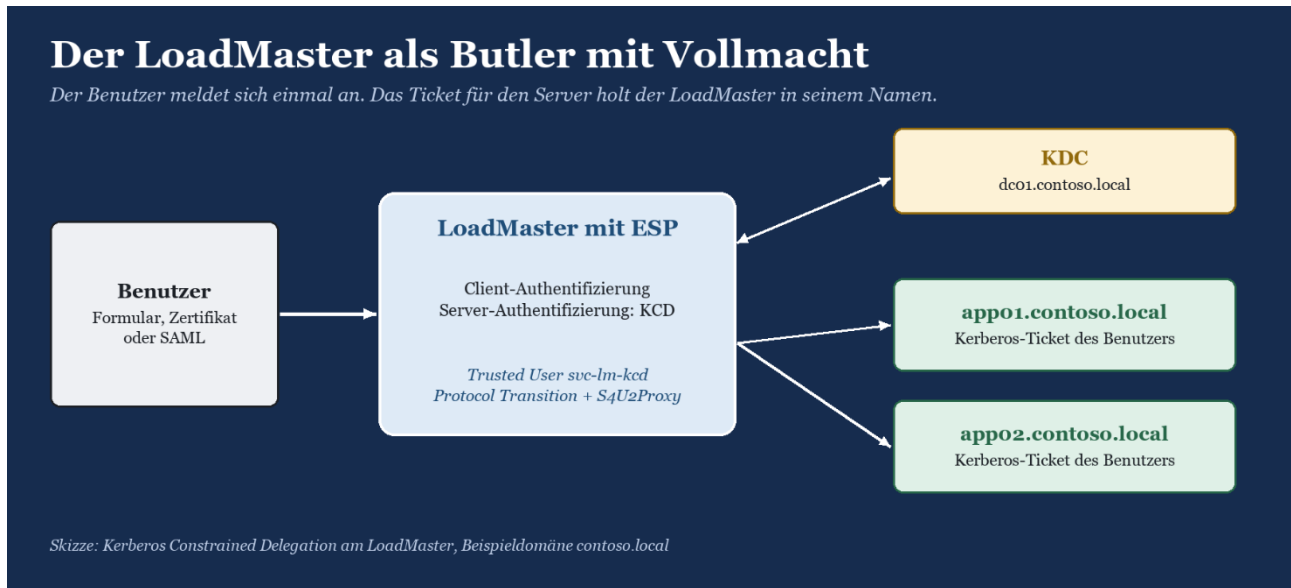


Kerberos Constrained Delegation am LoadMaster: KCD und RBCD sauber einrichten



WISSEN

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

› [Kemp LoadMaster](#)

BERATUNG

KCD scheitert mit kryptischen Meldungen? Wir gehen Konten, Delegation und SSO-Domänen gemeinsam durch.

› [Beratung zum Kemp LoadMaster](#)

SCHULUNG

Vorauthentifizierung und Kerberos-Delegation im Labor nachbauen, bis dein Team sie im Schlaf kann.

› [Kemp-LoadMaster-Schulung](#)

Kerberos Constrained Delegation hat einen Ruf wie ein alter Heizungskeller: Alle wissen, dass da unten etwas Wichtiges läuft, niemand geht freiwillig hinein, und wer es doch tut, kommt mit schwarzen Händen und einer Ereignis-ID zurück, die er nicht versteht. Am Progress Kemp LoadMaster wird es noch etwas spannender, denn hier treffen zwei Welten aufeinander, die sich gegenseitig gern ignorieren: die Kemp-Dokumentation, die erklärt, welche Felder du ausfüllen sollst, und Microsoft Learn, das erklärt, was der Domänencontroller davon hält. Dazwischen klafft eine Lücke, und genau in diese Lücke fallen Exchange-, SharePoint- und Eigenentwicklungsprojekte mit erstaunlicher Zuverlässigkeit.

Dieser Beitrag schließt die Lücke. Du erfährst, wie das Edge Security Pack (ESP) mit KCD als Server-Authentifizierung arbeitet, was ein Trusted User ist und warum er mehr darf als dein Domänen-Admin am Montagmorgen. Wir schauen uns an, warum Protocol Transition der eigentliche Trick ist, wo klassische KCD und ressourcenbasierte Delegation (RBCD) sich unterscheiden, und warum RBCD dir das Leben rettet, sobald deine Dienste unter Computerkonten laufen. Dazu gibt es PowerShell für die AD-Vorbereitung, ein anonymisiertes Praxisbeispiel mit zwei Real Servern hinter einem VIP und eine ehrliche Übersicht, welche Client-Authentifizierungen sich mit KCD vertragen und welche nicht.

Den Gesamtzusammenhang, also wo der LoadMaster vor Exchange, SharePoint, ADFS und RDS steht und welche Bausteine es sonst noch gibt, findest du auf der Übersichtsseite [Kemp LoadMaster](#). Die Kerberos-Grundlagen, von TGT bis PAC, stehen im Bereich [Microsoft Kerberos](#). Hier geht es um das, was dazwischen passiert.

FAKTEN · KCD am LoadMaster in fünf Sätzen

Der LoadMaster authentifiziert den Benutzer selbst, etwa per Formular, Zertifikat oder SAML.

Danach holt er mit einem eigenen Dienstkonto, dem Trusted User, beim KDC ein Kerberos-Ticket im Namen dieses Benutzers für den Real Server.

Weil der Benutzer gegenüber dem KDC nie ein Kennwort vorgelegt hat, braucht es Protocol Transition (S4U2Self) und anschließend Constrained Delegation (S4U2Proxy).

Ob das erlaubt ist, entscheidet ausschließlich Active Directory, entweder am Trusted User (klassische KCD) oder am Zielkonto (RBCD).

Der Real Server sieht eine ganz normale Kerberos-Anmeldung des Benutzers und merkt vom LoadMaster nichts.

