

TLS-Offload, Re-Encryption und Zertifikate: PKI trifft LoadMaster

PKI trifft LoadMaster

Zwei Zertifikate, zwei Ablaufdaten, ein Admin, der nur eines davon im Kalender hat.



Re-Encryption: zwei getrennte Verschlüsselungsstrecken, jede mit eigenem Zertifikat und eigenem Ablaufdatum

Skizze: Beispielwerte, Namen und Daten fiktiv

WISSEN

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

› [Kemp LoadMaster](#)

BERATUNG

Zertifikate, Cipher-Sets, Re-Encryption: Wir prüfen gemeinsam, wo deine TLS-Kette reißen kann, bevor sie es tut.

› [Beratung zum Kemp LoadMaster](#)

SCHULUNG

Grundlagen und Praxis für dein Admin-Team: vom ersten Virtual Service bis zum Zertifikatswechsel ohne Krisenrunde.

› [Kemp-LoadMaster-Schulung](#)

Es gibt in der IT genau zwei Arten von Zertifikaten: die, deren Ablaufdatum jemand im Kalender hat, und die, die an einem Montagmorgen um 7:12 Uhr ablaufen. Am LoadMaster hast du meistens beide Sorten gleichzeitig im Einsatz. Eines liegt am Virtual Service und zeigt nach außen, ein zweites liegt auf den Real Servern dahinter. Erneuert wird erfahrungsgemäß das, das man sieht. Das andere meldet sich dann selbst, meist mit einem Anruf aus der Geschäftsleitung.

Dieser Beitrag kümmert sich um alles, was beim kemp ssl offload zwischen Client, LoadMaster und Server passiert: die Wahl zwischen Offload, Re-Encryption und Passthrough und was sie für Kerberos, die Client-IP und die Web Application Firewall bedeutet. Dann die Zertifikate selbst: SAN, SNI, Zwischenzertifikate, interne AD-CA oder öffentliche CA, und warum auf dem LoadMaster UND auf den Real Servern etwas Gültiges liegen muss. Danach härteten wir das Frontend mit Cipher-Sets, TLS 1.3 und HSTS und schauen ehrlich hin, welche Scanner und ActiveSync-Altgeräte dabei auf der Strecke bleiben. Zum Schluss: das jährliche Ablauf-Chaos, das seit 2026 kein jährliches mehr ist.

Die Herstellerdokumentation erklärt dir, wo welcher Haken sitzt. Microsoft Learn erklärt dir, was Exchange oder ADFS erwarten. Die Lücke dazwischen, also warum ein Haken am LoadMaster eine Kennwortabfrage in Outlook auslöst, ist das Thema hier. Wenn dir Begriffe wie Virtual Service, SubVS oder Real Server noch nicht ganz geläufig sind, lies vorher kurz [Virtual Service, Real Server, SubVS: das Grundvokabular des LoadMasters](#). Den Gesamtüberblick über die Serie findest du im [Kemp LoadMaster: der Wissensbereich](#).

FAKTEN · Das Wichtigste in fünf Sätzen

Offload beendet TLS am LoadMaster und schickt Klartext ins Backend, Re-Encryption baut eine zweite TLS-Sitzung zum Server auf, Passthrough reicht TLS unangetastet durch.

Nur bei Offload und Re-Encryption sieht der LoadMaster HTTP: WAF, Edge Security Pack, X-Forwarded-For, Content Rules und Cookie-Persistence setzen das voraus.

Exchange mit Extended Protection verträgt kein Offload; bei Re-Encryption muss am LoadMaster dasselbe Zertifikat liegen wie auf den Exchange-Servern.

Seit dem 15. März 2026 sind öffentliche TLS-Zertifikate höchstens 200 Tage gültig, ab März 2027 100 Tage, ab März 2029 47 Tage.

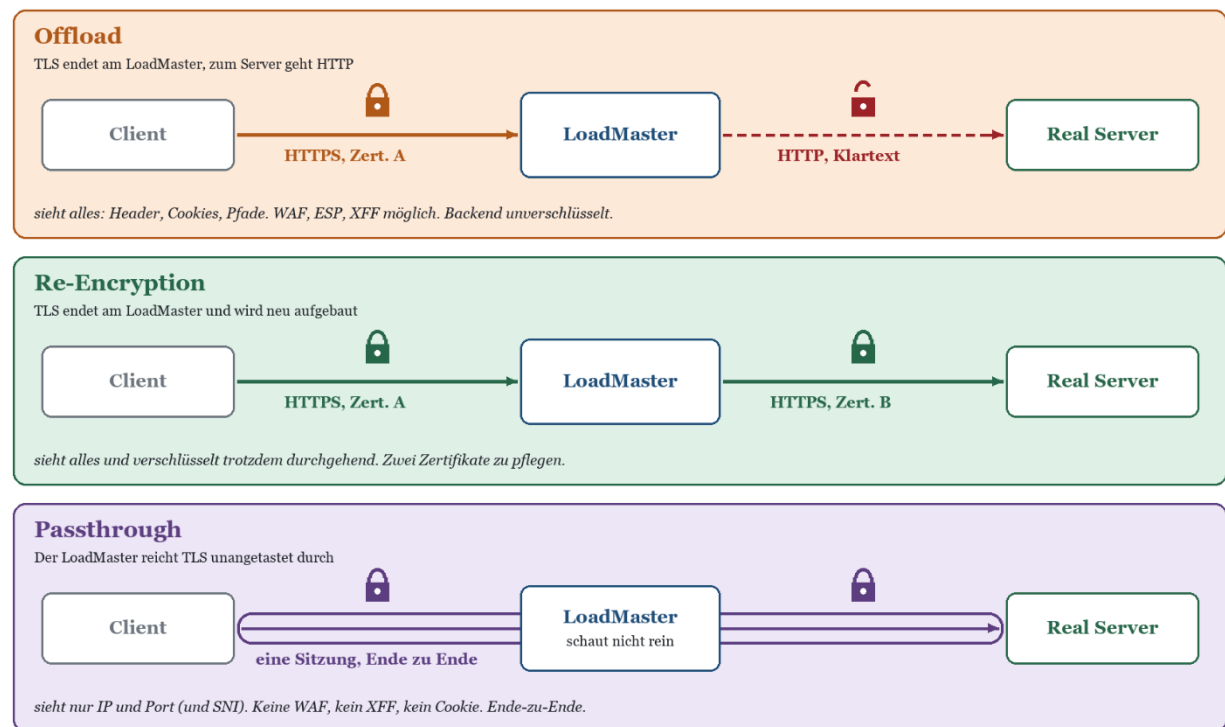
Jede Härtung des Frontends sperrt jemanden aus. Die Frage ist nur, ob du vorher weißt, wen.

