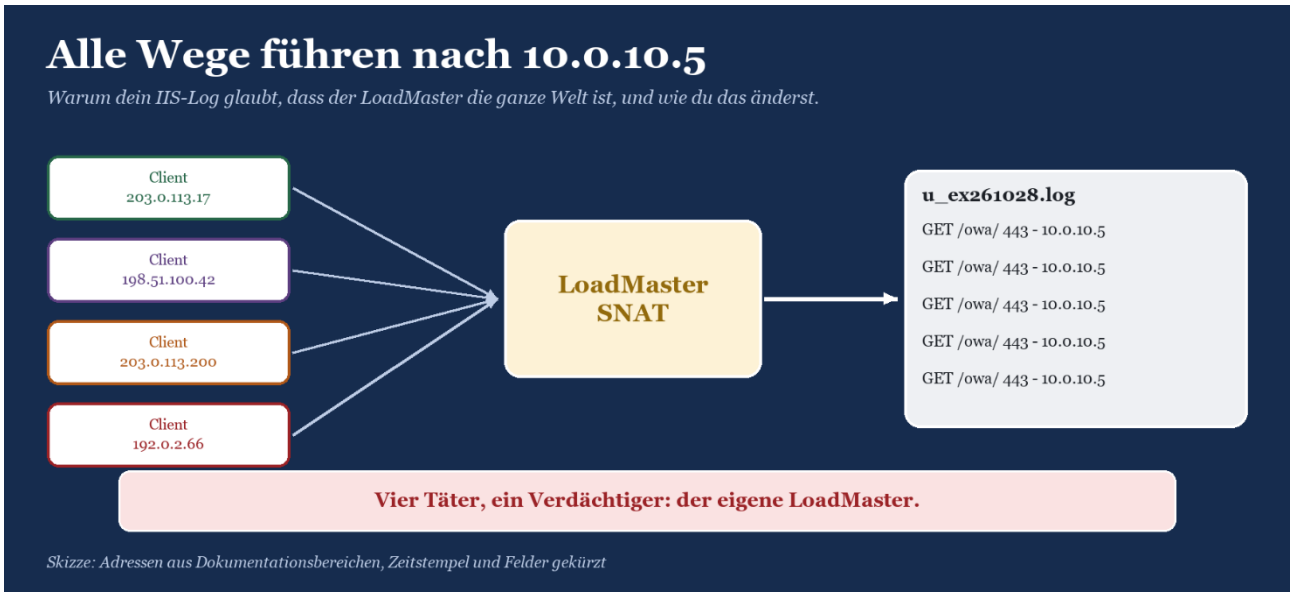
Client-IP sichtbar machen: Transparenz, SNAT und X-Forwarded-For



WISSEN

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

› **Kemp LoadMaster**

BERATUNG

Dein IIS-Log kennt nur noch einen Täter? Wir klären Transparenz, Routing und Header-Logging, bevor die nächste Sperrwelle kommt.

› **Beratung zum Kemp LoadMaster**

SCHULUNG

SNAT, Transparenz und X-Forwarded-For im Labor durchspielen, statt es beim Sicherheitsvorfall live zu lernen.

› **Kemp-LoadMaster-Schulung**

Es gibt einen Moment im Leben jeder Microsoft-Administratorin und jedes Microsoft-Administrators, der sich anfühlt wie die Szene im Krimi, in der der Kommissar merkt, dass der Mörder die ganze Zeit mit am Tisch saß. Montagmorgen, zwanzig gesperrte Konten, die Geschäftsführung will wissen, wer das war. Du öffnest das IIS-Log des Exchange-Servers, filterst auf die fehlgeschlagenen Anmeldungen und findest exakt eine Quelladresse. Tausende Male. Es ist die Adresse deines LoadMasters.

Der LoadMaster ist natürlich unschuldig. Er tut nur, was ein Reverse Proxy nun mal tut: Er nimmt die Verbindung des Clients an und baut eine eigene zum Server auf. Was dabei auf der Strecke bleibt, ist die Information, wer eigentlich angeklopft hat. Für Lastverteilung ist das egal. Für Forensik, Kontosperren, IP-basierte Zugriffsregeln, SMTP-Relays und ADFS Extranet Smart Lockout ist es fatal.

Dieser Beitrag aus der Serie rund um den Kemp LoadMaster erklärt, warum die Logs nach dem LoadMaster nur noch eine IP zeigen, welche zwei Wege es gibt, die echte Client-IP zu retten, und welche Rechnung dir jeder dieser Wege präsentiert. Transparenz kostet dich Routing, X-Forwarded-For kostet dich TLS-Offload und ein sauberes IIS-Logging. Dazu kommen die Eigenheiten von Exchange und ADFS, die in keiner Herstellerdoku zusammen auf einer Seite stehen, und die Frage, wie du dem Header überhaupt trauen kannst.

FAKTEN · Das Wichtigste in 30 Sekunden

SNAT: Der Server sieht die Adresse des LoadMasters. Routing ist unkompliziert, Logs sind blind.

Transparenz: Der Server sieht die echte Client-IP, muss aber über den LoadMaster antworten. Layer-4-Dienste sind beim LoadMaster grundsätzlich transparent, bei Layer 7 schaltest du es je Virtual Service ein.

X-Forwarded-For: Bei Layer 7 mit TLS-Offload oder Re-Encryption schreibt der LoadMaster die Client-IP in einen HTTP-Header. IIS protokolliert ihn erst, wenn du ein Custom Field anlegst.

Das IIS-Standardfeld c-ip bleibt immer die TCP-Quelle. Kein Header der Welt ändert das.

Vor WAP und ADFS, wo TLS durchgereicht wird, bleibt nur Transparenz.

Warum nach dem LoadMaster alle Clients gleich aussehen

Ein Virtual Service auf Layer 7 ist ein vollwertiger Proxy. Der Client spricht TCP und meist TLS mit der virtuellen Adresse des LoadMasters. Der LoadMaster nimmt die Anfrage entgegen, entscheidet anhand von Scheduling, Persistence und Content Rules, welcher Real Server sie bekommt, und öffnet dann eine zweite, völlig eigenständige TCP-Verbindung zu diesem Server. Für den Server ist der LoadMaster der Client. Wer die Begriffe Virtual Service, Real Server und SubVS noch einmal sortieren möchte, findet sie in Virtual Service, Real Server, SubVS: das Grundvokabular des LoadMasters.

