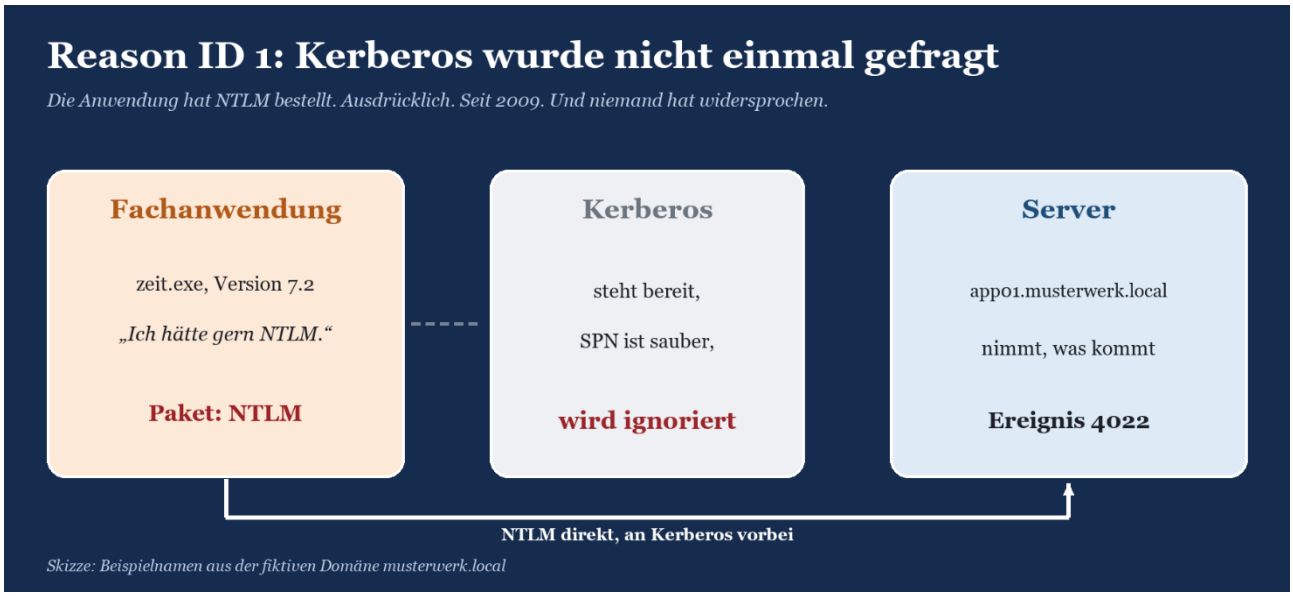
Wenn die Anwendung NTLM fest eingebaut hat: Hersteller richtig fragen



WISSEN
Was der NTLM Diagnostiker erfasst, wie er Prozess, Ziel und Häufigkeit einer Anwendung zeigt und was er kostet.
› **NTLM Diagnostiker**

BERATUNG
Fachanwendungen bewerten, Herstelleranfragen formulieren, Ausnahmen sauber begrenzen.
› **Beratung zur NTLM-Ablösung**

SCHULUNG
Reason-IDs lesen, Anwendungen als Ursache erkennen, Belege sammeln: damit dein Team die Fälle selbst klärt.
› **Schulung zum NTLM Diagnostiker**

Die meisten NTLM-Ursachen sind Konfigurationsfehler, die du selbst reparierst: ein fehlender SPN, ein Laufwerk auf eine IP-Adresse, ein lokales Konto in einer geplanten Aufgabe. Ärgerlich, aber dein Revier. Reason ID 1 ist anders. Hier hat Windows gar nicht erst versucht, Kerberos zu benutzen, weil die Anwendung das ausdrücklich nicht wollte. Sie hat beim Sicherheitssystem nicht „mach das Beste draus" bestellt, sondern „NTLM, bitte, und zwar sofort". Das Active Directory kann noch so sauber sein, die SPNs noch so gepflegt: Wer Kerberos nicht fragt, bekommt auch keins.

Im erweiterten NTLM-Auditing steht dann im Meldungstext „NTLM was called directly by the calling application." Übersetzt: Das Problem steckt im Programm, nicht im Netz. Und Programme gehören jemandem. Meistens einem Hersteller, manchmal einem Dienstleister, gelegentlich einem Kollegen, der 2011 „mal schnell was gebaut" hat und inzwischen Ziegen in Brandenburg züchtet.

Der Zeitdruck ist real: Seit dem Windows-Update vom 13.10.2026 blockieren Windows 11 ab 24H2 und Windows Server 2025 Single Sign-on mit NTLMv1 standardmäßig. Phase 2 bringt IAKerb und ein lokales KDC, und mit der nächsten Windows-Server-Version ist NTLM standardmäßig abgeschaltet. Eine Anwendung, die NTLM fest eingebaut hat, profitiert von keiner dieser Neuerungen. Sie fällt einfach aus. Den vollständigen Zeitplan findest du in NTLM-Abschaltung: Microsofts Fahrplan und was er für dich bedeutet.

Wer gerade das Fokus-Thema „Anwendung von NTLM auf Kerberos umstellen" auf dem Tisch hat, merkt schnell: Technisch ist das selten schwer, organisatorisch fast immer. Dieser Beitrag zeigt, was Reason ID 1 technisch bedeutet, welche Anwendungen typischerweise dahinterstecken, wie du

sie mit Bordmitteln findest, wie eine Herstelleranfrage aussieht, die nicht im Ticketsystem verhungert, und was du tust, wenn trotzdem nichts geht.

FAKTEN · Reason ID 1 auf einen Blick

Meldungstext laut Microsoft KB5064479: „NTLM was called directly by the calling application."

Fundort: Client-Ereignisse 4020 (Information) und 4021 (Warnung, etwa bei NTLMv1) im Protokoll Microsoft-Windows-NTLM/Operational. Wer NTLM bestellt hat, steht unter „Process Name" und „Process PID", das Ziel unter „Target Machine" und „Target Resource (SPN)", die Version unter „NTLM Version".