Offload, Re-Encryption, Passthrough: wo TLS endet

Der LoadMaster kann mit einer HTTPS-Verbindung drei Dinge tun. Er kann sie beenden, den Inhalt lesen und unverschlüsselt weiterreichen. Er kann sie beenden, lesen und für den Weg zum Server neu verschlüsseln. Oder er kann sie ungelesen durchschieben, wie ein Briefträger, der nicht in die Umschläge schaut. In der Kemp-Welt heißt das erste SSL Acceleration oder Offload, das zweite Re-Encryption (in der Oberfläche die Option Reencrypt, die erst erscheint, wenn SSL Acceleration aktiv ist), das dritte ist schlicht ein Virtual Service ohne SSL Acceleration, also Layer 4 mit TCP auf Port 443.

Drei Wege durch den LoadMaster

Wo TLS endet, entscheidet darüber, was der LoadMaster sehen und tun kann



Skizze: Offload, Re-Encryption und Passthrough im Vergleich, vereinfacht

Offload: schnell, bequem und ein bisschen nackt

Beim reinen Offload entschlüsselt der LoadMaster und spricht mit den Real Servern HTTP auf Port 80. Früher war das ein Leistungsargument: Die Server sparten sich die teure Kryptografie. Heute lacht jede halbwegs aktuelle Server-CPU über AES-GCM, und das Argument ist weitgehend tot. Übrig bleibt das Komfortargument: nur ein Zertifikat zu pflegen, das am LoadMaster. Der Preis

dafür ist Klartext im Backend-Netz, inklusive Kennwörter bei Basic-Authentifizierung, Formularanmeldungen und Session-Cookies.

Dazu kommt ein zweites, subtileres Problem: Die Anwendung weiß nicht mehr, dass der Client HTTPS benutzt hat. Sie generiert Weiterleitungen auf `http://`, setzt Cookies ohne Secure-Flag oder baut absolute Links mit dem falschen Schema. SharePoint löst das über Alternate Access Mappings, Eigenentwicklungen über einen ausgewerteten X-Forwarded-Proto-Header, und Exchange löst es gar nicht, weil Microsoft Offload dort inzwischen schlicht ausschließt, sobald Extended Protection aktiv ist.

Re-Encryption: der Normalfall, auch wenn er nach Arbeit klingt

Bei Re-Encryption entschlüsselt der LoadMaster, schaut hinein, tut, was er tun soll, und baut zum Real Server eine neue TLS-Verbindung auf. Damit bekommst du alle Layer-7-Funktionen und trotzdem durchgehende Verschlüsselung. Das ist in praktisch allen Microsoft-Szenarien die Empfehlung, und es ist auch das, was die Kemp-Templates für Exchange und SharePoint vorsehen. Der Haken: Es gibt jetzt zwei TLS-Strecken mit zwei Zertifikaten, zwei Cipher-Aushandlungen und zwei Ablaufdaten. Wer das als eine Strecke denkt, wird beim ersten Zertifikatswechsel daran erinnert, dass es zwei sind.

Wichtig zu wissen: Der LoadMaster handelt TLS zum Real Server unabhängig vom Frontend aus. Der Client kann TLS 1.3 sprechen, während der LoadMaster mit einem Windows Server 2019 TLS 1.2 spricht, weil HTTP.sys erst ab Windows Server 2022 TLS 1.3 beherrscht. Das ist kein Fehler, das ist der Sinn der Sache. Es bedeutet aber auch, dass eine Härtung des Frontends nichts über das Backend aussagt und umgekehrt.

Passthrough: der LoadMaster als diskreter Türsteher

Ohne SSL Acceleration sieht der LoadMaster nur IP-Adressen, Ports und, wenn er genau hinschaut, den SNI-Namen im Client Hello. Mehr nicht. Kein Cookie, kein Pfad, kein Header. Das klingt nach Rückschritt, ist aber für manche Dienste genau richtig: ADFS und der Web Application Proxy vertrauen einander über Client-Zertifikate, und ein Gerät dazwischen, das TLS aufbricht, zerstört diese Vertrauensbeziehung. Microsoft beschreibt das in den ADFS-Troubleshooting-Leitfäden unmissverständlich: SSL-Terminierung zwischen WAP und ADFS deaktivieren. Details dazu findest du im Beitrag [ADFS und Web Application Proxy mit dem LoadMaster: Farm, Probe, SNI](#).

Was die Wahl für Kerberos, Client-IP und WAF bedeutet

Jetzt wird es interessant, denn hier liegt der eigentliche Grund für diesen Beitrag. Die TLS-Entscheidung wirkt auf Dinge, die auf den ersten Blick nichts mit Verschlüsselung zu tun haben.

Kerberos. Ein Kerberos-Ticket reist bei HTTP im Authorization-Header. Der LoadMaster reicht diesen Header bei Offload und Re-Encryption unverändert durch, Kerberos an sich funktioniert also in allen drei Modi, solange der SPN zum Namen passt, den der Client benutzt, und beim richtigen Dienstkonto registriert ist. Wo es knallt, ist die Kanalbindung. Extended Protection for Authentication (EPA) verknüpft die Anmeldung mit dem TLS-Kanal, konkret mit einem Hash des Serverzertifikats, das der Client gesehen hat. Bei Offload sieht der Server gar keinen TLS-Kanal, die Bindung schlägt fehl. Bei Re-Encryption sieht er einen anderen Kanal, und die Bindung passt nur dann, wenn der LoadMaster dasselbe Zertifikat präsentiert wie der Server. Genau deshalb verlangt Microsoft für Exchange mit Extended Protection bei SSL Bridging dasselbe Zertifikat an beiden Stellen. Mehr zu SPNs hinter dem Balancer steht in [SPN-Probleme hinter dem Load Balancer: ein Name, mehrere Server](#), die Grundlagen im [Microsoft Kerberos: der Wissensbereich](#).