Welche Quelladresse der LoadMaster für diese zweite Verbindung verwendet, entscheidet sich an zwei Stellen. Ist Transparenz aus, nimmt er eine eigene Adresse. Welche genau, hängt an der Option Subnet Originating Requests: Ist sie aktiv, verwendet der LoadMaster seine Interface-Adresse im Subnetz des Real Servers, sonst die Adresse des Virtual Service. Ist Transparenz eingeschaltet, reicht er die Quelladresse des Clients durch. Laut Progress hat Transparenz dabei Vorrang: Ist sie am Virtual Service aktiv, gilt sie auch dann, wenn Subnet Originating Requests global eingeschaltet ist.

SNAT: bequem, routingneutral und blind

Im SNAT-Betrieb, also mit Source NAT auf eine Adresse des LoadMasters, ist das Leben einfach. Die Server antworten an den LoadMaster, weil der ja der Absender war. Dafür muss niemand Gateways verbiegen, Clients im selben Subnetz wie die Server funktionieren, und der LoadMaster darf im Prinzip irgendwo stehen, solange er die Server erreicht. Deshalb ist SNAT bei Layer-7-Diensten im Alltag der Normalfall und in vielen Vorlagen voreingestellt.

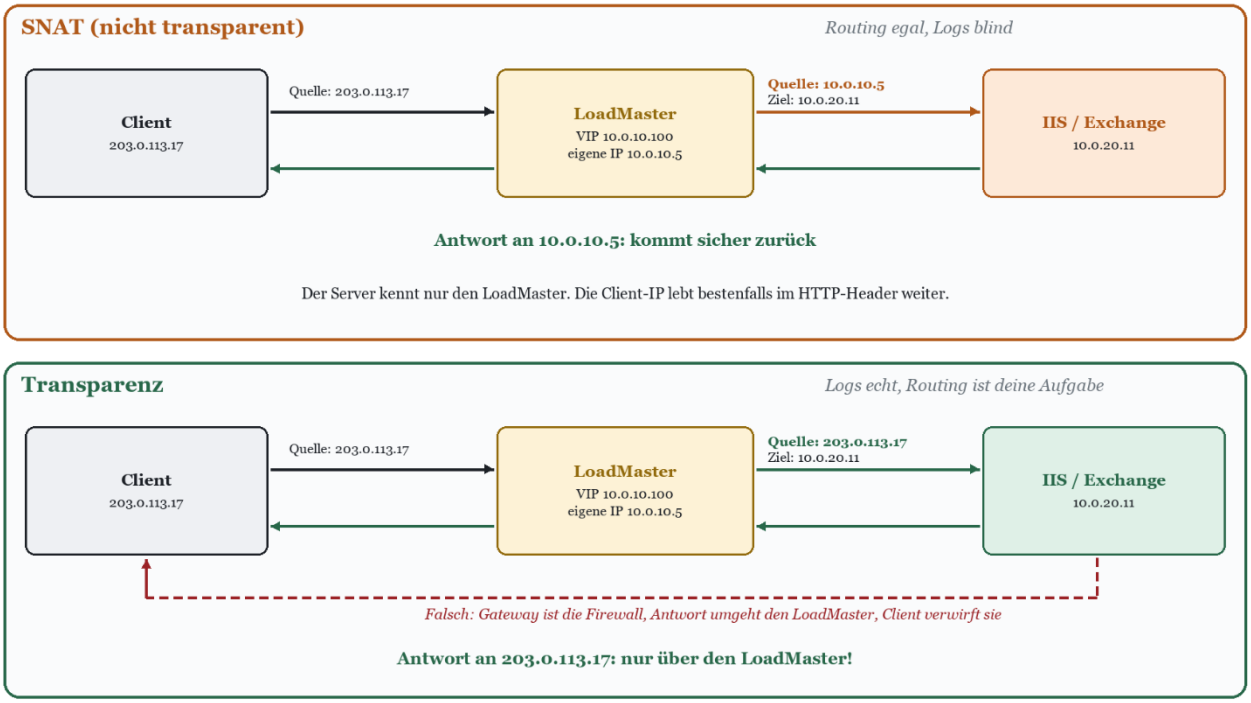
Der Preis: Alles, was der Server über den Client wissen will, muss jetzt anders transportiert werden. Das IIS-Log, die Windows-Sicherheitsereignisse, die Exchange-Protokolle, der Empfangsconnector, jede IP-Einschränkung im IIS: Alle sehen nur noch den LoadMaster. Und weil der LoadMaster in einem internen Netz steht, sieht er für viele Regeln aus wie ein besonders vertrauenswürdiger Kollege.

Transparenz: die echte Adresse, aber mit Routing-Pflicht

Im transparenten Betrieb baut der LoadMaster die Verbindung zum Server mit der Quelladresse des Clients auf. Der Server sieht 203.0.113.17 und antwortet folgerichtig an 203.0.113.17. Genau da beginnt das Problem: Diese Antwort muss wieder durch den LoadMaster, denn nur der kennt die Verbindung und weiß, dass er die Antwort mit der Adresse der VIP zurückschicken muss. Nimmt die Antwort den kurzen Weg über das Standard-Gateway, also meistens die Firewall, kommt beim Client ein Paket von einer Adresse an, mit der er nie gesprochen hat. Er verwirft es. Die Verbindung hängt, und zwar so, dass jede Fehlersuche zunächst beim Server, dann bei der Firewall und erst am Ende beim Routing landet.

SNAT gegen Transparenz: wer steht im Absenderfeld, und wer muss antworten?

Oben SNAT: zwei getrennte TCP-Verbindungen. Unten Transparenz: eine Adresse, aber Routing-Pflicht



Skizze: vereinfachtes Modell, Adressen fiktiv; Layer 4 ist am LoadMaster grundsätzlich transparent

Daraus folgen die Bedingungen, die Progress in der Funktionsbeschreibung zur Transparenz nüchtern auflistet und die in der Praxis jede für sich einen Supportfall produzieren: Das Standard-Gateway der Real Server ist der LoadMaster, oder der Rückweg wird anders erzwungen. Clients dürfen nicht im selben Subnetz wie die Real Server sitzen, weil deren Antworten sonst direkt zugestellt werden. Real Server in fremden Netzen gehen transparent nicht. Und Transparenz verträgt sich laut Progress-Dokumentation nicht mit TLS-Re-Encryption, was vor Exchange noch eine große Rolle spielen wird.

