beim ersten Lauf prüfen, ob LastSyncTime in Ortszeit oder UTC geliefert wird
    $alter = [int]((Get-Date) - $sync.LastSyncTime).TotalMinutes
    $grenze = [int](3 * $sync.PollDuration / 60)
    $status = if ($alter -le $grenze) { "OK" } else { "KRITISCH" }
    Melde "WID-Synchronisation" $status ("letzte Synchronisation vor {0} Minuten, Status {1}" -
f $alter, $sync.LastSyncStatus)
}

# 4. Probe lokal und Metadaten über den Dienstnamen
try {
    $p = Invoke-WebRequest -Uri "http://localhost/adfs/probe" -UseBasicParsing -TimeoutSec 10
    Melde "Probe lokal" "OK" "HTTP $($p.StatusCode)"
} catch { Melde "Probe lokal" "KRITISCH" $_.Exception.Message }

try {
    $m = Invoke-WebRequest -Uri "https://$DienstName/FederationMetadata/2007-
06/FederationMetadata.xml" `
        -UseBasicParsing -TimeoutSec 15
    Melde "Föderationsmetadaten" "OK" "HTTP $($m.StatusCode), $($m.RawContentLength) Byte"
} catch { Melde "Föderationsmetadaten" "KRITISCH" $_.Exception.Message }

# 5. Ausgabe und Exitcode
$datei = Join-Path $Protokollpfad ("adfs-health-{0:yyyyMMdd}.csv" -f (Get-Date))
$ergebnis | Export-Csv -Path $datei -NoTypeInfo -Encoding UTF8 -Append
$ergebnis | Where-Object Status -in 'WARNUNG', 'KRITISCH' | Format-Table -AutoSize

if ($ergebnis.Status -contains 'KRITISCH') { exit 2 }
elseif ($ergebnis.Status -contains 'WARNUNG') { exit 1 }
else { exit 0 }

```

Test-AdfsKnoten.ps1 – ein Beispielskript, kein fertiges Produkt. Erst im Testsystem ausprobieren, dann anpassen.

Ein paar Worte zu den Entscheidungen im Skript. Die Probe fragt es lokal über `http://localhost` ab, weil sie nur zeigen soll, dass der eigene Knoten lauscht. Die Metadaten dagegen holt es über den Dienstnamen und HTTPS – damit testest du den Weg, den auch ein Client nimmt, inklusive Load Balancer, SSL-Bindung und Zertifikat. Ein Abruf über `https://localhost` würde wegen der namensbasierten SSL-Bindung ohnehin scheitern und dir einen Fehlalarm bescheren. Und das sekundäre Token-Signing-Zertifikat meldet das Skript bewusst als Warnung, nicht als Fehler: Es ist kein Problem, sondern eine Einladung, die Anwendungsverantwortlichen zu informieren, solange noch Zeit ist.

TIPP — Als geplante Aufgabe mit gMSA ausführen

Richte das Skript als geplante Aufgabe ein und lass sie unter einem gruppenverwalteten Dienstkonto laufen, das auf den Farmknoten lokale Administratorrechte hat. Dann musst du kein Kennwort pflegen, das irgendwann abläuft und ausgerechnet dein Monitoring lahmlegt – die Ironie wäre zu schön. Wie gMSA bei ADFS funktioniert, steht in [gMSA als ADFS-Dienstkonto einrichten und umstellen](#).

Ergänzung für die WAP-Server

Auf den WAP-Servern gibt es kein ADFS-Modul, dafür das Modul WebApplicationProxy. Das kleine Gegenstück prüft den Zustand der Proxy-Komponenten, das Proxy-Trust-Zertifikat und die lokale Probe. Die Ausgabe kannst du mit denselben Funktionen Melde und Test-Laufzeit aufbereiten wie oben.

