

Access Control Policies in ADFS – Zugriffsregeln ohne Claim-Rule-Akrobatik

ADFS • Access Control Policies

Zugriffsregeln ohne Claim-Rule-Akrobatik

MFA für extern, Gruppen, Netzwerkstandort – per Vorlage statt per Regelsprache.

boddenberg.de

Früher

```
c:[Type == "_groupsid",
Value == "5-1-5-21-..."]
=> issue(Type =
"/_permit", ...);
exists([Type ==
"/x-ms-proxy"])
=> issue(... deny ...);
```

➔

Heute

- ☒ Gruppe: App
- ☒ Netz: Extranet
- ☒ MFA verlangen
- ☒ Ausnahme: Admins

WISSEN	BERATUNG	SCHULUNG
Grundlagen, Architektur und alle Praxisbeiträge rund um ADFS an einem Ort. ➤ Active Directory Federation Services	Alte Autorisierungsregeln lesen, in Policies übersetzen und Conditional Access gleich mitdenken. ➤ Consulting zu ADFS	Im Labor: Policies mit Parametern bauen, MFA für extern erzwingen, Ausnahmen absichtlich falsch setzen. ➤ ADFS-Schulungen

Es gibt in fast jeder ADFS-Farm eine Relying Party, deren Autorisierungsregeln niemand mehr anfassen will. Sieben Zeilen Claim-Regelsprache, zwei davon mit „deny“, eine mit einer SID, die zu einer Gruppe gehört, die es seit der letzten Umstrukturierung nicht mehr gibt. Die Regeln funktionieren – vermutlich. Getestet wurde zuletzt, als der Kollege noch da war, der sie geschrieben hat. Wenn du jetzt nach „adfs access control policies“ suchst, willst du wahrscheinlich genau das loswerden: die Akrobatik, um einfache Fragen wie „Nur diese Gruppe, und von außen bitte mit MFA“ zu beantworten.

Seit ADFS unter Windows Server 2016 gibt es dafür Access Control Policies, kurz Zugriffssteuerungsrichtlinien. Statt Regeln zu schreiben, klickst du Bedingungen zusammen: Gruppe, Netzwerkstandort, Gerätevertrauen, MFA, Ausnahmen. Die Policy hängst du an eine oder viele Relying Parties, und eine Änderung an der Vorlage wirkt überall dort, wo sie zugewiesen ist. Dieser Beitrag zeigt dir, wie die neue Welt gegenüber den klassischen Issuance Authorization Rules tickt, welche Vorlagen du für MFA von extern, Gruppenbeschränkung und Netzwerkstandort direkt nutzen kannst und wie du eine eigene Policy mit Parametern baust. Die Grundlagen zu Farm, Relying Parties und Claims setzen wir voraus – sie stehen auf der Übersichtsseite [Active Directory Federation Services](#).

FAKTEN — Access Control Policies auf einen Blick
Verfügbar ab ADFS unter Windows Server 2016, sobald das Farm Behavior Level mindestens 2016 ist. Permit-Modell: Ohne passende Regel gibt es kein Token. Ausnahmen verhindern ein Permit, ein explizites Deny gibt es nicht. Bedingungen innerhalb einer Regel sind UND-verknüpft, mehrere Regeln und mehrere Ausnahmen sind ODER-verknüpft. Je Relying Party gilt entweder eine Access Control Policy oder das alte Regelmodell – Microsoft

beschreibt beide als sich gegenseitig ausschließend.

Mitgelieferte Vorlagen sind nicht änderbar, eigene Policies schon. Parametrisierte Policies lassen sich nach dem Anlegen nicht mehr bearbeiten.

PowerShell: Get-, New-, Set- und Remove-AdfsAccessControlPolicy sowie -AccessControlPolicyName und -AccessControlPolicyParameters an der Relying Party.

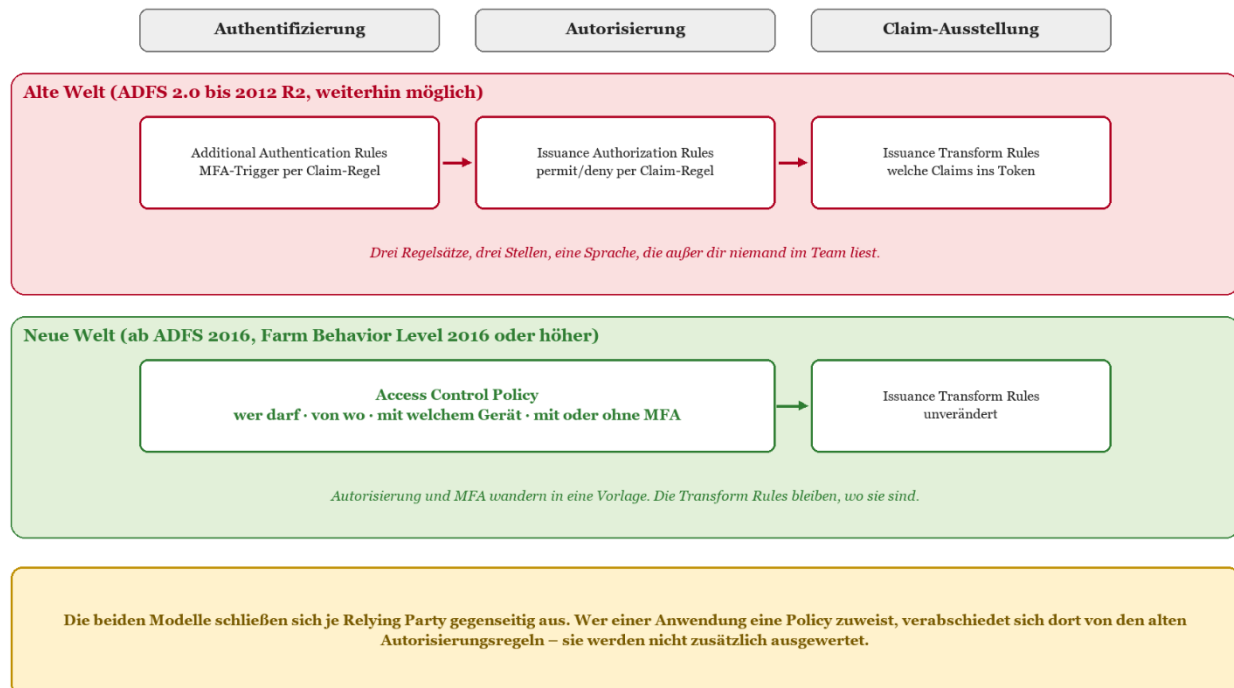
Alte Welt, neue Welt: Was Access Control Policies eigentlich ersetzen

Die ADFS-Pipeline hat drei Phasen: Authentifizierung, Autorisierung und Claim-Ausstellung. Bis ADFS 2012 R2 hast du jede davon mit eigenen Regeln bedient. Ob MFA nötig ist, entschieden die Additional Authentication Rules. Ob überhaupt ein Token ausgestellt wird, entschieden die Issuance Authorization Rules. Welche Claims drinstehen, regelten die Issuance Transform Rules. Drei Regelsätze in derselben Claim-Regelsprache, und gegenseitige Abhängigkeiten musstest du im Kopf behalten. Wie die Regelsprache funktioniert, erklärt der [ADFS Claim Rules Leitfaden](#) ausführlich – hier brauchst du nur die Erkenntnis, dass sie für einfache Zugriffsfragen überdimensioniert ist.