Kriterium	SNAT	Transparenz
Quell-IP am Server	Adresse des LoadMasters	echte Client-IP
Rückweg	automatisch zum LoadMaster	muss über den LoadMaster erzwungen werden
Standard-Gateway der Server	beliebig	LoadMaster oder passende Routen
Clients im Server-Subnetz	funktionieren	funktionieren nicht ohne Zusatzaufwand
Real Server in anderen Netzen	möglich	nicht möglich
Layer 4	nicht vorgesehen	Standardverhalten
Layer 7 mit Re-Encryption	möglich	laut Progress nicht kombinierbar
Client-IP für HTTP-Anwendungen	über X-Forwarded-For	direkt in der TCP-Verbindung
Client-IP für SMTP, RDP, ADFS über WAP	nicht verfügbar	verfügbar

WARNUNG · Der Server, der plötzlich niemanden mehr erreicht

Wer für Transparenz das Standard-Gateway eines Exchange- oder IIS-Servers auf den LoadMaster umbiegt, schickt damit auch dessen eigenen ausgehenden Verkehr über den LoadMaster: Windows Update, Zertifikatssperrlisten, Smarthost, Microsoft-365-Hybrid. Wenn der LoadMaster dafür keine Weiterleitung kennt, ist der Server nach dem Neustart des Dienstes zwar transparent, aber auch ziemlich einsam. Plane die ausgehenden Wege vorher, nicht nach dem Ticket.

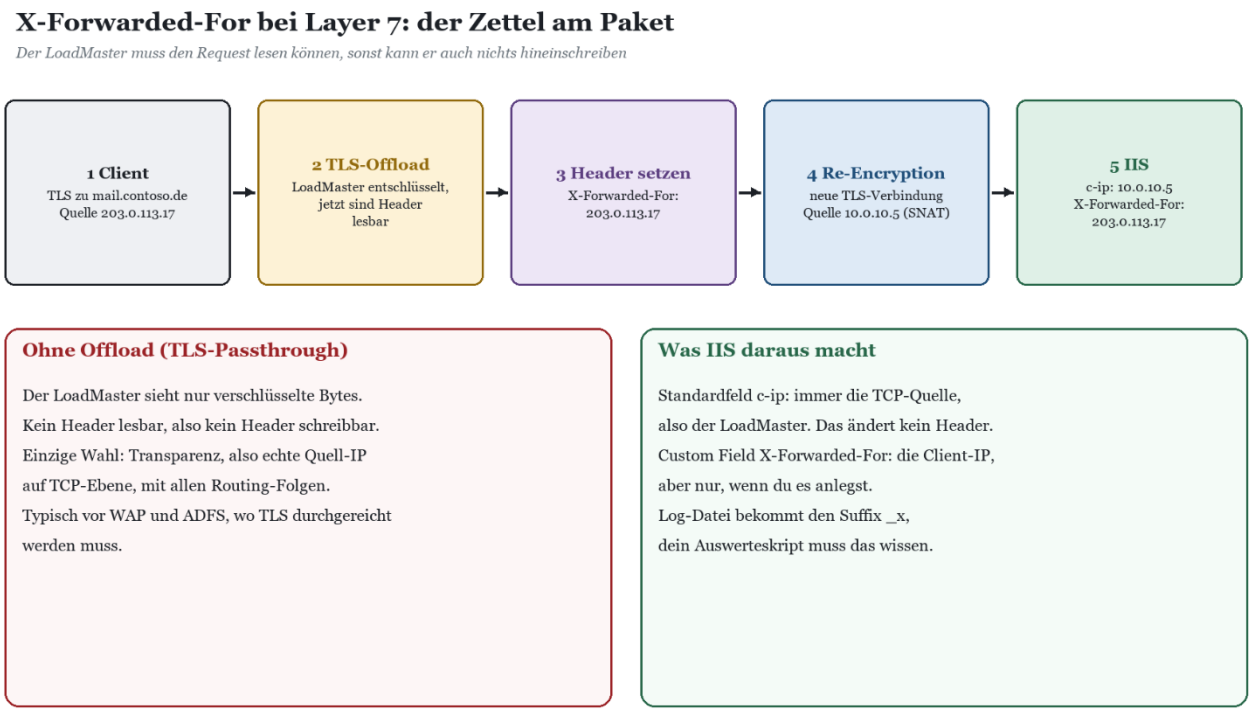
Wo Transparenz im Microsoft-Umfeld wirklich knallt

- **Admins im Server-Netz:** Die Administratorin testet OWA vom Jump-Host im selben Subnetz, der Server antwortet direkt, der Browser dreht sich. Draußen funktioniert alles. Das ist kein Fehler, das ist Physik.
- **Health Checks:** Die Prüfungen des LoadMasters laufen mit dessen eigener Adresse und funktionieren deshalb auch dann noch, wenn der transparente Kundenverkehr längst hängt. Grün im Dashboard, rot beim Benutzer.
- **Asymmetrie über die Firewall:** Stateful Firewalls verwerfen Antworten, deren Hinweg sie nie gesehen haben. Das sieht im Firewall-Log aus wie ein Angriff und ist doch nur dein eigener Rückweg.
- **HA-Paare:** Das Gateway der Server muss die gemeinsame, schwebende Adresse des Paares sein, nicht die Adresse eines Knotens. Sonst endet die Transparenz mit dem ersten Failover.
- **Mehrere Virtual Services, ein Server:** Ist ein Dienst transparent und ein anderer auf demselben Server nicht, funktionieren beide, solange das Routing sauber ist. Ist es das nicht, fällt nur einer aus, und zwar garantiert der, den du gerade nicht prüfst.

X-Forwarded-For bei Layer 7: der Zettel am Paket

Wenn Transparenz zu viel Routing-Ärger bedeutet, gibt es bei HTTP einen eleganteren Weg: Der LoadMaster schreibt die Client-IP, die er gesehen hat, in einen HTTP-Header und lässt die TCP-Verbindung zum Server per SNAT laufen. Der De-facto-Standard dafür heißt X-Forwarded-For. Er stammt aus der Welt der Caching-Proxys und ist nie offiziell standardisiert worden; der modernere Header Forwarded aus RFC 7239 hat sich im Microsoft-Umfeld bis heute kaum durchgesetzt. Praktisch alle Microsoft-Produkte, die überhaupt einen Proxy-Header auswerten, schauen auf X-Forwarded-For.