```
# Zustand der WAP-Komponenten (AD FS Proxy, Web Application Proxy Core)
Get-WebApplicationProxyHealth | Select-Object Component, HealthState

# Proxy-Trust-Zertifikat im Computerspeicher
$trust = Get-ChildItem -Path Cert:\LocalMachine\My |
    Where-Object Subject -like "CN=ADFS ProxyTrust*" |
    Sort-Object NotAfter -Descending | Select-Object -First 1
if ($trust) { "Proxy-Trust gültig bis {0:dd.MM.yyyy HH:mm}" -f $trust.NotAfter }
else { "KRITISCH: kein Proxy-Trust-Zertifikat gefunden" }

# Probe am WAP
(Invoke-WebRequest -Uri "http://localhost/adfs/probe" -UseBasicParsing -TimeoutSec
10).StatusCode
```

Prüfblock für den Web Application Proxy.

Abgleich mit Entra ID über Microsoft Graph PowerShell

Bleibt die Frage, ob Entra ID dasselbe Token-Signing-Zertifikat kennt wie deine Farm. Früher stand in jeder Anleitung ein Aufruf aus dem MSONline-Modul, das ist Geschichte. Heute nimmst du Microsoft Graph PowerShell und vergleichst das hinterlegte Signaturzertifikat der föderierten Domäne mit dem primären Token-Signing-Zertifikat von ADFS. Weichen sie voneinander ab, ist das ein Fall für [Microsoft-365-Federation-Trust nach Zertifikatswechsel aktualisieren](#) – und zwar bevor die Benutzer es merken.

```
# Auf einem Rechner mit ADFS-Modul und Microsoft Graph PowerShell
Connect-MgGraph -Scopes "Domain.Read.All"

$fed = Get-MgDomainFederationConfiguration -DomainId "contoso.de"
$aktuell = Get-AdfsCertificate -CertificateType Token-Signing | Where-Object IsPrimary
$adfsB64 = [Convert]::ToBase64String($aktuell.Certificate.RawData)

