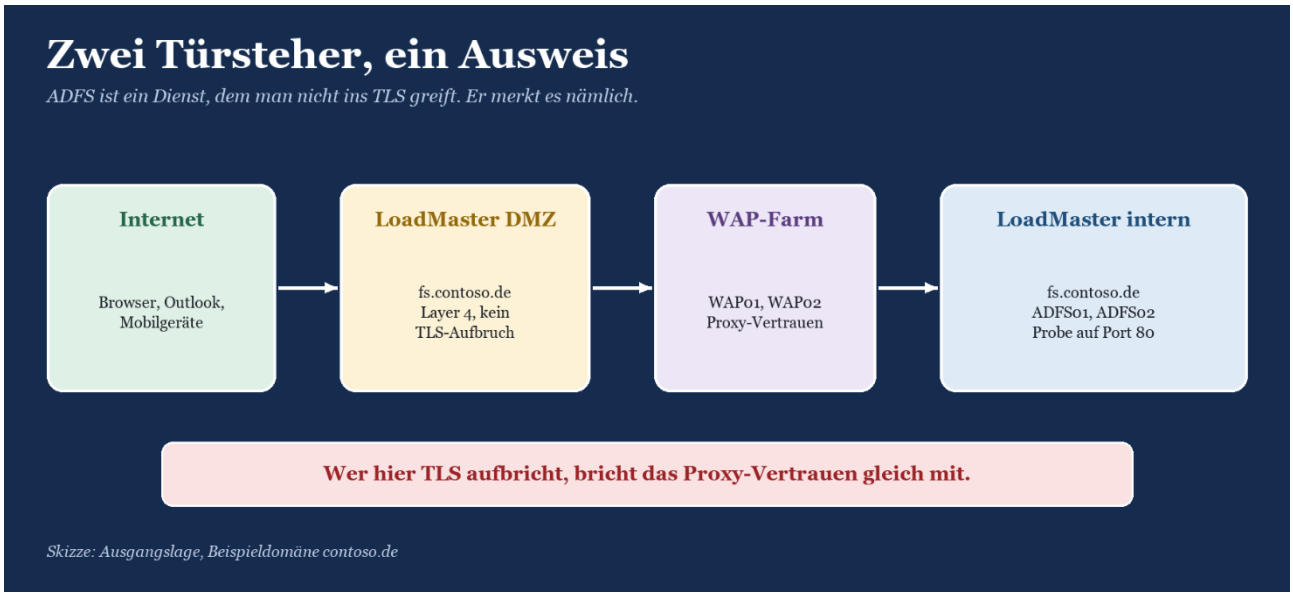
ADFS und Web Application Proxy mit dem LoadMaster: Farm, Probe, SNI



WISSEN

Alles zum Kemp LoadMaster mit Microsoft-Brille: Exchange, SharePoint, ADFS, RDS und Kerberos an einem Ort.

› **Kemp LoadMaster**

BERATUNG

ADFS-Farm hinter den LoadMaster bringen, WAP-Frage klären oder Extranet Lockout endlich mit echten Client-IPs betreiben? Wir schauen gemeinsam drauf.

› **Beratung zum Kemp LoadMaster**

SCHULUNG

Virtual Services, Health Checks und Transparenz für Microsoft-Workloads im Labor durchspielen, bevor es die Anmeldung aller Benutzer tut.

› **Kemp-LoadMaster-Schulung**

ADFS hat einen besonderen Platz im Rechenzentrum: Es ist der Dienst, über den jeder Benutzer morgens zuerst stolpert. Fällt Exchange aus, fluchen alle über E-Mail. Fällt ADFS aus, fluchen alle über alles, denn plötzlich gehen Microsoft 365, das Reisekostenportal, die Fachanwendung mit SAML und die halbe Intranet-Welt gleichzeitig nicht mehr. ADFS ist damit der Kandidat schlechthin für einen Load Balancer. Und gleichzeitig einer der empfindlichsten, weil ADFS sehr genau hinsieht, wer da eigentlich mit ihm spricht.

Der Klassiker in der Praxis sieht so aus: Der LoadMaster steht vor der ADFS-Farm, alles funktioniert, bis jemand im Rahmen einer Härtung den Health Check auf HTTPS umstellt. Ab da sind alle Real Server rot, obwohl ADFS quicklebendig ist. Oder der WAP meldet plötzlich, das Proxy-Vertrauen sei kaputt, nachdem jemand TLS-Offload am internen Virtual Service „nur mal kurz“ ausprobiert hat. Oder das Extranet Lockout sperrt nach einem Passwort-Spray nicht den Angreifer, sondern die gesamte Belegschaft, weil ADFS alle Anmeldungen von ein und derselben IP-Adresse kommen sieht. Der IP-Adresse des LoadMasters.

Dieser Beitrag gehört zur Serie rund um den Kemp LoadMaster und schließt die Lücke zwischen dem ADFS-Deployment-Guide von Progress Kemp und den ADFS-Seiten auf Microsoft Learn. Es geht um den Aufbau mit Farm und WAP, um /adfs/probe als Health Check, um SNI und die Frage, warum Clients ohne SNI vor verschlossener Tür stehen, um Zertifikate, um Extranet Lockout mit der echten Client-IP und am Ende um die unbequeme Frage, ob der LoadMaster den WAP ersetzen kann.

FAKTEN · Das Wichtigste in sechs Sätzen

Microsoft unterstützt kein TLS-Terminieren am Load Balancer vor ADFS, und zwar für keinen Anwendungsfall; das gilt vor der ADFS-Farm und vor den WAP-Servern.

Der Health Check gehört auf http://<server>/adfs/probe über Port 80; der Endpunkt antwortet mit 200 OK, lokal und ohne Abhängigkeit zu Backend-Diensten.

ADFS bindet sein Zertifikat seit Windows Server 2012 R2 per SNI an den Hostnamen; Clients und Prüfungen ohne SNI bekommen keinen Handshake, eine Fallback-Bindung auf 0.0.0.0:443 ist nur die Notlösung.

Microsoft rät von IP-basierter Sitzungsaffinität für ADFS ab; ADFS braucht keine Persistenz, weil der Anmeldezustand in Cookies und Token steckt.

Extranet Smart Lockout wertet die IP-Adressen aus, die der WAP in x-ms-forwarded-client-ip weiterreicht; sieht der WAP nur die Adresse des LoadMasters, lernt ADFS Unsinn.

Ein Drittanbieter-Proxy ersetzt den WAP nur dann vollwertig, wenn er das Protokoll MS-ADFSPIP spricht; ohne das hält ADFS jeden Zugriff für intern.