Die Voraussetzung ist so trivial wie gnadenlos: Der LoadMaster muss den HTTP-Request lesen und verändern können. Bei HTTPS heißt das TLS-Offload oder Re-Encryption. Wird TLS nur durchgereicht, sieht der LoadMaster lediglich verschlüsselte Bytes. Header schreiben ist dann so unmöglich wie das Ergänzen einer Fußnote in einem zugeklebten Briefumschlag. Wie Offload, Re-Encryption und die Zertifikate zusammenspielen, steht in TLS-Offload, Re-Encryption und Zertifikate: PKI trifft LoadMaster.



Skizze: Ablauf vereinfacht; Re-Encryption und Offload im Detail im Beitrag zu TLS

Den Header am LoadMaster einschalten

Am Virtual Service legst du in den erweiterten Eigenschaften über die Option Add HTTP Headers fest, ob und welchen Header der LoadMaster einfügt. Für Microsoft-Backends ist die Variante X-Forwarded-For ohne Via-Header die übliche Wahl. Der zusätzliche Via-Header verrät nur, dass ein Proxy im Spiel war, und das weißt du ja schon. Die Einstellung gilt je Virtual Service. Hängen unter einem Haupt-Service mehrere SubVS, prüfst du, ob die Einstellung dort ankommt, wo sie gebraucht wird, statt es zu vermuten.

Zwei Dinge solltest du bei der Gelegenheit klären. Erstens: Ergänzt dein LoadMaster einen vom Client mitgeschickten X-Forwarded-For-Header oder ersetzt er ihn? Das hängt von der Firmware-Version und der gewählten Variante ab, und es lässt sich in fünf Minuten testen: Schicke mit einem Testclient einen eigenen X-Forwarded-For-Header mit und sieh im IIS-Log nach, was ankommt. Zweitens: Wenn du über Content Rules weitere Header manipulierst, halte die Reihenfolge im Blick. Wie Header-Regeln im Detail funktionieren, erklärt Content Rules: Header umschreiben, Pfade umleiten, Dinge verstecken.

> **TIPP · Schneller Funktionstest ohne Spezialwerkzeug**
>
> Lege auf einem Real Server eine kleine Testseite an, die die Server-Variable REMOTE_ADDR und den Header X-Forwarded-For ausgibt, und rufe sie über die VIP auf. Dann siehst du in einem Bild, was die TCP-Verbindung meldet und was der Header behauptet. Danach die Seite wieder löschen. Wer nach einem Jahr eine vergessene Diagnoseseite im Pentest-Bericht findet, lacht selten.

IIS: X-Forwarded-For protokollieren mit Custom Fields

Der IIS weiß nichts davon, dass er hinter einem LoadMaster steht. Das Standardfeld c-ip im W3C-Log enthält immer die Quelladresse der TCP-Verbindung, hinter SNAT also den LoadMaster. Den Header protokollierst du zusätzlich als eigenes Feld. Seit IIS 8.5, also seit Windows Server 2012 R2, ist das mit Bordmitteln möglich und heißt bei Microsoft Enhanced Logging: Im IIS-Manager

wählst du an der Site die Protokollierung, dort die Felder und fügst ein benutzerdefiniertes Feld hinzu. Als Quelltyp nimmst du Request Header, als Quelle X-Forwarded-For.

Das funktioniert nur auf Site-Ebene und nur im W3C-Format. IIS legt danach neue Logdateien mit dem Suffix _x im Namen an, um zu kennzeichnen, dass sie benutzerdefinierte Felder enthalten. Den Feldnamen c-ip kannst du nicht verwenden, er kollidiert mit dem Standardfeld; nenne das Feld also schlicht X-Forwarded-For. Wer lieber skriptet, legt das Feld per PowerShell an:

```
Import-Module WebAdministration
Add-WebConfigurationProperty -PSPath 'MACHINE/WEBROOT/APPHOST' `
  -Filter "system.applicationHost/sites/site[@name='Default Web
Site']/logFile/customFields" `
  -Name '.' -Value @{logFieldName='X-Forwarded-For';sourceName='X-Forwarded-
For';sourceType='RequestHeader'}
```

Und dann kommt der Teil, den niemand macht: das Auswerteskript, den Log-Collector und das SIEM-Parsing anpassen. Neue Spalte, neuer Dateiname, und wer bisher stur c-ip ausgewertet hat, wertet weiterhin den LoadMaster aus, nur jetzt mit einer schönen zusätzlichen Spalte, die niemand liest.

WICHTIG · Advanced Logging ist Geschichte

Viele Anleitungen im Netz empfehlen noch das IIS Advanced Logging. Das war eine separat installierte Erweiterung aus der Zeit von IIS 7 und 7.5. Auf aktuellen Servern brauchst du sie nicht und solltest sie auch nicht mehr suchen: Die Custom Fields von Enhanced Logging erledigen dasselbe mit Bordmitteln, sind in der applicationHost.config dokumentiert und lassen sich per PowerShell oder AppCmd verteilen. Wenn eine Anleitung Advanced Logging erwähnt, ist sie mindestens zehn Jahre alt. Für den Rest ihres Inhalts gilt das vermutlich auch.