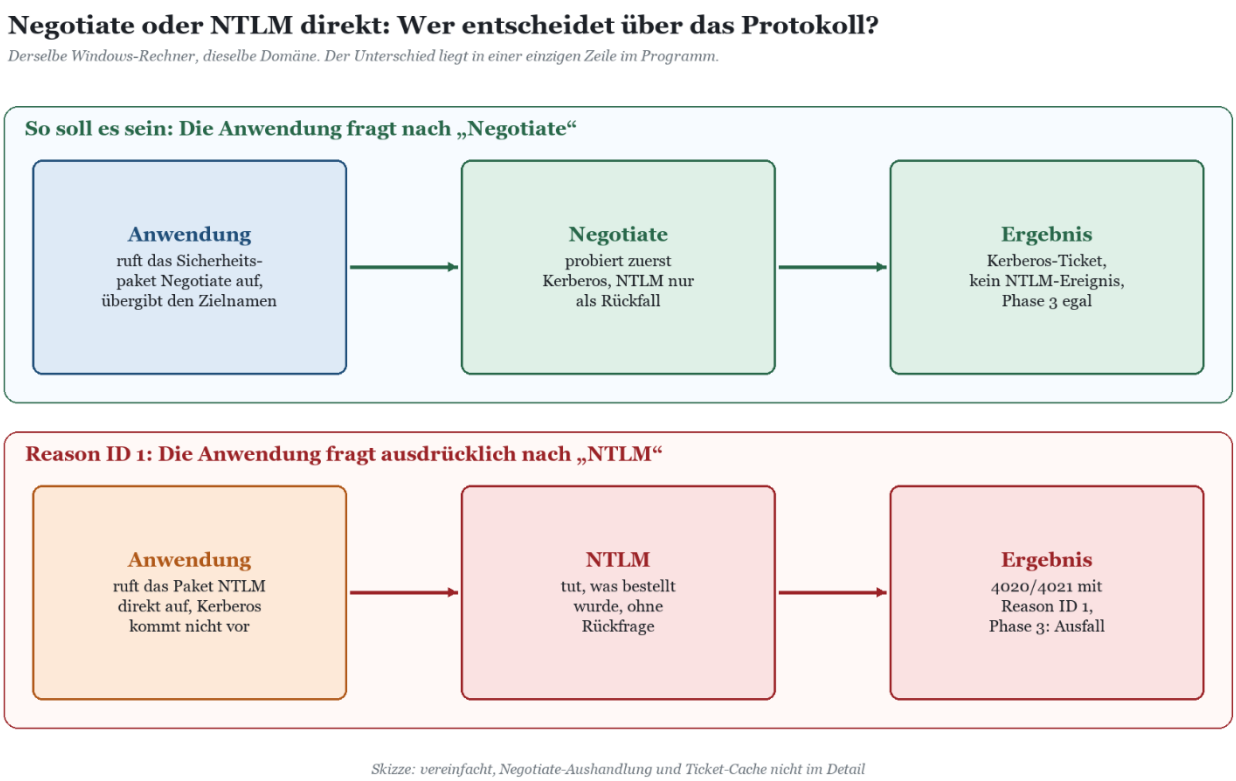
Gegenseite: Der Zielserver protokolliert die eingehende Anmeldung als 4022 oder 4023, der Domain Controller die weitergereichte Prüfung als 4032 oder 4033 (bei Domänenübergang 4030 oder 4031). Die Reason ID kennt nur der Client.

Systeme mit erweitertem Auditing: Windows 11 ab 24H2, Windows Server 2025 sowie Windows Server 2022 und 2019 mit Updates ab dem 13.10.2026. Ältere Systeme liefern nur die klassischen Restrict-NTLM-Ereignisse 8001 bis 8004, ohne Begründung und meist ohne Prozessnamen.

Negotiate oder NTLM direkt: der kleine Unterschied mit großer Wirkung

Windows-Anwendungen authentifizieren sich in aller Regel nicht selbst, sondern über die Sicherheitsschnittstelle des Betriebssystems, SSPI. Dort wählt das Programm ein Sicherheitspaket aus. Die vernünftige Wahl heißt Negotiate: Windows probiert Kerberos, und nur wenn das nicht klappt, fällt es auf NTLM zurück. Dann erscheint im Audit eine der anderen Reason-IDs, etwa 6 für einen nicht auflösbaren Zielnamen oder 7 für eine IP-Adresse. Wie Kerberos dabei im Detail arbeitet, steht im Kompetenzbereich Microsoft Kerberos.

Die unvernünftige Wahl heißt NTLM. Dann wird Kerberos übersprungen, ohne Versuch, ohne Rückfrage. Windows protokolliert das ehrlich als Reason ID 1, und genau das macht die ID so wertvoll: Sie sagt dir, dass du mit SPNs, DNS und Firewall-Regeln nichts gewinnst. Du kannst den Kerberos-Teppich ausrollen, so rot du willst; die Anwendung läuft daneben her.



Warum Anwendungen NTLM direkt aufrufen

Selten aus Bosheit, meist aus Geschichte. Viele Fachanwendungen stammen aus einer Zeit, in der NTLM „einfach funktionierte“ und Kerberos als launische Diva galt, die SPNs, Zeitsynchronisation und korrekte Namen verlangt. Ein Entwickler, der eine Fehlermeldung im Kundenbetrieb loswerden wollte, hat dann eben NTLM fest eingetragen. Das Ticket wurde geschlossen, der Code blieb. Andere Gründe: Bibliotheken, die nur NTLM können, Beispielcode aus einem Forum, der „NTLM“ statt „Negotiate“ setzte, oder Webserver, die nur den NTLM-Anbieter aktiviert haben und den Client damit zwingen, ebenfalls NTLM zu sprechen.

Abgrenzung zu den Nachbar-Ursachen

Reason ID 1 wird gern mit anderen Begründungen verwechselt, weil sie sich in der Praxis ähnlich anfühlen: immer derselbe Prozess, immer dasselbe Ziel. Die Unterscheidung entscheidet über den Lösungsweg. Die vollständige Liste aller zwölf IDs steht in NTLM-Reason-IDs erklärt: warum Windows nicht Kerberos benutzt hat.