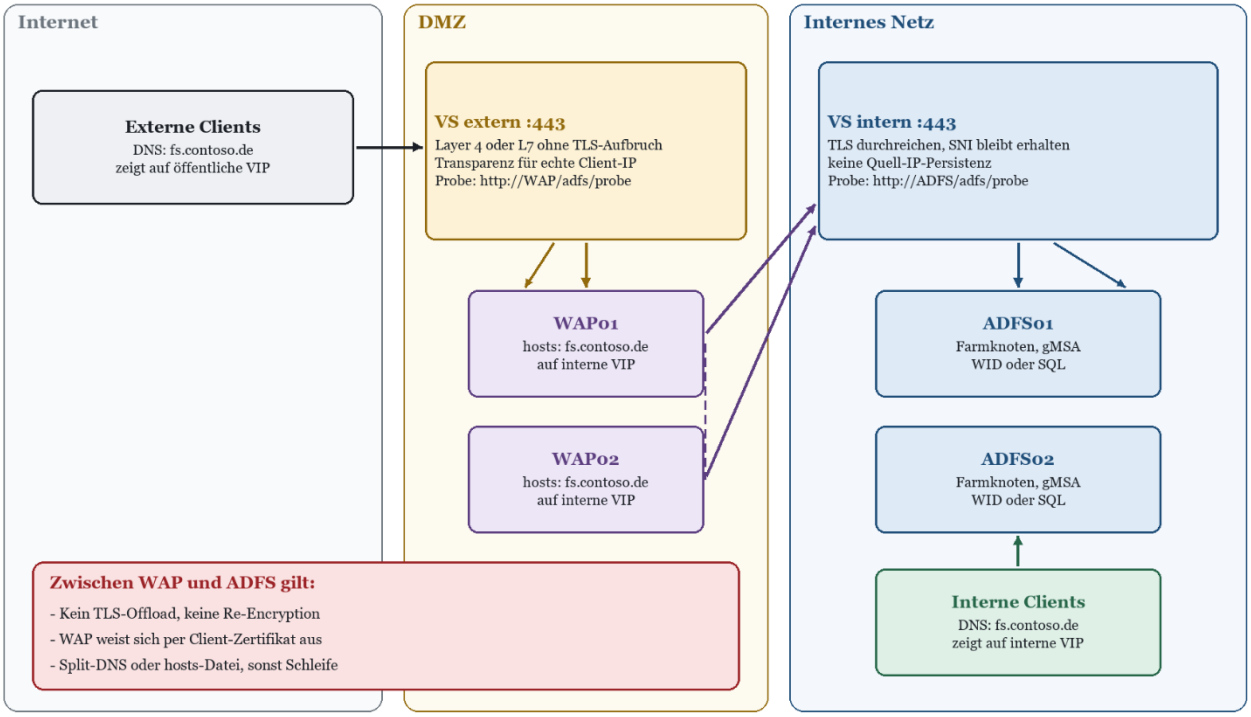
Die Architektur: Farm innen, WAP in der DMZ, LoadMaster an beiden Stellen

Eine ADFS-Bereitstellung im Mittelstand oder in der Verwaltung besteht fast immer aus zwei Schichten. Innen steht die ADFS-Farm, zwei oder mehr Server mit derselben Konfiguration, entweder auf Windows Internal Database oder auf SQL Server. Außen, in der DMZ, stehen die Web Application Proxy Server, kurz WAP, die Anfragen aus dem Internet annehmen und an die Farm weiterreichen. Microsoft schreibt für Zugriffe von außen den WAP oder einen kompatiblen Drittanbieter-Proxy vor. Der WAP ist nicht nur ein Reverse Proxy, sondern ein Mitglied der Familie: Er wird bei der Einrichtung per Proxy-Vertrauen an die Farm gekoppelt und weist sich bei jeder Verbindung mit einem Zertifikat aus.

Der LoadMaster taucht in diesem Bild typischerweise zweimal auf, auch wenn es physisch dasselbe HA-Paar ist. Ein Virtual Service in der DMZ verteilt die Zugriffe aus dem Internet auf die WAP-Server. Ein zweiter Virtual Service im internen Netz verteilt die Zugriffe der WAP-Server und der internen Clients auf die ADFS-Knoten. Beide Virtual Services hören auf denselben Namen, zum Beispiel fs.contoso.de, nur eben in unterschiedlichen DNS-Welten.

Die Referenzarchitektur: zwei Virtual Services, zwei Welten

Extern über WAP, intern direkt; beide Wege enden am selben Namen fs.contoso.de



Skizze: Referenzaufbau, Namen fiktiv; HA-Paar der LoadMaster nicht dargestellt

Split-DNS oder hosts-Datei: der Name muss überall woanders hinzeigen

Dieser Punkt sorgt in Projekten regelmäßig für den ersten Ausfall. Externe Clients lösen fs.contoso.de auf die öffentliche Adresse vor den WAP-Servern auf. Interne Clients lösen denselben Namen auf die interne VIP vor der Farm auf. Und die WAP-Server selbst? Die müssen fs.contoso.de ebenfalls zur internen VIP auflösen, sonst schicken sie die Anfrage brav wieder ins Internet, wo sie beim externen Virtual Service landet, der sie wieder an einen WAP gibt. Das ist die Endlosschleife, die man in keinem Monitoring sieht, aber in jeder Anmeldeseite. Microsoft beschreibt beide Wege: einen eigenen DNS-Server für die DMZ oder einen Eintrag in der hosts-Datei jedes WAP-Servers. In kleinen Umgebungen gewinnt die hosts-Datei, weil sie ehrlich ist und niemand sie versehentlich umkonfiguriert.

Für die Windows-integrierte Anmeldung intern gibt Microsoft außerdem vor, dass der Dienstname per A-Eintrag aufgelöst wird, nicht per CNAME. Ein CNAME auf einen generischen Load-Balancer-Namen sieht ordentlich aus, führt aber dazu, dass manche Clients ein Kerberos-Ticket für den falschen Namen anfordern. Der SPN gehört bei ADFS an das Dienstkonto, in modernen Farmen ein gMSA, und ist auf allen Knoten gleich. Der LoadMaster ist für Kerberos damit unsichtbar, solange niemand TLS aufbricht. Mehr zum Prinzip findest du auf der Themenseite Microsoft Kerberos.

Wie die Virtual Services aussehen

Weil Microsoft TLS-Terminierung ausschließt, bleiben für beide Virtual Services im Kern zwei Varianten: ein Layer-4-Virtual-Service, der TCP 443 unverändert durchreicht, oder ein Layer-7-Virtual-Service ohne TLS-Beschleunigung, der zwar Verbindungen verwaltet, aber nicht in den verschlüsselten Datenstrom hineinsieht. Content Rules, Header-Manipulation und Cookie-Persistenz fallen damit weg. Das ist kein Mangel, sondern der Preis dafür, dass ADFS mit Client-Zertifikaten, Channel Binding und Proxy-Vertrauen arbeitet. Progress Kemp liefert für ADFS Templates und einen Deployment Guide, die einen guten Startpunkt bieten. Prüfe die Vorlage

trotzdem gegen die Microsoft-Vorgaben, bevor du sie übernimmst, vor allem bei Persistenz und TLS-Einstellungen.