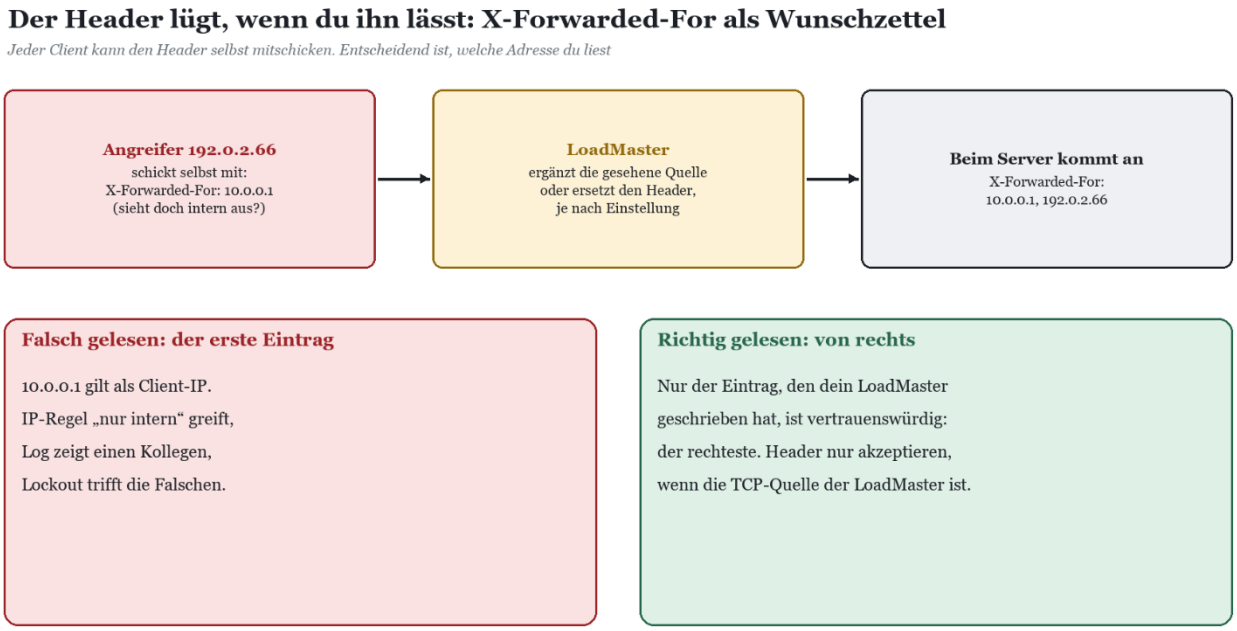
Feld oder Quelle	Inhalt hinter SNAT	Inhalt bei Transparenz	Bemerkung
c-ip (Standardfeld)	LoadMaster	Client	nicht durch Header änderbar
Custom Field X-Forwarded-For	Client, ggf. Liste	leer oder nur vom Client gesetzt	erst nach Anlegen vorhanden
REMOTE_ADDR in der Anwendung	LoadMaster	Client	das, was Code ohne Anpassung nutzt
Windows-Ereignis 4625	je nach Anmeldeweg Server oder LoadMaster	eher Client	für Forensik mit IIS-Zeitstempel korrelieren
LoadMaster-Logs und Syslog	Client	Client	unabhängig von Server-Konfiguration

Eigenentwicklungen: REMOTE_ADDR ist nicht die Wahrheit

Was fürs Log gilt, gilt erst recht für Code. Eine Anwendung, die REMOTE_ADDR liest, um Benutzer zu protokollieren, IP-Sperren zu verhängen oder Rate-Limits zu setzen, sieht hinter SNAT nur den LoadMaster. Im schlimmsten Fall sperrt sie bei zu vielen Fehlversuchen die Adresse des LoadMasters, und damit alle. Das ist der Moment, in dem ein Rate-Limit plötzlich sehr wirkungsvoll wird. Unter ASP.NET Core gibt es dafür die Forwarded-Headers-Middleware, der du ausdrücklich sagst, welchen Proxys du vertraust. Andere Frameworks brauchen eigenen Code. Wie du solche Anwendungen insgesamt sauber hinter den LoadMaster bringst, behandelt IIS-Eigenentwicklungen und http.sys-Dienste veröffentlichen.

Dem Header vertrauen: nur von rechts und nur vom LoadMaster

X-Forwarded-For ist ein Wunschzettel. Jeder Client kann ihn selbst mitschicken, mit jeder Adresse, die ihm gerade gefällt. Ergänzt der LoadMaster den Header, statt ihn zu ersetzen, steht am Ende eine Liste, und nur der rechteste Eintrag stammt aus einer Quelle, der du vertraust. Wer stattdessen den ersten Eintrag liest, lässt den Angreifer selbst entscheiden, als wer er im Log stehen möchte.



Skizze: ob der LoadMaster anhängt oder überschreibt, prüfst du mit einem eigenen Test-Header

- **Nur vom LoadMaster:** Werte den Header nur aus, wenn die TCP-Quelle eine Adresse des LoadMasters ist. Server, die auch direkt erreichbar sind, bekommen sonst gefälschte Header frei Haus.
- **Von rechts lesen:** Bei einer Liste zählt der Eintrag, den dein letzter vertrauenswürdiger Proxy angehängt hat.
- **Keine Sicherheitsentscheidung allein auf Basis des Headers:** Zugriffsregeln wie „nur intern“ gehören an den LoadMaster, der die echte Quelle kennt, nicht an einen Header.
- **Mehrstufige Ketten dokumentieren:** Steht vor dem LoadMaster noch ein CDN, eine Cloud-WAF oder ein weiterer Proxy, verschiebt sich der vertrauenswürdige Eintrag um eine Stelle. Wer das nicht aufschreibt, liest in einem Jahr die falsche Spalte.

Exchange und ADFS: zwei Sonderfälle mit Ansage

Exchange und ADFS sind die beiden Dienste, bei denen die Client-IP in der Praxis am häufigsten fehlt und am schmerzlichsten vermisst wird. Beide haben Eigenheiten, die den naheliegenden Weg versperren.

Exchange: Extended Protection schließt den bequemen Weg

Seit Microsoft Extended Protection für Exchange Server zum Standard gemacht hat, ist TLS-Offload vor Exchange nicht mehr unterstützt. Die Verbindung muss am Exchange-Server mit dessen Zertifikat enden; erlaubt ist Re-Encryption, auch SSL Bridging genannt, mit demselben Zertifikat auf LoadMaster und Server. Damit fällt bei HTTPS-Diensten wie OWA, ECP, EWS, MAPI und ActiveSync die Transparenz faktisch weg, denn sie verträgt sich laut Progress nicht mit Re-Encryption. Übrig bleibt X-Forwarded-For auf einem Layer-7-Service mit Re-Encryption. Für Exchange ist das ohnehin die empfohlene Architektur; die Details zu Templates, Namespaces und

Health Checks findest du in Exchange Server hinter dem LoadMaster: Templates, Namespaces, Health Checks.

Exchange selbst protokolliert auf mehreren Ebenen. Das Default Web Site des Frontends sieht die Verbindung vom LoadMaster und bekommt dort dein Custom Field. Das Exchange Back End sieht dagegen nur noch den eigenen oder einen anderen Exchange-Server, weil das Frontend weiter proxyt. Das Custom Field gehört deshalb auf das Default Web Site aller Exchange-Server, nicht auf das Back End. Zusätzlich lohnt ein Blick in die HttpProxy-Protokolle unter dem Logging-Verzeichnis der Exchange-Installation: Prüfe dort an einem Testaufruf, welche Adresse in deinen Builds tatsächlich landet, bevor du dich in einem Vorfall darauf verlässt.