Access Control Policies fassen die ersten beiden Phasen zusammen. Eine Policy beantwortet: Wer darf, von wo, mit welchem Gerät, und muss er oder sie dafür einen zweiten Faktor vorzeigen? Die Transform Rules bleiben unberührt, denn welche Claims eine Anwendung braucht, ist eine andere Frage als die, wer sie benutzen darf.

Skizze 1: Zwei Welten in derselben Pipeline

Pro Relying Party gilt entweder die alte Regelwelt oder eine Access Control Policy – nie beides



Skizze 1: Alte Regelwelt und Access Control Policy in derselben ADFS-Pipeline

So sah es früher aus

Damit du später weißt, was du übersetzt, hier die drei Klassiker der alten Welt. Erstens: Nur Mitglieder einer Gruppe dürfen rein. Zweitens: Von außen gar nicht. Drittens: Von außen nur mit MFA – das stand nicht in den Autorisierungsregeln, sondern in den Additional Authentication Rules.

```
# 1) Issuance Authorization Rule: nur eine Gruppe (per SID)
c:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/groupsid",
  Value == "S-1-5-21-1111111111-2222222222-3333333333-4711"]
=> issue(Type = "http://schemas.microsoft.com/authorization/claims/permit", Value = "true");

# 2) Issuance Authorization Rule: Anfragen über den WAP abweisen
exists([Type == "http://schemas.microsoft.com/2012/01/requestcontext/claims/x-ms-proxy"])
=> issue(Type = "http://schemas.microsoft.com/authorization/claims/deny", Value = "true");

# 3) Additional Authentication Rule: MFA, wenn nicht im Firmennetz
c:[Type == "http://schemas.microsoft.com/ws/2012/01/insid corporatenetwork", Value == "false"]
=> issue(Type =
"http://schemas.microsoft.com/ws/2008/06/identity/claims/authenticationmethod",
  Value = "http://schemas.microsoft.com/claims/multipleauthn");
```

Drei typische Regeln der alten Welt – korrekt, funktionsfähig und für die Kollegen im Helpdesk eine Fremdsprache

Daran ist nichts falsch. Die Regeln tun, was sie sollen, und wer sie geschrieben hat, versteht sie auch. Das Problem ist der Betrieb: Jede Relying Party hat ihre eigene Kopie, jede Kopie altert anders, und nach fünf Jahren hast du vierzig leicht unterschiedliche Varianten von „nur Gruppe X, extern mit MFA“. Eine zentrale Änderung, etwa eine neue Ausnahmegruppe, bedeutet vierzig Mal dieselbe Operation am offenen Herzen.

Gegenüberstellung: Claim-Regeln gegen Policy

Aspekt	Issuance Authorization Rules (alt)	Access Control Policies (ab ADFS 2016)
Konfiguration	Claim-Regelsprache, je Relying Party	Bedingungen im Editor oder per PowerShell, als Vorlage
Wiederverwendung	Kopieren und Einfügen	Eine Policy, beliebig viele Relying Parties
Zentrale Änderung	Jede Relying Party einzeln anfassen	Policy ändern, wirkt sofort überall (nur ohne Parameter)
MFA	Separat in Additional Authentication Rules	Teil der Regel: „und MFA verlangen“
Netzwerkstandort	Claims x-ms-proxy bzw. insid corporatenetwork auswerten	Bedingung „aus dem Intranet“ oder „aus dem Extranet“
Gruppen	Meist per SID hart codiert	Gruppe auswählen oder als Parameter bei der Zuweisung
Ablehnung	Explizites deny, Reihenfolge beachten	Kein Deny, nur Permit plus Ausnahmen
Sonderlogik	Alles, was die Regelsprache hergibt	Begrenzt auf die angebotenen Bedingungen
Lesbarkeit	Für Eingeweihte	Fast als Satz lesbar
Voraussetzung	Jede ADFS-Version	Farm Behavior Level 2016 oder höher

Tabelle 1: Alte und neue Welt der ADFS-Autorisierung im direkten Vergleich

WICHTIG – Upgrade allein stellt nichts um

Wenn du eine Farm von 2012 R2 hochziehst, behalten alle bestehenden Relying Parties ihre alten Regeln. Die Access Control Policies stehen erst zur Verfügung, wenn das Farm Behavior Level angehoben ist – wie das ohne Ausfall geht, zeigt der Beitrag [ADFS-Farm upgraden – Farm Behavior Level anheben ohne Ausfall](#). Danach entscheidest du je Anwendung, ob und wann du umstellst. Niemand zwingt dich, alles an einem Wochenende zu migrieren.

Die mitgelieferten Vorlagen: MFA für extern, Gruppe, Netzwerkstandort

ADFS bringt eine Handvoll fertiger Policies mit, die die häufigsten Fälle abdecken. Sie lassen sich nicht bearbeiten, aber kopieren. In einer frischen Farm findest du unter „Zugriffssteuerungsrichtlinien“ beziehungsweise „Access Control Policies“ diese Einträge – die englischen Namen sind auch die, die du in PowerShell angibst:

Vorlage	Was sie tut	Parameter
Permit everyone	Jeder authentifizierte Benutzer bekommt ein Token	keiner
Permit everyone and require MFA	Jeder, aber immer mit zweitem Faktor	keiner
Permit everyone and require MFA from extranet access	Intern ohne, über den WAP mit MFA	keiner
Permit everyone and require MFA from unauthenticated devices	MFA, wenn das Gerät nicht registriert ist	keiner
Permit everyone and require MFA, allow automatic device registration	MFA mit Weg zur Geräteregistrierung	keiner
Permit everyone for intranet access	Nur aus dem internen Netz, extern kein Zugriff	keiner
Permit specific group	Nur Mitglieder der angegebenen Gruppe(n)	Gruppe
Permit everyone and require MFA for specific group	Alle dürfen, die angegebene Gruppe nur mit MFA	Gruppe