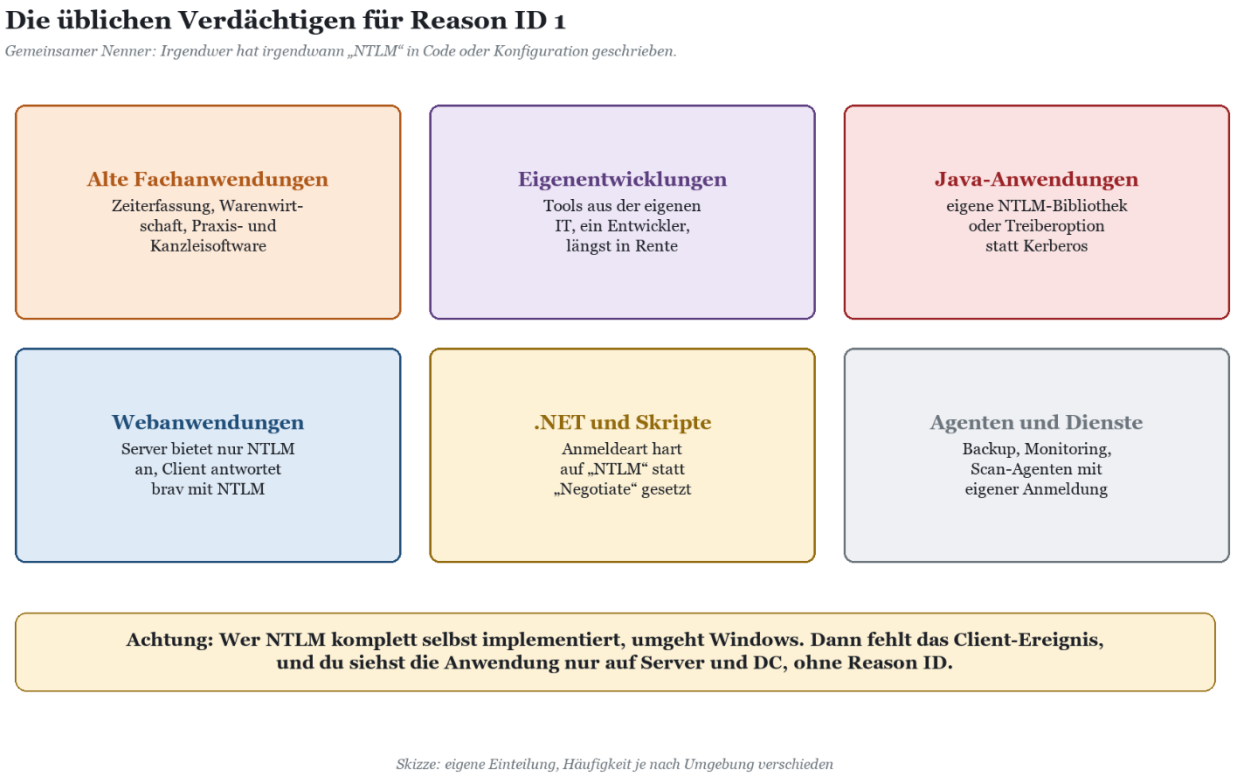
Reason ID	Meldungstext (KB5064479)	Wo liegt das Problem	Wer behebt es
1	NTLM was called directly by the calling application.	im Programm oder seiner Konfiguration	Hersteller, Entwickler, manchmal du per Einstellung
5	The target name was missing or empty.	Programm übergibt keinen Zielnamen	meist Hersteller
6	The target name could not be resolved by Kerberos or other protocols.	SPN oder DNS	du, mit setspn und DNS
7	The target name contains an IP address.	Pfad oder Konfiguration mit IP	du, Umstellung auf Namen
2	Authenticating a Local Account.	Konto statt Programm	du, Umstellung auf Domänenkonto

WARNUNG · Kein Client-Ereignis heißt nicht kein NTLM

Reason ID 1 erscheint nur, wenn die Anwendung NTLM über Windows anfordert. Bringt ein Programm seine eigene NTLM-Implementierung mit, etwa eine Java- oder Python-Bibliothek, läuft die Anmeldung an der Windows-Sicherheitsschnittstelle des Clients vorbei. Dann siehst du auf dem Client nichts, auf dem Zielserver aber Ereignis 4022 oder 4023 und auf dem DC 4032, 4033 oder 8004. Wer nur Client-Ereignisse auswertet, übersieht genau diese Anwendungen.

Typische Kandidaten: wer NTLM fest eingebaut hat

Nach etlichen Auswertungen in Mittelstand, Kliniken und Verwaltungen ist das Bild erstaunlich stabil. Reason ID 1 kommt selten von Microsoft-Bordmitteln, sondern fast immer von Software, die jemand zusätzlich installiert hat. Die Gruppen unterscheiden sich darin, wie gut du selbst eingreifen kannst.



Skizze: eigene Einteilung, Häufigkeit je nach Umgebung verschieden

Alte Fachanwendungen

Zeiterfassung, Warenwirtschaft, Praxisverwaltung, Kanzleisoftware, Laborsysteme. Gemeinsam ist ihnen ein Client, der sich per Windows-Anmeldung an einem Anwendungsserver oder direkt an einer Datenbank meldet, und ein Hersteller, der die Anmeldeschicht seit Jahren nicht angefasst hat. Oft gibt es irgendwo eine Einstellung wie „Authentifizierung: Windows (NTLM)“ und daneben, gut versteckt, „Windows (Kerberos)“ oder „Integriert“. Lies die Administrationsdokumentation, bevor du den Hersteller anrufst; manchmal ist die Lösung ein Häkchen. In Kliniken ist die Lage besonders heikel, weil Medizinprodukte oft nicht ohne Weiteres aktualisiert werden dürfen. Mehr dazu steht in NTLM in der Klinik: Medizintechnik, Labor und Fachanwendungen.

Eigenentwicklungen

Das Inventar-Tool der IT, das PowerShell-Skript mit fest verdrahteter Anmeldeart, die C#-Anwendung, die per CredentialCache ausdrücklich „NTLM“ statt „Negotiate“ anfordert. Hier ist die gute Nachricht: Der Quellcode liegt bei dir. Die schlechte: Niemand weiß mehr, wo. Bei

Eigenentwicklungen bist du selbst der Hersteller, und die Herstelleranfrage geht an die eigene Entwicklungsabteilung oder den Dienstleister, der damals den Auftrag hatte. Die Änderung ist meist klein: das Paket auf Negotiate umstellen und dafür sorgen, dass ein korrekter Zielname übergeben wird.

Java-Anwendungen

Java ist ein Sonderfall, weil viele Java-Programme Windows gar nicht nach der Anmeldung fragen. Sie bringen eigene Bibliotheken mit, die NTLM selbst sprechen, oder nutzen Treiber mit konfigurierbarer Anmeldeart. Der Microsoft-JDBC-Treiber für SQL Server etwa kennt für die integrierte Anmeldung verschiedene Verfahren, darunter Kerberos und NTLM; welches greift, steht in der Verbindungszeichenfolge. Das ist praktisch, weil du es selbst ändern kannst, und tückisch, weil eine selbst implementierte NTLM-Anmeldung kein Client-Ereignis erzeugt. Für Kerberos braucht die Java-Seite außerdem eine saubere Konfiguration, und ein falsches Detail dort endet gern in einem stillen Rückfall. Prüfe hier immer auch die Server-Ereignisse.