Client-IP. Bei Layer-7-Verarbeitung, also Offload und Re-Encryption, ist der LoadMaster ein Proxy. Der Real Server sieht standardmäßig die Adresse des LoadMasters als Absender. Dafür kann

der LoadMaster einen X-Forwarded-For-Header einfügen, den IIS-Logs und Anwendungen auswerten können. Bei Passthrough gibt es keinen Header, in den man etwas schreiben könnte. Dort bleibt nur die transparente Weiterleitung mit dem LoadMaster als Gateway der Real Server, mit allen Folgen für das Routing. Ausführlich: [Client-IP sichtbar machen: Transparenz, SNAT und X-Forwarded-For](#).

WAF. Eine Web Application Firewall, die verschlüsselten Verkehr prüfen soll, ist ein Sicherheitsbeauftragter mit verbundenen Augen. Die WAF des LoadMasters braucht entschlüsselten Verkehr, also Offload oder Re-Encryption. Wer Passthrough fährt und gleichzeitig die WAF einschaltet, hat ein gutes Gefühl und sonst nichts. Wie du sie ohne Fehlalarmflut betreibst, steht in [Web Application Firewall und Geoblocking am LoadMaster: Schutz ohne Fehlalarm-Hölle](#).

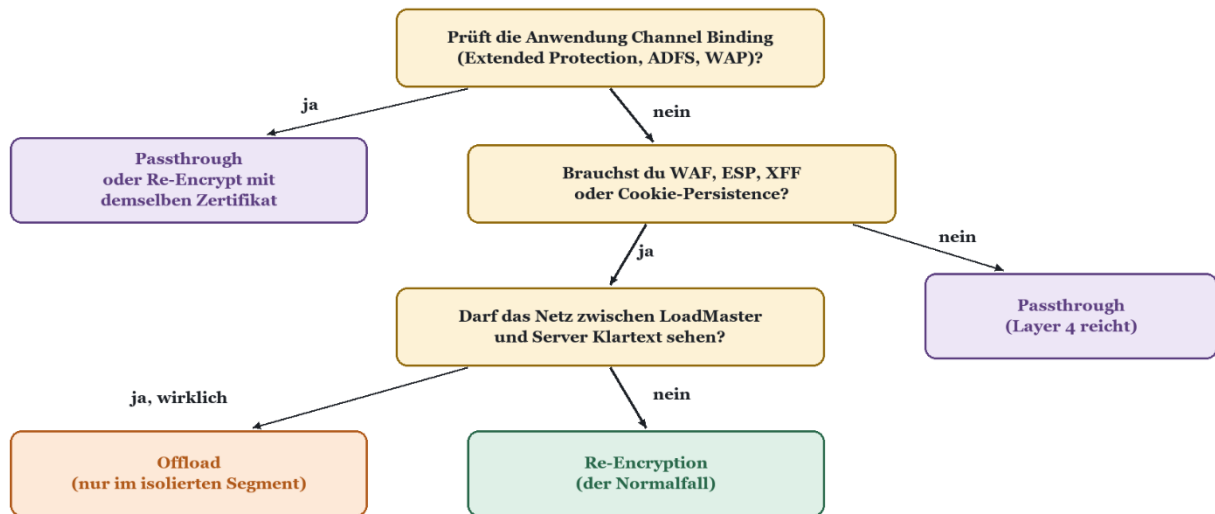
Aspekt	Offload	Re-Encryption	Passthrough
Backend verschlüsselt	nein	ja	ja (Ende zu Ende)
Zertifikate zu pflegen	eines (LoadMaster)	zwei (LoadMaster und Server)	eines (Server)
Kerberos (Header)	funktioniert	funktioniert	funktioniert
Channel Binding / EPA	bricht	nur mit identischem Zertifikat	funktioniert
X-Forwarded-For	ja	ja	nein
WAF, ESP, Content Rules	ja	ja	nein
Cookie-Persistence	ja	ja	nein (nur Source IP)
Client-Zertifikate an den Server	nur über Header-Weitergabe	nur über Header-Weitergabe	direkt
Typische Microsoft-Kandidaten	selten sinnvoll	Exchange, SharePoint, RD Web, IIS	ADFS, WAP-Backend, Sonderfälle

WARNUNG · Offload und Exchange vertragen sich nicht mehr

Seit Exchange Server 2019 CU14 ist Extended Protection standardmäßig aktiv. Microsoft schreibt in der Dokumentation zu [Extended Protection in Exchange Server](#) klar, dass SSL Offloading in diesem Fall nicht unterstützt wird und dass bei SSL Bridging am Load Balancer dasselbe Zertifikat wie auf Exchange liegen muss. Wer ein altes Template mit Offload übernimmt, bekommt Kennwortabfragen in Outlook und fehlschlagende Anmeldungen, die aussehen wie ein Kerberos-Problem und keines sind.

Offload, Re-Encryption oder Passthrough?

Eine Entscheidungshilfe mit Microsoft-Brille, ohne Anspruch auf Vollständigkeit



Faustregel: Re-Encryption ist der Standard. Offload nur, wenn du Klartext im Backend begründen kannst. Passthrough, wenn die Anwendung Ende-zu-Ende-TLS verlangt.

Skizze: vereinfachter Entscheidungsbaum

Workload	Empfehlung	Begründung
Exchange Server (OWA, EWS, MAPI, ActiveSync)	Re-Encryption, identisches Zertifikat	Extended Protection prüft die Kanalbindung gegen das Serverzertifikat.
SharePoint Server SE	Re-Encryption	Layer 7 für Persistence und Header, interne CA im Backend möglich, AAM bzw. öffentliche URL auf https.
ADFS-Farm	Passthrough oder Re-Encryption mit SNI	Zertifikatsbasierte Vertrauensstellung zum WAP; Health Probe über HTTP-Endpunkt.
Web Application Proxy (extern)	Passthrough oder Re-Encryption	Hängt davon ab, ob vor dem WAP noch eine WAF gewünscht ist.
RD Gateway	Re-Encryption (HTTPS-Teil)	UDP 3391 lässt sich nicht als TLS terminieren und läuft separat als Layer 4.
RD Web	Re-Encryption	Cookie-Persistence und Header-Steuerung benötigen Layer 7.
IIS-Eigenentwicklung	Re-Encryption	Offload nur, wenn die Anwendung X-Forwarded-Proto sauber auswertet.

