

ADFS hinter dem Load Balancer – Health Probe, SNI und Persistenz

WISSEN

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

› [Active Directory Federation Services](#)

BERATUNG

Load Balancer, WAP und Farm sauber aufeinander abstimmen – bevor der Health Check die Farm für tot erklärt.

› [Consulting zu ADFS](#)

SCHULUNG

Probe, SNI, Bindungen und Client-IP im Workshop an einer echten Farm mit Load Balancer durchspielen.

› [ADFS-Schulungen](#)

Es gibt eine besondere Sorte Montagmorgen. Die ADFS-Farm läuft, alle vier Server sind grün, der Dienst antwortet, das Ereignisprotokoll ist langweilig wie immer. Trotzdem kommt niemand an Microsoft 365 heran. Der Grund steht zwei Racks weiter: Der Load Balancer hat über das Wochenende ein Firmware-Update bekommen, prüft seine Real Server jetzt per HTTPS ohne Servernamen im Handshake und hat daraufhin alle Knoten für verstorben erklärt. Er meint es gut. Er schützt die Anwender vor Servern, die in Wahrheit kerngesund sind – und zwar vollständig.

Dieser Beitrag kümmert sich um genau diese Nahtstelle. Du erfährst, welchen Health-Probe-Endpunkt ADFS und der Web Application Proxy mitbringen und warum er auf Port 80 läuft, wo SNI bei älteren Load Balancern zuschlägt, ob du Persistenz brauchst (Spoiler: fast nie), warum SSL-Bridging für ADFS keine Option ist und wie die echte Client-IP bis zu Extranet Smart Lockout durchkommt. Das Praxisbeispiel nutzt einen Kemp LoadMaster, die Regeln gelten aber für jedes Gerät. Grundlagen zu Farm, Proxy und Vertrauensstellungen setzen wir voraus; die findest du auf der Pillar-Seite [Active Directory Federation Services](#) und im Beitrag [ADFS: Architektur, Einsatzszenarien und Hochverfügbarkeit](#).

FAKTEN – Die Kurzfassung für Eilige

Health Probe: HTTP (nicht HTTPS) auf Port 80, Pfad /adfs/probe – auf ADFS-Servern und auf dem Web Application Proxy. Die Antwort ist 200 OK und wird lokal erzeugt, ohne Abhängigkeit zu Back-End-Diensten. Unter Windows Server 2012 R2 setzt der Endpunkt das Update-Rollup vom August 2014 (KB2975719) voraus.

TLS: Der Load Balancer darf TLS nicht terminieren. Microsoft unterstützt das für keinen ADFS-Anwendungsfall, weil Zertifikatsanmeldung und das Proxy-Vertrauen zwischen WAP und ADFS daran zerbrechen.

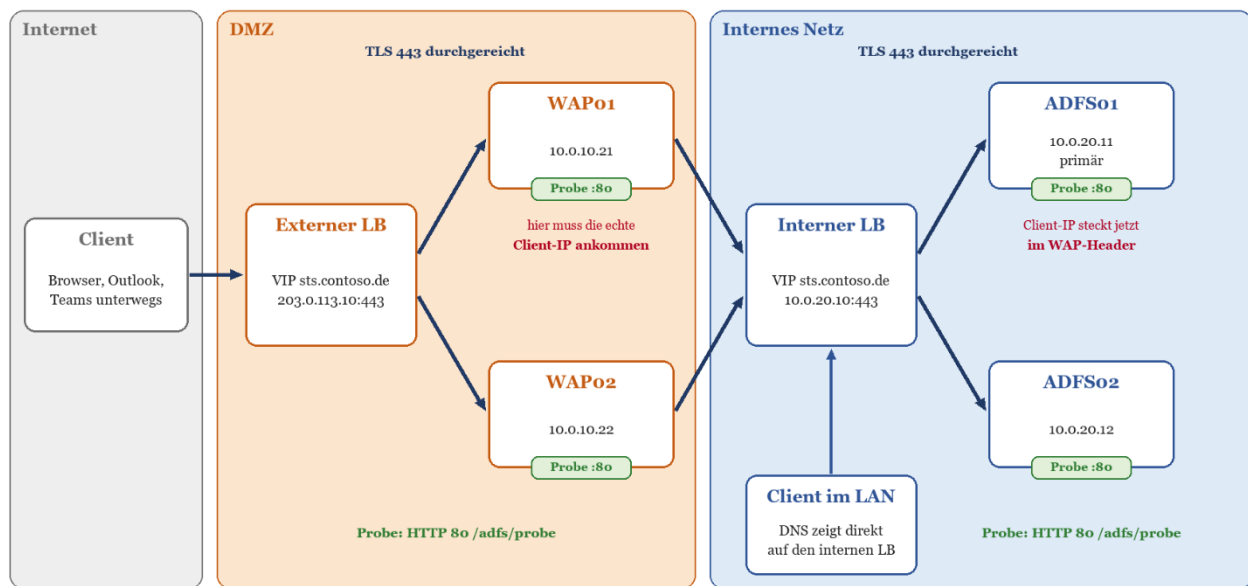
SNI: Der Load Balancer sollte SNI unterstützen. Wenn nicht, hilft eine o.o.o.o-Rückfallbindung auf ADFS- oder WAP-Server.

Persistenz: IP-basierte Session Affinity oder Sticky Sessions empfiehlt Microsoft für ADFS nicht. ADFS ist für die Anmeldung zustandslos.

Client-IP: Die Quelladresse des Clients muss beim WAP ankommen. Kann der Load Balancer sie nicht erhalten, muss er sie in X-Forwarded-For eintragen – sonst arbeiten Extranet Smart Lockout und gesperrte IP-Adressen mit falschen Daten.

Skizze 1: Der Weg einer Anmeldung durch zwei Load Balancer

Client, externer LB, WAP, interner LB, ADFS – und was an jeder Station passieren muss



Zwei Load Balancer, dieselbe Regel: TLS nicht aufbrechen, per HTTP auf Port 80 prüfen, keine Persistenz erzwingen. Der Unterschied liegt bei der Client-IP: Vor dem WAP entscheidet sie über Extranet Lockout, hinter dem WAP trägt sie der Proxy ohnehin im Header mit.