Wie ESP mit KCD arbeitet: der Butler mit Vollmacht

Das Edge Security Pack trennt die Authentifizierung in zwei Hälften. Auf der Clientseite legst du fest, wie sich der Benutzer gegenüber dem LoadMaster ausweist. Auf der Serverseite legst du fest, wie der LoadMaster sich anschließend gegenüber dem Real Server ausweist, und zwar im Namen des Benutzers. KCD ist eine dieser serverseitigen Optionen, neben None, Basic Authentication, Form Based und Server Token. Konfiguriert wird sie über eine SSO-Domäne mit dem Konfigurationstyp für die Serverseite, die Kemp als Outbound Configuration bezeichnet. Was eine SSO-Domäne grundsätzlich ist und wie Vorauthentifizierung am LoadMaster aufgebaut ist, beschreibt [Edge Security Pack: Vorauthentifizierung, SSO-Domänen und der TMG-Ersatz](#).

Der Trusted User: ein Konto mit Vollmacht

Der Trusted User ist ein gewöhnliches Domänenkonto mit ungewöhnlichen Befugnissen. Der LoadMaster meldet sich mit dessen Benutzernamen und Kennwort am KDC an, bekommt ein TGT und darf damit Tickets für andere Benutzer anfordern. Kemp bezeichnet ihn in der Oberfläche als Kerberos Trusted User; in der Outbound-SSO-Domäne trägst du außerdem den Kerberos Realm und den Key Distribution Center ein. Seit Firmware 7.2.51 sind zwei KDCs pro Domäne möglich, der zweite springt ein, wenn der erste ausfällt.

Das Konto braucht einen Service Principal Name. Nicht, weil irgendwer ein Ticket für diesen SPN bestellen würde, sondern weil Active Directory die Delegationseinstellungen erst anbietet, wenn ein SPN existiert, und weil S4U2Self nur für Konten mit SPN funktioniert. Die Kemp-Anleitung verwendet dafür einen symbolischen Wert wie http/trusteduser. Das klappt, solange der SPN eindeutig ist. Wir nehmen lieber einen sprechenden Namen, der niemals mit einem echten Dienst kollidiert, etwa HTTP/svc-lm-kcd.contoso.local.

WICHTIG · Windows-Autorisierungszugriffsgruppe nicht vergessen

Kemp verlangt, dass der Trusted User Mitglied der Gruppe Windows Authorization Access Group ist, auf deutschen Domänencontrollern Windows-Autorisierungszugriffsgruppe. Ohne diese Mitgliedschaft kann das Konto die Gruppenzugehörigkeit der Benutzer nicht vollständig auslesen, und S4U2Self liefert Tickets mit lückenhaften Berechtigungen. Die Anmeldung klappt dann, aber der Benutzer sieht plötzlich nur die Hälfte seiner Dokumente. Ein Fehler, der sich als Berechtigungsproblem tarnt und tagelang beim falschen Team liegt.

Protocol Transition: der eigentliche Zaubertrick

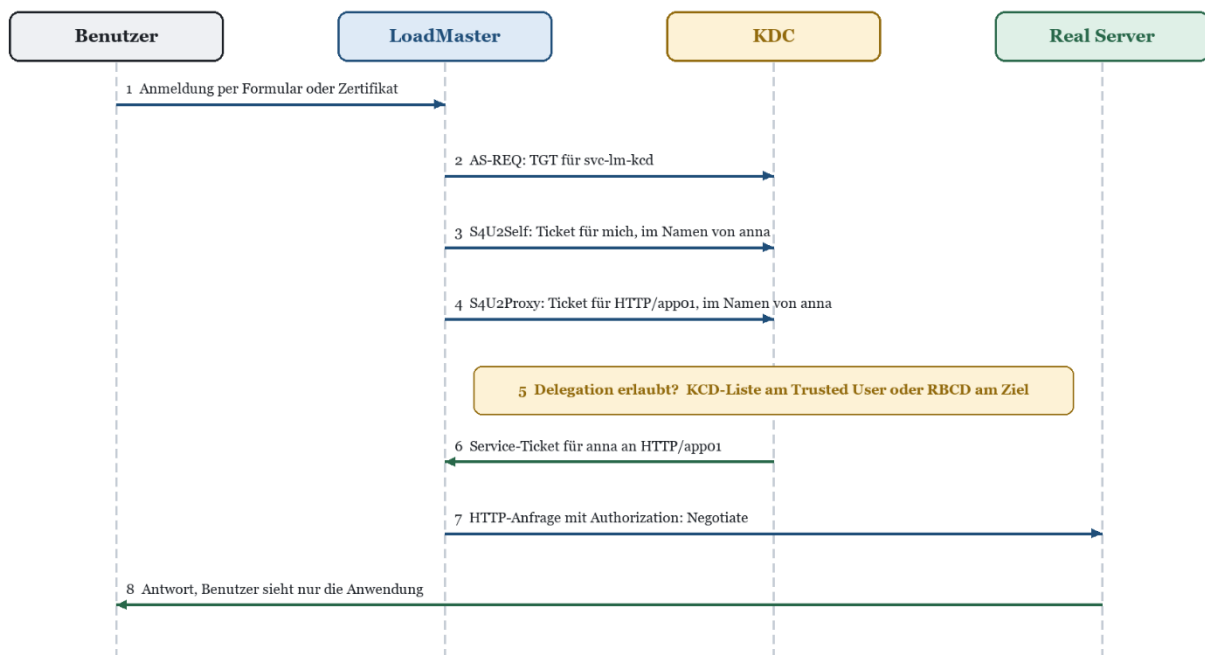
Klassische Delegation setzt voraus, dass der Benutzer sich per Kerberos am Frontend angemeldet hat. Am LoadMaster ist das in aller Regel nicht der Fall. Der Benutzer kommt aus dem Internet,

tippt Benutzernamen und Kennwort in ein Formular oder zeigt ein Zertifikat vor. Der LoadMaster hat also kein Kerberos-Ticket des Benutzers, das er weiterreichen könnte. Hier kommt Service for User to Self ins Spiel, kurz S4U2Self: Der Trusted User bittet den KDC um ein Ticket, das bestätigt, dass Benutzer anna sich bei ihm angemeldet hat. Der KDC glaubt das, weil das Konto ausdrücklich dafür freigeschaltet ist. Microsoft nennt das Protocol Transition, in der Oberfläche von Active Directory heißt die Einstellung „Beliebige Authentifizierungsprotokolle verwenden“.