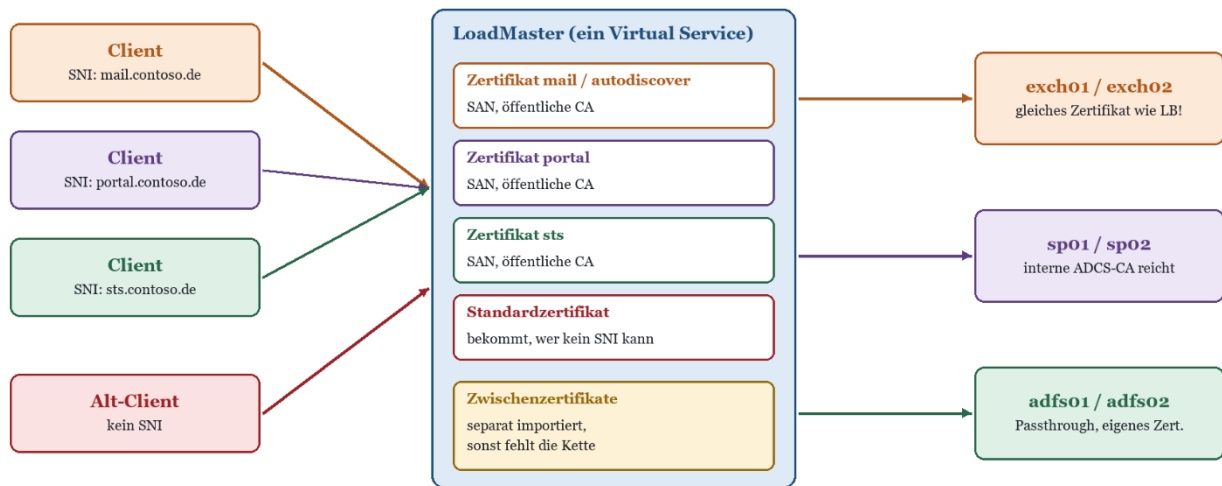
Wenn du den LoadMaster zusätzlich als Vorauthentifizierung mit [Edge Security Pack: Vorauthentifizierung, SSO-Domänen und der TMG-Ersatz](#) nutzt, ist Layer 7 ohnehin gesetzt. Dann holt sich der LoadMaster per KCD selbst ein Kerberos-Ticket für den Benutzer, und das Thema Kanalbindung verschiebt sich auf die Strecke zwischen LoadMaster und Real Server. Wie das sauber geht, steht in [Kerberos Constrained Delegation am LoadMaster: KCD und RBCD sauber einrichten](#).

Zertifikate am LoadMaster: SAN, SNI, Kette und Herkunft

Ein Zertifikat am LoadMaster ist technisch dasselbe wie eines im IIS. Der Unterschied liegt im Drumherum: Der LoadMaster hat keinen Windows-Zertifikatsspeicher, kein Autoenrollment und keine Gruppenrichtlinie, die ihm vertrauenswürdige Stammzertifizierungsstellen verteilt. Alles, was Windows dir stillschweigend abnimmt, musst du hier bewusst tun. Importiert werden Zertifikate unter Certificates & Security, die Serverzertifikate mit privatem Schlüssel unter SSL Certificates, die Zwischenzertifikate getrennt davon unter Intermediate Certs. Den Export aus Windows erledigst du als PFX mit privatem Schlüssel und vollständiger Kette, das Kennwort fragt der Import ab.

Was am LoadMaster liegt und was am Real Server

Ein Virtual Service, mehrere Namen, zwei Zertifikatebenen und eine Kette dazwischen



Frontend: Zertifikat A, wird nach SNI-Namen ausgewählt

Re-Encrypt: zweite TLS-Sitzung, Zertifikat B

Skizze: Zertifikatsablage bei Offload, Re-Encryption und Passthrough, Namen beispielhaft

SAN: der Common Name ist Geschichte

Browser werten seit Jahren ausschließlich den Subject Alternative Name aus. Der Common Name ist ein Relikt, das Zertifikatsassistenten aus Höflichkeit noch befüllen. Für den LoadMaster heißt das: Jeder Name, unter dem ein Virtual Service erreichbar ist, gehört als DNS-Eintrag in den SAN. Bei Exchange sind das typischerweise mail.contoso.de und autodiscover.contoso.de, gegebenenfalls ergänzt um einen separaten Namen für ActiveSync oder MAPI. Wildcard-Zertifikate sind am LoadMaster kein Problem, decken aber nur eine Ebene ab und machen aus jedem Schlüsseldiebstahl einen Flächenbrand.

SNI: mehrere Zertifikate auf einer IP

Server Name Indication schickt den gewünschten Hostnamen im Client Hello mit, bevor das Zertifikat ausgetauscht wird. Damit kann ein Virtual Service auf einer IP-Adresse mehrere Zertifikate tragen und je nach Name das passende ausliefern. Am LoadMaster weist du dem Virtual Service dafür mehrere Zertifikate zu; die Zuordnung erfolgt über die Namen im Zertifikat. Mit der Option Require SNI hostname lehnt der LoadMaster Clients ab, die keinen oder einen unbekannten Namen schicken. Das ist sauber, weil es Scans gegen die nackte IP abblitzen lässt, und gefährlich, weil es genau die Alt-Clients trifft, die du am wenigsten auf dem Schirm hast.

Bei Re-Encryption gibt es SNI ein zweites Mal: Der LoadMaster muss beim Verbindungsaufbau zum Real Server einen Namen mitschicken, wenn das Backend SNI-Bindungen verwendet. ADFS ist der Klassiker: Es bindet HTTP.sys standardmäßig per SNI an den Federation-Service-Namen. Ein Health Check oder eine Re-Encryption ohne passenden SNI-Namen landet dann ins Leere, oft mit einem Verbindungsabbruch ohne brauchbare Fehlermeldung. Wie du Prüfungen baust, die so etwas erkennen, erklärt [Health Checks, die wirklich prüfen: warum „Port 443 offen“ nicht reicht.](#)

Zwischenzertifikate: die Kette, die keiner sieht

Der häufigste Zertifikatsfehler am LoadMaster ist keiner, den du am eigenen Rechner siehst. Windows-Clients holen fehlende Zwischenzertifikate über die AIA-Erweiterung selbst nach und zeigen ein grünes Schloss. Android-Geräte, Java-Anwendungen, Python-Skripte und viele Embedded-Clients tun das nicht. Sie brechen ab, und der Benutzer meldet „Mail geht auf dem Handy nicht“. Deshalb gehört das Zwischenzertifikat der ausstellenden CA zwingend auf den LoadMaster. Und wenn eine öffentliche CA ihre Zwischenzertifikate wechselt, was in den letzten Jahren häufiger passiert ist, als irgendjemandem lieb war, musst du auch das neue importieren.