Skizze 1: Datenfluss vom Client über den externen Load Balancer zum WAP, weiter über den internen Load Balancer zur ADFS-Farm. Beide Load Balancer reichen TLS durch und prüfen per HTTP auf Port 80.

Der Datenfluss: zwei Load Balancer, zwei Aufgaben

In einer typischen Umgebung stehen zwei Load Balancer im Weg einer Anmeldung, auch wenn sie physisch dasselbe Gerät mit zwei virtuellen Diensten sein können. Der externe verteilt Anfragen aus dem Internet auf die WAP-Server in der DMZ. Der interne verteilt Anfragen auf die ADFS-Knoten – und zwar sowohl die, die vom WAP kommen, als auch die der Clients im LAN. Beide tragen denselben Namen: sts.contoso.de. Nur die DNS-Auflösung unterscheidet sich je nachdem, wo der anfragende Client steht.

Außen: vor dem Web Application Proxy

Der externe Load Balancer ist die Stelle, an der Anmeldungen aus dem Internet eintreffen – legitime ebenso wie die eines Botnetzes, das gerade Passwort-Spray übt. Hier entscheidet sich, welche Quell-IP der WAP zu sehen bekommt. Der WAP schreibt genau diese Adresse in den Header x-ms-forwarded-client-ip, den ADFS für Extranet Smart Lockout, Zugriffsrichtlinien und die Protokollierung auswertet. Was hier verloren geht, kommt später nicht zurück.

Innen: zwischen WAP und ADFS

Der WAP muss sts.contoso.de auf die interne VIP auflösen – nicht auf die externe, sonst schickt er Anfragen im Kreis. In der DMZ geschieht das über einen eigenen DNS-Server oder, deutlich häufiger, über die HOSTS-Datei. Zwischen WAP und ADFS läuft außerdem das Proxy-Vertrauen: Der WAP authentifiziert sich per Client-Zertifikat an der Farm. Deshalb gilt innen dieselbe Regel wie außen: TLS durchreichen, nicht aufbrechen. Für die internen Clients ist zudem wichtig, dass der Dienstname als A-Eintrag auf die VIP zeigt und nicht als CNAME – sonst fällt die Windows-

integrierte Anmeldung auf das Formular zurück. Das ganze Thema vertieft [SPN, DNS und Kerberos für ADFS – warum die interne Anmeldung auf Formular zurückfällt](#).

Thema	Externer LB (vor dem WAP)	Interner LB (vor ADFS)
Clients	Internet, Mobilgeräte, Homeoffice	WAP-Server und Clients im LAN
Real Server	WAP01, WAP02 auf 443	ADFS01, ADFS02 auf 443
Health Probe	HTTP 80, /adsf/probe auf jedem WAP	HTTP 80, /adsf/probe auf jedem ADFS-Knoten
TLS	durchreichen	durchreichen – Proxy-Vertrauen per Client-Zertifikat
Persistenz	keine	keine
Client-IP	muss erhalten bleiben oder in X-Forwarded-For stehen	SNAT unkritisch für Extranet-Anfragen, für LAN-Clients aber sichtbar
Namensauflösung	öffentliches DNS zeigt auf die externe VIP	internes DNS und HOSTS-Datei der WAPs zeigen auf die interne VIP

Tabelle 1: Was der externe und der interne Load Balancer jeweils leisten müssen

WICHTIG – Der WAP ist kein beliebiger Reverse Proxy

Wer den Web Application Proxy durch ein anderes Produkt ersetzen will, braucht einen Proxy, der das Protokoll MS-ADFSPIP beherrscht. Ein normaler Reverse Proxy oder ein Load Balancer mit hübscher Weboberfläche reicht nicht, weil ADFS vom Proxy erwartet, dass er sich per Zertifikat ausweist und die Client-IP im vorgesehenen Header mitliefert.

Die Architektur und die Frage, womit du den WAP irgendwann ablöst, behandelt [Web Application Proxy: Architektur und Ablöse](#).

Health Probe: /adsf/probe auf Port 80

Der Pfad ist schnell erklärt: Jeder ADFS-Server und jeder WAP beantwortet HTTP-Anfragen an /adsf/probe auf Port 80 mit 200 OK. Das funktioniert über den Servernamen, die IP-Adresse oder den Dienstnamen – also etwa adfs01.contoso.local/adsf/probe oder 10.0.20.11/adsf/probe, jeweils per HTTP. Unverschlüsselt, ohne Zertifikat, ohne SNI. Genau darum geht es: Microsoft empfiehlt den HTTP-Endpunkt ausdrücklich, damit der Health Check nicht an TLS-Details scheitert.

TIPP – Port 80 ist kein Sicherheitsproblem

Über Port 80 läuft ausschließlich die Probe. Anmeldeseiten, Token-Endpunkte und Metadaten gibt es bei ADFS nur über HTTPS; die Cookies sind als secure markiert. Port 80 muss auch nicht aus dem Internet erreichbar sein – nur vom Load Balancer aus. Beschränke die Firewall-Regel auf dessen Adressen, und die Sicherheitsabteilung kann weiterschlafen.

Was die Probe kann – und was nicht

Die Probe beantwortet genau eine Frage: Läuft der Dienst auf diesem Server und nimmt Anfragen an? Ob die Konfigurationsdatenbank erreichbar ist, ein Domänencontroller antwortet oder das Token-Signing-Zertifikat noch gültig ist, prüft die Probe nicht. Das ist Absicht. Würde die Probe von Back-End-Diensten abhängen, würde ein kurzer Aussetzer eines Domänencontrollers alle Knoten gleichzeitig aus dem Pool werfen – und aus einer leichten Störung würde ein Totalausfall.