Mit diesem Ticket in der Hand folgt Service for User to Proxy, kurz S4U2Proxy. Der LoadMaster legt das S4U2Self-Ticket vor und bittet um ein Service-Ticket für HTTP/app01.contoso.local, ausgestellt auf anna. Jetzt prüft der KDC, ob der Trusted User an genau diesen Dienst delegieren darf. Ist das erlaubt, bekommt der LoadMaster ein Ticket, das sich für den Real Server in nichts von einem Ticket unterscheidet, das anna selbst bestellt hätte.

Was zwischen Anmeldung und Ticket passiert

Protocol Transition (S4U2Self) und Constrained Delegation (S4U2Proxy) aus Sicht des LoadMasters



Skizze: vereinfachter Ablauf, Benutzer anna@contoso.local, Ziel app01.contoso.local

WARNUNG · Das ist ein Generalschlüssel, behandle ihn so

Ein Konto mit Protocol Transition kann für jeden Benutzer der Domäne ein Ticket an den freigegebenen Diensten erzeugen, ohne dessen Kennwort zu kennen. Wer das Kennwort des Trusted Users kennt, kann sich also als beliebiger Benutzer an app01 ausgeben, auch als Domänen-Admin, sofern der nicht geschützt ist. Darum: langes, zufälliges Kennwort, keine interaktive Anmeldung, Delegation nur an die tatsächlich benötigten SPNs und privilegierte Konten in Protected Users oder mit „Konto ist vertraulich und kann nicht delegiert werden“. Dann prallt die Delegation an genau den Konten ab, bei denen ein Missbrauch richtig weh tun würde.

Welchen SPN der LoadMaster eigentlich bestellt

Hier steckt die erste Praxisfalle. Der LoadMaster bestellt das Ticket nicht für den Namen der VIP, sondern für den Real Server. Er kennt den Real Server aber nur als IP-Adresse und ermittelt den Namen per Reverse-DNS. Die Kemp-Anleitung prüft das ausdrücklich: Im Debug-Log des SSO-Managers muss sichtbar sein, dass die Adresse des Real Servers in einen Hostnamen aufgelöst

wird. Fehlt der PTR-Eintrag oder zeigt er auf einen veralteten Namen, bestellt der LoadMaster ein Ticket für einen SPN, den es nicht gibt oder der jemand anderem gehört.

Das ist der wesentliche Unterschied zum Kerberos ohne ESP. Dort holt der Client das Ticket für den VIP-Namen, und alle Real Server müssen sich ein Konto teilen. Mit KCD am LoadMaster bekommt jeder Real Server ein Ticket für seinen eigenen Namen, und die Identitätsfrage stellt sich pro Server. Warum das gemeinsame Konto ohne KCD unverzichtbar ist, erklärt [SPN-Probleme hinter dem Load Balancer: ein Name, mehrere Server](#). Mit KCD wird aus dieser Pflicht eine Option, und genau deshalb passt RBCD so gut zu Computerkonten.

Baustein	Wo konfiguriert	Wozu
Kerberos Realm	Outbound-SSO-Domäne am LoadMaster	Name der Domäne in Großbuchstaben, zum Beispiel CONTOSO.LOCAL
Kerberos KDC	Outbound-SSO-Domäne	Ein oder zwei Domänencontroller, ab Firmware 7.2.51 mit Ausfallumschaltung
Trusted User Name und Kennwort	Outbound-SSO-Domäne	Konto, das S4U2Self und S4U2Proxy ausführt
AES256-Option	Verwaltung der SSO-Domänen	Schaltet das Verfahren für KCD auf AES256 SHA1, sonst RC4 und DES-Varianten
SPN am Trusted User	Active Directory	Voraussetzung für Delegation und S4U2Self
Delegation	Active Directory, am Trusted User oder am Zielkonto	Legt fest, an welche Dienste delegiert werden darf
PTR-Einträge	DNS	Liefern dem LoadMaster den Namen, für den er das Ticket bestellt
Zeitsynchronisation	LoadMaster und Domänencontroller	Kerberos toleriert standardmäßig fünf Minuten Abweichung