Einstellung	VS extern (vor WAP)	VS intern (vor ADFS)
Port	TCP 443, ggf. 49443 für Zertifikatsanmeldung	TCP 443
TLS	durchreichen, kein Offload	durchreichen, kein Offload, keine Re-Encryption
Health Check	HTTP Port 80, /adfs/probe	HTTP Port 80, /adfs/probe
Persistenz	keine oder kurz, keine Quell-IP-Pflicht	keine Quell-IP-Persistenz
Scheduling	Least Connection oder Round Robin	Least Connection oder Round Robin
Client-IP	transparent, damit der WAP sie sieht	unkritisch, Header kommt vom WAP
Quellen	Internet über Firewall	WAP-Server und interne Clients

WARNUNG · TLS-Offload ist hier keine Optimierung, sondern ein Defekt

Microsoft formuliert es in den ADFS-Anforderungen ungewohnt deutlich: Der Load Balancer darf TLS nicht terminieren, das ist für keinen Anwendungsfall unterstützt. Der Grund ist handfest. Der WAP authentifiziert sich bei der Farm mit einem Client-Zertifikat im TLS-Handshake. Bricht der LoadMaster die Verbindung auf, kommt dieses Zertifikat nie beim ADFS an, und das Proxy-Vertrauen ist dahin. Dasselbe gilt für Zertifikatsanmeldung von Benutzern und Geräten und für Extended Protection bei Windows-Anmeldung. Re-Encryption hilft nicht, denn auch dabei beendet der LoadMaster die ursprüngliche TLS-Sitzung.

Wie TLS-Offload und Re-Encryption bei Diensten funktionieren, bei denen sie erlaubt sind, und warum ADFS hier die Ausnahme bildet, beschreibt der Beitrag TLS-Offload, Re-Encryption und Zertifikate: PKI trifft LoadMaster.

Persistenz: Microsoft sagt nein, und zwar mit Grund

ADFS speichert den Anmeldezustand nicht im Arbeitsspeicher eines Knotens, sondern in verschlüsselten Cookies beim Client und in der gemeinsamen Konfiguration. Jeder Knoten kann jeden Benutzer weiterbedienen. Persistenz ist also nicht nötig. Microsoft rät sogar ausdrücklich davon ab, IP-basierte Affinität einzusetzen. Das Szenario dahinter ist lehrreich: Wenn Exchange Online Legacy-Anmeldungen im Namen von Outlook-Clients an ADFS weiterreicht, kommen sehr viele Anfragen von wenigen Microsoft-Adressen. Mit Quell-IP-Persistenz landen die alle auf einem Knoten, und der wird zum Flaschenhals. Bei internen Zugriffen über einen NAT-Router oder einen Proxy der Zentrale passiert dasselbe im Kleinen. Warum Quell-IP-Persistenz grundsätzlich mit Vorsicht zu genießen ist, steht im Beitrag Persistence und Scheduling: wer hängt an welchem Server und warum.

/adfs/probe und SNI: der Health Check, der funktioniert, und der, der nicht

Seit Windows Server 2012 R2 bringen ADFS und WAP einen eigenen Prüfendpunkt mit: /adfs/probe. Er antwortet über HTTP auf Port 80 mit 200 OK und wird lokal bedient, ohne Abhängigkeit zu Backend-Diensten. Microsoft empfiehlt ausdrücklich, diesen HTTP-Endpunkt für Health Checks des Load Balancers zu verwenden, nicht HTTPS. Erreichbar ist er unter Servername oder IP-Adresse, also http://adfs01.contoso.local/adfs/probe ebenso wie über die nackte Adresse. Auf dem WAP heißt er genauso und verhält sich genauso.

Für den LoadMaster bedeutet das: Der Virtual Service lauscht auf 443, der Health Check prüft aber auf Port 80. Die Real-Server-Prüfung des LoadMasters erlaubt es, einen vom Dienstport abweichenden Prüfport festzulegen, und als Prüfmethode einen HTTP-GET auf eine URL mit Bewertung des Statuscodes. Mehr braucht es nicht. Die Firewall zwischen LoadMaster und Real Servern muss Port 80 natürlich durchlassen, und die Windows-Firewall auf den Servern auch. Das klingt banal, ist aber in gehärteten DMZ-Netzen der häufigste Grund, warum der Check zwar richtig konfiguriert, aber trotzdem rot ist.

Was die Probe beweist und was nicht

Die Probe hat eine Eigenschaft, die man kennen muss: Sie prüft, ob der ADFS-Dienst läuft und HTTP-Anfragen annimmt. Sie prüft nicht, ob die Konfigurationsdatenbank erreichbar ist, ob ein Domänencontroller antwortet oder ob das Token-Signaturzertifikat noch gültig ist. Auf dem WAP sagt sie dir, dass der Dienst lebt, aber nicht zwingend, dass das Proxy-Vertrauen zur Farm intakt ist. Ein WAP mit abgelaufenem Vertrauen kann also grün im LoadMaster stehen und trotzdem jede Anmeldung mit einem Fehler beantworten. Das ist so, als würde der Pförtner bestätigen, dass das Gebäude steht, während innen gerade die Treppe fehlt.

Das ist kein Grund, die Probe zu verwerfen, sondern ein Grund, sie zu ergänzen. Der LoadMaster nimmt einen Knoten aus dem Spiel, wenn der Dienst steht; das deckt Neustarts, Patches und abgestürzte Dienste ab. Für Datenbank, Zertifikate und Proxy-Vertrauen gehört ein Monitoring daneben, das echte Anmeldungen prüft, zum Beispiel einen regelmäßigen Abruf der Metadaten unter /FederationMetadata/2007-06/FederationMetadata.xml über den externen Namen oder einen synthetischen Anmeldetest. Warum „Port offen“ nie reicht und wie man Health Checks grundsätzlich aufbaut, steht im Beitrag Health Checks, die wirklich prüfen: warum „Port 443 offen“ nicht reicht.

Prüfmethode	Ergebnis bei ADFS	Bewertung
TCP 443 offen	grün, solange http.sys lauscht	erkennt keinen gestoppten ADFS-Dienst zuverlässig
HTTPS GET / ohne SNI	rot, Handshake scheitert	falsch negativ, Knoten sind gesund
HTTPS GET /adfs/ls/ mit SNI	200 oder Umleitung, je nach Version	aufwendig, kein Mehrwert
HTTP GET /adfs/probe Port 80	200 OK, lokal bedient	Microsoft-Empfehlung, Standard
Synthetische Anmeldung	beweist Datenbank, AD, Zertifikate	gehört ins Monitoring, nicht in den LoadMaster