Webanwendungen

Ein Klassiker mit vertauschten Rollen: Der Browser auf dem Client würde gern Kerberos sprechen, aber der Webserver bietet nur NTLM an. Der Client antwortet höflich in der angebotenen Sprache. In der Auswertung erscheint dann der Browser als Prozess und der Webserver als Ziel. Die Lösung liegt auf der Serverseite: Negotiate als Anbieter, ein SPN für den Webnamen und gegebenenfalls das richtige Dienstkonto für den Anwendungspool. Die Details stehen in NTLM in IIS und Webanwendungen: Negotiate statt NTLM. Läuft der Anwendungspool unter einem eigenen Konto, hilft dir Dienstkonten als NTLM-Quelle: finden, prüfen, umstellen weiter.

Agenten, Dienste und Skripte

Backup-Agenten, Monitoring, Scan-Software, Druckverwaltung: Programme, die im Hintergrund laufen, sich regelmäßig anmelden und dabei gern ihre eigene Meinung zur Authentifizierung haben. Sie fallen weniger durch die Zahl der Rechner auf als durch die Frequenz. Ein Agent, der sich alle fünf Minuten meldet, produziert auf 300 Rechnern über 86.000 Ereignisse am Tag. Die Ereignisanzeige freut sich, du nicht.

Kandidat	Woran du ihn erkennst	Selbst änderbar?	Erster Schritt
Alte Fachanwendung	Prozess aus Programmverzeichnis des Herstellers, festes Ziel	manchmal per Einstellung	Admin-Doku lesen, dann Hersteller
Eigenentwicklung	unbekannter Prozessname, interner Pfad, keine Signatur	ja, mit Quellcode	Quellcode suchen, Paket auf Negotiate
Java-Anwendung	java.exe oder javaw.exe, oft nur Server-Ereignisse	oft per Konfiguration	Treiber- und Bibliotheksoptionen prüfen
Webanwendung	Browser als Prozess, Webserver als Ziel	ja, auf dem Server	Anbieter im Webserver auf Negotiate
Agent oder Dienst	sehr hohe Frequenz, Systemkonto oder Dienstkonto	selten	Version prüfen, Hersteller fragen

Reason ID 1 mit Bordmitteln finden

Bevor du einem Hersteller schreibst, brauchst du Belege. Und zwar nicht „irgendwas mit NTLM“, sondern Prozess, Ziel und Häufigkeit. Die gute Nachricht: Das erweiterte Auditing liefert genau diese Felder. Falls du es noch nicht eingeschaltet hast, steht der Weg in Erweitertes NTLM-Auditing einschalten: Schritt für Schritt; das NTLM-Audit-GPO-Skript nimmt dir die Gruppenrichtlinie dafür ab.

Ereignisanzeige: der erste Blick

Auf einem verdächtigen Client öffnest du in der Ereignisanzeige „Anwendungs- und Dienstprotokolle“, dann Microsoft, Windows, NTLM, Operational. Filtere auf 4020 und 4021 und schau dir den Meldungstext an. Interessant sind „Process Name“, „Target Machine“, „Target Resource (SPN)“, „Reason ID“ und „NTLM Version“. Für einen Rechner reicht das. Wie du die Felder im Detail liest, steht in Ereignis 4020 und 4021: ausgehende NTLM-Anmeldungen lesen.

PowerShell: Prozesse und Ziele zählen

Für mehr als einen Rechner hilft PowerShell. Das folgende Beispiel liest die Client-Ereignisse der letzten sieben Tage, filtert auf Reason ID 1 und gruppiert nach Prozess und Ziel. Es wertet den Meldungstext aus, weil der in jeder Sprache dieselben englischen Feldbezeichnungen trägt; prüfe die Muster trotzdem an einem echten Ereignis aus deiner Umgebung.

```
$start = (Get-Date).AddDays(-7)
$ev = Get-WinEvent -FilterHashtable @{
    LogName   = 'Microsoft-Windows-NTLM/Operational'
    Id        = 4020, 4021
    StartTime = $start
} -ErrorAction SilentlyContinue

$ev | Where-Object { $_.Message -match 'Reason ID:\s*1\s' } | ForEach-Object {
    [pscustomobject]@{
        Prozess = ([regex]::Match($_.Message, 'Process Name:\s*(.+)')).Groups[1].Value.Trim()
        Ziel    = ([regex]::Match($_.Message, 'Target Machine:\s*(.+)')).Groups[1].Value.Trim()
        Version = ([regex]::Match($_.Message, 'NTLM Version:\s*(.+)')).Groups[1].Value.Trim()
    }
} | Group-Object Prozess, Ziel, Version | Sort-Object Count -Descending |
    Select-Object Count, Name -First 20
```

Für mehrere Rechner kannst du das Ganze mit Invoke-Command verteilen oder die Ereignisse vorher zentral einsammeln. Wie das mit Bordmitteln geht, beschreibt NTLM-Ereignisse zentral sammeln mit Windows Event Forwarding. Gegen Java-Programme und andere Eigenbau-NTLM-Clients hilft das allerdings nicht, denn die erzeugen kein Client-Ereignis. Dort schaust du auf den Zielserver: Ereignis 4022 und 4023 nennen „Client Machine“ und den Prozess auf der Serverseite, der die Anmeldung angenommen hat. Mehr dazu steht in Ereignis 4022 und 4023: eingehende NTLM-Anmeldungen auf Servern.

TIPP · Version und Pfad gleich mitnehmen

Der Prozessname allein ist für einen Hersteller wenig wert. Ermittle auf einem betroffenen Rechner den vollständigen Pfad und die Dateiversion, etwa mit (Get-Item 'C:\Programme\Hersteller\zeit.exe').VersionInfo. Damit weiß der Support sofort, über welchen Stand ihr redet, und kann dir nicht mit „bitte erst aktualisieren“ ausweichen, wenn du schon auf der aktuellen Version bist.