TIPP • AES-Haken setzen, und zwar sofort

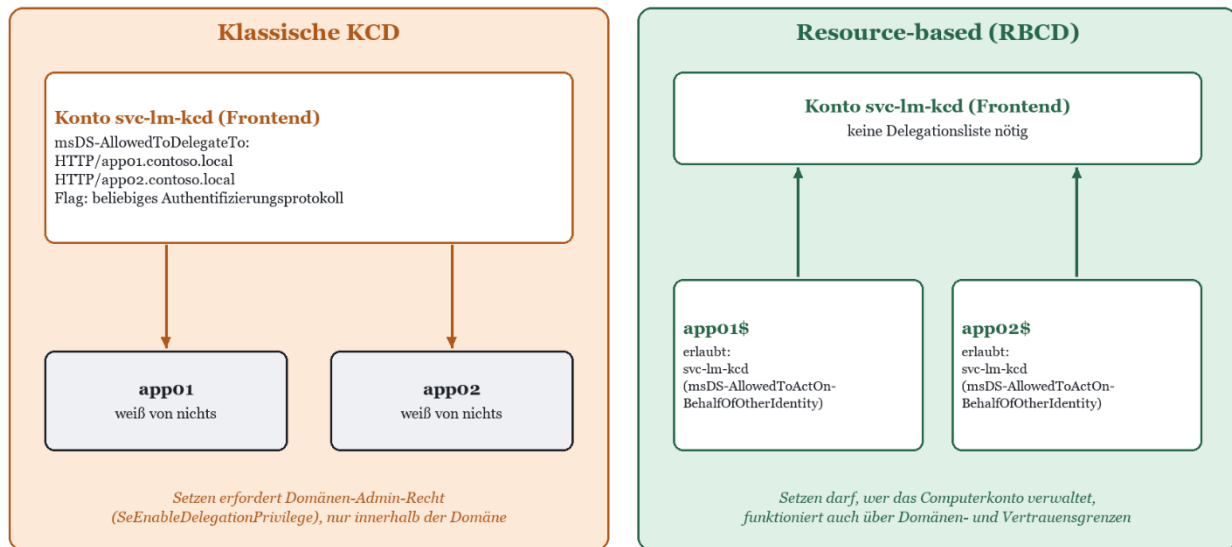
Laut Kemp verwendet KCD ohne die Option für AES256 die Verfahren RC4, DES und DES3. In einer Domäne, die RC4 bereits einschränkt, endet das mit Fehlern, die nach allem aussehen, nur nicht nach Verschlüsselung. Der Trusted User sollte in msDS-SupportedEncryptionTypes AES ausweisen und sein Kennwort nach dem Setzen einmal ändern, damit AES-Schlüssel existieren. Die ganze Geschichte steht in [Die RC4-Falle: KCD-Keytab trifft AES-only-Domäne](#).

Klassische KCD oder RBCD: wer gibt die Erlaubnis?

Constrained Delegation gibt es in Active Directory seit Windows Server 2003, die ressourcenbasierte Variante seit Windows Server 2012. Beide beantworten dieselbe Frage, nämlich ob Konto A im Namen eines Benutzers an Dienst B delegieren darf. Sie speichern die Antwort aber an entgegengesetzten Enden der Leitung, und daraus folgt fast alles, was in der Praxis zählt.

Wer erteilt die Erlaubnis?

Klassische KCD schreibt am Frontend, RBCD schreibt am Backend



Skizze: Richtung der Vertrauensbeziehung bei KCD und RBCD

Klassische KCD: die Liste am Frontend

Bei klassischer KCD steht am Trusted User das Attribut msDS-AllowedToDelegateTo mit der Liste aller SPNs, an die er delegieren darf. Für Protocol Transition kommt das Kontoattribut TRUSTED_TO_AUTH_FOR_DELEGATION hinzu, das ist die Option „Beliebiges Authentifizierungsprotokoll verwenden“ auf der Registerkarte Delegation. So beschreibt es auch die Kemp-Anleitung: Registerkarte öffnen, „Benutzer bei Delegierungen angegebener Dienste vertrauen“ wählen, beliebiges Protokoll erlauben, die Real Server mit dem Diensttyp http hinzufügen.

Das funktioniert, hat aber drei Haken. Erstens braucht das Bearbeiten dieser Liste das Benutzerrecht SeEnableDelegationPrivilege, das standardmäßig nur Domänen-Admins haben. Jeder neue Real Server heißt also: Ticket an das AD-Team, warten, nachfragen, warten. Zweitens funktioniert klassische KCD nur innerhalb einer Domäne, Frontend und Dienst müssen in derselben Domäne liegen. Drittens weiß der Server, an den delegiert wird, nichts davon. Wer app01 betreut, sieht an seinem Konto nicht, dass ein LoadMaster sich als beliebiger Benutzer an ihm anmelden darf.

RBCD: die Erlaubnis am Backend

Bei RBCD dreht sich die Richtung um. Am Zielkonto steht das Attribut msDS-AllowedToActOnBehalfOfOtherIdentity, eine Sicherheitsbeschreibung mit den Konten, die im Namen von Benutzern an diesen Dienst delegieren dürfen. In PowerShell setzt du das bequem über den Parameter PrincipalsAllowedToDelegateToAccount von Set-ADComputer, Set-ADUser oder Set-ADServiceAccount. Bearbeiten darf es, wer Schreibrechte auf das Zielkonto hat; ein Domänen-Admin-Recht ist dafür nicht nötig. Und weil der KDC der Zieldomäne entscheidet, funktioniert RBCD auch über Domänen- und Gesamtstrukturgrenzen hinweg, sofern eine Vertrauensstellung besteht.

Ein Detail, das bei Microsoft im Protokolldokument MS-SFU steht und das Gespräche mit Sicherheitsleuten zuverlässig verlängert: Bei RBCD akzeptiert der KDC das S4U2Self-Ticket auch dann, wenn es nicht als weiterleitbar markiert ist. Das Frontend braucht für RBCD also streng genommen weder eine Delegationsliste noch das Flag für Protocol Transition. Der LoadMaster macht dabei nichts anders, er stellt dieselben S4U-Anfragen, und der KDC prüft einfach an anderer

Stelle. Die Kemp-Dokumentation beschreibt RBCD nicht eigens; teste die Kombination deshalb vor dem Produktivbetrieb mit deiner Firmware im Labor.

Warum RBCD zu Computerkonten passt

Viele Dienste laufen gar nicht unter einem eigenen Dienstkonto. Ein IIS-Anwendungspool mit ApplicationPoolIdentity, ein Windows-Dienst als Netzwerkdienst, Exchange ohne ASA oder eine Eigenentwicklung als http.sys-Dienst unter LocalSystem: Sie alle authentifizieren sich im Netz als Computerkonto. Der SPN HTTP/app01.contoso.local ist dann über den HOST-SPN des Computerkontos APP01\$ bereits abgedeckt, ganz ohne setspn.