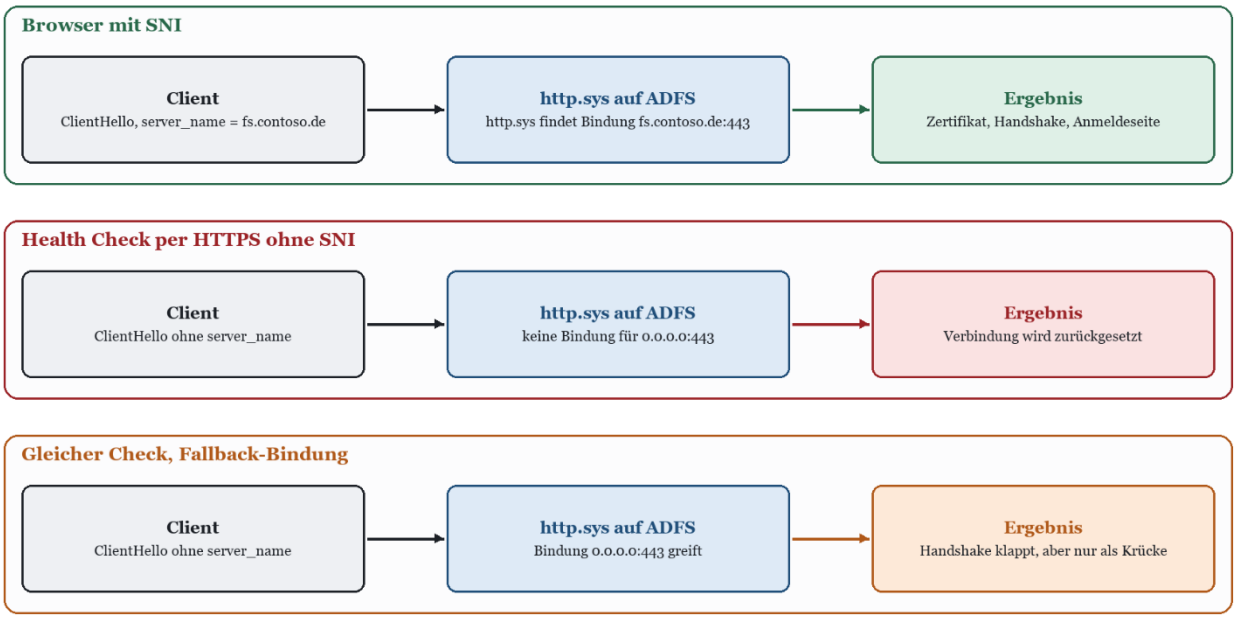
SNI: warum Clients ohne Namen scheitern

Ab Windows Server 2012 R2 läuft ADFS nicht mehr im IIS, sondern direkt auf http.sys. Die Zertifikatsbindung legt ADFS dabei nicht als klassische IP-und-Port-Bindung an, sondern als Hostname-und-Port-Bindung, also mit Server Name Indication. Der Client muss im TLS-ClientHello mitteilen, welchen Namen er erreichen will, damit http.sys weiß, welches Zertifikat es vorzeigen soll. Schickt der Client keinen Namen, findet http.sys keine passende Bindung und beendet die Verbindung. Kein Fehlerzertifikat, keine Fehlerseite, nur ein Reset.

Moderne Browser und Betriebssysteme senden SNI seit über einem Jahrzehnt. Wer scheitert, sind die Exoten: sehr alte Mobilgeräte, eingebettete Clients in Fachanwendungen, Java- oder .NET-Bibliotheken in betagten Versionen, Scanner-Firmware, Monitoring-Skripte, die eine IP-Adresse statt eines Namens ansprechen, und eben Health Checks von Load Balancern, die per HTTPS auf die Adresse des Real Servers zugreifen. Microsoft nennt SNI-Unterstützung deshalb sowohl als Browser-Anforderung als auch als Anforderung an den Load Balancer.

SNI: warum ein Client ohne Namen keine Antwort bekommt

http.sys bindet das ADFS-Zertifikat an Hostname und Port, nicht an IP und Port



Skizze: Prinzip; Microsoft empfiehlt die HTTP-Probe, damit SNI für den Check keine Rolle spielt

Die Fallback-Bindung: Krücke mit Nebenwirkungen

Wenn ein Client partout kein SNI kann, beschreibt Microsoft als Ausweg eine zusätzliche Bindung auf 0.0.0.0:443 auf den ADFS- oder WAP-Servern. Diese Bindung greift für alle Verbindungen ohne Namen und liefert dasselbe Zertifikat aus. Eingerichtet wird sie mit netsh http add sslcert, mit dem Fingerabdruck des ADFS-Zertifikats und der Anwendungs-ID der bestehenden Bindung, die du vorher mit netsh http show sslcert ausliest. Das funktioniert, hat aber zwei Nebenwirkungen. Erstens musst du die Bindung bei jedem Zertifikatswechsel mitpflegen, denn Set-AdfsSslCertificate kennt sie nicht. Zweitens gilt sie für alles, was auf dem Server per http.sys auf 443 lauscht. Auf einem dedizierten ADFS-Server ist das egal, auf einem WAP, der weitere Anwendungen mit eigenen Zertifikaten veröffentlicht, kann es zu überraschenden Zertifikaten führen.

TIPP · Erst den Client fixen, dann die Bindung

Bevor du eine Fallback-Bindung anlegst, prüfe, wer eigentlich ohne SNI kommt. Ein Mitschnitt am LoadMaster oder am Server zeigt im ClientHello, ob die Erweiterung server_name fehlt. In neun von zehn Fällen ist es ein Health Check oder ein Skript, das eine IP-Adresse anspricht. Das korrigierst du in fünf Minuten. Die Fallback-Bindung ist für den zehnten Fall gedacht: das Gerät, das sich nicht ändern lässt und trotzdem bleiben muss.

Zertifikatsbindung: eins für alle, und zwar wirklich

Microsoft empfiehlt, auf allen ADFS- und WAP-Servern dasselbe TLS-Zertifikat zu verwenden. Für zwei Fälle ist das sogar Pflicht: Wenn der WAP Anfragen mit Windows-integrierter Anmeldung weiterreicht und wenn die ADFS-Eigenschaft ExtendedProtectionTokenCheck aktiv ist, was der Standard ist. Dann muss das Zertifikat auf dem WAP denselben Schlüssel haben wie auf der Farm. Das Zertifikat braucht den Dienstnamen im Antragstellernamen oder in den alternativen Namen, für Zertifikatsanmeldung auf Port 443 zusätzlich certauth.fs.contoso.de und für die Geräteregistrierung älterer Clients enterpriseregistration.contoso.de je UPN-Suffix.

Der LoadMaster selbst braucht für ADFS kein Zertifikat, weil er TLS nicht anfasst. Das ist ein Vorteil, den man beim Zertifikatswechsel zu schätzen lernt: Ein Ort weniger, an dem das neue Zertifikat vergessen werden kann. Die Stellen, an denen es nicht vergessen werden darf, sind die ADFS-Knoten per Set-AdfsSslCertificate, die WAP-Server per Set-WebApplicationProxySslCertificate und eine eventuell angelegte Fallback-Bindung.