WARNUNG · SMTP kennt keine Header-Notlösung

SMTP ist kein HTTP. Es gibt keinen Header, in den der LoadMaster die Client-IP schreiben könnte, den Exchange vor der Annahme auswertet. Empfangsconnectors entscheiden über RemoteIPRanges anhand der TCP-Quelle. Hinter SNAT ist jeder Absender der LoadMaster, und ein Relay-Connector für das LoadMaster-Netz wird zum offenen Relay für alle. Wie du das mit Transparenz oder getrennten VIPs löst, steht in SMTP über den LoadMaster: Relay, Empfangsconnectors und die echte Absender-IP.

ADFS: der WAP kann nur weitergeben, was er sieht

Bei ADFS liegt die Sache anders. Der Web Application Proxy schreibt die Adresse, die er als Quelle der Verbindung sieht, in den Header x-ms-forwarded-client-ip und markiert die Anfrage mit x-ms-proxy als von außen kommend. ADFS verwendet diese Werte für Extranet Smart Lockout, für Claim-Regeln und im Sicherheitsüberwachungsprotokoll. Steht der LoadMaster mit SNAT vor dem WAP, sieht der WAP für alle Clients die Adresse des LoadMasters und meldet sie gewissenhaft weiter. Smart Lockout lernt dann, dass sämtliche Benutzer von einem einzigen Ort kommen, und der ist zufällig auch der Ort, von dem jeder Angreifer kommt.

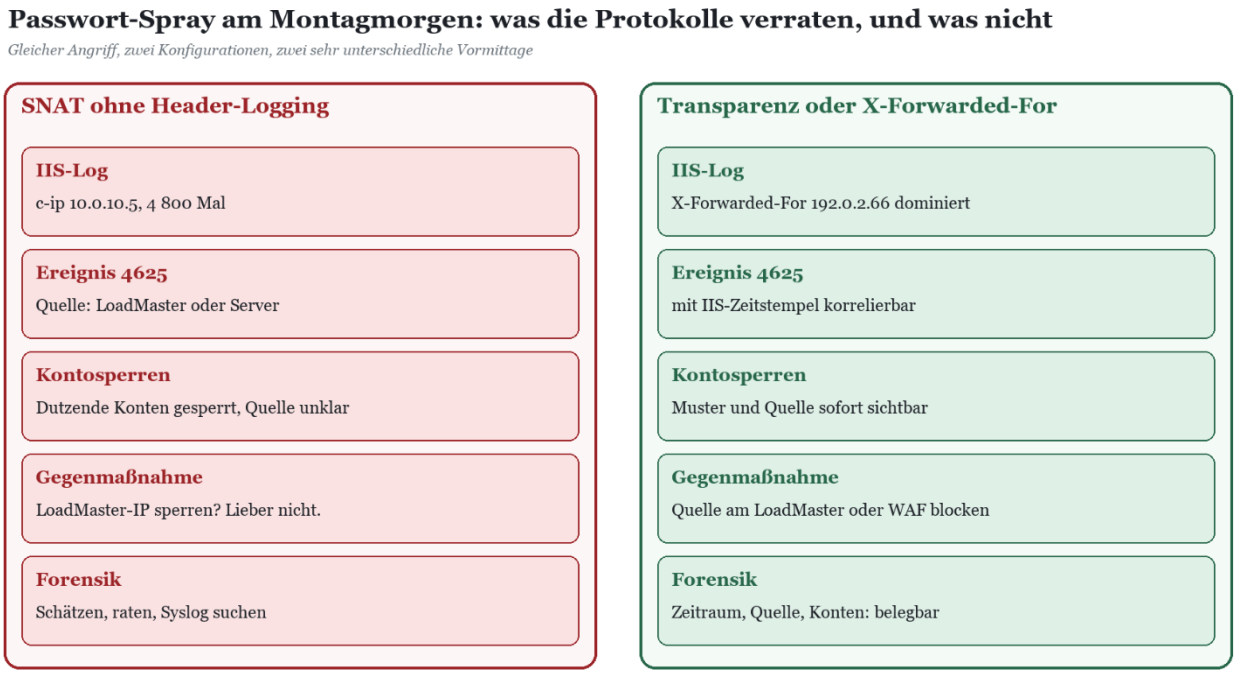
Weil die Verbindung zum WAP verschlüsselt durchgereicht wird, kann der LoadMaster dort keinen X-Forwarded-For-Header setzen. Der Weg ist Transparenz auf dem externen Virtual Service vor dem WAP, mit dem LoadMaster als Rückweg der WAP-Server. Zwischen WAP und ADFS-Farm ist SNAT dagegen unkritisch, weil ADFS die Client-IP aus dem Header des WAP bezieht. Die vollständige Architektur mit Probe-Endpunkt und SNI steht in ADFS und Web Application Proxy mit dem LoadMaster: Farm, Probe, SNI.

Dienst	Empfohlener Weg	Wo die Client-IP landet	Typische Falle
Exchange OWA, EWS, MAPI, EAS	Layer 7, Re-Encryption, X-Forwarded-For	Custom Field im Default Web Site	Offload bricht Extended Protection
Exchange SMTP	Layer 4 transparent oder getrennte VIPs	Empfangsconnector, Protokoll	Relay-Connector für das LoadMaster-Netz
ADFS über WAP	Layer 4 transparent vor dem WAP	x-ms-forwarded-client-ip, Ereignisse	SNAT macht Smart Lockout blind
SharePoint	Layer 7 mit Offload oder Re-Encryption, X-Forwarded-For	Custom Field im IIS, ULS nur mit Aufwand	Logauswertung liest weiter c-ip
RD Gateway	Layer 7 für HTTPS, Layer 4 für UDP	RD-Gateway-Protokoll, abhängig vom Weg	UDP-Transport und HTTPS-Verkehr mit unterschiedlicher Sicht
IIS-Eigenentwicklung	Layer 7, X-Forwarded-For, Middleware	Anwendungslog	Rate-Limit sperrt den LoadMaster

Für die Remote Desktop Services lohnt sich ein eigener Blick, weil dort HTTPS und UDP nebeneinanderlaufen und jede Transportart die Client-IP anders zu Gesicht bekommt. Mehr dazu in Remote Desktop Services: RD Gateway, RD Web und Connection Broker balancieren.

Forensik und Lockout: warum die IP mehr ist als Kosmetik

Man kann jahrelang mit blinden Logs leben. Die Lastverteilung funktioniert, die Anwender sind zufrieden, und das Thema steht auf der Liste der Dinge, die man irgendwann macht. Irgendwann ist dann meistens ein Montagmorgen mit einem Passwort-Spray, einer Datenschutzanfrage oder einem Pentest-Bericht.