Genau hier spielt RBCD seine Stärke aus. Du hängst die Erlaubnis direkt an das Computerkonto jedes Real Servers. Kommt ein dritter Server dazu, setzt die Person, die seine Serverobjekte verwaltet, eine Zeile PowerShell, ohne das AD-Team mit dem Delegationsrecht zu bemühen. Der Trusted User bleibt unverändert, und das ist am LoadMaster ein echter Vorteil, denn laut Kemp unterstützt der LoadMaster nur eine KCD-Konfiguration auf der Serverseite. Ein Trusted User für alle Virtual Services, viele Ziele, die ihm jeweils selbst vertrauen: Das ist das Modell, das auch nach dem fünften Projekt noch überschaubar bleibt.

Kriterium	Klassische KCD	RBCD
Attribut	msDS-AllowedToDelegateTo am Trusted User	msDS-AllowedToActOnBehalfOfOtherIdentity am Ziel
Protocol Transition	Flag „Beliebiges Authentifizierungsprotokoll“ nötig	Funktioniert auch ohne das Flag
Wer darf setzen	Domänen-Admins (SeEnableDelegationPrivilege)	Wer Schreibrechte am Zielkonto hat
Domänengrenzen	Nur innerhalb einer Domäne	Auch über Vertrauensstellungen
Neuer Real Server	Liste am Trusted User erweitern	Eine Zeile am neuen Computerkonto
Werkzeug	Registerkarte Delegierung in ADUC	Nur PowerShell, keine GUI-Registerkarte
Sichtbarkeit	Am Frontend, Ziel weiß nichts	Am Ziel, Frontend weiß nichts
Voraussetzung	Windows Server 2003 und neuer	KDC ab Windows Server 2012

WARNUNG · Beides gleichzeitig ist keine Redundanz, sondern Rätselraten

Wenn am Trusted User noch eine alte Delegationsliste steht und du zusätzlich RBCD setzt, funktioniert es oft, aber niemand weiß mehr, warum. Beim Aufräumen der Altlast fällt dann plötzlich ein Dienst aus, der nur über die Liste lief. Entscheide dich pro Ziel für ein Modell und dokumentiere es.

FAKTEN · Was RBCD nicht aushebelt

Benutzer, die mit „Konto ist vertraulich und kann nicht delegiert werden“ markiert oder Mitglied von Protected Users sind, werden auch bei RBCD nicht delegiert. Das ist gewollt. Teste deshalb nie mit deinem Admin-Konto, sondern mit einem normalen Testbenutzer, sonst jagst du einen Fehler, der eigentlich ein Schutzmechanismus ist.

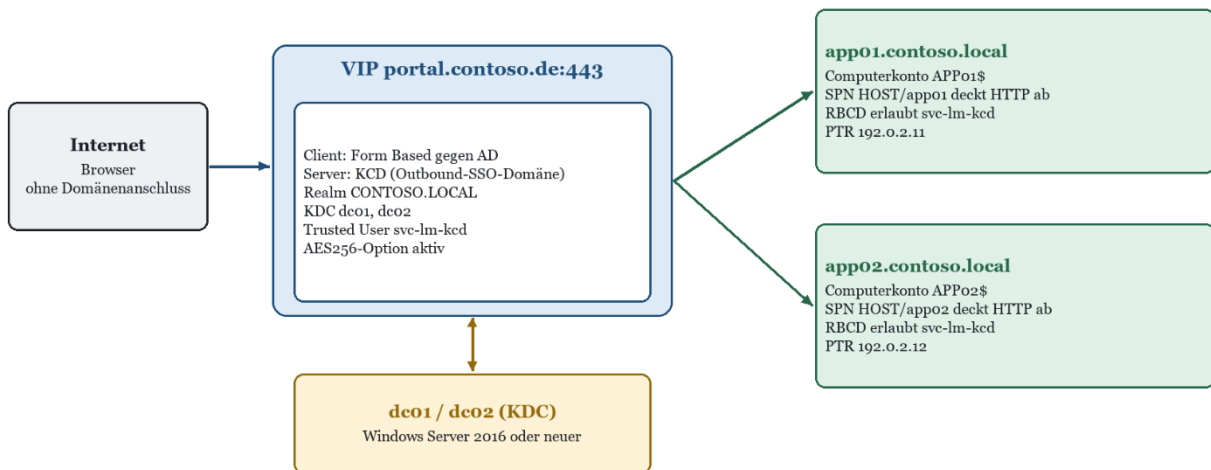
AD-Vorbereitung per PowerShell und ein Praxisbeispiel

Genug Theorie. Das folgende Beispiel ist aus mehreren Projekten zusammengesetzt und anonymisiert. Eine Verwaltung veröffentlicht ein internes Fachportal, eine Eigenentwicklung auf

IIS, für Mitarbeiterinnen und Mitarbeiter im Homeoffice. Das Portal läuft auf zwei Servern, app01 und app02, jeweils mit ApplicationPoolIdentity, also unter den Computerkonten. Nach außen gibt es einen VIP mit dem Namen portal.contoso.de, intern heißt die Domäne contoso.local. Die Benutzer melden sich am LoadMaster per Formular an, die Anwendung selbst erwartet Windows-Authentifizierung.

Praxisbeispiel: zwei Real Server, zwei Computerkonten, ein VIP

portal.contoso.de außen, app01 und app02 innen, die Anwendung läuft als ApplicationPoolIdentity