Zertifikat	Wo	Was der LoadMaster damit zu tun hat
TLS-Zertifikat fs.contoso.de	alle ADFS- und WAP-Server, gleicher Schlüssel	nichts, er reicht durch
certauth.fs.contoso.de im SAN	für Zertifikatsanmeldung über Port 443	Name muss auf die VIP zeigen
enterpriseregistration im SAN	Geräteregistrierung älterer Clients	Name muss auf die VIP zeigen
Proxy-Vertrauenszertifikat	WAP, automatisch von ADFS ausgestellt	darf nicht unterbrochen werden
Token-Signatur und -Entschlüsselung	ADFS-Farm intern	keine Berührung

WICHTIG · Port 49443 nicht vergessen

Bei ADFS auf Windows Server 2012 R2 und standardmäßig auch bei späteren Versionen läuft die Zertifikatsanmeldung von Benutzern über TCP 49443, sofern du nicht auf das Verfahren mit certauth über Port 443 umgestellt hast. Wenn deine Anwender sich mit Smartcard oder Benutzerzertifikat von außen anmelden, braucht der externe Virtual Service auch diesen Port, selbstverständlich ebenfalls ohne TLS-Aufbruch. Zwischen WAP und Farm ist 49443 nicht nötig.

Extranet Lockout und die Client-IP: wer sperrt hier wen?

ADFS kennt zwei Generationen von Sperren für Anmeldungen von außen. Das klassische Extranet Soft Lockout aus Windows Server 2012 R2 zählt Fehlversuche je Konto und lässt ab einem Schwellwert für eine Beobachtungszeit keine Anmeldungen von außen mehr zu. Damit schützt es das Active-Directory-Konto vor der echten Sperre, sperrt aber den Benutzer von außen trotzdem aus. Das Extranet Smart Lockout, kurz ESL, das für ADFS auf Windows Server 2016 mit den Updates ab Juni 2018 verfügbar und ab Windows Server 2019 eingebaut ist, ist klüger: Es merkt sich für jeden Benutzer bis zu 20 vertraute IP-Adressen und führt getrennte Zähler für vertraute und unbekannte Orte. Ein Angreifer aus einem fremden Netz sperrt sich damit selbst aus, der Benutzer aus dem Homeoffice kommt weiter rein.

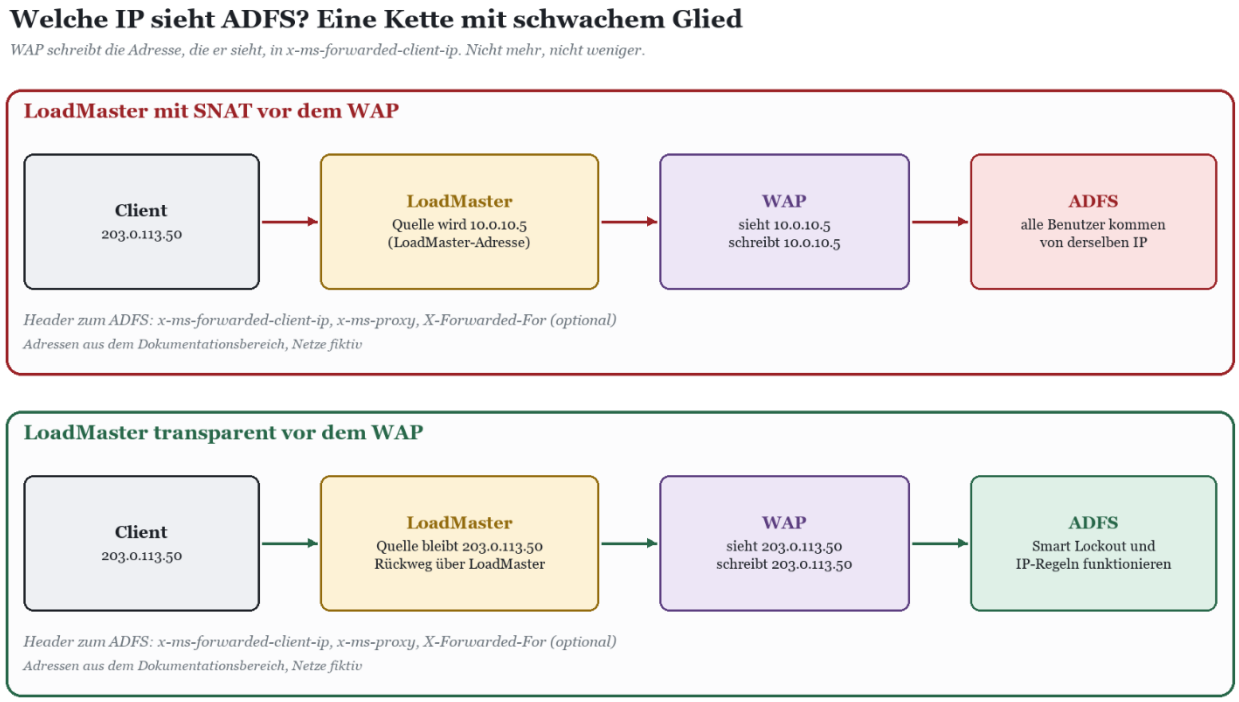
Wichtig ist eine Einschränkung, die Microsoft ausdrücklich nennt: ESL wirkt nur für Anmeldungen mit Benutzername und Kennwort, die über den WAP oder einen MS-ADFSPIP-fähigen Drittanbieter-Proxy hereinkommen. Interne Zugriffe direkt auf die Farm sind nicht betroffen. Und damit ist ESL komplett davon abhängig, welche IP-Adressen der WAP an die Farm meldet.

Wie die Client-IP zum ADFS kommt

Der WAP nimmt die Verbindung des Clients an und baut eine eigene Verbindung zur Farm auf. Damit ADFS trotzdem weiß, wer da eigentlich anklopft, schreibt der WAP die Adresse, die er als Quelle der Verbindung sieht, in den Header x-ms-forwarded-client-ip. Zusätzlich markiert er die Anfrage mit x-ms-proxy als von außen kommend. ADFS stellt diese Werte als Claims bereit, protokolliert sie im Sicherheitsüberwachungsprotokoll und verwendet sie für ESL. Laut Microsoft prüft ESL dabei alle vorgelegten Adressen: die Netzwerk-IP, die weitergereichte IP und optional

eine Adresse aus X-Forwarded-For. Nur wenn alle Adressen einer Anfrage in der Liste vertrauter Orte stehen, gilt die Anfrage als vertraut.

Und jetzt kommt der LoadMaster vor dem WAP ins Spiel. Arbeitet der externe Virtual Service mit SNAT, also mit der eigenen Adresse des LoadMasters als Quelle, sieht der WAP für jeden Client dieselbe Adresse. Der WAP schreibt sie gewissenhaft in x-ms-forwarded-client-ip. ADFS lernt, dass jeder Benutzer von 10.0.10.5 kommt. Vertraute Orte werden damit zu einem einzigen Ort, und der ist zufällig der, von dem auch jeder Angreifer kommt.