TIPP · Kette prüfen, nicht glauben

Prüfe nach jedem Import von außen, mit einem Werkzeug, das keine Ketten nachlädt. Unter Windows eignet sich ein Aufruf von openssl s_client mit -showcerts und -servername, sofern verfügbar, oder ein kleines PowerShell-Skript, das die vom Server gesendete Kette ausliest. Am Mobiltelefon ohne WLAN testen ist die schwarzhumorige, aber erstaunlich zuverlässige Alternative.

Interne ADACS-CA oder öffentliche CA?

Die Frage ist weniger eine Glaubensfrage als eine Frage, wer dem Zertifikat vertrauen muss. Domänenmitglieder vertrauen deiner ADACS-CA, weil die Gruppenrichtlinie ihnen das Stammzertifikat verteilt. Private Smartphones, Partner und externe Dienste wie Entra ID oder Exchange Online tun das nicht. Für alles, was von außen erreichbar ist, brauchst du also am Frontend eine öffentliche CA. Für die Strecke zwischen LoadMaster und Real Server reicht oft eine interne CA, weil dort nur der LoadMaster vertrauen muss. Dafür importierst du das Stammzertifikat deiner ADACS-CA auf dem LoadMaster.

Kriterium	Interne ADACS-CA	Öffentliche CA
Vertrauen	nur Geräte mit verteiltem Stammzertifikat	praktisch alle Geräte
Laufzeit	frei wählbar über die Vorlage	seit 15.03.2026 max. 200 Tage, ab 15.03.2027 max. 100 Tage
Interne Namen (contoso.local)	möglich	nicht möglich
Automatisierung auf Windows	Autoenrollment über Vorlage	ACME-Clients oder Werkzeuge der CA
Automatisierung am LoadMaster	Skript über REST-API	ACME, sofern deine Firmware es unterstützt, sonst Skript über REST-API
Typischer Einsatz	Backend bei Re-Encryption, interne Dienste	Frontend aller extern erreichbaren Virtual Services
Sonderfall Exchange mit EP	nur, wenn auch das Frontend intern ist	Frontend UND Exchange-Server mit identischem Zertifikat

WICHTIG · Exchange ist die Ausnahme von der Faustregel

Die elegante Trennung, außen öffentlich und innen ADCS, funktioniert bei Exchange mit Extended Protection nicht. Weil die Kanalbindung das Zertifikat vergleicht, das der Client gesehen hat, muss das öffentliche Zertifikat auch auf den Exchange-Servern liegen. Mit identischem Thumbprint, nicht nur mit gleichem Namen. Das verdoppelt die Arbeit bei jeder Erneuerung und ist der Hauptgrund, warum Exchange-Umgebungen beim Zertifikatswechsel so zuverlässig in Flammen aufgehen. Die Namensraum- und Template-Seite dazu beleuchtet der Beitrag zu [Exchange Server hinter dem LoadMaster: Templates, Namespaces, Health Checks](#).

Zertifikat auf LoadMaster UND Real Server

Bei Re-Encryption braucht jeder Real Server ein eigenes gültiges Zertifikat. Wie streng der LoadMaster dieses Backend-Zertifikat prüft, hängt von deiner Konfiguration und Firmware ab. Viele Installationen prüfen es gar nicht, und das ist ein zweischneidiges Schwert: Ein abgeladenes Backend-Zertifikat stört dann zunächst niemanden, bis jemand die Prüfung einschaltet, der Health Check strenger wird oder, bei Exchange, die Kanalbindung zuschlägt. Dann fällt die ganze Farm auf einmal um, nicht ein Server nach dem anderen.

Und denk an den HA-Partner. Zertifikate werden im HA-Paar synchronisiert, aber nur, wenn die Synchronisierung läuft und du am aktiven Gerät importierst. Wer im Stress am passiven Knoten arbeitet, hat nach dem nächsten Failover das alte Zertifikat wieder. Wie die Synchronisierung im Detail funktioniert, steht in [HA-Paar, Failover und Konfig-Sync: zwei LoadMaster, ein Gedanke](#).

Frontend härten: Cipher-Sets, TLS 1.3 und HSTS

Härten ist am LoadMaster angenehm zentral. Statt auf jedem IIS-Server mit Registry-Schlüsseln oder IIS Crypto zu hantieren, legst du am Virtual Service fest, welche Protokolle und welche Cipher-Suiten nach außen angeboten werden. Die Real Server dahinter können ihr eigenes Tempo gehen. Das ist ein echter Vorteil, solange du nicht vergisst, dass die Härtung nur für den Abschnitt gilt, den der LoadMaster terminiert.

Cipher-Sets: vorgefertigte Kompromisse

Der LoadMaster bündelt Cipher-Suiten zu benannten Cipher-Sets, die du dem Virtual Service zuweist. Für TLS 1.3 gibt es seit LMOS 7.2.56 eine eigene Auswahl der fünf TLS-1.3-Suiten, standardmäßig sind alle aktiv. Die Sets für TLS 1.2 und älter unterscheiden sich vor allem darin, wie viel Vergangenheit sie zulassen.

Cipher-Set	Gedacht für	Einschätzung
BestPractices	Standard für neue Virtual Services	Von Progress empfohlen und gelegentlich an aktuelle Empfehlungen angepasst, zuletzt um schwache Suiten bereinigt.
Intermediate_compatibility	Umgebungen mit älteren Browsern und Diensten	Kompromiss, wenn BestPractices nachweislich Clients aussperrt.
Backward_compatibility	Clients bis zurück zu Windows XP und IE6	Nur als letzter Ausweg und zeitlich befristet, am besten auf einem eigenen Virtual Service.
Eigenes Set	Sonderanforderungen, etwa Vorgaben des Informationssicherheitsbeauftragten	Sauber dokumentieren, sonst versteht in drei Jahren niemand mehr, warum.

WARNUNG · Firmware-Updates verändern Cipher-Sets

Weil Progress das Set BestPractices bei Firmware-Updates nachschärft, können nach einem Update

Clients ausgesperrt sein, die gestern noch funktioniert haben. Die Release Notes weisen ausdrücklich darauf hin, die verwendeten Sets vorher zu sichern. Wer auf dem Altgerät im Lager eine Ausnahme braucht, baut sich ein eigenes Set, statt beim nächsten Update überrascht zu werden.