Skizze: Beispielumgebung, Adressen aus dem Dokumentationsbereich 192.0.2.0/24

Schritt 1: den Trusted User anlegen

Das Konto bekommt ein langes Kennwort, das nicht abläuft, einen eindeutigen SPN, AES als Verschlüsselungstyp und die Mitgliedschaft in der Windows-Autorisierungszugriffsgruppe. Den Wert 24 für msDS-SupportedEncryptionTypes musst du nicht auswendig lernen: Er steht für AES 128 plus AES 256. Achte darauf, das Kennwort erst nach dem Setzen der Verschlüsselungstypen zu vergeben oder anschließend einmal zu ändern.

```

New-ADUser -Name 'svc-lm-kcd' -SamAccountName 'svc-lm-kcd' `
  -UserPrincipalName 'svc-lm-kcd@contoso.local' -Path
  'OU=Dienstkonten,DC=contoso,DC=local' `
  -AccountPassword (Read-Host 'Kennwort' -AsSecureString) -Enabled $true `
  -PasswordNeverExpires $true -CannotChangePassword $true

Set-ADUser svc-lm-kcd -ServicePrincipalNames @{Add='HTTP/svc-lm-kcd.contoso.local'} `
  -KerberosEncryptionType AES128,AES256

# Windows Authorization Access Group, SID S-1-5-32-560
Add-ADGroupMember -Identity 'S-1-5-32-560' -Members svc-lm-kcd

Set-ADAccountPassword svc-lm-kcd -Reset -NewPassword (Read-Host 'Neues Kennwort' -
  AsSecureString)
    
```

TIPP • Interaktive Anmeldung verbieten

Der Trusted User braucht keine Anmeldung an Servern und Clients. Verweigere ihm per Gruppenrichtlinie die lokale Anmeldung und die Anmeldung über Remotedesktop, und nimm ihn in dein Tiering-Modell auf. Gruppenverwaltete Dienstkonten kommen hier übrigens nicht infrage, weil

der LoadMaster kein Windows-System ist und das Kennwort im Klartext in der SSO-Domäne hinterlegt wird.

Schritt 2: RBCD an den Computerkonten setzen

Jetzt bekommen die beiden Real Server die Erlaubnis. Statt das Konto direkt einzutragen, nehmen wir eine Gruppe. Das hält die Sicherheitsbeschreibung kurz und macht den Eintrag lesbar. Ein Haken dabei: Die Gruppenmitgliedschaft des Trusted Users wird aus dessen TGT gelesen. Nimmst du ihn neu in die Gruppe auf, muss der LoadMaster ein neues TGT holen, im Zweifel hilft das erneute Setzen des Kennworts in der SSO-Domäne oder ein wenig Geduld.

```
New-ADGroup -Name 'GG-KCD-LoadMaster' -GroupScope Global -GroupCategory Security `
-Path 'OU=Gruppen,DC=contoso,DC=local'
Add-ADGroupMember 'GG-KCD-LoadMaster' -Members svc-lm-kcd

$frontend = Get-ADGroup 'GG-KCD-LoadMaster'
foreach ($srv in 'APP01','APP02') {
    Set-ADComputer $srv -PrincipalsAllowedToDelegateToAccount $frontend
}

Get-ADComputer APP01 -Properties PrincipalsAllowedToDelegateToAccount |
Select-Object Name, PrincipalsAllowedToDelegateToAccount
```

WARNUNG · PrincipalsAllowedToDelegateToAccount ersetzt, es ergänzt nicht

Der Parameter schreibt die komplette Liste neu. Wenn an app01 bereits ein anderes Frontend delegieren durfte, etwa ein Reporting-Dienst, ist dessen Eintrag nach deinem Befehl weg. Lies die vorhandenen Einträge vorher aus und übergib alle gewünschten Konten gemeinsam. Der Fehler fällt erfahrungsgemäß nicht dir auf, sondern dem anderen Team, und zwar am Monatsende.

Schritt 3: DNS, Zeit und Real Server prüfen

Für beide Real Server müssen Vorwärts- und Rückwärtsauflösung stimmen, also app01.contoso.local auf die Adresse und die Adresse zurück auf app01.contoso.local. Der LoadMaster muss die internen DNS-Server verwenden und eine passende Suchdomäne kennen; die Kemp-Anleitung prüft ausdrücklich, ob einfache Hostnamen auflösbar sind. Die Uhr des LoadMasters gehört an denselben Zeitgeber wie die Domänencontroller. Auf den Real Servern ist im IIS die Windows-Authentifizierung mit dem Anbieter Negotiate aktiv. Weil die Anwendung unter dem Computerkonto läuft, kann die Kernelmodus-Authentifizierung eingeschaltet bleiben; das spart dir die Debatte um useAppPoolCredentials, die bei gemeinsamen Dienstkonten unvermeidlich ist.

```
Resolve-DnsName app01.contoso.local
Resolve-DnsName 192.0.2.11 -Type PTR
setspn -Q HTTP/app01.contoso.local
w32tm /stripchart /computer:dc01.contoso.local /samples:3
```

Schritt 4: den LoadMaster konfigurieren

In der Verwaltung der SSO-Domänen legst du eine serverseitige Konfiguration an, setzt die Option für den AES256-SHA1-Algorithmus, trägt CONTOSO.LOCAL als Realm ein, dco1 und dco2 als KDC sowie svc-lm-kcd mit Kennwort als Trusted User. Im Virtual Service aktivierst du ESP, wählst als Client-Authentifizierung Form Based mit einer clientseitigen SSO-Domäne gegen das Active Directory und als Server-Authentifizierung KCD mit der eben angelegten Outbound-Konfiguration.

Die Real Server trägst du mit ihren Adressen ein; den Namen holt sich der LoadMaster selbst über DNS.

Eine Stellschraube, die erst auffällt, wenn Formulare in der Anwendung abgeschickt werden: Seit Firmware 7.2.51 gibt es die L7-Einstellung L7 Wait after POST. Sie legt fest, wie lange der LoadMaster bei einem POST auf die 401-Antwort des Servers wartet, bevor er den Rest des Inhalts sendet. Der Standard sind 2000 Millisekunden. Bei Anwendungen mit vielen kleinen POST-Anfragen fühlt sich das an wie ein Portal mit Schluckauf; ein kleinerer Wert, im Labor ausprobiert, hilft.