Wo es mühsam wird

Auf einem Testrechner ist das alles nett. In einer echten Umgebung mit 400 Clients, 60 Servern und einem Agenten, der sich alle fünf Minuten meldet, sieht es anders aus. Du sammelst Hunderttausende Ereignisse, musst Meldungstexte zerlegen, Prozesse über Rechnergrenzen zusammenführen und Client- mit Server-Sicht abgleichen, damit dir die Java-Anwendung nicht durchrutscht. Dazu kommen Rechner, die nie ein Ereignis liefern, weil das Auditing dort gar nicht aktiv ist. Und am Ende soll daraus ein Dokument werden, das ein Hersteller ernst nimmt. Das ist der Punkt, an dem das Skript aufhört, ein Werkzeug zu sein, und anfängt, ein Projekt zu werden.

SO GEHT'S MIT DEM NTLM DIAGNOSTIKER · Prozess, Ziel und Häufigkeit als Beleg

Der NTLM Diagnostiker arbeitet ausschließlich lesend. Er liest die NTLM-Ereignisse direkt von den Domain Controllern, über einen Windows Event Collector oder per Direktabfrage und führt sie zusammen. Für Reason ID 1 bedeutet das:

Die Ursachen-Engine ordnet die Ereignisse ihren Reason-IDs zu und trennt Anwendungsfälle von SPN- und DNS-Problemen, die sie gesondert prüft.

Die Kommunikationsmatrix zeigt, welcher Prozess von welchen Rechnern zu welchem Ziel spricht und wie oft. Die Konten-Sicht ergänzt, unter welchem Konto das passiert. Genau diese drei Angaben braucht deine Herstelleranfrage.

Die NTLMv1-Hitliste zeigt, ob eine Anwendung zusätzlich noch NTLMv1 spricht, was seit dem 13.10.2026 doppelt dringend ist. Sicherheitsflaggen markieren auffällige Anmeldungen.

Der Abdeckungsmonitor und die OS-Landkarte zeigen, welche Rechner gar keine erweiterten Ereignisse liefern, damit dir keine Anwendung durch ein Funkloch rutscht. Für Eigenbau-NTLM ohne Client-Ereignis hilft der Wireshark-Import.

Die Abschalt-Simulation zeigt, was ausfallen würde, wenn NTLM für dieses Ziel abgeschaltet wäre. Der Word-Bericht und der Excel- oder CSV-Export liefern dir die Zahlen als Anlage für den Hersteller.

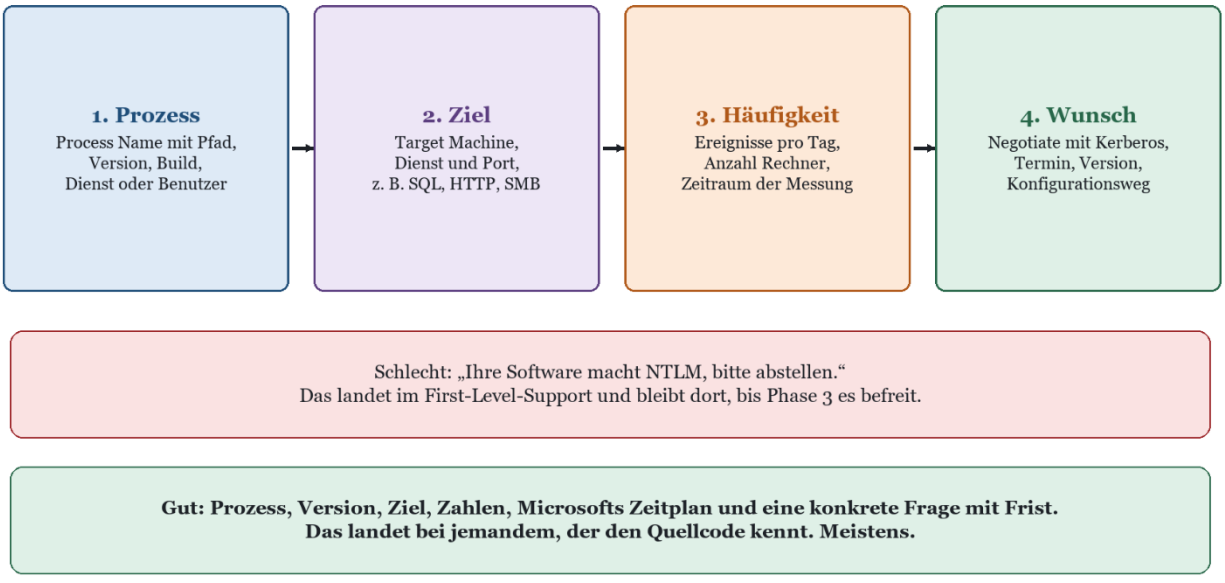
Mit der kostenlosen Edition Express hast du einen Fünf-Minuten-Befund, der Testzeitraum läuft 7 Tage, lizenziert wird pro aktivem Computerkonto. Vorlagen für Herstelleranfragen und ein Anwendungskatalog sind als Ausbaustufen geplant, heute noch nicht enthalten.

Die Herstelleranfrage: was rein muss, damit sie beantwortet wird

Hersteller-Support funktioniert wie ein Krankenhausflur am Wochenende: Wer am lautesten jammert, wird nicht zuerst behandelt, sondern wer die besten Befunde mitbringt. Eine Anfrage nach dem Muster „Ihre Software macht NTLM, bitte abstellen“ landet beim First-Level-Support, der dir freundlich erklärt, dass die Software „voll Active-Directory-kompatibel“ sei. Das stimmt sogar. NTLM ist ja Teil von Active Directory. Noch.

Vier Bausteine einer Herstelleranfrage, die beantwortet wird

Ohne Belege bekommst du „bei uns geht das“. Mit Belegen bekommst du einen Ansprechpartner.



Skizze: eigenes Vorgehensmodell

Die vier Pflichtangaben

- **Prozessname:** vollständiger Pfad, Dateiversion, Produktname und Lizenz- oder Kundennummer. Dazu, ob der Prozess als Dienst oder in der Benutzersitzung läuft.
- **Ziel:** Zielserver und Dienst, also etwa Datenbank, Webdienst oder Dateifreigabe, mit Port, wenn bekannt. Der Inhalt von „Target Machine“ und „Target Resource (SPN)“ gehört hinein, ohne dass du dabei interne Hostnamen preisgibst, die der Hersteller nicht kennen muss.
- **Häufigkeit:** Ereignisse pro Tag, Anzahl betroffener Rechner, Messzeitraum. Zahlen machen aus einem Einzelfall ein Problem, das eine Produktmanagerin oder ein Produktmanager ernst nimmt.
- **Gewünschtes Verhalten:** Authentifizierung über Negotiate mit Kerberos, ohne festen Aufruf von NTLM. Dazu die konkrete Frage: Gibt es eine Einstellung, eine Version oder einen Termin?