Die Kehrseite: Ein WAP, dessen Proxy-Vertrauen abgelaufen ist, kann dem Load Balancer weiterhin fröhlich 200 OK melden, obwohl er keine einzige Anmeldung mehr durchreicht. Für solche Fälle brauchst du ein echtes Monitoring mit synthetischen Anmeldungen oder zumindest einem Abruf der Federation-Metadaten über den externen Namen. Was dafür infrage kommt, steht in [ADFS überwachen – Health Checks, Diagnose-Toolbox und Entra Connect Health](#); was beim

abgelaufenen Proxy-Vertrauen hilft, in [WAP-Proxy-Vertrauen erneuern – wenn der Web Application Proxy plötzlich nicht mehr will](#).

Skizze 2: Zwei Arten, einen gesunden Server für tot zu erklären

Die HTTP-Probe auf Port 80 fragt nur, ob ADFS lebt – die HTTPS-Probe ohne SNI fragt, ob du Glück hast

Empfohlen: HTTP-Probe



Kein Zertifikat, kein Handshake, keine SNI-Frage.
Die Antwort kommt lokal vom Dienst, ohne Abhängigkeit zu Datenbank oder Domänencontroller.
Läuft der ADFS-Dienst nicht, kommt keine 200 – und der Knoten fliegt raus.

Falle: HTTPS-Check ohne SNI



http.sys findet keine passende Bindung und bricht ab.
Der LB markiert alle Knoten als ausgefallen, obwohl jeder Browser problemlos anmelden könnte.
Ergebnis: Totalausfall durch Übervorsicht.

Die Probe misst, ob der Dienst antwortet – nicht, ob eine Anmeldung klappt. Für Letzteres gibt es synthetische Tests im Monitoring. Wer beides verwechselt, wird entweder zu früh oder zu spät geweckt.

Skizze 2: Die HTTP-Probe auf Port 80 kommt ohne Handshake aus. Ein HTTPS-Check ohne SNI trifft auf http.sys ohne passende Bindung – und erklärt einen gesunden Server für ausgefallen.

Die Probe von Hand testen

Bevor du den Load Balancer anfasst, prüfe die Probe von einem System aus, das im selben Netz wie der Load Balancer steht. Das Windows-eigene curl.exe ist dafür hervorragend geeignet, weil es dir auch den Handshake zeigt, wenn es später um SNI geht.

```

# Probe auf jedem Knoten direkt abfragen – erwartet wird StatusCode 200
foreach ($n in 'adfs01.contoso.local','adfs02.contoso.local','10.0.10.21','10.0.10.22') {
    try {
        $r = Invoke-WebRequest -Uri ("http://{0}/adfs/probe" -f $n) -UseBasicParsing -
        TimeoutSec 5
        "{0,-24} {1}" -f $n, $r.StatusCode
    } catch {
        "{0,-24} FEHLER: {1}" -f $n, $_.Exception.Message
    }
}

# Ist Port 80 überhaupt offen?
Test-NetConnection -ComputerName adfs01.contoso.local -Port 80

# Läuft der Dienst? (auf dem ADFS-Server bzw. dem WAP)
Get-Service adfssrv # ADFS
Get-Service approxysvc # Web Application Proxy
    
```

Listing 1: Probe-Endpoint auf ADFS und WAP prüfen. Liefert ein Knoten keine 200, ist er für den Load Balancer zu Recht außer Dienst.

HEAD, HTTPS und andere Kreativität

Viele Load Balancer bieten für Health Checks eine Auswahl an Methoden und Protokollen. Die Versuchung ist groß, das „sicherste“ zu nehmen: HTTPS, am besten gleich auf die Anmeldeseite. Leider ist das bei ADFS die unsicherste Variante – nicht für die Daten, sondern für die Verfügbarkeit.

Prüfmethode	Was passiert	Bewertung
HTTP GET auf Port 80,	200 OK, lokal erzeugt, kein TLS	empfohlen

/adfs/probe		
HTTP HEAD auf /adfs/probe	ADFS unterstützt HEAD-Anfragen laut Microsoft grundsätzlich nicht; Antworten können unerwartet oder verzögert sein	vermeiden, auf GET umstellen
HTTPS auf 443 ohne SNI	http.sys findet keine Hostname-Bindung, Handshake bricht ab	Totalausfall vorprogrammiert
HTTPS auf 443 mit SNI, /adfs/ls/	funktioniert, erzeugt aber Last und ggf. Einträge im Protokoll	unnötig, fehleranfällig
Reiner TCP-Check auf 443	sieht nur, ob der Port offen ist	zu grob – ein hängender Dienst bleibt im Pool
ICMP-Ping	sieht nur, ob der Server eingeschaltet ist	Dekoration

Tabelle 2: Health-Check-Varianten für ADFS und WAP im Vergleich

WARNUNG — Der Check auf die Anmeldeseite

Ein Health Check, der alle paar Sekunden die Seite /adfs/ls/idpinitiatedsignon.aspx per HTTPS abrufen, sieht auf den ersten Blick gründlich aus. In Wahrheit hängt er an Zertifikat, SNI, einer Einstellung, die unter Windows Server 2016 und später standardmäßig ausgeschaltet ist, und im ungünstigsten Fall an Back-End-Diensten. Fällt eine davon aus, fliegen alle Knoten gleichzeitig raus. Wenn du eine Ende-zu-Ende-Prüfung willst: gern – aber im Monitoring, das dich alarmiert, nicht im Load Balancer, der dann eigenmächtig abschaltet.