Tabelle 2: Mitgelieferte Access Control Policies – prüfe den Bestand deiner Farm mit `Get-AdfsAccessControlPolicy`

Welche Vorlagen in deiner Farm tatsächlich vorhanden sind und wie oft sie verwendet werden, siehst du schneller per PowerShell als im Snap-In:

```
# Alle Policies mit Kennzeichen und Verwendungszähler
Get-AdfsAccessControlPolicy |
  Sort-Object Name |
  Format-Table Name, IsBuiltIn, RpUsageCount -AutoSize

# Welche Relying Party nutzt was – und wer hängt noch an alten Regeln?
Get-AdfsRelyingPartyTrust |
  Select-Object Name, AccessControlPolicyName,
    @{ n = 'AlteAutorisierung'; e = { [bool]$_._IssuanceAuthorizationRules } } |
  Sort-Object AccessControlPolicyName, Name |
  Format-Table -AutoSize
```

Bestandsaufnahme: Policies und ihre Zuweisung auf einen Blick

MFA für extern: der häufigste Fall

„Permit everyone and require MFA from extranet access“ ist die Policy, die in den meisten Farmen den größten Teil der Anwendungen abdecken könnte. Intern meldet sich die Belegschaft per Kerberos still an, von außen kommt nach dem Kennwort der zweite Faktor. Voraussetzung ist, dass in der Farm überhaupt ein MFA-Anbieter als zusätzliche Authentifizierungsmethode aktiviert ist. Ohne Anbieter hat die Policy nichts, womit sie MFA verlangen könnte – wie du Entra-MFA als Adapter einbindest, steht im Beitrag [ADFS mit Entra-MFA koppeln – Adapter, Zertifikat und Fallstricke](#).

```
Set-AdfsRelyingPartyTrust -TargetName "Reisekosten" `
  -AccessControlPolicyName "Permit everyone and require MFA from extranet access"
```

MFA für Zugriffe über den Web Application Proxy – ein Einzeiler statt einer Additional Authentication Rule

WARNUNG — Extranet ist, was über den WAP kommt

Für ADFS ist eine Anfrage genau dann „extern“, wenn sie über einen Web Application Proxy eintrifft. Zeigt dein VPN-Client per Split-DNS auf die internen ADFS-Server, sind die VPN-Nutzer für jede Policy im Intranet und bekommen keine MFA-Abfrage. Umgekehrt landen interne Clients, die wegen

falschem DNS über den WAP gehen, plötzlich in der MFA. Bevor du einer Policy die Schuld gibst, prüfe die Namensauflösung von sts.contoso.de von beiden Seiten. Mehr zur Rolle des Proxys im Beitrag [Web Application Proxy: Architektur und Ablöse](#).

Gruppenbeschränkung: Permit specific group

Die zweite Standardfrage lautet: „Nur die Leute aus der Buchhaltung.“ Dafür gibt es die parametrisierte Vorlage „Permit specific group“. Bei der Zuweisung gibst du die Gruppe an – im Assistenten über die Gruppenauswahl, in PowerShell als Name. Liegt die Gruppe in der Domäne der ADFS-Farm, reicht der reine Gruppenname, ansonsten nimmst du die Form DOMÄNE\Gruppe.

```
# Eine Gruppe
Set-AdfsRelyingPartyTrust -TargetName "Reisekosten" `
  -AccessControlPolicyName "Permit specific group" `
  -AccessControlPolicyParameters "CONTOSO\APP-Reisekosten-Nutzer"

# Mehrere Gruppen (ODER-verknüpft)
Set-AdfsRelyingPartyTrust -TargetName "Reisekosten" `
  -AccessControlPolicyName "Permit specific group" `
  -AccessControlPolicyParameters ("CONTOSO\APP-Reisekosten-Nutzer", "CONTOSO\APP-Reisekosten-Freigabe")
```

Gruppenbeschränkung per Parameter – die Policy bleibt eine, die Gruppe wechselt je Anwendung

TIPP – Eine Gruppe pro Anwendung, nicht pro Abteilung

Weise Policies an Anwendungsgruppen zu (APP-Reisekosten-Nutzer) und nicht direkt an Abteilungsgruppen. Dann steuert der Fachbereich die Mitgliedschaft, und du musst die Relying Party nie wieder anfassen, wenn die Buchhaltung beschließt, dass die Revision auch mal reinschauen darf.

Netzwerkstandort: intern ja, extern nein

„Permit everyone for intranet access“ ist die Policy für Anwendungen, die von außen schlicht nicht erreichbar sein sollen – etwa ein Administrationsportal. Sie ersetzt die klassische x-ms-proxy-Deny-Regel. Feinere Unterscheidungen nach IP-Bereichen bildet der Netzwerkstandort nicht ab, er kennt im Kern nur innen und außen. Wer wirklich nach Client-IP-Adressen filtern muss, landet wieder bei Claim-Bedingungen oder bei Regeln – und sollte sich fragen, ob diese Logik nicht ohnehin besser in die Firewall gehört.