Zusätzlich hilft der Hinweis auf Microsofts Zeitplan. Hersteller kennen KB-Nummern, und ein nüchterner Verweis auf die angekündigte Standardabschaltung mit der nächsten Windows-Server-Version wirkt oft besser als jede Eskalation. Für NTLMv1-Fälle kommt der Verweis auf die seit Oktober 2026 geltende Blockade dazu; die Einzelheiten stehen in NTLMv1 blockiert: was seit Oktober 2026 nicht mehr geht.

MUSTERTEXT · Herstelleranfrage zu fest eingebautem NTLM

Betreff: Authentifizierung von [Produkt, Version] verwendet NTLM statt Kerberos, Bitte um Stellungnahme bis [Datum]

Sehr geehrte Damen und Herren,

wir setzen [Produkt] in Version [x.y, Build z] unter Kundennummer [Nummer] ein. Im Rahmen der NTLM-Ablösung in unserer Active-Directory-Umgebung haben wir das erweiterte NTLM-Auditing nach Microsoft KB5064479 aktiviert. Dabei fällt Ihre Anwendung mit folgendem Befund auf:

Prozess: [Pfad\Programm.exe], Ausführung als [Dienst unter Konto X oder Benutzersitzung]. Ziel: [Anwendungsserver oder Datenbank], Dienst [z. B. SQL Server, HTTP, SMB]. Häufigkeit: rund [Anzahl] Anmeldungen pro Tag von [Anzahl] Arbeitsplätzen, gemessen vom [Datum] bis [Datum]. Windows protokolliert dazu Reason ID 1: „NTLM was called directly by the calling application.“ Die Anwendung fordert NTLM also ausdrücklich an, statt über Negotiate Kerberos zu verwenden. Die Kerberos-Voraussetzungen auf unserer Seite (SPN, DNS-Auflösung) sind geprüft.

Microsoft hat angekündigt, NTLM mit der nächsten Windows-Server-Version standardmäßig abzuschalten. Wir bitten Sie daher um Antwort auf folgende Fragen:

1. Gibt es eine Konfiguration, mit der [Produkt] über Negotiate oder Kerberos authentifiziert? 2. Falls nicht: Ab welcher Version ist das geplant, und zu welchem Termin? 3. Welche SPNs oder Dienstkonten setzt Kerberos bei Ihrer Anwendung voraus? 4. Unterstützt die Anwendung NTLMv2, falls bis zur Umstellung eine befristete Ausnahme nötig ist?

Eine Auswertung mit Prozess, Ziel und Häufigkeit liegt bei. Für Rückfragen steht [Name, Rolle, Telefon] zur Verfügung.

Mit freundlichen Grüßen

Antworten richtig einordnen

Hersteller antworten erfahrungsgemäß in einer von fünf Tonarten. Wichtig ist, dass du aus jeder Antwort eine Entscheidung ableitest und sie dokumentierst. „Wir prüfen das“ ist keine Antwort, sondern eine Wartezimmer-Nummer.

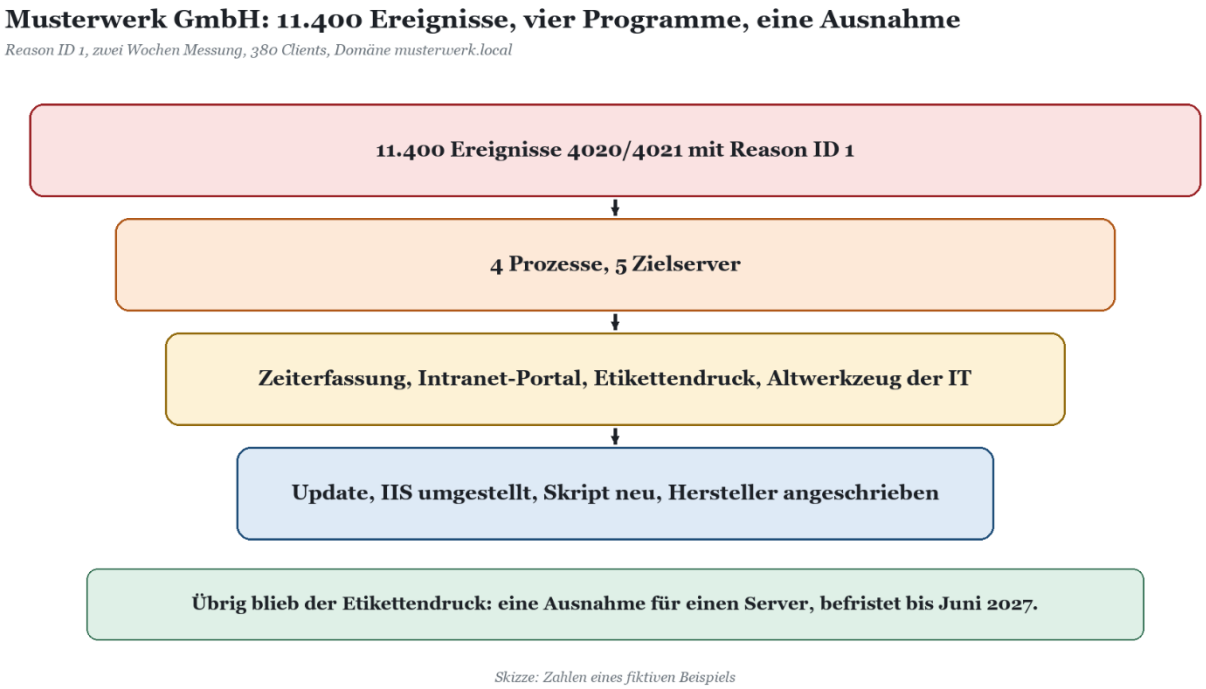
Antwort des Herstellers	Was sie bedeutet	Dein nächster Schritt
„Stellen Sie Option X	Bestes Ergebnis, lag die ganze	Testsystem umstellen, Ereignisse 4020

auf Kerberos.“	Zeit da	prüfen, ausrollen
„Ab Version 9 unterstützt.“	Lösung existiert, kostet Update und Test	Update in den Projektplan, bis dahin überwachen
„Ist geplant für Q3 2027.“	Zusage mit Termin, Risiko bleibt	Termin schriftlich, befristete Ausnahme vorbereiten
„Bei uns geht das.“	Anfrage nicht verstanden oder Ausweichen	Belege nachreichen, Eskalation über Vertrieb oder Account-Management
„Das Produkt ist abgekündigt.“	Kein Fix kommt, nie	Ausnahme eng begrenzen, Ablösung des Produkts planen

AUS DER PRAXIS · Musterwerk GmbH: vier Programme, eine Ausnahme

Bei der Musterwerk GmbH, Domäne musterwerk.local, ergab eine zweiwöchige Messung 11.400 Ereignisse mit Reason ID 1. Dahinter steckten genau vier Prozesse. Die Zeiterfassung beherrschte Kerberos seit zwei Versionen; das Update war ohnehin überfällig. Das Intranet-Portal lief auf IIS mit nur dem NTLM-Anbieter, nach der Umstellung auf Negotiate und einem SPN für den Webnamen war Ruhe. Das Altwerkzeug der IT war ein PowerShell-Skript mit fest eingetragener Anmeldeart, eine Zeile Änderung.