AUS DER PRAXIS · Der PTR-Eintrag aus dem letzten Jahrzehnt

Im Pilotbetrieb lief alles auf app01, auf app02 scheiterte jede Anmeldung mit einem 401. Delegation korrekt, Zeit korrekt, IIS identisch. Das Debug-Log des SSO-Managers verrät den Täter: Die Adresse von app02 wurde in den Namen eines Servers aufgelöst, der vor Jahren abgebaut worden war. Der PTR-Eintrag war nie gelöscht worden, und das Computerkonto des Altserver existierte auch noch. Der LoadMaster bestellte also brav ein Ticket für einen Server, den es nicht mehr gab. Zwei Minuten DNS-Hygiene, und das Portal lief. Die Stunden davor zählen wir lieber nicht.

Welche Client-Authentifizierung passt zu KCD?

KCD ist nur die zweite Hälfte der Geschichte. Die erste Hälfte ist, wie der Benutzer sich am LoadMaster ausweist, und nicht jede Methode lässt sich mit KCD kombinieren. Die Regel dahinter ist einfacher, als die Auswahlliste vermuten lässt: KCD braucht eine verifizierte Benutzeridentität, die der LoadMaster einem AD-Konto zuordnen kann. Alles, was diese Identität liefert, passt grundsätzlich. Alles, was die Anmeldung ungeprüft an den Server weiterreicht oder nur ein Kennwort durchschleift, passt nicht.

Client-Authentifizierung	Mit KCD?	Anmerkung
Form Based	Ja	Der Klassiker. LoadMaster prüft gegen AD, KCD holt das Ticket.
Client Certificate	Ja	Paradefall für Protocol Transition, es gibt gar kein Kennwort. Nicht zusammen mit WAF und KCD auf derselben Ebene.
SAML	Ja	Identität kommt vom IdP, etwa ADFS oder Entra ID; der Benutzer muss im AD existieren und zuordenbar sein.
NTLM und NTLM-Proxy	Ja	Kemp empfiehlt dafür den NTLM-Proxy-Modus; der Real Server muss NTLM unterstützen.
OIDC und OAuth	Versionsabhängig	Vor dem Einsatz in der Firmware-Dokumentation prüfen.
Basic Authentication	Nein	Der LoadMaster stellt die Serverseite automatisch auf Basic. Basic gibt es nur mit Basic.
Delegate to Server	Nein	Keine Vorauthentifizierung, die Serverseite steht auf None.
Pass Post	Nein	Reicht die Anmeldedaten durch, statt eine Identität zu liefern.

Basic Authentication: nur mit Basic

Die Regel ist in der Kemp-Oberfläche fest verdrahtet: Wählst du Basic Authentication auf der Clientseite, setzt der LoadMaster die Serverseite automatisch auf Basic Authentication. KCD steht dann nicht zur Auswahl. Das wirkt zunächst willkürlich, ist aber konsequent. Bei Basic hat der LoadMaster das Kennwort des Benutzers, er kann es einfach weiterreichen, und Clients wie

ActiveSync oder alte Skripte erwarten genau dieses Verhalten. Delegation wäre hier ein Umweg mit zusätzlichem Risiko.

In der Praxis heißt das: Wer OWA per Formular mit KCD und ActiveSync per Basic in einem Namespace betreibt, braucht getrennte SubVS mit unterschiedlichen ESP-Einstellungen. Das ist kein Fehler im Design, sondern genau der Grund, warum es SubVS gibt. Und weil Basic gegenüber Exchange Online ohnehin Geschichte ist und auch On-Premises kaum noch verteidigt werden kann, ist jedes Basic-SubVS eine gute Gelegenheit, über moderne Anmeldung nachzudenken, etwa mit [SAML, OIDC und Entra ID: der LoadMaster als moderner Identitätsvermittler](#).

NTLM und Client-Zertifikat: die Sonderwege

NTLM auf der Clientseite und KCD auf der Serverseite klingt zunächst paradox: Warum sollte der Benutzer sich per NTLM anmelden, nur damit der LoadMaster danach Kerberos spricht? Der Anwendungsfall ist das interne Netz mit Single Sign-on am Domänenclient, bei dem das Backend Kerberos für die Weiterdelegation braucht. Kemp empfiehlt dafür den NTLM-Proxy-Modus in den L7-Einstellungen. Dann prüft der Real Server das NTLM-Handshake, bevor der LoadMaster per KCD weitermacht. Was dabei schiefgehen kann und warum NTLM gern heimlich gewinnt, steht in [NTLM, NTLM-Proxy und der Kerberos-Fallback, den keiner bemerkt](#).

Beim Client-Zertifikat ordnet der LoadMaster das Zertifikat einem AD-Konto zu und holt per Protocol Transition das Ticket. Das ist die eleganteste Variante, weil nirgends ein Kennwort fließt. Die Einschränkung: Client-Zertifikat, WAF und KCD zusammen unterstützt Kemp nicht.

WICHTIG · KCD verträgt sich nicht mit allem

Seit Firmware 7.2.40 lassen sich WAF und KCD nicht auf derselben Ebene eines Virtual Service aktivieren. Der unterstützte Weg ist WAF am übergeordneten Virtual Service und ESP mit KCD im SubVS. Regeln, die den Antwortinhalt verändern, sind mit KCD ebenfalls nicht kombinierbar. Plane das, bevor du Content Rules und Firewall auf denselben Dienst stapelst. Mehr zur Firewall-Seite in [Web Application Firewall und Geoblocking am LoadMaster: Schutz ohne Fehlalarm-Hölle](#).

Wenn es knallt: Diagnose in fünf Stufen

KCD-Fehler haben die unangenehme Eigenschaft, sich alle gleich anzufühlen: Der Benutzer meldet sich an und bekommt trotzdem keinen Zugang. Gehe deshalb streng in Reihenfolge vor. Am LoadMaster lohnt das Debug-Log des SSO-Managers, dort erscheinen die Meldungen der Kerberos-Bibliothek. Auf den Domänencontrollern zeigt das Ereignis 4769 jede Ticketanforderung samt Dienstname, Verschlüsselungstyp und, bei Delegation, den übertragenen Diensten. Auf dem Real Server verrät das Anmeldeereignis 4624, ob wirklich Kerberos angekommen ist.