Skizze: konstruiertes Beispiel; Ereignis 4625 nennt je nach Anmeldeweg unterschiedliche Quellen

Kontosperren: die Suche nach der Quelle

Die klassische Sperrsuche führt über Ereignis 4740 auf dem PDC-Emulator zum Caller Computer Name und von dort zu den Ereignissen 4625 auf dem Server, der die Anmeldung entgegengenommen hat. Bei OWA, ActiveSync und EWS ist das der Exchange-Server. Welche Netzwerkadresse dort steht, hängt vom Anmeldeweg ab, und hinter SNAT ist es eben nicht die des Angreifers. Der eigentliche Beleg steckt im IIS-Log: Zeitstempel, Benutzername, Statuscode 401 und, wenn du vorgesorgt hast, die Client-IP im X-Forwarded-For-Feld. Ohne dieses Feld bleibt die Spur am LoadMaster hängen.

Die zweite Spur führt über den LoadMaster selbst. Er kennt die echte Quelladresse in jedem Fall und kann sie per Syslog an ein zentrales System liefern. Für die Korrelation brauchst du dann aber synchrone Uhren und ein SIEM, das beide Quellen zusammenführt. Wie du den LoadMaster dafür sauber anbindest, beschreibt Monitoring, Syslog und SNMP: den LoadMaster ins Monitoring holen.

Lockout-Mechanismen, die an der IP hängen

Mechanismus	Was er braucht	Folge hinter SNAT
ADFS Extranet Smart Lockout	echte Client-IP über den WAP	vertraute und unbekannte Orte verschmelzen zu einem
IIS-Einschränkungen nach	TCP-Quelle oder Proxy-Modus	alles erlaubt oder alles gesperrt

IP und Domäne		
Exchange-Empfangsconnectors	TCP-Quelle	Relay für alle oder für niemanden
Rate-Limits in Eigenentwicklungen	Client-IP aus REMOTE_ADDR oder Header	erster Angreifer sperrt alle Benutzer
Fail2ban-artige Skripte auf Windows	Client-IP aus Ereignissen	sperren den LoadMaster
WAF und Geoblocking am LoadMaster	Quelle am LoadMaster	funktioniert, weil der LoadMaster sie sieht

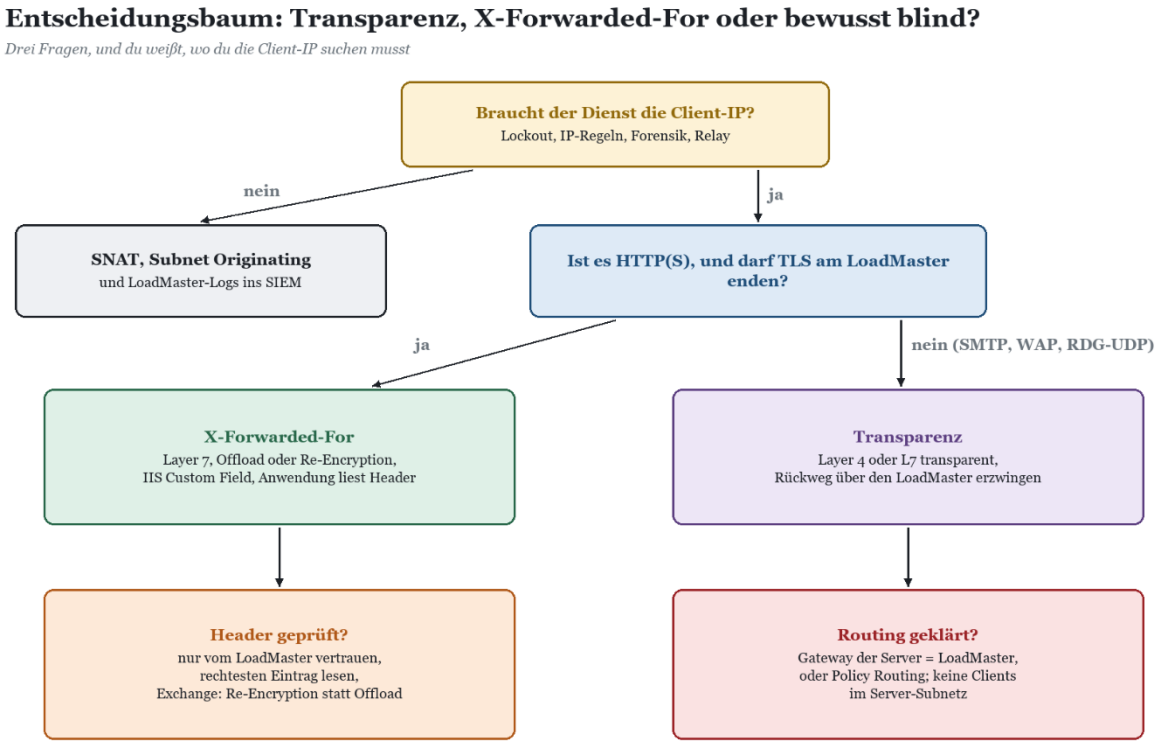
AUS DER PRAXIS · Die Firewall-Regel, die das ganze Haus aussperrte

In einer Verwaltung mit rund 800 Postfächern hatte ein findiger Kollege ein Skript gebaut, das fehlgeschlagene OWA-Anmeldungen aus dem Ereignisprotokoll las und auffällige Quelladressen per Windows-Firewall sperrte. Funktionierte prima, bis der LoadMaster eingeführt wurde. Beim ersten Passwort-Spray sperrte das Skript sehr zuverlässig genau eine Adresse: die des LoadMasters. Der Angreifer war damit tatsächlich ausgesperrt. Alle anderen auch. Die Lösung: das Skript abgeschaltet, X-Forwarded-For im IIS protokolliert und die Abwehr dorthin verlagert, wo die echte Quelle bekannt ist, an die WAF des LoadMasters.

Datenschutz: auch die Client-IP ist ein personenbezogenes Datum

Wer die echte Client-IP wieder in die Logs holt, protokolliert ein personenbezogenes Datum. Das ist kein Grund, darauf zu verzichten, denn Sicherheitsprotokollierung ist ein anerkannter Zweck. Es ist aber ein Grund, Aufbewahrungsfristen festzulegen und mit dem Datenschutz abzustimmen, gerade in Verwaltungen. Sinnvoll ist, das gleich beim Umbau zu klären, statt beim ersten Auskunftsersuchen festzustellen, dass die IIS-Logs seit 2019 unverändert auf dem Laufwerk D liegen.