Blieb der Etikettendruck in der Produktion. Der Hersteller antwortete nach drei Wochen und einer freundlichen Erinnerung mit „ist für die nächste Hauptversion geplant“. Ergebnis: eine Ausnahme für genau einen Server, befristet bis Juni 2027, mit Namen des Verantwortlichen im Ausnahmeregister.



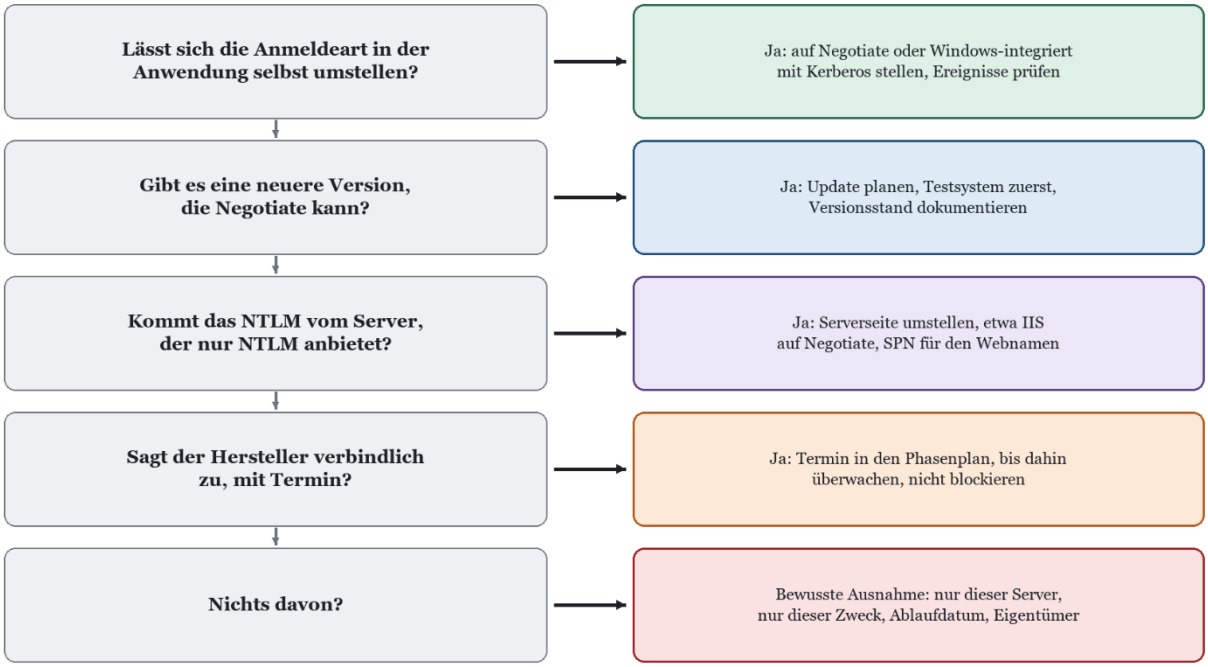
Skizze: Zahlen eines fiktiven Beispiels

Wenn nichts geht: die bewusste, begrenzte Ausnahme

Manchmal kommt keine Lösung. Der Hersteller ist insolvent, das Produkt abgekündigt, das Medizingerät zertifiziert und unantastbar, der Entwickler bei den Ziegen. Dann brauchst du eine Ausnahme. Das ist keine Niederlage, sondern Risikomanagement, solange sie drei Eigenschaften hat: Sie ist bewusst entschieden, sie ist eng begrenzt, und sie hat ein Ablaufdatum.

Reason ID 1 auf dem Tisch: in dieser Reihenfolge vorgehen

Von oben nach unten. Die Ausnahme steht ganz am Ende, nicht am Anfang.



Skizze: eigenes Vorgehensmodell

Wie eine Ausnahme technisch aussieht

Solange NTLM nur überwacht wird, brauchst du keine technische Ausnahme, sondern einen Eintrag auf deiner Liste. Ernst wird es, sobald du NTLM per Richtlinie einschränkst. Die Restrict-NTLM-Richtlinien kennen dafür Ausnahmelisten: „Netzwerksicherheit: NTLM beschränken: Serverausnahmen in dieser Domäne hinzufügen“ auf den Domain Controllern und „Netzwerksicherheit: NTLM beschränken: Remoteserverausnahmen für die NTLM-Authentifizierung hinzufügen“ auf den Clients. Dort trägst du genau den Zielserver ein, den die Anwendung braucht, nicht die halbe Serverlandschaft. Wie die Richtlinien zusammenspielen, erklärt Restrict-NTLM-Richtlinien erklärt: Überwachen, Ausnahmen, Verweigern; wie du Ausnahmen pflegst, ohne dass sie zum Dauerzustand werden, steht in Server-Ausnahmen für NTLM: sauber pflegen statt Löcher bohren.

Für die künftige Standardabschaltung hat Microsoft ebenfalls Steuerungsmöglichkeiten angekündigt. Plane trotzdem nicht mit Hoffnung, sondern mit Inventar: Wer heute weiß, welcher Server welche Ausnahme braucht, kann sie morgen in jedes neue Werkzeug übertragen. Wer es nicht weiß, sucht am Morgen nach der Umstellung, während das Telefon klingelt.

WICHTIG · Pflichtangaben für jede Ausnahme

Ziel: genau ein Server oder Dienst, keine Platzhalter für ganze Domänen.

Anwendung und Begründung: Prozess, Version, Herstellerantwort mit Datum.

Eigentümer: eine verantwortliche Person aus dem Fachbereich, nicht „die IT“.

Ablaufdatum: spätestens zur nächsten Hauptversion oder nach zwölf Monaten, mit Wiedervorlage.