SNI und TLS: Passthrough statt Bridging

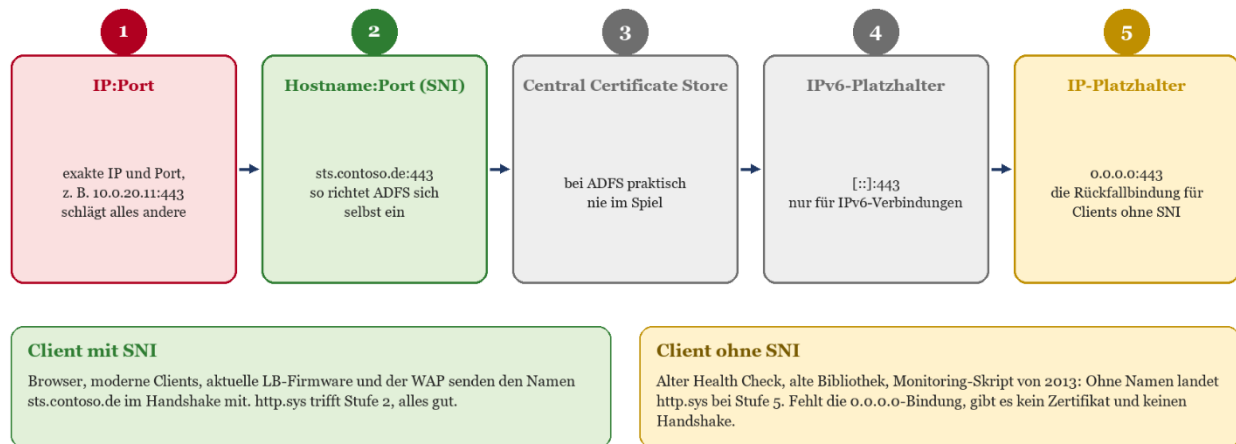
Seit Windows Server 2012 R2 bindet ADFS sein Zertifikat in http.sys nicht mehr an eine IP-Adresse, sondern an den Hostnamen: sts.contoso.de:443, dazu localhost:443 und bei Zertifikatsanmeldung weitere Bindungen. Das ist sauber, setzt aber voraus, dass der Client im TLS-Handshake mitteilt, welchen Namen er sprechen will. Das ist Server Name Indication, kurz SNI. Browser tun das seit Jahren, Microsoft nennt SNI ausdrücklich als Browser-Anforderung für ADFS.

Warum ältere Load Balancer stolpern

Das Problem sind selten die Browser, sondern alles dazwischen: Load Balancer mit älterer Firmware, deren HTTPS-Health-Check keinen Namen sendet. Monitoring-Skripte, die sich per IP-Adresse verbinden. Alte Bibliotheken in Fachanwendungen. Alle landen bei http.sys ohne Namen – und http.sys hat ohne Namen keine passende Bindung, also kein Zertifikat, also keinen Handshake. Aus Sicht des Clients sieht das aus wie ein Server, der die Verbindung grußlos zuknallt.

Skizze 3: Wie http.sys das Zertifikat auswählt

Reihenfolge der TLS-Bindungen – und wo die o.o.o.o.-Rückfallbindung ins Spiel kommt



Eine IP-spezifische Bindung (Stufe 1) ist der klassische Querschläger: Ein anderes Programm legt sie an, und plötzlich liefert der ADFS-Server auf 443 ein fremdes Zertifikat aus. netsh http show sslcert zeigt es.

Skizze 3: http.sys prüft die TLS-Bindungen in fester Reihenfolge. Clients ohne SNI landen bei der o.o.o.o.-Rückfallbindung – oder, wenn es keine gibt, im Nichts.

```

# Welche TLS-Bindungen gibt es auf diesem Server?
netsh http show sslcert

# Handshake mit SNI (so wie ein Browser)
$fs = 'sts.contoso.de'; $ip = '10.0.20.11'
curl.exe -v --resolve "${fs}:443:$ip" "https://$fs/FederationMetadata/2007-06/FederationMetadata.xml" -o NUL

# Handshake ohne SNI (so wie ein alter Health Check)
curl.exe -vk "https://$ip/" -o NUL
    
```

Listing 2: SNI-Verhalten eines ADFS-Knotens testen. Bricht nur der zweite Aufruf im Handshake ab, fehlt die Rückfallbindung – und du weißt, warum der alte Health Check scheitert.

Die o.o.o.o.-Rückfallbindung

Kann dein Load Balancer kein SNI und lässt er sich nicht auf die HTTP-Probe umstellen, bietet Microsoft eine Umgehung an: eine zusätzliche Bindung auf o.o.o.o.:443. Diese Bindung greift nur, wenn keine genauere Bindung passt, also genau für Clients ohne Servernamen. Die Application ID des ADFS-Dienstes ist {5d89a20c-beab-4389-9447-324788eb944a}; auf dem WAP übernimmst du die Application ID aus der vorhandenen Hostname-Bindung, die dir netsh http show sslcert anzeigt.

```

# Auf jedem ADFS-Knoten: aktuelles SSL-Zertifikat ermitteln
Get-AdfsSslCertificate

# Rückfallbindung für Clients ohne SNI anlegen (Thumbprint ersetzen)
netsh http add sslcert ipport=0.0.0.0:443 certhash=<THUMBPRINT>appid="{5d89a20c-beab-4389-9447-324788eb944a}" certstorename=MY

# Auf dem WAP: Thumbprint und Application ID der bestehenden Bindung ablesen
Get-WebApplicationProxySslCertificate
netsh http show sslcert hostnameport=sts.contoso.de:443
    
```

Listing 3: o.o.o.o.-Rückfallbindung anlegen. Das ist eine Krücke für alte Geräte, keine Dauerlösung – und sie muss bei jedem Zertifikatswechsel mitgedacht werden.

WARNUNG — Die vergessene Bindung beim Zertifikatswechsel

Die Rückfallbindung gehört nicht zu dem, was ADFS von sich aus einrichtet. Wenn du nächstes Jahr das SSL-Zertifikat tauschst, zeigt sie womöglich weiter auf das alte Zertifikat. Die Browser merken nichts, weil sie per SNI die richtige Bindung treffen – der alte Health Check dagegen bekommt ein abgelaufenes Zertifikat und schaltet die Farm ab. Am liebsten an einem Freitagabend.

Nimm die Bindung deshalb in dein Runbook für [SSL-Zertifikat auf ADFS und Web Application Proxy tauschen](#) auf. Oder besser: Stell den Health Check auf die HTTP-Probe um, dann brauchst du die Krücke gar nicht.