Skizze: Prinzip; Details zu Transparenz und SNAT im Geschwisterbeitrag zur Client-IP

Folge von SNAT vor dem WAP	Was konkret passiert
Smart Lockout wird blind	Angreifer und Benutzer teilen sich den vertrauten Ort; der Angreifer kommt aus Sicht von ADFS von einer bekannten Adresse
Soft Lockout trifft trotzdem nur Konten	der Zähler läuft je Konto, die Sperre gilt dann für den Benutzer von überall außen
IP-basierte Claim-Regeln kippen	Regeln wie „Zugriff nur aus Netz X“ oder „MFA außer aus dem Firmennetz“ werten die LoadMaster-Adresse aus; liegt sie im erlaubten Bereich, ist die Regel ausgehebelt
Protokolle werden wertlos	Ereignisse 411, 1203 und 516 nennen alle dieselbe Client-IP, Forensik nach einem Passwort-Spray wird zur Schätzung
Entra Connect Health liefert Unsinn	der Bericht zu riskanten IP-Adressen zeigt eine einzige, eigene Adresse als Täter

Transparenz statt Header: die Lösung am LoadMaster

Weil der externe Virtual Service TLS nicht aufbrechen darf, kann der LoadMaster auch keinen X-Forwarded-For-Header in die Anfrage schreiben. Header liegen im verschlüsselten Teil. Der einzige saubere Weg ist deshalb Transparenz: Der LoadMaster leitet die Verbindung mit der originalen Quelladresse des Clients an den WAP weiter. Der WAP sieht die echte Adresse und

meldet sie an ADFS. Bei Layer-4-Virtual-Services ist das beim LoadMaster das übliche Verhalten, bei Layer 7 lässt es sich je Virtual Service einschalten.

Transparenz hat eine Bedingung, die in DMZ-Netzen gern übersehen wird: Die Antworten der WAP-Server müssen wieder über den LoadMaster zurücklaufen. Sonst antwortet der WAP dem Client direkt, der Client verwirft das Paket, weil er es von der VIP erwartet, und die Verbindung hängt. In der Praxis heißt das: Das Standard-Gateway der WAP-Server ist der LoadMaster, oder du arbeitest mit passenden Routen. Weil der WAP für den Weg zur Farm ohnehin eine Route ins interne Netz braucht, lohnt sich hier eine kleine Skizze des Routings, bevor du das erste Mal auf Übernehmen klickst. Alle Varianten mit ihren Vor- und Nachteilen erklärt der Beitrag Client-IP sichtbar machen: Transparenz, SNAT und X-Forwarded-For.

Variante	Client-IP am WAP	Voraussetzung	Für ADFS geeignet
Layer 4, transparent	echt	Rückweg über den LoadMaster	ja, Standardempfehlung
Layer 7 ohne TLS, transparent	echt	Rückweg über den LoadMaster	ja
Layer 7 ohne TLS, SNAT	LoadMaster-Adresse	keine	nein, ESL wird blind
TLS-Offload mit X-Forwarded-For	im Header	TLS-Terminierung	nein, nicht unterstützt
Kein LoadMaster, DNS Round Robin	echt	keine	nein, Microsoft rät ab

WARNUNG · Der interne Virtual Service ist das kleinere Problem

Zwischen WAP und Farm ist SNAT unkritisch, denn ADFS bekommt die Client-IP aus dem Header des WAP und nicht aus der TCP-Verbindung. Kritisch sind dort nur interne Clients, wenn du intern IP-basierte Regeln einsetzt. Die Zugriffe intern laufen aber ohnehin nicht durch ESL. Konzentriere die Transparenz-Arbeit also auf den externen Virtual Service vor dem WAP, dort entscheidet sich, ob das Lockout funktioniert.

ESL sauber einschalten

Wenn die Client-IP stimmt, ist ESL schnell aktiviert. Microsoft empfiehlt, zuerst den Modus AdfsSmartLockoutLogOnly für drei bis sieben Tage laufen zu lassen, damit ADFS die vertrauten Orte der Benutzer lernt, und erst danach auf AdfsSmartLockoutEnforce umzustellen. Als Schwellwert gilt die Faustregel: ExtranetLockoutThreshold auf die Hälfte der AD-Sperrschwelle, damit ADFS sperrt, bevor das Active Directory es tut. Die Beobachtungszeit liegt typischerweise bei 30 Minuten. Mit ExtranetLockoutRequirePDC auf $false fällt ADFS bei nicht erreichbarem PDC-Emulator auf einen anderen Domänencontroller zurück. Und bevor du den Lernmodus startest, prüfe einmal mit Get-AdfsAccountActivity für ein Testkonto, ob dort echte öffentliche Adressen auftauchen und nicht die des LoadMasters. Wer erst lernt und dann prüft, hat sieben Tage lang die falsche Adresse eingelernt.

AUS DER PRAXIS · Das Lockout, das alle aussperrte

Eine Kommunalverwaltung mit etwa 1.200 Benutzerinnen und Benutzern betreibt ADFS für Microsoft 365. Vor den beiden WAP-Servern steht ein LoadMaster mit einem Layer-7-Virtual-Service ohne TLS-Beschleunigung, angelegt aus einer älteren Vorlage, Quelladresse per SNAT. Extranet Soft Lockout ist aktiv. An einem Montagmorgen läuft ein Passwort-Spray über die Anmeldeseite. Die

Sperren treffen Dutzende Konten, die Ereignisse im Sicherheitsprotokoll nennen als Quelle ausnahmslos eine interne Adresse aus dem DMZ-Netz. Das Sicherheitsteam vermutet zuerst einen kompromittierten Server im eigenen Netz. Erst der Blick auf die Konfiguration des Virtual Service zeigt, wessen Adresse das ist.

Die Lösung: Umstellung auf Transparenz, Anpassung des Standard-Gateways der WAP-Server, danach ESL im Lernmodus und eine Woche später im Erzwingungsmodus. Der nächste Spray-Versuch zwei Monate später sperrt nur noch den Angreifer aus. Die Benutzer haben davon nichts mitbekommen, was das größte Kompliment ist, das man einem Lockout machen kann.

Kann der LoadMaster den WAP ersetzen? Eine ehrliche Abwägung

Die Frage kommt in jedem zweiten Projekt, meistens vom Betrieb und meistens mit nachvollziehbarer Motivation: Zwei WAP-Server sind zwei weitere Windows-Server in der DMZ, die gepatcht, überwacht und bei jedem Zertifikatswechsel angefasst werden wollen. Der LoadMaster steht ohnehin da und kann mit dem Edge Security Pack sogar Vorauthentifizierung. Warum also nicht einfach die externen Anfragen vom LoadMaster direkt auf die interne Farm schicken?