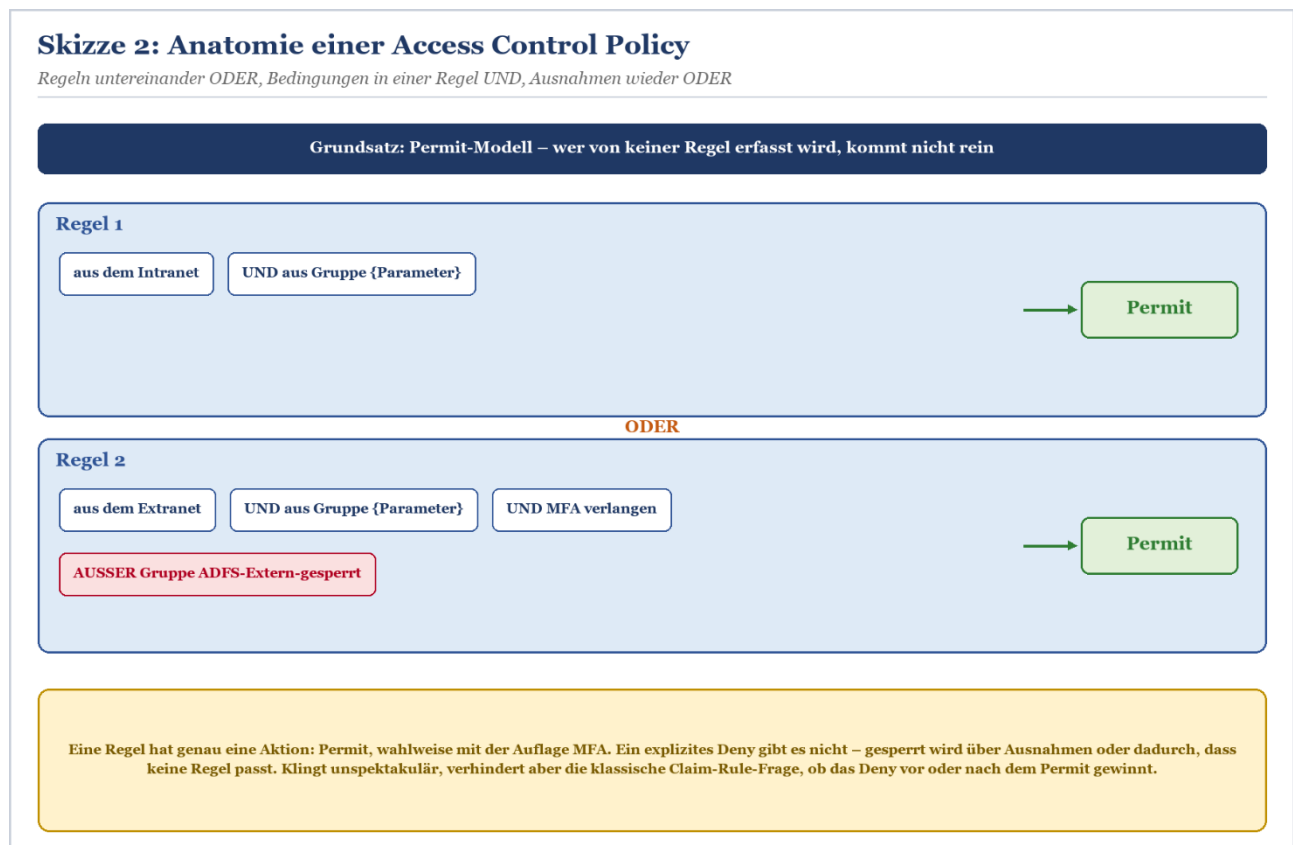
Anforderung	Alte Welt	Passende Policy
Externer Zugriff nur mit MFA	Additional Authentication Rule auf insidecorporatenetwork	Permit everyone and require MFA from extranet access
Nur eine Gruppe	Authorization Rule auf groupsid	Permit specific group
Extern gar nicht	Deny-Regel auf x-ms-proxy	Permit everyone for intranet access
Alle, eine Gruppe mit MFA	Additional Authentication Rule auf groupsid	Permit everyone and require MFA for specific group
Gruppe, extern mit MFA, Admins nie von außen	Mehrere Regeln in zwei Regelsätzen	Eigene Policy (nächstes Kapitel)

Tabelle 3: Typische Anforderungen und ihre Übersetzung

Eigene Access Control Policy mit Parametern bauen

Spätestens bei der Anforderung „Nur Gruppe X, von außen mit MFA, und die Domänen-Admins grundsätzlich nicht über das Internet“ reicht keine Vorlage mehr. Dann baust du eine eigene

Policy. Der Editor sieht aus wie eine Outlook-Regel aus dem Jahr 2010, und das ist ausdrücklich ein Kompliment: Du kannst ihn lesen.



Skizze 2: Aufbau einer Access Control Policy – Regeln, Bedingungen, Ausnahmen, Aktion

Bausteine im Editor

Bedingung („Permit users ...“)	Bedeutung	Als Parameter möglich
from specific network	Intranet oder Extranet, also direkt oder über den WAP	ja
from specific groups	Mitgliedschaft in einer oder mehreren AD-Gruppen	ja
from devices with specific trust levels	Gerät authentifiziert, verwaltet oder konform	ja
with specific claims in the request	Beliebiger Claim mit Operator und Wert, etwa Abteilung	ja
and require multi-factor authentication	Permit nur nach zweitem Faktor	–

Tabelle 4: Bedingungen im Policy-Editor; dieselben Kategorien stehen auch als Ausnahmen zur Verfügung

Jedes unterstrichene „specific“ im Editor ist ein Platzhalter. Du kannst ihn sofort mit einem Wert füllen oder „Parameter specified when the access control policy is assigned“ wählen. Im zweiten Fall wird der Wert erst bei der Zuweisung an eine Relying Party abgefragt. Daraus ergeben sich zwei Sorten von Policies mit sehr unterschiedlichem Verhalten im Betrieb.

Eigenschaft	Ohne Parameter	Mit Parametern
Werte	Fest in der Policy	Je Relying Party bei der Zuweisung
Nachträglich bearbeitbar	Ja – Änderung wirkt auf alle zugewiesenen RPs	Nein – neue Policy anlegen und umhängen
Typischer Einsatz	Farmweite Linie: „extern immer MFA“	Muster mit wechselnder Gruppe je Anwendung

Risiko	Ein Klick ändert viele Anwendungen gleichzeitig	Viele Zuweisungen mit unterschiedlichen Werten
---------------	---	--

Tabelle 5: Parametrisierte und nicht parametrisierte Policies im Vergleich

WARNUNG – Die Reichweite einer Änderung

Eine nicht parametrisierte Policy wirkt auf alle Relying Parties, denen sie zugewiesen ist – sofort, ohne Rückfrage. Bevor du an „Contoso – Standard extern“ eine Ausnahme ergänzt, schau dir an, wie viele Anwendungen daran hängen. Der Zähler RpUsageCount aus Get-AdfsAccessControlPolicy ist dafür der schnellste Blick, und ein frisches Backup der Konfiguration der zweitschnellste.

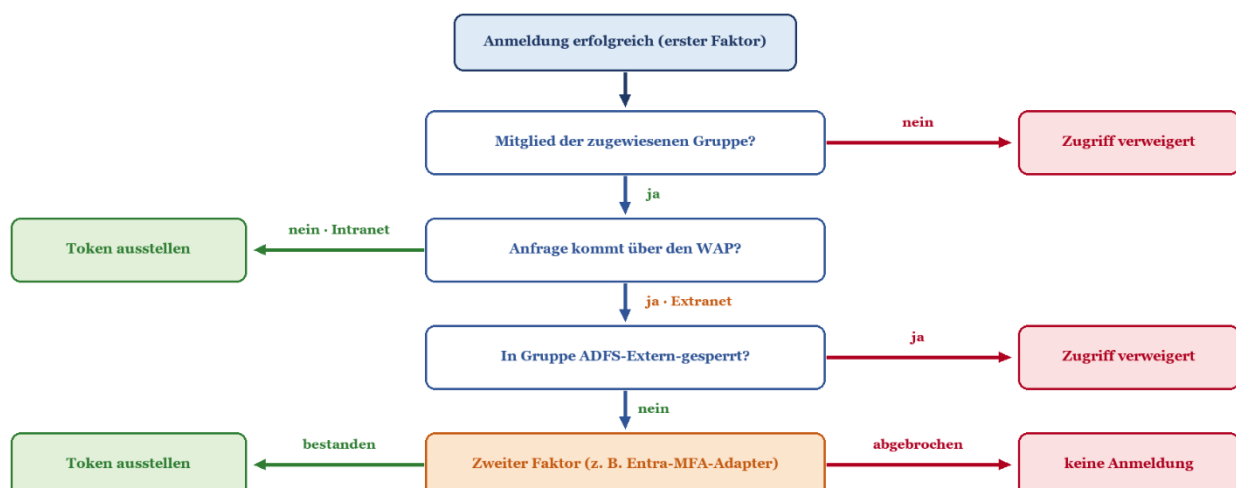
Beispiel: „Contoso – Gruppe, extern mit MFA“