Und falls netsh http show sslcert eine IP-spezifische Bindung wie 10.0.20.11:443 zeigt, die du nicht angelegt hast: Die schlägt alle anderen Bindungen. Finde heraus, welches Programm sie angelegt hat, bevor du sie löschst – sonst kommt sie beim nächsten Neustart wieder.

SSL-Bridging gegen Passthrough

Viele Load Balancer können TLS terminieren, den Inhalt prüfen, Header einfügen und die Verbindung zum Real Server neu verschlüsseln. Das heißt je nach Hersteller SSL-Bridging, Re-Encryption oder Offloading mit Rückverschlüsselung. Für Webshops ist das großartig. Für ADFS ist es ein Ausschlusskriterium: Microsoft schreibt, dass der Load Balancer TLS nicht terminieren darf und dass das für keinen Anwendungsfall unterstützt wird.

Der Grund ist nicht Pedanterie, sondern Kryptografie. Bei der Zertifikatsanmeldung und bei der Geräteauthentifizierung legt der Client sein Zertifikat im TLS-Handshake vor. Endet der Handshake am Load Balancer, sieht ADFS das Zertifikat nie. Das Proxy-Vertrauen zwischen WAP und ADFS funktioniert genauso – wer innen bridged, kappt dem WAP die Verbindung zur Farm. Und Extended Protection, das bei ADFS standardmäßig aktiv ist, erwartet, dass Proxy und ADFS denselben Schlüssel verwenden und die TLS-Sitzung nicht von einem Dritten neu aufgebaut wird.

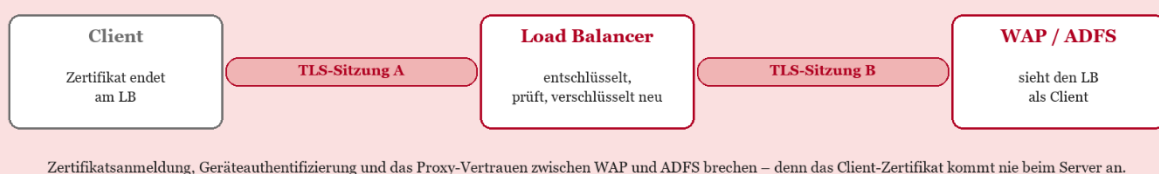
Skizze 4: SSL-Passthrough und SSL-Bridging im Vergleich

Oben reicht der LB den Tunnel durch, unten baut er einen neuen – und ADFS merkt den Unterschied

Passthrough (Layer 4 oder Layer 7 ohne TLS-Aufbruch): unterstützt



Bridging / Offloading (LB terminiert TLS und baut neu auf): nicht unterstützt



Der LB kann beim Bridging hübsch Header einfügen und Cookies lesen. Dafür kann ADFS danach weniger. Microsoft unterstützt das Terminieren am LB für keinen ADFS-Anwendungsfall.

Skizze 4: Beim Passthrough läuft ein TLS-Tunnel vom Client bis zum Server. Beim Bridging gibt es zwei Sitzungen – und alles, was an der ersten hängt, erreicht ADFS nie.

Aspekt	Passthrough	Bridging / Re-Encryption
Wo endet TLS?	auf WAP bzw. ADFS	auf dem Load Balancer, danach neue

		Sitzung
Zertifikat auf dem LB	nicht nötig	nötig, inklusive privatem Schlüssel
Client-Zertifikatsanmeldung	funktioniert	bricht
Proxy-Vertrauen WAP zu ADFS	funktioniert	bricht, wenn innen gebridged wird
Header einfügen (X-Forwarded-For)	nicht möglich, Verkehr ist verschlüsselt	möglich
Cookie-Persistenz	nicht möglich	möglich
Support durch Microsoft	ja	für keinen Anwendungsfall

Tabelle 3: Passthrough und Bridging aus Sicht von ADFS

Die Tabelle zeigt auch das Dilemma, das viele erst beim Praxistest bemerken: Die bequemen Funktionen – Header einfügen, Cookies lesen – gibt es nur beim Bridging. Beim Passthrough sieht der Load Balancer ausschließlich TCP und den SNI-Namen. Das hat direkte Folgen für die Frage, wie die Client-IP bis zum WAP kommt. Dazu gleich mehr.

Persistenz und Client-IP

Persistenz: in aller Regel nein

ADFS ist für die Anmeldung zustandslos. Was zwischen zwei Schritten einer Anmeldung gebraucht wird, steckt in Cookies und Formularfeldern, die jeder Knoten der Farm lesen kann. Ein Client kann die Anmeldeseite von ADFS01 bekommen, das Kennwort an ADFS02 schicken und die Weiterleitung von ADFS01 erhalten – es ist allen Beteiligten egal. Persistenz ist deshalb nicht nötig.

Persistenz ist sogar schädlich, zumindest in der IP-basierten Variante. Microsoft rät ausdrücklich von IP-Affinität ab: Bei Legacy-Authentifizierung für Exchange Online kommen viele Anfragen aus wenigen Microsoft-Adressbereichen, und eine Quell-IP-Persistenz nagelt sie alle auf denselben Knoten. Der arbeitet dann für die ganze Farm, während die anderen Kaffee trinken.

Persistenzmodus	Mit Passthrough möglich?	Für ADFS geeignet?
Keine	ja	ja – die Standardempfehlung
Quell-IP-Affinität	ja	nein, führt zu ungleicher Last
Cookie-basiert	nein, Cookies sind verschlüsselt	nicht erreichbar ohne Bridging
TLS-Session-ID	technisch teilweise	unnötig, zudem unzuverlässig bei modernen TLS-Versionen

Tabelle 4: Persistenzmodi im Überblick

TIPP – Wenn ein Hersteller doch Persistenz verlangt

Manche Zusatzprodukte, etwa ältere MFA-Adapter von Drittanbietern, halten Zustand im Arbeitsspeicher eines Knotens. Dann – und nur dann, wenn der Hersteller es dokumentiert – kann eine kurze Persistenz sinnvoll sein. Frag in dem Fall zuerst, ob es eine neuere Version ohne diese Einschränkung gibt. Für die Kopplung mit der Entra-MFA gilt das nicht.