TLS 1.0 und 1.1 abschalten

Am Virtual Service wählst du die erlaubten Protokollversionen einzeln aus. Die Empfehlung von Progress lautet TLS 1.2 und TLS 1.3, und genau so sollte es bei jedem neuen Virtual Service aussehen. Microsoft zieht auf Windows-Seite in dieselbe Richtung: TLS 1.0 und 1.1 sind in neuen Windows-Versionen von Client und Server standardmäßig deaktiviert, und Microsoft kündigt die vollständige Entfernung für künftige Versionen an. Wer am LoadMaster noch TLS 1.0 anbietet, verteidigt also eine Position, die selbst der Betriebssystemhersteller aufgegeben hat.

TLS 1.3: schneller, schlanker, weniger Stellschrauben

TLS 1.3 verkürzt den Handshake, entfernt statischen RSA-Schlüsselaustausch und erlaubt nur noch AEAD-Suiten. Am Frontend spricht nichts dagegen. Zwei Dinge solltest du trotzdem wissen. Erstens: Persistence über die SSL Session ID ist unter TLS 1.3 praktisch wertlos, weil die Session ID dort nur noch aus Kompatibilitätsgründen existiert. Zweitens: Im Backend entscheidet das Betriebssystem der Real Server. Windows Server 2019 kann kein TLS 1.3, Windows Server 2022 und 2025 schon. Der LoadMaster handelt das bei Re-Encryption automatisch passend aus.

HSTS: ein Versprechen, das der Browser sich merkt

HTTP Strict Transport Security weist den Browser an, eine Domäne für eine festgelegte Zeit nur noch per HTTPS anzusprechen. Am LoadMaster kannst du den Header je nach Firmware über eine eigene Option in den Virtual-Service-Eigenschaften setzen oder über eine Content Rule, die den Strict-Transport-Security-Header in die Antwort einfügt. Wie solche Header-Regeln aufgebaut sind, erklärt [Content Rules: Header umschreiben, Pfade umleiten, Dinge verstecken](#).

HSTS ist einer der wenigen Sicherheitsmechanismen mit eingebautem Langzeitgedächtnis, und genau das macht ihn gefährlich. Ein max-age von einem Jahr bedeutet: Wenn du dich vertan hast, dauert die Korrektur ein Jahr, bei jedem Browser, der den Header einmal gesehen hat. Mit includeSubDomains auf contoso.de erklärst du gleich alle Subdomänen für HTTPS-pflichtig, auch das Druckerportal, die Zeiterfassung und das NAS-Webinterface, die bisher fröhlich per HTTP liefen. Fang deshalb mit kurzen Laufzeiten an, etwa einem Tag, und verlängere schrittweise.

Alt-Clients: Scanner, ActiveSync-Geräte und andere Fossilien

Jetzt der unangenehme Teil. Härtung ist kein abstraktes Sicherheitsthema, sie sperrt konkrete Geräte aus. In Verwaltungen und im Mittelstand sind das selten Browser, sondern Multifunktionsgeräte, die per Scan-to-Web auf SharePoint ablegen, Smartphones aus dem Lager-Rotationspool, die ActiveSync gegen Exchange sprechen, Zeiterfassungsterminals, Telefonanlagen mit Exchange-Kalenderanbindung und Fachanwendungen auf einer alten .NET-Version.

Wer bleibt draußen, wenn du härtest?

Jede Stufe der Härtung sperrt eine Generation von Geräten aus, manche davon unerwartet wichtig



Skizze: typische Betroffene je Härtungsstufe, keine abschließende Liste

Client	Typisches Problem	Was du tun kannst
Multifunktionsgerät, Scanner	Firmware spricht nur TLS 1.0 oder alte CBC-Suiten	Firmware aktualisieren; sonst eigener interner Virtual Service mit gelockertem Set, nicht der externe.
Android vor Version 5	TLS 1.2 nicht standardmäßig aktiv	Gerät ersetzen. Ehrlich.
ActiveSync auf Altgeräten	fehlendes SNI, fehlende moderne Suiten, unvollständige Kette	Kette prüfen, Standardzertifikat am Virtual Service passend wählen, Geräteinventar abgleichen.
.NET-Fachanwendung	nutzt standardmäßig alte Protokolle	SchUseStrongCrypto und SystemDefaultTlsVersions auf dem Client setzen oder Anwendung aktualisieren.
Skripte, Monitoring-Agenten	fehlendes SNI, alte Bibliotheken	Bibliothek aktualisieren, beim Monitoring SNI-Namen konfigurieren.
Windows 7 und Server 2008 R2	TLS 1.2 nur nach Update und Registry-Eingriff	Nicht mehr unterstützt; als Risiko dokumentieren und ablösen.

AUS DER PRAXIS · Der Scanner, der nicht mehr wollte

Eine Verwaltung schaltete am externen und internen SharePoint-Virtual-Service gleichzeitig TLS 1.0 und 1.1 ab. Die Browser liefen, die Tests waren grün. Am nächsten Morgen standen auf drei Etagen Mitarbeiterinnen und Mitarbeiter mit Papierstapeln vor Multifunktionsgeräten, die per Scan-to-SharePoint ablegen sollten und nur noch Fehlercodes zeigten. Die Lösung war ein eigener interner Virtual Service für die Geräte, mit gelockertem Cipher-Set, nur aus dem Druckernetz erreichbar, und ein Ticket an die Beschaffung. Der externe Dienst blieb hart. So soll es sein.

TIPP · Erst messen, dann härten

Bevor du Protokolle abschaltest, schau in die Statistiken und Logs des LoadMasters, welche Protokollversionen und Suiten tatsächlich ausgehandelt werden. Auf Windows-Seite helfen die Schannel-Ereignisse der Real Server und die IIS-Logs mit den erweiterten TLS-Feldern. Was keiner benutzt, kann weg. Was jemand benutzt, braucht einen Plan, keinen Überraschungseffekt.