Unser Beispiel soll für viele Fachanwendungen taugen: Zugriff nur für eine anwendungsspezifische Gruppe, die als Parameter übergeben wird. Intern ohne weitere Hürde, extern nur mit MFA. Und Mitglieder der Gruppe ADFS-Extern-gesperrt – privilegierte Konten, Dienstkonto mit interaktiver Anmeldung – kommen von außen überhaupt nicht rein, egal wie viele Faktoren sie vorzeigen.

- In der ADFS-Verwaltung „Access Control Policies“ auswählen und „Add Access Control Policy“ klicken.
- Name „Contoso – Gruppe, extern mit MFA“ und eine Beschreibung vergeben, die in einem Jahr noch jemand versteht.
- Regel 1 hinzufügen: „from specific network“ auf Intranet setzen, „from specific groups“ auf „Parameter specified when the access control policy is assigned“.
- Regel 2 hinzufügen: „from specific network“ auf Extranet, „from specific groups“ wieder als Parameter, dazu „and require multi-factor authentication“.
- In Regel 2 unter „Except“ die Option „from specific groups“ wählen und fest ADFS-Extern-gesperrt eintragen.
- Speichern, im Vorschaufenster den Satz einmal laut vorlesen. Klingt er falsch, ist er falsch.

Skizze 3: So entscheidet die Beispiel-Policy „Contoso – Gruppe, extern mit MFA“

Eine Anmeldung an sts.contoso.de: Gruppe, Netzwerk, Ausnahme, zweiter Faktor – und nur zwei Ausgänge



„Extranet“ heißt für ADFS: Die Anfrage kam über einen Web Application Proxy. Wer von zu Hause per VPN direkt auf die interne ADFS-Adresse zeigt, ist für die Policy im Intranet – und bekommt keine MFA-Abfrage. Das ist kein Fehler der Policy, sondern eines DNS-Designs.

Skizze 3: Entscheidungsweg der Beispiel-Policy bei einer Anmeldung

Die Ausnahme hängt bewusst nur an Regel 2. Würdest du sie an Regel 1 hängen, könnten die gesperrten Konten auch intern nicht mehr auf die Anwendung – das wäre ein anderes Sicherheitskonzept, und vermutlich eines, nach dem um 7:45 Uhr das Telefon klingelt. Weil Regeln ODER-verknüpft sind, reicht es, dass eine Regel ein Permit liefert. Die Ausnahme verhindert nur das Permit ihrer eigenen Regel.

Zuweisen per Assistent und per PowerShell

Im Snap-In wählst du die Relying Party, klickst auf „Edit Access Control Policy“, nimmst die neue Policy und bekommst für jeden Parameter einen Dialog zur Gruppenauswahl. Bei neuen Relying Parties fragt der Assistent die Policy direkt mit ab – Details dazu im Beitrag [Relying Party Trust anlegen – per Metadaten und von Hand](#). In PowerShell übergibst du die Werte über -AccessControlPolicyParameters. Hat eine Policy mehrere Parameter, erwartet das Cmdlet eine Hashtable, deren Schlüssel die Parameternamen aus der Policy sind. Den einfachsten Weg, an diese Namen zu kommen, liefert ADFS selbst: einmal per Assistent zuweisen, dann auslesen.

```
# Einmal per Assistent zugewiesen – jetzt die Parameterstruktur ansehen
$rp = Get-AdfsRelyingPartyTrust -Name "Reisekosten"
$rp.AccessControlPolicyName
$rp.AccessControlPolicyParameters
```

```
# Dieselbe Struktur per Skript auf weitere Anwendungen übertragen
# (Schlüsselnamen aus der Ausgabe oben übernehmen)
Set-AdfsRelyingPartyTrust -TargetName "Zeiterfassung" `
    -AccessControlPolicyName "Contoso – Gruppe, extern mit MFA" `
    -AccessControlPolicyParameters $parameterAusAusgabeOben
```

Erst ansehen, dann skripten: Parameternamen stammen aus der Policy, nicht aus deiner Fantasie

Für Policies mit genau einem Parameter genügt der einfache Wert, wie oben bei „Permit specific group“. Microsoft zeigt in der Cmdlet-Hilfe außerdem die Hashtable-Form für Claim-Bedingungen mit ClaimType, Operator und Value – praktisch, wenn deine Policy etwa nach Abteilung filtert.

Kopieren statt bearbeiten

Weil parametrisierte Policies nach dem Anlegen eingefroren sind und eingebaute Vorlagen ohnehin nicht geändert werden dürfen, ist Kopieren dein wichtigstes Werkzeug. New-AdfsAccessControlPolicy kann eine bestehende Policy als Quelle nehmen. Die Kopie bearbeitest du, testest sie an einer Pilotanwendung und hängst danach die übrigen Relying Parties um.

```
# Kopie einer bestehenden Policy anlegen
New-AdfsAccessControlPolicy -Name "Contoso – Gruppe, extern mit MFA v2" `
    -SourceName "Contoso – Gruppe, extern mit MFA" `
    -Description "v2: zusätzlich nur registrierte Geräte von extern"

# Pilot umhängen, Rest später
Set-AdfsRelyingPartyTrust -TargetName "Reisekosten-Test" `
    -AccessControlPolicyName "Contoso – Gruppe, extern mit MFA v2" `
    -AccessControlPolicyParameters $parameterAusAusgabeOben

# Alte Version erst löschen, wenn RpUsageCount bei 0 steht
Get-AdfsAccessControlPolicy -Name "Contoso – Gruppe, extern mit MFA" |
    Select-Object Name, RpUsageCount
```