Wie die Anbindung der Entra-MFA sauber aussieht, zeigt [ADFS mit Entra-MFA koppeln – Adapter, Zertifikat und Fallstricke](#).

Client-IP für Extranet Smart Lockout

Jetzt wird es spannend. Extranet Smart Lockout merkt sich pro Konto die vertrauten Orte, also die IP-Adressen, von denen aus schon erfolgreich angemeldet wurde. Fehlversuche von unbekannten Orten zählen getrennt von denen aus vertrauten Orten. So sperrt ein Angreifer mit Passwort-Spray

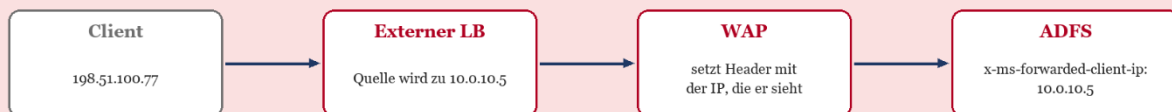
nicht gleich den Anwender mit, der im Büro sitzt. Das funktioniert allerdings nur, wenn ADFS die echten Adressen sieht.

Die Kette sieht so aus: Der WAP nimmt die Verbindung an und schreibt die Quell-IP, die er auf seiner Netzwerkkarte sieht, in den Header x-ms-forwarded-client-ip. ADFS übernimmt diesen Wert für Extranet-Anfragen in den Claim userip. Macht der externe Load Balancer Source-NAT, sieht der WAP nur die Adresse des Load Balancers – und ab da haben alle Anwender dieser Welt denselben Wohnort. Microsoft beschreibt genau dieses Symptom: Tauchen in Berichten die Adressen des Load Balancers auf, sendet der externe Load Balancer die Client-IP nicht weiter.

Skizze 5: Welche IP sieht ADFS? SNAT gegen transparente Weiterleitung

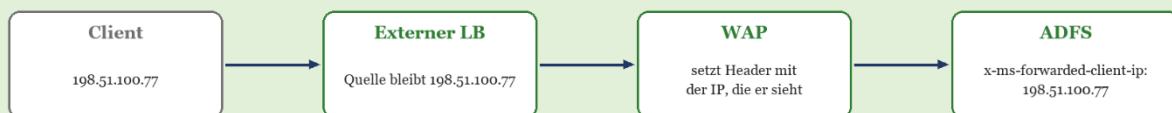
Extranet Smart Lockout zählt pro Ort – wenn alle Orte gleich aussehen, zählt es falsch

Externer LB mit SNAT (Standard vieler Geräte)



Alle Anmeldungen kommen scheinbar vom LB. Bekannte und unbekannte Orte verschwimmen, ein Angreifer sperrt im schlimmsten Fall alle mit.

Externer LB transparent (Quell-IP bleibt erhalten)



ADFS sieht die echte Adresse. Familiar Locations, Banned IPs und Auswertungen im Protokoll stimmen.

Der interne LB zwischen WAP und ADFS darf übrigens SNAT machen: Die Client-IP reist dort schon verschlüsselt im Header des WAP mit. Entscheidend ist ausschließlich, was der WAP auf seiner Netzwerkkarte sieht.

Skizze 5: Mit SNAT am externen Load Balancer sieht der WAP nur den Load Balancer. Transparent weitergeleitet, landet die echte Client-IP im Header und damit bei Extranet Smart Lockout.

Microsoft formuliert die Anforderung so: Der Load Balancer soll die Quelladresse des Clients erhalten. Kann er das nicht, muss er sie in den Standard-Header X-Forwarded-For schreiben; ADFS ab Windows Server 2016 mit aktuellen Updates wertet ihn aus. Die Krux: Header einfügen geht nur, wenn der Load Balancer TLS aufbricht – und das ist ja gerade nicht unterstützt. In der Praxis bleibt daher für den externen Load Balancer fast immer nur die Layer-3-Lösung.

Variante	Wie es funktioniert	Bewertung
Transparente Weiterleitung	LB tauscht nur die Ziel-IP; der WAP sieht die echte Quelle und antwortet über den LB als Gateway	empfohlen, braucht passendes Routing in der DMZ
Direct Server Return	Antworten gehen am LB vorbei direkt zum Client	möglich, aufwendiger im Netzdesign
SNAT plus X-Forwarded-For	LB bricht TLS auf und fügt den Header ein	nicht unterstützt, siehe Bridging
SNAT ohne alles	WAP sieht nur den LB	Extranet Smart Lockout arbeitet blind

Tabelle 5: Wege, die Client-IP bis zum WAP zu retten

Der interne Load Balancer ist entspannter. Der Header des WAP reist im durchgereichten TLS-Tunnel unverändert mit, auch wenn der interne Load Balancer SNAT macht. Für Extranet-

Anfragen spielt die Quell-IP innen deshalb keine Rolle. Für Clients im LAN dagegen schon: Deren Adresse landet in X-MS-Client-IP, und mit SNAT stehen dort nur noch Load-Balancer-Adressen. Wenn du Zugriffsrichtlinien nach internen Netzen unterscheidest oder im Protokoll nachvollziehen willst, welcher Rechner sich angemeldet hat, lohnt sich auch innen die transparente Variante.

```
# Ist Extranet Smart Lockout aktiv, und in welchem Modus?
Get-AdfsProperties | Select-Object ExtranetLockoutEnabled, ExtranetLockoutMode,
    ExtranetLockoutThreshold, ExtranetObservationWindow

# Welche Orte kennt ADFS für ein Konto? Stehen hier nur LB-Adressen, stimmt die Kette nicht.
Get-AdfsAccountActivity -Identity 'CONTOSO\m.muster'
```

Listing 4: Kontrolle, ob Extranet Smart Lockout echte Client-Adressen sieht.

FAKTEN — Wo die IP-Adressen auftauchen

X-MS-Client-IP: Netzwerkadresse des Geräts, das mit ADFS verbunden ist. Bei Extranet-Anfragen immer die IP des WAP.