Entscheidungshilfe und Checkliste



Skizze: Faustregel für Microsoft-Workloads, kein Ersatz für einen Blick auf dein Netz

- **Inventur:** Für jeden Virtual Service festhalten, ob Layer 4 oder 7, SNAT oder transparent, ob Add HTTP Headers aktiv ist und wer die Client-IP braucht.
- **IIS:** Custom Field X-Forwarded-For auf allen betroffenen Sites aller Server anlegen, per Skript, nicht per Hand auf drei von vier Servern.
- **Auswertung:** Log-Collector und SIEM auf die _x-Dateien und das neue Feld umstellen.
- **Vertrauen:** Header nur von LoadMaster-Adressen akzeptieren, rechtesten Eintrag lesen, Testfall mit gefälschtem Header durchspielen.
- **Transparenz:** Rückweg dokumentieren, Gateway auf die schwebende Adresse des HA-Paares, ausgehende Wege der Server klären, Test aus dem Server-Subnetz bewusst ausschließen.
- **Syslog:** Logs des LoadMasters zentral sammeln, als zweite unabhängige Quelle.
- **Fehlersuche:** Bei hängenden transparenten Verbindungen zuerst den Rückweg per Paketmitschnitt prüfen; das Vorgehen beschreibt Troubleshooting am LoadMaster: Logs, L7-Debug und TCP-Dump.

FAQ: häufige Fragen zu X-Forwarded-For, IIS und Transparenz

Warum zeigt mein IIS-Log trotz X-Forwarded-For weiterhin die LoadMaster-IP?

Weil c-ip immer die TCP-Quelle ist. Der Header steht nur dann im Log, wenn du ihn als Custom Field anlegst, und zwar in einer neuen Spalte. Prüfe außerdem, ob du die neuen Logdateien mit dem Suffix _x auswertest und nicht die alten.

Kann ich X-Forwarded-For ohne TLS-Offload verwenden?

Nein. Ohne Entschlüsselung kann der LoadMaster den HTTP-Request weder lesen noch verändern. Du brauchst Offload oder Re-Encryption. Wo TLS durchgereicht werden muss, bleibt nur Transparenz.

Gibt es noch IIS Advanced Logging?

Als Erweiterung für IIS 7 und 7.5 gab es das, auf aktuellen Windows-Servern brauchst du es nicht. Seit IIS 8.5 erledigen die Custom Fields des Enhanced Logging dieselbe Aufgabe.

Kann ich das Feld c-ip durch den Header ersetzen?

Nicht mit Bordmitteln. Ein Custom Field mit dem Namen c-ip führt zu einem Konflikt mit dem Standardfeld. Lege ein eigenes Feld an und passe die Auswertung an.

Warum funktioniert OWA im transparenten Modus von meinem Admin-PC nicht?

Vermutlich sitzt der PC im selben Subnetz wie die Exchange-Server. Der Server antwortet dann direkt statt über den LoadMaster, und der Client verwirft die Antwort. Teste aus einem anderen Netz oder nimm für Admin-Zugriffe einen eigenen, nicht transparenten Virtual Service.

Ist Transparenz vor Exchange überhaupt noch sinnvoll?

Für SMTP oft ja, wenn Empfangsconnectors die echte Quelle brauchen. Für die HTTPS-Dienste kaum, weil Extended Protection Offload ausschließt und Transparenz sich laut Progress nicht mit Re-Encryption kombinieren lässt. Dort ist X-Forwarded-For mit Re-Encryption der Weg.

Hilft X-Forwarded-For bei ADFS Extranet Smart Lockout?

Vor dem WAP nicht, weil der LoadMaster dort TLS durchreicht und keinen Header setzen kann. Der WAP meldet die Adresse, die er auf TCP-Ebene sieht. Die Lösung ist Transparenz vor dem WAP.

Darf ich X-Forwarded-For für Zugriffsregeln im IIS verwenden?

Nur mit großer Vorsicht und nie als einzige Kontrolle. Der Header lässt sich vom Client fälschen, wenn der Server auch ohne LoadMaster erreichbar ist oder der LoadMaster den Header nur ergänzt. Zugriffsregeln nach Quelladresse setzt du besser am LoadMaster um.

Fazit: Die IP ist weg, aber nicht verloren

Dass nach dem LoadMaster nur noch eine Adresse in den Logs steht, ist kein Fehler, sondern die logische Folge eines Proxys. Du hast zwei Werkzeuge, um die Client-IP zurückzuholen, und beide haben ihren Preis. Transparenz liefert die echte Adresse auf TCP-Ebene, für SMTP und ADFS über den WAP der einzige Weg, verlangt dafür aber ein Routing, das jede Antwort zurück über den LoadMaster zwingt. X-Forwarded-For lässt das Routing in Ruhe, braucht dafür Layer 7 mit TLS-Terminierung, ein Custom Field im IIS und eine Auswertung, die den Header auch tatsächlich liest, und zwar von rechts.

Für die typische Microsoft-Landschaft heißt das: Exchange-HTTPS, SharePoint und Eigenentwicklungen mit X-Forwarded-For und Re-Encryption, SMTP und WAP transparent, der LoadMaster als zweite, unabhängige Quelle per Syslog. Richte das ein, bevor du es brauchst. Denn der Montagmorgen mit zwanzig gesperrten Konten kommt bestimmt, und es ist deutlich angenehmer, wenn der Hauptverdächtige dann nicht wieder dein eigener LoadMaster ist.

TIPP · Unterstützung bei Planung und Umbau

Wenn du Transparenz einführen willst, ohne das Routing deiner DMZ zu zerlegen, oder das IIS-Logging über eine ganze Exchange-Organisation vereinheitlichen möchtest, hilft die Beratung zum Kemp LoadMaster. Wer SNAT, Transparenz und X-Forwarded-For mit dem eigenen Team im Labor durchspielen möchte, findet das in der Kemp-LoadMaster-Schulung.

WEITERLESEN · Weiter im Thema

› SMTP über den LoadMaster: Relay, Empfangsconnectors und die echte Absender-IP
› ADFS und Web Application Proxy mit dem LoadMaster: Farm, Probe, SNI
› Monitoring, Syslog und SNMP: den LoadMaster ins Monitoring holen
› Exchange Server hinter dem LoadMaster: Templates, Namespaces, Health Checks
› TLS-Offload, Re-Encryption und Zertifikate: PKI trifft LoadMaster
› Troubleshooting am LoadMaster: Logs, L7-Debug und TCP-Dump