Versionieren mit Kopien – der Weg um die Unveränderlichkeit herum

TIPP – Versionsnummer in den Namen

Ein „v2“ im Policy-Namen wirkt bürokratisch, rettet dich aber beim nächsten Audit. Du siehst auf einen Blick, welche Anwendung schon auf der neuen Linie ist, und das Zurückhängen auf v1 ist ein

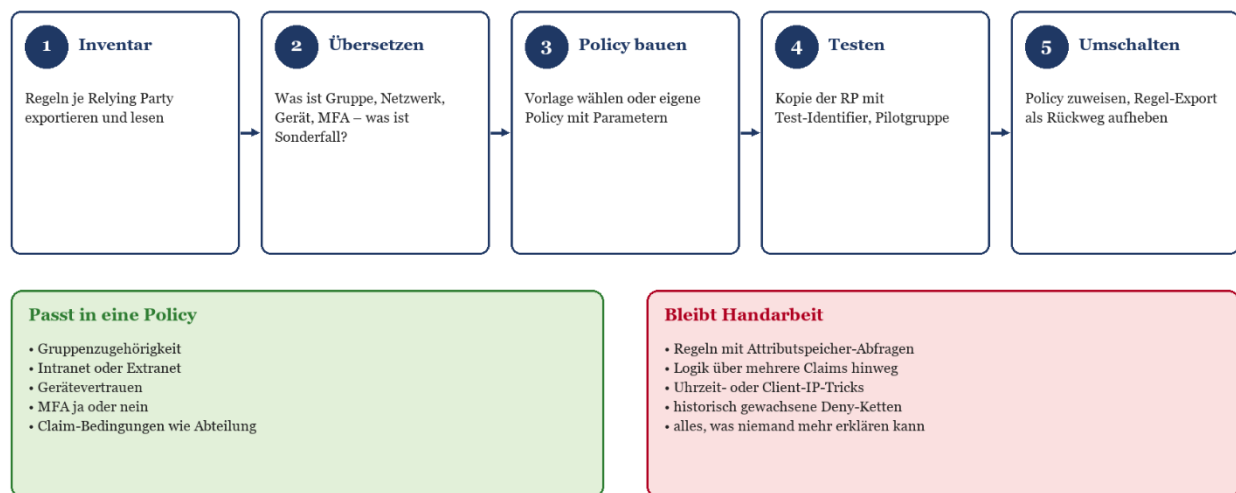
einzigster Befehl.

Umstellen ohne Montagsfrust: von alten Regeln zur Policy

Die Umstellung einer bestehenden Relying Party ist technisch ein Befehl und organisatorisch ein kleines Projekt. Der Befehl ist schnell getippt, das Projekt besteht darin, vorher zu wissen, was die alten Regeln eigentlich tun. Das ist nicht immer offensichtlich, und gelegentlich findest du dabei Regeln, die seit Jahren das Gegenteil dessen tun, was ihr Kommentar behauptet.

Skizze 4: Von alten Autorisierungsregeln zur Policy – in fünf Schritten

Erst lesen, dann nachbauen, dann an einer Test-Relying-Party beweisen



Skizze 4: Umstellung einer Relying Party in fünf Schritten

Alte Regeln sichern und lesen

```

$ziel = "C:\ADFS-Export\Regeln"
New-Item -ItemType Directory -Path $ziel -Force | Out-Null

Get-AdfsRelyingPartyTrust | ForEach-Object {
    $datei = Join-Path $ziel ($_.Name -replace '[\\/:*?"<>|]', '_')
    $_.IssuanceAuthorizationRules | Set-Content "$datei.authz.txt" -Encoding UTF8
    $_.AdditionalAuthenticationRules | Set-Content "$datei.mfa.txt" -Encoding UTF8
    $_.IssuanceTransformRules | Set-Content "$datei.transform.txt" -Encoding UTF8
}

```

Export aller Regelsätze je Relying Party – Rückweg und Lesestoff in einem

Mit dem Export hast du zweierlei: einen Rückweg, falls die Policy doch nicht das tut, was die Regeln taten, und das Material für die Übersetzung. Gehe jede Regel durch und ordne sie einer Kategorie zu: Gruppe, Netzwerk, Gerät, MFA, Claim-Bedingung – oder Sonderfall. Die Sonderfälle sind die interessanten. Wenn eine Anwendung eine Regel hat, die einen Attributspeicher abfragt oder Claims über mehrere Bedingungen verknüpft, passt sie nicht in eine Policy. Dann bleibt diese Anwendung vorerst in der alten Welt, und das ist völlig in Ordnung. Für den großen Rundumblick über alle Anwendungen lohnt der Beitrag [Relying Parties inventarisieren – die Bestandsaufnahme vor der Entra-Migration](#).

Umschalten und zurückschalten

```

# Umschalten: Policy zuweisen
Set-AdfsRelyingPartyTrust -TargetName "Reisekosten" `

```

```
-AccessControlPolicyName "Permit specific group" `
-AccessControlPolicyParameters "CONTOSO\APP-Reisekosten-Nutzer"

# Notfall: Policy entfernen und alte Regeln aus dem Export zurückspielen
Set-AdfsRelyingPartyTrust -TargetName "Reisekosten" -AccessControlPolicyName $null
Set-AdfsRelyingPartyTrust -TargetName "Reisekosten" `
-IssuanceAuthorizationRulesFile "C:\ADFS-Export\Regeln\Reisekosten.authz.txt" `
-AdditionalAuthenticationRulesFile "C:\ADFS-Export\Regeln\Reisekosten.mfa.txt"
```

Hin und zurück – das Entfernen der Policy mit \$null ist in der Microsoft-Hilfe dokumentiert

WARNUNG – Ohne Regeln kein Zugang

Wenn du die Policy entfernst und keine Autorisierungsregeln zurückspielst, hat die Relying Party weder das eine noch das andere. Im alten Modell bedeutet das: Kein Permit, kein Token. Das ist sicher, aber auch genau die Art Sicherheit, für die dich am nächsten Morgen niemand lobt. Spiel den Rückweg einmal in der Testumgebung durch, bevor du ihn in Produktion brauchst.

Testen, bevor es ernst wird

Bewährt hat sich eine Kopie der Relying Party mit einem eigenen Test-Identifizier, auf die eine kleine Pilotgruppe zugreift – das funktioniert bei vielen Anwendungen, aber nicht bei allen, weil manche nur einen Identifizier kennen. Teste immer beide Wege: einmal intern, einmal über den WAP, jeweils mit einem Konto in der Gruppe, einem außerhalb und einem in der Ausnahmegruppe. Neun Anmeldungen, zehn Minuten, und du weißt mehr als nach jeder Diskussion über die Regel.