if ($fed.SigningCertificate -eq $adfsB64) {
    "OK: Entra ID kennt das primäre Token-Signing-Zertifikat."
} else {
    "WARNUNG: Entra ID und ADFS sind sich beim Token-Signing-Zertifikat nicht einig."
}
```

Token-Signing-Zertifikat in ADFS und Entra ID vergleichen.

WICHTIG – Während eines Rollovers sind zwei Zertifikate normal

In der Phase zwischen Erzeugung und Beförderung eines neuen Token-Signing-Zertifikats kann Entra ID das neue Zertifikat bereits als nächstes Signaturzertifikat kennen. Prüfe in dieser Zeit zusätzlich die Eigenschaft NextSigningCertificate der Federation-Konfiguration, bevor du Alarm schlägst. Und schalte keine automatischen Korrekturen auf Basis dieses Vergleichs – eine falsch aktualisierte Federation-Konfiguration sperrt alle föderierten Benutzer aus, gründlicher als jeder Angreifer.

WEITER – Wenn die Prüfung rot wird

- › [ADFS-Anmeldung schlägt fehl – die häufigsten Ursachen und ihre Lösung](#) – wenn trotz grüner Lampen niemand reinkommt
- › [ADFS-Ereignisprotokolle lesen – Admin-Log, Debug-Tracing und Auditing](#) – Admin-Log, Debug-Tracing und Auditing richtig lesen
- › [ADFS sichern und wiederherstellen – Rapid Restore Tool in der Praxis](#) – damit ein kaputter Knoten

kein Drama wird

Fazit: Lieber ein langweiliges Monitoring als ein spannender Montag

ADFS ist nicht tot. In vielen Unternehmen läuft die Farm seit über zehn Jahren und stellt zuverlässig Tokens aus – gerade deshalb schaut keiner mehr hin. Das ist der eigentliche Fehler. Die gute Nachricht: Ein solides Monitoring ist kein Großprojekt. Fünf Dauerbrenner gehören auf die Liste – Zertifikatslaufzeiten, Dienststatus, Synchronisation der Farm, WAP-Trust und Probe-Endpunkt –, ergänzt um einen echten Funktionstest über HTTPS und den Abgleich mit Entra ID. Ein eigenes Skript erledigt das täglich, die ADFS Toolbox liefert bei Änderungen und Störungen den breiten Prüfkatalog von Microsoft, und Entra Connect Health schaut dauerhaft aus der Cloud zu, wenn du die Lizenzen ohnehin hast.

Ob deine Farm noch zehn Jahre läuft oder der Umstieg auf Entra ID schon geplant ist: Solange sie in Betrieb ist, verdient sie ein Monitoring, das rechtzeitig Bescheid sagt. Wenn du dabei Unterstützung willst oder eine zweite Meinung zu deinen Schwellwerten brauchst, findest du uns im [Consulting zu ADFS \(Active Directory Federation Services\)](#). Wer Zertifikatswechsel, Sync-Abbrüche und kaputte Proxy-Trusts lieber einmal im Labor erlebt als nachts in der Produktion, ist in den [ADFS-Schulungen](#) richtig. Betrieb, Hochverfügbarkeit und Fehlersuche vertieft außerdem das Buch – mehr dazu unter [ADFS in der Praxis – das Buch im Detail](#).

WEITER – Passende Beiträge zum Betrieb der Farm

- › [ADFS Token-Signing-Zertifikat erneuern – mit und ohne AutoCertificateRollover](#) – damit der Rollover keine Überraschung wird
- › [SSL-Zertifikat auf ADFS und Web Application Proxy tauschen](#) – der Tausch, den dein Monitoring ankündigt
- › [ADFS hinter dem Load Balancer – Health Probe, SNI und Persistenz](#) – Probe, SNI und Persistenz richtig einstellen
- › [ADFS: Architektur, Einsatzszenarien und Hochverfügbarkeit](#) – das große Bild zu Farm, Proxy und Hochverfügbarkeit

FAQ: ADFS Monitoring und Health Check

Wie prüfe ich, ob ADFS läuft?

Für einen schnellen Lebenszeichen-Test rufst du `http://<server>/adfs/probe` auf, die Antwort sollte HTTP 200 sein. Ob ADFS wirklich arbeitet, zeigt erst der Abruf der Föderationsmetadaten über HTTPS und den Dienstnamen oder eine Testanmeldung.

Was prüft der Endpunkt /adfs/probe genau?

Nur, ob der ADFS- oder WAP-Server erreichbar ist und der Dienst antwortet. Die Antwort wird lokal erzeugt, ohne Abhängigkeit von Datenbank, Domänencontrollern oder Zertifikaten. Für Load Balancer ist das genau richtig, für ein vollständiges Monitoring reicht es nicht.

Warum sollte der Load Balancer die Probe über HTTP statt HTTPS abfragen?

Weil Microsoft es so empfiehlt: Der HTTP-Endpunkt auf Port 80 vermeidet Probleme mit SNI, die bei einer HTTPS-Abfrage ohne passenden Hostnamen auftreten.

Wie erkenne ich, ob die WID-Synchronisation in der Farm funktioniert?

Mit Get-AdfsSyncProperties auf einem Sekundärknoten. Das Cmdlet zeigt Rolle, Primärknoten, Abfrageintervall sowie Zeitpunkt und Status der letzten Synchronisation. Standardmäßig fragt jeder Sekundärknoten alle fünf Minuten nach Änderungen.

Was ist die ADFS Toolbox?

Ein PowerShell-Modul von Microsoft namens ADFSToolbox, das du aus der PowerShell Gallery installierst. Es enthält unter anderem Test-AdfsServerHealth für einen Prüfkatalog häufiger Konfigurationsfehler und Export-AdfsDiagnosticsFile für eine Diagnose über alle Farmknoten.

Gibt es AD FS Help noch?

Nein, die Website wurde eingestellt. Der Diagnostics Analyzer liegt heute auf GitHub und gehört zu den AD FS Offline Tools.

Welche Lizenz brauche ich für Entra Connect Health for ADFS?

Microsoft Entra ID P1 oder P2. Connect Health ist eine Funktion dieser Lizenzstufen.

Auf welchen Servern muss der Connect-Health-Agent installiert werden?

Auf jedem ADFS-Server und jedem Web Application Proxy. Unterstützt sind Windows Server 2016, 2019, 2022 und 2025, nicht aber Server Core.

Warum zeigt Connect Health keine Nutzungsdaten an?

Meist fehlt das Auditing: Das Dienstkonto braucht das Recht zum Generieren von Sicherheitsüberwachungen, die Unterkategorie Anwendungsgeneriert muss überwacht werden, und die Überwachungsstufe sollte auf Verbose stehen. Prüfe auch, ob eine Gruppenrichtlinie das Auditing abschaltet.

Wie oft sollte ich die Zertifikate von ADFS prüfen?

Täglich und automatisiert. Für das SSL-Zertifikat haben sich Warnschwellen von 60 Tagen für die Bestellung, 30 Tagen für die Warnung und 10 Tagen für den kritischen Alarm bewährt. Beim Token-Signing-Zertifikat sollte dich zusätzlich das Auftauchen eines sekundären Zertifikats informieren.