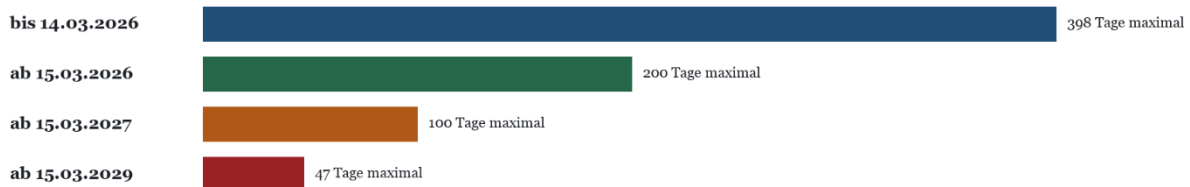
Praxis: das jährliche Ablauf-Chaos, wenn nur eine Seite erneuert wird

Es beginnt immer harmlos. Die Überwachung meldet, dass das Zertifikat für mail.contoso.de in 14 Tagen abläuft. Jemand bestellt ein neues, importiert es am LoadMaster, weist es dem Virtual

Service zu, testet mit dem Browser, sieht ein grünes Schloss und schließt das Ticket. Auf den Exchange-Servern liegt das alte Zertifikat noch, das in 14 Tagen abläuft. Oder es liegt dort schon das neue, weil eine Kollegin parallel auf den Servern erneuert hat, aber der LoadMaster hat noch das alte. Beide Varianten enden gleich: mit einem Anruf.

Das jährliche Ablauf-Chaos, jetzt zweimal im Jahr

Öffentliche Zertifikatslaufzeiten schrumpfen, das Gedächtnis der IT-Abteilung nicht im gleichen Tempo



Und so läuft eine typische Erneuerung, wenn nur eine Seite dran denkt:



Skizze: Laufzeiten nach CA/Browser-Forum-Beschluss SC-081; Ablauf der Erneuerung beispielhaft

Warum das seit 2026 kein jährliches Problem mehr ist

Im April 2025 hat das CA/Browser Forum mit dem Beschluss SC-081 die maximale Laufzeit öffentlicher TLS-Zertifikate stufenweise verkürzt. Seit dem 15. März 2026 gilt ein Maximum von 200 Tagen, ab dem 15. März 2027 sind es 100 Tage, ab dem 15. März 2029 nur noch 47 Tage. Was früher einmal im Jahr schiefging, geht jetzt zweimal im Jahr schief, nächstes Jahr viermal und ab 2029 ungefähr achtmal. Spätestens dann ist manuelle Erneuerung kein Prozess mehr, sondern ein Hobby.

Die vier Klassiker

Szenario	Symptom	Ursache
Nur LoadMaster erneuert, Exchange alt	Outlook fragt nach Kennwörtern, ActiveSync-Anmeldungen schlagen fehl	Kanalbindung vergleicht unterschiedliche Zertifikate (Extended Protection).
Nur Real Server erneuert, LoadMaster alt	Öffentliche Zertifikatswarnung am Ablauftag, Server selbst wirken gesund	Das Frontend-Zertifikat liegt allein am LoadMaster.
Neues Zertifikat, neues Zwischenzertifikat vergessen	Desktop funktioniert, Mobilgeräte und Skripte brechen ab	Die CA hat ihre Zwischen-CA gewechselt, die Kette ist unvollständig.
Backend-Zertifikat abgelaufen, lange unbemerkt	Nach Firmware-Update oder Änderung am Health Check plötzlich alle Real Server down	Die Prüfung des Backend-Zertifikats war vorher lax oder deaktiviert.

Ablaufüberwachung, die den Namen verdient

Der LoadMaster zeigt in der Zertifikatsübersicht die Ablaufdaten an. Das ist hilfreich, wenn jemand hinschaut, was erfahrungsgemäß niemand tut. Überwachung heißt deshalb: Beide Seiten werden regelmäßig maschinell abgefragt, und das Ergebnis landet in einem System, das dich rechtzeitig weckt.

- **Von außen:** Ein Monitoring-Check gegen jeden öffentlichen Namen mit SNI, der Ablaufdatum und Kette prüft. Das ist die Sicht deiner Benutzer.
- **Am LoadMaster:** Ein Skript über die REST-API, das alle installierten Zertifikate und ihre Ablaufdaten ausliest. Wie das mit PowerShell geht, steht in der Automatisierungsanleitung.
- **Auf den Real Servern:** Ein Check gegen jeden Real Server direkt, nicht über den Virtual Service, sonst misst du nur wieder das Frontend.
- **In der ADCS-CA:** Ausgestellte Zertifikate aus Vorlagen ohne Autoenrollment regelmäßig auswerten, etwa über die Datenbankabfrage der CA.
- **Im HA-Paar:** Nach jedem Import am passiven Knoten prüfen, ob das neue Zertifikat angekommen ist.

Die Anbindung des LoadMasters an Syslog und SNMP beschreibt [Monitoring, Syslog und SNMP: den LoadMaster ins Monitoring holen](#), die Skriptseite [Automatisierung per REST-API und PowerShell](#).

Eine Checkliste, die erst schließt, wenn alles erledigt ist

Schritt	Erledigt, wenn
1. Inventar	Für jeden Virtual Service ist bekannt, welches Zertifikat am Frontend und welches auf jedem Real Server liegt, inklusive Herkunft und Ablaufdatum.
2. Bestellen oder ausstellen	Alle SAN-Einträge stimmen, bei Exchange mit EP derselbe Schlüssel für LoadMaster und Server.
3. Kette importieren	Neues Zwischenzertifikat am LoadMaster vorhanden, alte bleiben bis zum Ablauf.
4. Frontend wechseln	Zertifikat am Virtual Service und allen SubVS bzw. SNI-Zuweisungen getauscht.
5. Backend wechseln	Bindungen in IIS bzw. Exchange-Dienste aktualisiert, Thumbprint verglichen.
6. HA prüfen	Passiver Knoten zeigt das neue Zertifikat.
7. Von außen testen	Kette und Ablaufdatum mit einem Werkzeug geprüft, das keine Zwischenzertifikate nachlädt, plus ein echtes Mobilgerät.
8. Altes entfernen	Altes Zertifikat nach Wartezeit gelöscht, Überwachung auf das neue Ablaufdatum umgestellt.

FAKTEN · Was Automatisierung heute leisten kann

Auf Windows-Seite deckt Autoenrollment über ADCS-Vorlagen das Backend ab, für öffentliche Zertifikate gibt es ACME-Clients. Am LoadMaster lassen sich Zertifikate per REST-API hochladen und zuweisen; neuere Firmware-Versionen bringen zusätzlich ACME-Unterstützung mit, deren Umfang du für deine Version prüfen solltest. Der eigentliche Engpass ist selten die Technik, sondern die Frage, wer im Haus für die Kette von Bestellung bis Überwachung verantwortlich ist.