Symptom	Wahrscheinliche Ursache	Prüfen
Externe Nutzer sehen keine MFA-Abfrage	Anfrage läuft nicht über den WAP (Split-DNS, VPN)	Namensauflösung von sts.contoso.de, WAP-Zugriffe im Log
Policy mit MFA lässt sich zuweisen, Anmeldung scheitert	Kein MFA-Anbieter als zusätzliche Methode aktiv	Authentifizierungsmethoden der Farm
Gruppenmitglied wird abgewiesen	Falsche Domäne beim Parameter, Mitgliedschaft noch nicht im Ticket	Parameter an der RP, Abmelden und neu anmelden
Policy taucht nicht auf	Farm Behavior Level noch unter 2016	Get-AdfsFarmInformation
Änderung trifft unerwartet viele Apps	Nicht parametrisierte Policy mehrfach zugewiesen	RpUsageCount vor der Änderung

Tabelle 6: Typische Stolpersteine nach der Umstellung

Wenn die Fehlerseite trotzdem nur „An error occurred“ sagt, liefert das Admin-Log der ADFS-Server mit der Activity-ID die eigentliche Begründung. Wie du die Einträge liest und das Debug-Tracing gezielt einschaltest, erklärt der Beitrag [ADFS-Ereignisprotokolle lesen – Admin-Log, Debug-Tracing und Auditing](#). Für die übrigen Ursachen fehlgeschlagener Anmeldungen hilft [ADFS-Anmeldung schlägt fehl – die häufigsten Ursachen und ihre Lösung](#).

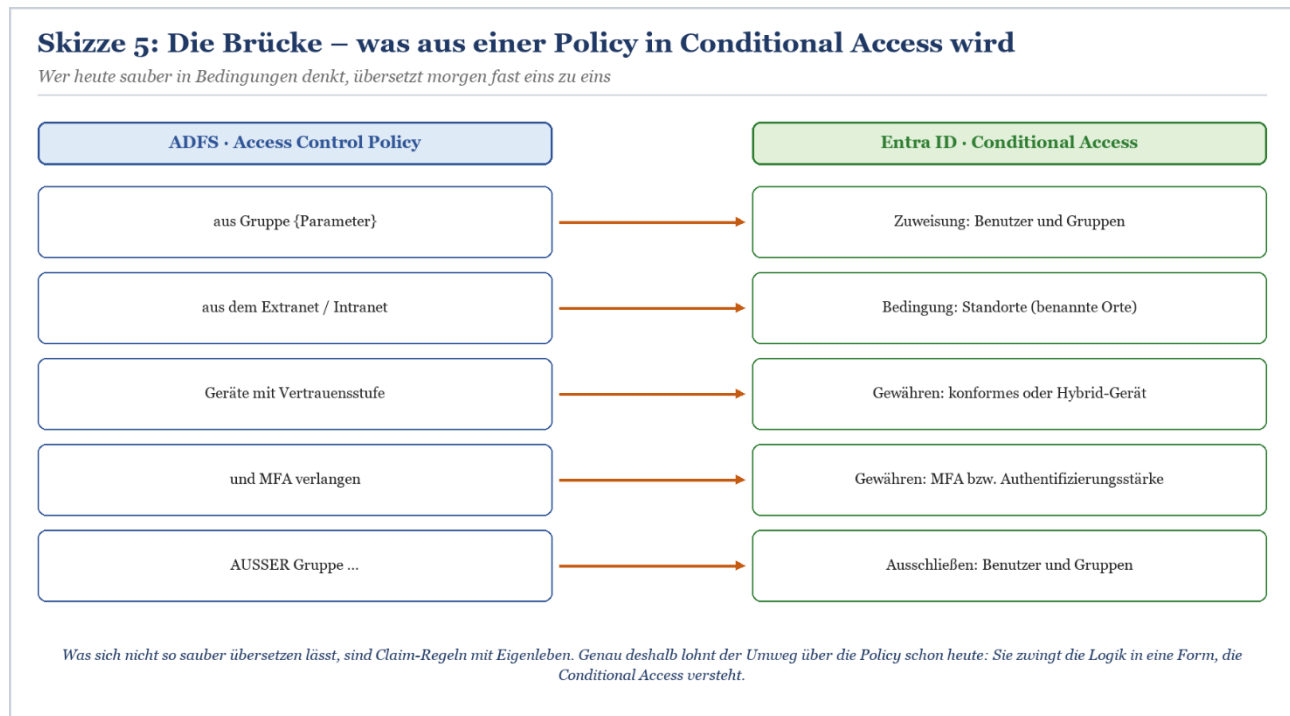
FAKTEN – Policies sind kein Ersatz für Härtung

Eine Access Control Policy entscheidet nach der Kennworteingabe. Gegen Passwort-Spray auf den Extranet-Endpunkt hilft sie nicht, dafür ist Extranet Smart Lockout zuständig – siehe [Extranet Smart Lockout – ADFS gegen Passwort-Spray schützen](#). Den Gesamtblick auf Angriffe und Gegenmaßnahmen liefert [ADFS-Sicherheit: Angriffe, Härtung, Golden SAML und MFA](#).

Die Brücke zu Conditional Access

ADFS läuft in vielen Unternehmen seit über zehn Jahren, und es gibt keinen Grund, eine stabile Farm in Panik abzureißen. Trotzdem planen die meisten früher oder später den Weg nach Entra

ID. Und genau hier zählt sich die Umstellung auf Access Control Policies ein zweites Mal aus: Eine Policy ist bereits in Bedingungen gedacht – Wer, Von wo, Welches Gerät, Welche Auflage. Das ist ziemlich genau die Struktur einer Conditional-Access-Richtlinie in Entra ID.



Skizze 5: Bausteine einer Access Control Policy und ihre Entsprechung in Conditional Access

Access Control Policy	Conditional Access	Anmerkung
from specific groups	Zuweisung: Benutzer und Gruppen	Gruppe muss in Entra ID vorhanden sein, etwa per Synchronisation
from Intranet / Extranet	Bedingung: Standorte	Aus „über den WAP“ werden benannte Orte mit IP-Bereichen
devices with trust level	Gewähren: konformes Gerät oder Hybrid-Join	Setzt Geräteverwaltung bzw. Registrierung voraus
require MFA	Gewähren: MFA oder Authentifizierungsstärke	Der MFA-Adapter in ADFS entfällt
Except ...	Ausschließen: Benutzer, Gruppen, Orte	Notfallkonten gehören hier immer dazu
Zuweisung an Relying Party	Zielressourcen: Unternehmensanwendung	Erst nach Umzug der Anwendung nach Entra ID

Tabelle 7: Übersetzung von Policy-Bausteinen in Conditional Access

Die Übersetzung im Detail – auch für die Regeln, die nicht in eine Policy gepasst haben – beschreibt der Geschwisterbeitrag [Claim Rules nach Conditional Access übersetzen](#). Dort geht es auch um die Fälle, die Conditional Access anders löst als ADFS, etwa weil „Extranet“ in der Cloud keine Eigenschaft der Anfrage mehr ist, sondern eine Frage der IP-Adresse. Den Überblick über die Zielplattform findest du auf der Seite [Microsoft Entra ID](#).