Wenn es knallt: Prüfreihenfolge für KCD am LoadMaster

Von links nach rechts, weil Kerberos Uhrzeit und Namen vor allem anderen braucht



Benutzer bekommt Fehler 401 trotz Anmeldung	Zeit, PTR oder fehlende Delegation
SSOMGR-Log: KDC has no support for encryption type	AES-Option oder Kontoschlüssel
SSOMGR-Log: KDC can't fulfill requested option	Delegation nicht erlaubt oder Benutzer gesperrt
Ticket kommt an, Anwendung lehnt ab	SPN am falschen Konto, siehe SPN-Beitrag

Skizze: typische Meldungen sinngemäß, der Wortlaut hängt von Firmware und Debug-Stufe ab

Für die tieferen Werkzeuge am LoadMaster, also L7-Debug, Mitschnitte und die richtigen Logstufen, gibt es [Troubleshooting am LoadMaster: Logs, L7-Debug und TCP-Dump](#).

FAQ: KCD und RBCD am Kemp LoadMaster

Brauche ich für RBCD am LoadMaster eine besondere Firmware-Option?

Nein. Der LoadMaster stellt dieselben S4U-Anfragen wie bei klassischer KCD. Ob die Delegation über die Liste am Trusted User oder über das Attribut am Zielkonto erlaubt wird, entscheidet der KDC. Kemp beschreibt RBCD nicht eigens, deshalb gehört ein Labortest dazu.

Kann ich mehrere Trusted User für verschiedene Virtual Services anlegen?

Laut Kemp unterstützt der LoadMaster nur eine KCD-Konfiguration auf der Serverseite. Plane also mit einem Trusted User für alle Dienste. RBCD macht das handhabbar, weil die Erlaubnis an den Zielen hängt.

Warum funktioniert die Anmeldung mit meinem Admin-Konto nicht?

Vermutlich, weil es Mitglied von Protected Users ist oder als vertraulich markiert wurde. Solche Konten werden bewusst nicht delegiert. Teste mit einem normalen Benutzer.

Kann ich ein gMSA als Trusted User verwenden?

Nein. Der LoadMaster braucht Benutzernamen und Kennwort im Klartext für die SSO-Domäne. Ein gruppenverwaltetes Dienstkonto gibt sein Kennwort nur an berechtigte Windows-Systeme heraus.

Muss der SPN des Real Servers am Computerkonto oder am Dienstkonto hängen?

An dem Konto, unter dem der Dienst die Tickets entschlüsselt. Bei ApplicationPoolIdentity oder Netzwerkdienst ist das das Computerkonto, und der HOST-SPN deckt HTTP bereits ab. Bei einem eigenen Dienstkonto setzt du den SPN dort und RBCD ebenfalls am Dienstkonto.

Warum geht KCD nicht zusammen mit Basic Authentication?

Weil der LoadMaster bei Basic auf der Clientseite die Serverseite fest auf Basic setzt. Er hat das Kennwort ohnehin und reicht es weiter. Brauchst du beides, trennst du die Pfade in SubVS.

Ist RBCD sicherer als klassische KCD?

Es ist anders verteilt, nicht automatisch sicherer. Wer Schreibrechte auf ein Computerkonto hat, kann RBCD setzen. Kontrolliere deshalb, wer Computerobjekte bearbeiten darf, und überwache Änderungen an msDS-AllowedToActOnBehalfOfOtherIdentity.

Fazit: Vollmacht ja, aber mit Unterschrift am richtigen Ende

Kerberos Constrained Delegation am LoadMaster ist kein schwarzes Loch, es ist eine Vollmacht mit klar definierten Grenzen. Der Benutzer weist sich am LoadMaster aus, der Trusted User holt per Protocol Transition und S4U2Proxy das Ticket, und Active Directory entscheidet, ob das erlaubt ist. Die Fehler entstehen fast nie im LoadMaster selbst, sondern drumherum: im DNS, in der Uhrzeit, in fehlenden AES-Schlüsseln, in einer vergessenen Gruppenmitgliedschaft oder in einem PTR-Eintrag, der seit Jahren auf einen Server zeigt, der längst Elektroschrott ist.

Wenn deine Dienste unter Computerkonten laufen, ist RBCD die sauberere Wahl. Die Erlaubnis hängt dort, wo sie hingehört, an der Ressource, du brauchst kein Domänen-Admin-Recht für jeden neuen Server, und der eine Trusted User, den der LoadMaster kennt, bleibt unangetastet. Setze AES von Anfang an, teste mit normalen Benutzern statt mit deinem Admin-Konto, und dokumentiere, welches Ziel welches Modell nutzt. Dann bleibt der Heizungskeller ein Ort, an dem einfach nur die Heizung läuft.

TIPP · Unterstützung bei Einrichtung und Review

Wenn du KCD, RBCD und Vorauthentifizierung in einer gewachsenen Umgebung sauber aufsetzen oder bestehende Delegationen prüfen lassen willst, hilft dir die [Beratung zum Kemp LoadMaster](#). Wer das Zusammenspiel von ESP und Kerberos mit dem eigenen Team im Labor durchspielen möchte, findet das in der [Kemp-LoadMaster-Schulung](#).

WEITERLESEN · Weiter im Thema

- › [Microsoft Kerberos](#)
- › [Edge Security Pack: Vorauthentifizierung, SSO-Domänen und der TMG-Ersatz](#)
- › [SPN-Probleme hinter dem Load Balancer: ein Name, mehrere Server](#)
- › [Die RC4-Falle: KCD-Keytab trifft AES-only-Domäne](#)
- › [NTLM, NTLM-Proxy und der Kerberos-Fallback, den keiner bemerkt](#)
- › [SAML, OIDC und Entra ID: der LoadMaster als moderner Identitätsvermittler](#)
- › [Troubleshooting am LoadMaster: Logs, L7-Debug und TCP-Dump](#)