X-MS-Forwarded-Client-IP: mehrwertig, enthält die Adresse, die der WAP als Client gesehen hat, sowie Werte, die Exchange Online weiterreicht.

userip: bei Extranet-Anfragen der Wert aus X-MS-Forwarded-Client-IP, bei Intranet-Anfragen der Wert aus X-MS-Client-IP.

Wie du Extranet Smart Lockout einrichtest und im Modus „Protokollieren“ zuerst beobachtest, beschreibt [Extranet Smart Lockout – ADFS gegen Passwort-Spray schützen](#); wie du Angriffe in den Protokollen erkennst, [Passwort-Spray und Brute Force auf ADFS erkennen](#).

Praxisbeispiel: Kemp LoadMaster, herstellerneutral übersetzt

Im folgenden Beispiel aus einem Projekt – anonymisiert, versteht sich – stehen zwei WAP-Server und zwei ADFS-Knoten hinter einem Kemp LoadMaster. Die Begriffe unten stammen aus dessen Oberfläche; in der zweiten Spalte steht, wie die Einstellung bei anderen Herstellern ungefähr heißt. Mehr zum Gerät selbst findest du auf der Themenseite [Kemp LoadMaster](#).

Einstellung (Kemp)	Herstellerneutral	Wert für ADFS und WAP
Virtual Service, Port 443, TCP	virtueller Server, Listener	VIP extern 203.0.113.10, intern 10.0.20.10
Service Type	Profil, Protokollmodus	Generic bzw. Layer 4 – kein HTTP-Profil mit TLS-Aufbruch
SSL Acceleration	TLS-Offloading, Client-SSL-Profil	aus
Re-Encrypt	Server-SSL-Profil, Bridging	aus
Persistence Mode	Persistenz, Affinity, Stickiness	None
Scheduling Method	Verteilverfahren	Round Robin oder Least Connection
Real Server Check Method	Health Monitor, Probe	HTTP (nicht HTTPS)
Checked Port	Monitor-Port	80
URL	Monitor-Pfad	/adfs/probe
HTTP Method	Monitor-Methode	GET
Transparency	Quell-IP erhalten, kein SNAT	extern an, Real Server nutzen den LB als Gateway
Real Servers	Pool-Mitglieder	WAP01/02 bzw. ADFS01/02, jeweils Port 443

Tabelle 6: Einstellungen am Beispiel Kemp LoadMaster mit herstellerneutraler Entsprechung

WICHTIG — Vorsicht bei Herstellervorlagen

Viele Hersteller liefern fertige Vorlagen für ADFS und den WAP aus, darunter oft auch Varianten mit TLS-Offloading oder Re-Encryption, weil sich damit Header und Cookies bearbeiten lassen. Die Vorlage spart Tipparbeit, nimmt dir aber nicht die Prüfung ab: Schau nach dem Import nach, ob TLS wirklich durchgereicht wird, ob der Health Check auf HTTP 80 und /adfs/probe zeigt und ob Persistenz ausgeschaltet ist.

Was im Projekt schief lief

Die Farm lief seit 2016 zuverlässig. Dann zog der externe virtuelle Dienst auf ein neues Gerätepaar um, und das Team übernahm die alte Konfiguration – nicht ganz eins zu eins. Drei Dinge fielen in den folgenden Wochen auf, und keines davon meldete sich mit einer verständlichen Fehlermeldung.

- Der Health Check war auf HTTPS gestellt, weil „HTTP auf der DMZ-Firewall doch zu ist“. Er funktionierte, bis die Firmware ein Update bekam, das den Servernamen im Check nicht mehr mitschickte. Umstellung auf HTTP 80 plus eine Firewall-Regel nur für die Adressen des Load Balancers – erledigt.
- Transparency war aus. Extranet Smart Lockout lief im Erzwingungsmodus und zählte alle externen Anmeldungen als einen einzigen Ort. Die ersten Sperren trafen Außendienstler, die sich nichts vorzuwerfen hatten. Nach dem Umschalten tauchten in Get-AdfsAccountActivity endlich wieder echte Adressen auf.
- Persistenz war auf Quell-IP gestellt. Einer der beiden WAP trug an manchen Tagen den Großteil der Last, weil ein Großteil der Anfragen aus wenigen Adressbereichen kam. Persistenz aus, Last gleichmäßig.

FAKTEN — Die Lehre aus dem Projekt

Keiner der drei Fehler hätte beim Umzug auffallen können, wenn nur geprüft wird, ob die Anmeldeseite lädt. Die lädt nämlich in allen drei Fällen. Deshalb gehören zu jeder Änderung am Load Balancer drei Tests: Probe per HTTP auf jedem Real Server, Anmeldung von extern mit Blick auf die protokollierte IP-Adresse und Kontrolle der Lastverteilung über einige Stunden.

Fehlerbilder und ihre Ursache

Symptom	Wahrscheinliche Ursache	Prüfung
Alle Knoten im LB als ausgefallen markiert, Server gesund	HTTPS-Check ohne SNI oder Port 80 gesperrt	Listing 1 und 2, Firewall zwischen LB und Servern
Zertifikatsanmeldung scheitert, Kennwort geht	TLS wird am LB terminiert	Service-Konfiguration: Offloading und Re-Encryption
WAP meldet Verbindungsprobleme zur Farm	Bridging am internen LB oder abgelaufenes Proxy-Vertrauen	interner virtueller Dienst, Ereignisprotokoll am WAP
Extranet Lockout sperrt Unbeteiligte	SNAT am externen LB, alle Anmeldungen haben dieselbe IP	Get-AdfsAccountActivity, Transparency
Ein Knoten dauerhaft überlastet	Quell-IP-Persistenz	Persistence Mode
Interne Clients bekommen das Formular statt SSO	CNAME statt A-Eintrag oder SPN-Problem	Beitrag SPN, DNS und Kerberos für ADFS – warum die interne Anmeldung auf Formular zurückfällt
Nach Zertifikatswechsel fällt nur der LB-Check aus	o.o.o.o-Bindung zeigt noch auf das alte Zertifikat	netsh http show sslcert

Tabelle 7: Typische Fehlerbilder an der Nahtstelle zwischen Load Balancer und ADFS

Wenn eine Anmeldung trotz korrektem Load Balancer scheitert, liegt die Ursache meist woanders. Einen strukturierten Weg durch die übrigen Verdächtigen bietet [ADFS-Anmeldung schlägt fehl – die häufigsten Ursachen und ihre Lösung](#).