Absicherung: nur NTLMv2, Kanalbindung und Signierung, wo möglich, und laufende Überwachung.

Die Absicherung ist kein Nebensatz. Eine NTLM-Ausnahme ist ein Einfallstor für Relay-Angriffe, wenn der Zielserver weder Signierung noch Kanalbindung erzwingt. Was du dagegen tun kannst, steht in NTLM-Relay verhindern: MIC, Kanalbindung und erweiterter Schutz.

WARNUNG · Die Ausnahme, die bleibt

Jede Organisation hat sie: die „vorübergehende“ Ausnahme von 2014, die keiner mehr erklären kann. Ausnahmen ohne Ablaufdatum werden zu Inventar, und Inventar wird nicht mehr hinterfragt. Wenn du beim nächsten Audit nicht erklären willst, warum der halbe Serverraum NTLM darf, setz das Datum gleich beim Anlegen.

Einbetten in den Gesamtplan

Reason-ID-1-Fälle sind die langsamsten im ganzen Projekt, weil sie von Dritten abhängen. Deshalb gehören sie an den Anfang, nicht ans Ende: Je früher die Anfrage beim Hersteller liegt, desto eher steht die neue Version im Wartungsfenster. Wie du Phasen, Rollen und Aufwand planst, beschreibt NTLM-Ablösung als Projekt: Zeitplan, Rollen, Aufwand; die Reihenfolge der technischen Schritte steht in NTLM schrittweise abschalten, ohne dass etwas ausfällt.

Wenn du Herstelleranfragen und Ausnahmen lieber mit jemandem durchgehst, der schon einige Dutzend davon geschrieben und gegengelesen hat, hilft dir die Beratung zur NTLM-Ablösung. Soll dein Team Reason-IDs und Anwendungsursachen künftig selbst auswerten, ist die Schulung zum NTLM Diagnostiker der kürzere Weg.

FAQ: Anwendungen mit fest eingebautem NTLM

Was bedeutet Reason ID 1 im NTLM-Ereignis?

Die Anwendung hat NTLM ausdrücklich angefordert, statt über Negotiate zuerst Kerberos zu versuchen. Der Meldungstext lautet „NTLM was called directly by the calling application.“ Mit SPNs oder DNS lässt sich das nicht beheben.

Kann ich eine Anwendung selbst von NTLM auf Kerberos umstellen?

Manchmal. Viele Fachanwendungen, Datenbanktreiber und Webserver haben eine Einstellung für die Anmeldeart. Ist NTLM im Programmcode fest eingetragen, hilft nur eine neue Version oder eine Codeänderung.

Warum sehe ich die Anwendung nur auf dem Server, nicht auf dem Client?

Weil sie NTLM vermutlich selbst implementiert und Windows dafür nicht fragt. Das ist bei Java- und Python-Bibliotheken häufig. Dann gibt es kein Ereignis 4020 auf dem Client, aber 4022 oder 4023 auf dem Zielserver.

Was muss in eine Herstelleranfrage?

Prozessname mit Pfad und Version, Ziel mit Dienst, Häufigkeit mit Messzeitraum und das gewünschte Verhalten: Authentifizierung über Negotiate mit Kerberos. Dazu konkrete Fragen mit Frist und der Verweis auf Microsofts Zeitplan.

Hilft IAKerb bei Reason ID 1?

Nein. IAKerb hilft Clients, die keinen Domain Controller sehen. Eine Anwendung, die NTLM ausdrücklich anfordert, fragt Kerberos gar nicht erst, egal ob mit oder ohne IAKerb.

Ist eine Ausnahme für NTLM ein Sicherheitsproblem?

Sie ist ein bewusst akzeptiertes Risiko. Begrenze sie auf genau einen Server, erzwinge NTLMv2, Signierung und Kanalbindung, wo möglich, und gib ihr ein Ablaufdatum und einen Eigentümer.

Webanwendung mit NTLM: Liegt das am Browser?

Selten. Meist bietet der Webserver nur NTLM an, und der Browser antwortet entsprechend. Stell auf dem Server Negotiate als Anbieter ein und registriere einen SPN für den Webnamen.

Schreibt der NTLM Diagnostiker die Herstelleranfrage für mich?

Heute nicht. Er liefert Prozess, Ziel und Häufigkeit als Beleg, dazu Word-Bericht und Excel- oder CSV-Export als Anlage. Vorlagen für Herstelleranfragen sind als Ausbaustufe geplant. Er arbeitet ausschließlich lesend und ändert nichts an deiner Umgebung.

Fazit: Belege schlagen Beschwerden

Reason ID 1 ist die ehrlichste aller NTLM-Ursachen. Sie sagt dir klar, dass dein Active Directory unschuldig ist und das Problem im Programm steckt. Das macht die Lösung nicht leichter, aber den Weg klarer: Erst prüfen, ob sich die Anmeldeart selbst umstellen lässt, dann den Hersteller mit Prozess, Ziel, Häufigkeit und einer konkreten Frage konfrontieren, und erst wenn alles scheitert, eine bewusste, eng begrenzte Ausnahme mit Ablaufdatum anlegen.

Fang damit früh an. Hersteller haben Releasezyklen, Kliniken Zertifizierungen, Verwaltungen Vergabeverfahren. Wer seine Anfragen heute verschickt, hat zur Standardabschaltung eine Antwort. Wer wartet, hat am ersten Morgen danach eine stillstehende Zeiterfassung und eine Belegschaft, die erstmals seit Jahren pünktlich ins Büro kommt, aber nicht stempeln kann.

WEITERLESEN · Weiter in dieser Serie

› NTLM-Reason-IDs erklärt: warum Windows nicht Kerberos benutzt hat
› NTLM in IIS und Webanwendungen: Negotiate statt NTLM
› Ereignis 4020 und 4021: ausgehende NTLM-Anmeldungen lesen
› Ereignis 4022 und 4023: eingehende NTLM-Anmeldungen auf Servern
› Restrict-NTLM-Richtlinien erklärt: Überwachen, Ausnahmen, Verweigern
› Server-Ausnahmen für NTLM: sauber pflegen statt Löcher bohren
› Dienstkonten als NTLM-Quelle: finden, prüfen, umstellen
› NTLM im Netzwerkmitschnitt finden: Wireshark richtig einsetzen
› NTLM in der Klinik: Medizintechnik, Labor und Fachanwendungen
› NTLM-Abschaltung: Microsofts Fahrplan und was er für dich bedeutet