AUS DER PRAXIS · Zwei Teams, ein Zertifikat, null Absprache

Im Haus eines Mittelständlers betreute das Netzwerkteam den LoadMaster, das Messaging-Team die Exchange-Server. Beide hatten das Ablaufdatum im Kalender, beide bestellten ein neues Zertifikat, jeweils mit eigenem Schlüssel. Ergebnis: zwei gültige, aber unterschiedliche Zertifikate, Extended Protection schlug Alarm, Outlook fragte zwei Tage lang nach Kennwörtern. Die Lösung war banal: ein

gemeinsames Inventar, ein Verantwortlicher pro Zertifikat und die Regel, dass Exchange-Zertifikate nur einmal bestellt und auf beide Seiten verteilt werden.

Wenn du bei der Bestandsaufnahme Unterstützung brauchst oder den Zertifikatsprozess für LoadMaster und Microsoft-Server gemeinsam neu aufsetzen willst, ist das ein typisches Thema für eine [Beratung zum Kemp LoadMaster](#).

FAQ

Was ist der Unterschied zwischen SSL-Offload und Re-Encryption am LoadMaster?

Beim Offload endet TLS am LoadMaster, und der Verkehr geht unverschlüsselt zum Real Server. Bei Re-Encryption endet TLS ebenfalls am LoadMaster, wird aber für den Weg zum Server neu aufgebaut. In beiden Fällen kann der LoadMaster den HTTP-Verkehr lesen und bearbeiten.

Funktioniert Kerberos mit TLS-Offload?

Die Kerberos-Anmeldung selbst ja, denn das Ticket steckt im HTTP-Header und wird durchgereicht. Was nicht funktioniert, ist Kanalbindung über Extended Protection, weil der Server bei Offload keinen TLS-Kanal sieht und bei Re-Encryption einen anderen als der Client.

Muss bei Exchange auf dem LoadMaster und den Servern dasselbe Zertifikat liegen?

Ja, sobald Extended Protection aktiv ist, was seit Exchange Server 2019 CU14 standardmäßig der Fall ist. Microsoft unterstützt dann kein SSL Offloading, und bei SSL Bridging muss das Zertifikat am Load Balancer mit dem auf Exchange übereinstimmen.

Reicht eine interne ADCS-CA für den LoadMaster?

Für das Backend bei Re-Encryption und für rein interne Dienste ja. Für alles, was private Geräte, Partner oder Cloud-Dienste erreichen, brauchst du am Frontend ein Zertifikat einer öffentlichen CA.

Warum funktioniert mein Zertifikat am PC, aber nicht auf dem Smartphone?

Fast immer fehlt das Zwischenzertifikat am LoadMaster. Windows lädt fehlende Zwischenzertifikate selbst nach, viele Mobilgeräte und Anwendungen nicht.

Welches Cipher-Set sollte ich verwenden?

Für neue Virtual Services BestPractices, zusammen mit TLS 1.2 und TLS 1.3. Gelockerte Sets wie `Intermediate_compatibility` oder `Backward_compatibility` nur dort, wo nachweislich Clients es brauchen, und am besten auf einem eigenen internen Virtual Service.

Kann ich TLS 1.0 und 1.1 einfach abschalten?

Technisch ja, organisatorisch erst nach einem Blick in die Logs. Betroffen sind meist Scanner, Altgeräte mit ActiveSync und Fachanwendungen auf alten .NET-Versionen, nicht die Browser.

Ist HSTS am LoadMaster riskant?

Nur, wenn du mit langer Laufzeit und `includeSubDomains` startest. Beginne mit einem kurzen `max-age` und prüfe, ob unter der Domäne noch Dienste laufen, die nur HTTP können.

Wie oft muss ich öffentliche Zertifikate jetzt erneuern?

Seit dem 15. März 2026 spätestens alle 200 Tage, ab dem 15. März 2027 alle 100 Tage und ab dem 15. März 2029 alle 47 Tage. Ohne Automatisierung wird das unbequem.

Fazit: zwei Strecken, zwei Zertifikate, ein Prozess

Die Wahl zwischen Offload, Re-Encryption und Passthrough ist keine Geschmacksfrage. Sie entscheidet, ob Kanalbindung funktioniert, ob der Server die echte Client-IP erfährt und ob deine

WAF überhaupt etwas sieht. In der Microsoft-Welt ist Re-Encryption der Normalfall, Passthrough die Ausnahme für ADFS und Co., und reines Offload ein Auslaufmodell, das Exchange inzwischen aktiv bestraft.

Bei den Zertifikaten gilt: Was am LoadMaster liegt, muss nach außen passen, mit vollständigem SAN, passendem SNI und lückenloser Kette. Was auf den Real Servern liegt, muss für den LoadMaster gültig sein und bei Exchange mit dem Frontend identisch. Härte das Frontend mit BestPractices, TLS 1.2 und 1.3 und vorsichtig dosiertem HSTS, aber erst, nachdem du weißt, welcher Scanner im dritten Stock noch TLS 1.0 spricht. Und weil die Laufzeiten öffentlicher Zertifikate weiter schrumpfen, ist ein sauberer, überwachter und möglichst automatisierter Erneuerungsprozess keine Kür mehr. Wer das im eigenen Team durchspielen will, findet den Rahmen dafür in der [Kemp-LoadMaster-Schulung](#).

WEITERLESEN · Weiter im Thema

- › [Virtual Service, Real Server, SubVS: das Grundvokabular des LoadMasters](#)
- › [Exchange Server hinter dem LoadMaster: Templates, Namespaces, Health Checks](#)
- › [ADFS und Web Application Proxy mit dem LoadMaster: Farm, Probe, SNI](#)
- › [Web Application Firewall und Geoblocking am LoadMaster: Schutz ohne Fehlalarm-Hölle](#)
- › [Content Rules: Header umschreiben, Pfade umleiten, Dinge verstecken](#)
- › [Client-IP sichtbar machen: Transparenz, SNAT und X-Forwarded-For](#)
- › [SPN-Probleme hinter dem Load Balancer: ein Name, mehrere Server](#)
- › [Automatisierung per REST-API und PowerShell](#)