Technisch geht das. Progress Kemp beschreibt im eigenen ADFS-Deployment-Guide sogar ausdrücklich ein Szenario, in dem der LoadMaster in der DMZ externe Anfragen ohne WAP an die ADFS-Server weiterreicht. Im selben Atemzug schränkt der Hersteller ein: Das gilt nur, wenn keine erweiterten Claim-Regeln wie Netzwerk- oder Vertrauensstufen genutzt werden. Sonst kann der LoadMaster neben der WAP-Farm stehen, aber nicht als Ersatz dienen. Diese Einschränkung ist der Kern der Sache, und sie reicht weiter, als der Satz vermuten lässt.

Was ohne WAP fehlt

Microsoft verlangt von Drittanbieter-Proxys, die den WAP ersetzen sollen, die Unterstützung des Protokolls MS-ADFSPIP. Darüber registriert sich der Proxy mit Zertifikat bei der Farm, setzt die Header x-ms-proxy und x-ms-forwarded-client-ip und übernimmt Aufgaben wie die Weiterleitung der Zertifikatsanmeldung. Ein Proxy, der MS-ADFSPIP nicht spricht, ist für ADFS ein ganz normaler interner Client. ADFS kann eine Anfrage, die über einen solchen Proxy kommt, nicht von einer Anfrage aus dem Büro unterscheiden. Der LoadMaster findet sich nach unserem Kenntnisstand nicht unter den Proxys, die Microsoft für MS-ADFSPIP aufführt; prüfe das für deine Firmware-Version im Zweifel bei Progress nach.

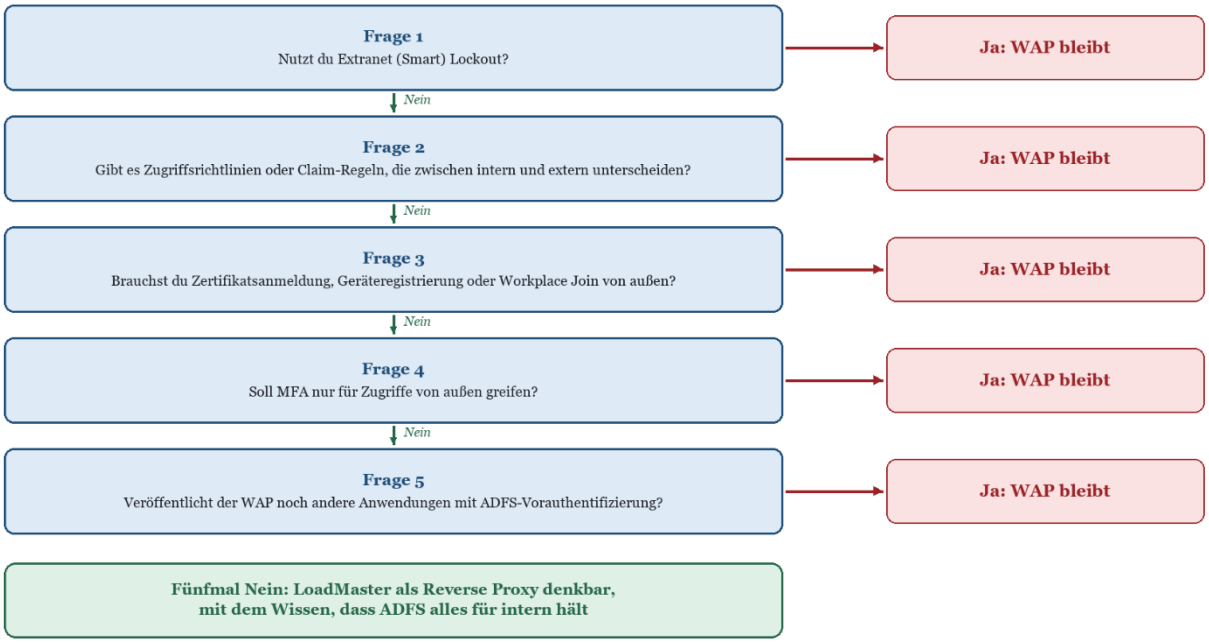
- **Extranet Lockout fällt weg.** ESL und Soft Lockout greifen nur für Anfragen über einen MS-ADFSPIP-Proxy. Ohne WAP schützt dich nur noch die AD-Kontosperre, und die sperrt den Benutzer gleich mit.
- **Intern und extern verschwimmen.** Der Claim insidecorporatenetwork steht auf wahr, Zugriffssteuerungsrichtlinien wie „Von überall mit MFA, im Firmennetz ohne“ lassen jeden Zugriff von außen ohne zweiten Faktor durch. Das ist keine Kleinigkeit, sondern eine Einladung.
- **Anmeldemethoden je Netz greifen nicht.** ADFS bietet extern typischerweise Formularanmeldung und intern Windows-Anmeldung an. Ohne WAP bekommen externe Browser die interne Variante angeboten und sehen ein Anmeldefenster, das mit ihrem Gerät nichts anfangen kann.
- **Zertifikatsanmeldung und Geräteregistrierung leiden.** Ohne TLS-Aufbruch kommen Client-Zertifikate zwar beim ADFS an, aber die Einordnung als externer Zugriff fehlt, und Szenarien wie Workplace Join von außen verhalten sich anders als geplant.

- **Andere WAP-Veröffentlichungen brauchen ein neues Zuhause.** Wenn der WAP auch Anwendungen mit ADFS-Vorauthentifizierung veröffentlicht, musst du diese auf LoadMaster-Vorauthentifizierung umbauen oder einen schlanken WAP dafür behalten.

Zum letzten Punkt: Die Vorauthentifizierung am LoadMaster mit SSO-Domänen ist ein vollwertiger Ersatz für klassische TMG- und WAP-Veröffentlichungen von Webanwendungen. Details stehen im Beitrag Edge Security Pack: Vorauthentifizierung, SSO-Domänen und der TMG-Ersatz. Für ADFS selbst ist sie aber kein Ersatz, denn ADFS will wissen, dass ein Proxy dazwischen steht, und nicht nur, dass jemand vorab ein Kennwort geprüft hat.

Ersetzt der LoadMaster den WAP? Die ehrliche Entscheidungstreppe

Jede Frage mit Ja schiebt dich zurück zum WAP oder zu einem MS-ADFSPIP-fähigen Proxy



Skizze: Entscheidungshilfe nach Praxiserfahrung, keine Herstellervorgabe

Kriterium	WAP hinter dem LoadMaster	LoadMaster statt WAP
Unterstützung durch Microsoft	ja, Standardarchitektur	nicht als ADFS-Proxy, kein MS-ADFSPIP
Extranet Smart Lockout	ja, bei transparenter Client-IP	nein
Unterscheidung intern und extern	ja, über x-ms-proxy	nein, alles gilt als intern
MFA nur extern	ja	nein, bzw. für alle oder keinen
Zertifikatsanmeldung von außen	ja, inklusive Port 49443	eingeschränkt
Server in der DMZ	zwei zusätzliche Windows-Server	keine zusätzlichen
Betriebsaufwand	Patches, Zertifikate, Proxy-Vertrauen	geringer, aber Risiko in den Richtlinien
Geeignet für	jede ADFS-Umgebung mit Zugriff von außen	Labor, Sonderfälle ohne netzabhängige Regeln

Die ehrliche Antwort

Für eine produktive ADFS-Umgebung mit Zugriff aus dem Internet lautet die Antwort: Nein, der LoadMaster ersetzt den WAP nicht, er ergänzt ihn. Der LoadMaster sorgt für Verfügbarkeit, Verteilung, gegebenenfalls Geoblocking vor dem WAP und eine saubere Weitergabe der Client-IP.