Wenn du schon jetzt wissen willst, was auf der Entra-Seite bereits existiert, reicht ein lesender Blick mit Microsoft Graph PowerShell. Die früher in jeder Anleitung gezeigten Module MSONline und AzureAD sind dafür nicht mehr der Weg.

```
Connect-MgGraph -Scopes "Policy.Read.All"
Get-MgIdentityConditionalAccessPolicy |
  Select-Object DisplayName, State |
  Sort-Object DisplayName
```

Bestehende Conditional-Access-Richtlinien nur lesen – als Abgleich zu deinen ADFS-Policies

TIPP — Policies jetzt, Migration später

Wer eine Anwendung heute von zwölf Zeilen Claim-Regeln auf „Permit specific group“ umstellt, hat damit die halbe Migrationsanalyse erledigt. Die Frage „Was tut diese Regel eigentlich?“ ist dann beantwortet, und sie ist im Migrationsprojekt die teuerste.

Fazit: Weniger Regelsprache, mehr Lesbarkeit

Access Control Policies sind kein revolutionäres Feature, sondern ein sehr vernünftiges. Sie ersetzen die Issuance Authorization Rules und die Additional Authentication Rules durch etwas, das man lesen, wiederverwenden und zentral ändern kann. Für die drei Standardfragen – MFA von extern, Gruppenbeschränkung, Netzwerkstandort – reichen die mitgelieferten Vorlagen. Für alles darüber baust du eigene Policies, mit Parametern dort, wo die Gruppe je Anwendung wechselt, und ohne Parameter dort, wo eine farmweite Linie gelten soll.

Die Umstellung bestehender Anwendungen folgt einem simplen Muster: Regeln exportieren, lesen, übersetzen, an einer Kopie testen, umschalten, Rückweg aufheben. Was nicht passt, bleibt in der alten Welt, bis die Migration nach Entra ID ansteht – und dann bist du für Conditional Access schon gut vorbereitet. Wenn du deine Regelbestände gemeinsam durchgehen und eine saubere Policy-Landschaft aufbauen willst, unterstützen wir dich im [Consulting zu ADFS \(Active Directory Federation Services\)](#). Wer Policies, Ausnahmen und MFA lieber im Labor zerlegt, ist in den [ADFS-Schulungen](#) richtig. Die Autorisierungslogik von ADFS, alte wie neue Welt, vertieft außerdem das Buch – mehr dazu unter [ADFS in der Praxis – das Buch im Detail](#).

WEITER — Passend zum Thema

- › [Claim Rules nach Conditional Access übersetzen](#) – der nächste Schritt nach der Policy
- › [ADFS Claim Rules Leitfaden](#) – die Regelsprache, die du hier ablöst
- › [ADFS mit Entra-MFA koppeln – Adapter, Zertifikat und Fallstricke](#) – ohne MFA-Anbieter keine MFA-Policy
- › [ADFS-Farm upgraden – Farm Behavior Level anheben ohne Ausfall](#) – Voraussetzung für Access Control Policies
- › [Relying Party Trust anlegen – per Metadaten und von Hand](#) – Policy-Auswahl beim Anlegen einer Anwendung
- › [ADFS sichern und wiederherstellen – Rapid Restore Tool in der Praxis](#) – bevor du eine Policy mit vielen Zuweisungen änderst

FAQ: Access Control Policies in ADFS

Ab welcher ADFS-Version gibt es Access Control Policies?

Ab ADFS unter Windows Server 2016. Die Farm muss dafür auf Farm Behavior Level 2016 oder höher laufen; in einer Farm, die noch auf 2012-R2-Niveau steht, siehst du sie nicht.

Kann ich Access Control Policies und Issuance Authorization Rules kombinieren?

Nicht an derselben Relying Party. Microsoft beschreibt die beiden Modelle als sich gegenseitig ausschließend. In derselben Farm kannst du aber problemlos Anwendungen mit Policy und Anwendungen mit alten Regeln betreiben.

Wie erzwingen wir MFA nur für externe Zugriffe?

Mit der Vorlage „Permit everyone and require MFA from extranet access“. Voraussetzung ist ein aktivierter MFA-Anbieter in der Farm und ein DNS-Design, bei dem externe Zugriffe tatsächlich über den WAP laufen.

Wie beschränke ich eine Anwendung auf eine AD-Gruppe?

Mit „Permit specific group“ und der Gruppe als Parameter, etwa per Set-AdfsRelyingPartyTrust -AccessControlPolicyName "Permit specific group" -AccessControlPolicyParameters "CONTOSO\APP-Nutzer".

Warum kann ich meine eigene Policy nicht mehr bearbeiten?

Weil sie Parameter enthält. Parametrisierte Policies sind nach dem Anlegen nicht mehr änderbar. Lege mit New-AdfsAccessControlPolicy -SourceName eine Kopie an, passe sie an und hänge die Relying Parties um.

Kann ich mit einer Policy ausdrücklich ablehnen?

Nein, das Modell kennt nur Permit. Gesperrt wird über Ausnahmen an einer Regel oder dadurch, dass keine Regel zutrifft.

Kann eine Policy nach IP-Adressbereichen unterscheiden?

Der Netzwerkstandort unterscheidet Intranet und Extranet, also direkten Zugriff und Zugriff über den WAP. Feinere IP-Logik bleibt Sache von Claim-Bedingungen, Regeln oder besser der Firewall.

Wie entferne ich eine Policy wieder von einer Relying Party?

Mit Set-AdfsRelyingPartyTrust -TargetName "App" -AccessControlPolicyName \$null. Danach musst du die alten Autorisierungsregeln zurückspielen, sonst bekommt niemand mehr ein Token.

Ändert eine Policy-Anpassung sofort alle zugewiesenen Anwendungen?

Bei Policies ohne Parameter ja, ohne weitere Rückfrage. Prüfe vorher mit Get-AdfsAccessControlPolicy, wie viele Relying Parties die Policy verwenden.

Hilft mir die Umstellung bei der Migration nach Entra ID?

Ja. Policy-Bausteine wie Gruppe, Standort, Gerät und MFA lassen sich weitgehend direkt in Conditional Access übertragen. Die Analyse, was eine Anwendung eigentlich verlangt, ist dann schon erledigt.