Fazit

Ein Load Balancer vor ADFS muss erstaunlich wenig können – und genau das ist die Kunst. Er soll TLS durchreichen statt aufbrechen, per HTTP auf Port 80 den Pfad /adfs/probe prüfen, auf Persistenz verzichten und die Client-IP am externen Übergang nicht wegübersetzen. Jede zusätzliche Intelligenz, die das Gerät gern beisteuern würde, kostet bei ADFS eine Funktion: Offloading die Zertifikatsanmeldung, IP-Persistenz die gleichmäßige Last, SNAT den Schutz durch Extranet Smart Lockout, der HTTPS-Check im ungünstigsten Moment die ganze Farm.

ADFS läuft in vielen Unternehmen seit über zehn Jahren, und es wird dort noch eine Weile laufen, auch wenn der Weg nach Entra ID schon geplant ist. Bis dahin verdient es einen Load Balancer, der ihm nicht im Weg steht. Wenn du die Konfiguration gemeinsam durchgehen oder einen Umzug auf neue Geräte absichern willst, ist das ein typischer Fall für [Consulting zu ADFS \(Active Directory Federation Services\)](#). Farm, Proxy und Load Balancer im Zusammenspiel kannst du in den [ADFS-Schulungen](#) selbst aufbauen und kaputt machen. Und wer Hochverfügbarkeit, Proxy und Zertifikate im Zusammenhang nachlesen möchte, findet das ausführlich in [ADFS in der Praxis – das Buch im Detail](#).

WEITER – Passend zu diesem Beitrag

- › [Extranet Smart Lockout – ADFS gegen Passwort-Spray schützen](#)
- › [SSL-Zertifikat auf ADFS und Web Application Proxy tauschen](#)
- › [WAP-Proxy-Vertrauen erneuern – wenn der Web Application Proxy plötzlich nicht mehr will](#)
- › [ADFS überwachen – Health Checks, Diagnose-Toolbox und Entra Connect Health](#)
- › [Web Application Proxy: Architektur und Ablöse](#)
- › [Kemp LoadMaster](#)

FAQ

Welchen Health-Probe-Endpunkt hat ADFS?

Jeder ADFS-Server und jeder Web Application Proxy beantwortet HTTP-Anfragen auf Port 80 unter /adfs/probe mit 200 OK. Die Antwort wird lokal erzeugt und hängt nicht von Datenbank oder Domänencontrollern ab. Unter Windows Server 2012 R2 brauchst du dafür das Update-Rollup vom August 2014.

Warum HTTP und nicht HTTPS für den Health Check?

Weil der HTTP-Check jede SNI- und Zertifikatsfrage umgeht. Microsoft empfiehlt ausdrücklich die HTTP-Probe. Ein HTTPS-Check ohne Servernamen im Handshake schlägt an ADFS fehl und nimmt gesunde Server aus dem Pool.

Muss Port 80 aus dem Internet erreichbar sein?

Nein. Port 80 muss nur vom Load Balancer zu den Real Servern offen sein. Beschränke die Firewall-Regel auf die Adressen des Load Balancers. Anmeldeverkehr läuft ausschließlich über 443.

Was tun, wenn mein Load Balancer kein SNI kann?

Stell den Health Check auf die HTTP-Probe um. Wenn Clients ohne SNI auch produktiv zugreifen müssen, lege auf ADFS und WAP eine 0.0.0.0:443-Rückfallbindung an und denke bei jedem Zertifikatswechsel daran, sie zu aktualisieren.

Darf der Load Balancer TLS terminieren?

Nein. Microsoft unterstützt das Terminieren am Load Balancer für keinen ADFS-Anwendungsfall. Zertifikatsanmeldung, Geräteauthentifizierung und das Proxy-Vertrauen zwischen WAP und ADFS funktionieren nur, wenn TLS bis zum Server durchgereicht wird.

Brauche ich Persistenz oder Sticky Sessions?

In aller Regel nicht. ADFS ist für die Anmeldung zustandslos, jeder Knoten kann jeden Schritt bearbeiten. IP-basierte Persistenz rät Microsoft sogar ab, weil sie Knoten ungleichmäßig belastet.

Warum sieht Extranet Smart Lockout nur die IP des Load Balancers?

Weil der externe Load Balancer Source-NAT macht. Der WAP schreibt die Adresse, die er sieht, in x-ms-forwarded-client-ip – und das ist dann die des Load Balancers. Abhilfe schafft eine transparente Weiterleitung, bei der die Quell-IP erhalten bleibt.

Kann ich die Client-IP nicht einfach per X-Forwarded-For weitergeben?

ADFS ab Windows Server 2016 mit aktuellen Updates wertet X-Forwarded-For aus. Einfügen kann der Load Balancer den Header aber nur, wenn er TLS aufbricht – und das ist nicht unterstützt. Deshalb ist die Layer-3-Lösung der praktikable Weg.

Darf der interne Load Balancer zwischen WAP und ADFS SNAT machen?

Für Extranet-Anfragen ja, denn die Client-IP steckt bereits im Header des WAP und reist unverändert im TLS-Tunnel mit. Für Clients im LAN verlierst du mit SNAT allerdings die echte Adresse im Protokoll und in Zugriffsrichtlinien.

Reicht DNS-Round-Robin statt Load Balancer?

Microsoft empfiehlt es nicht. DNS-Round-Robin kennt keine Health Probe und verteilt weiter auf einen ausgefallenen Knoten, bis jemand den Eintrag von Hand entfernt.