Der WAP sorgt dafür, dass ADFS weiß, mit wem es spricht. Wer den WAP streicht, spart zwei Server und verliert dafür das Lockout, die Netzunterscheidung und die externe MFA-Logik. Das ist ein schlechter Tausch, und er fällt meistens erst dann auf, wenn jemand die Anmeldeprotokolle nach einem Vorfall liest.

Die eigentlich interessante Frage ist eine andere: Wie lange brauchst du ADFS überhaupt noch? Viele Umgebungen, die ADFS nur für Microsoft 365 betreiben, können auf Kennwort-Hash-Synchronisierung oder Pass-Through-Authentifizierung mit Entra ID umsteigen und Fachanwendungen per SAML direkt an Entra ID anbinden. Der LoadMaster kann dabei selbst als SAML-Dienstanbieter vor klassischen Webanwendungen stehen. Wie das aussieht, beschreibt der Beitrag SAML, OIDC und Entra ID: der LoadMaster als moderner Identitätsvermittler. Wer die Architekturfrage grundsätzlich stellen will, findet Orientierung in der Entscheidungshilfe: LoadMaster, Entra Application Proxy, Azure Application Gateway oder Reverse Proxy?.

TIPP · Wenn der WAP schon bleibt, dann gut

Halte die WAP-Server schlank: keine weiteren Rollen, nur die nötigen Firewall-Regeln, Updates im selben Zyklus wie die Farm. Prüfe das Proxy-Vertrauen regelmäßig, zum Beispiel mit einem geplanten Abruf der Metadaten über den externen Namen, denn ein WAP, der länger offline war, kann sein Vertrauen verlieren und muss dann per Install-WebApplicationProxy neu registriert werden. Am LoadMaster gehört jeder WAP vor der Wartung aus dem Virtual Service, damit laufende Anmeldungen sauber auslaufen.

FAQ: Häufige Fragen zu Kemp und ADFS

Darf ich TLS am LoadMaster vor ADFS terminieren, wenn ich wieder verschlüssele?

Nein. Microsoft unterstützt TLS-Terminierung am Load Balancer vor ADFS für keinen Anwendungsfall, und Re-Encryption ist technisch eine Terminierung. Das Proxy-Vertrauen des WAP, Zertifikatsanmeldung und Extended Protection gehen dabei kaputt.

Warum ist mein HTTPS-Health-Check gegen ADFS immer rot?

Weil der LoadMaster beim Prüfen der Real-Server-Adresse kein SNI mitschickt und ADFS sein Zertifikat nur an den Hostnamen bindet. Stelle den Check auf HTTP Port 80 und den Pfad /adfs/probe um, wie Microsoft es empfiehlt.

Funktioniert /adfs/probe auch auf dem WAP?

Ja. Der Endpunkt existiert auf ADFS- und WAP-Servern gleichermaßen und antwortet über HTTP mit 200 OK. Er sagt allerdings nichts darüber, ob das Proxy-Vertrauen zur Farm intakt ist.

Brauche ich Persistenz für ADFS am LoadMaster?

Nein. ADFS hält den Anmeldezustand in Cookies und Token, jeder Knoten kann jede Anfrage bedienen. Microsoft rät von IP-basierter Affinität sogar ab, weil sie einzelne Knoten überlasten kann.

Warum zeigt ADFS bei allen Anmeldungen dieselbe Client-IP?

Weil der Virtual Service vor dem WAP mit SNAT arbeitet und der WAP nur die Adresse des LoadMasters sieht. Stelle auf Transparenz um und sorge dafür, dass die Antworten der WAP-Server über den LoadMaster zurücklaufen.

Kann ich auf den WAP verzichten, wenn der LoadMaster Vorauthentifizierung macht?

Technisch ja, fachlich selten. Ohne MS-ADFSPIP-fähigen Proxy hält ADFS jeden Zugriff für intern. Extranet Lockout, netzabhängige Zugriffsrichtlinien und MFA nur für externe Zugriffe funktionieren dann nicht mehr.

Welche Clients scheitern an der SNI-Pflicht?

Moderne Browser und Betriebssysteme nicht. Betroffen sind sehr alte Geräte, eingebettete Bibliotheken, Skripte und Health Checks, die eine IP-Adresse statt des Dienstnamens ansprechen. Für Unbelehrbare gibt es die Fallback-Bindung auf 0.0.0.0:443.

Fazit: Hände weg vom TLS, Augen auf bei der Client-IP

ADFS hinter dem LoadMaster ist kein schwieriges Projekt, wenn man drei Regeln verinnerlicht. Erstens: TLS bleibt zu, an beiden Virtual Services, ohne Ausnahme und ohne „nur mal kurz testen“. Zweitens: Der Health Check läuft per HTTP auf /adfs/probe, weil SNI sonst jeden gesunden Knoten für tot erklärt, und er wird von einem Monitoring ergänzt, das echte Anmeldungen prüft. Drittens: Vor dem WAP arbeitet der LoadMaster transparent, damit ADFS die echte Client-IP sieht und das Smart Lockout den Angreifer aussperrt und nicht das eigene Personal.

Den WAP ersetzt der LoadMaster dabei nicht. Er macht ihn verfügbar, verteilt die Last und reicht die Wahrheit über die Herkunft der Anfragen weiter. Das klingt nach einer bescheidenen Rolle, ist aber genau die, bei der ein Load Balancer glänzt: Wenn er seine Arbeit richtig macht, redet niemand über ihn. Wenn nicht, redet die ganze Belegschaft über ADFS, und zwar ab 7:45 Uhr.

TIPP · Unterstützung bei Planung und Umsetzung

Wenn du ADFS und WAP hinter den LoadMaster bringen, die Client-IP für das Smart Lockout geradeziehen oder den Abschied von ADFS planen willst, hilft dir die Beratung zum Kemp LoadMaster. Wer Virtual Services, Transparenz und Health Checks mit dem eigenen Team im Labor durchspielen möchte, findet das in der Kemp-LoadMaster-Schulung.

WEITERLESEN · Weiter im Thema

› Health Checks, die wirklich prüfen: warum „Port 443 offen“ nicht reicht
› Client-IP sichtbar machen: Transparenz, SNAT und X-Forwarded-For
› SAML, OIDC und Entra ID: der LoadMaster als moderner Identitätsvermittler
› TLS-Offload, Re-Encryption und Zertifikate: PKI trifft LoadMaster
› Edge Security Pack: Vorauthentifizierung, SSO-Domänen und der TMG-Ersatz