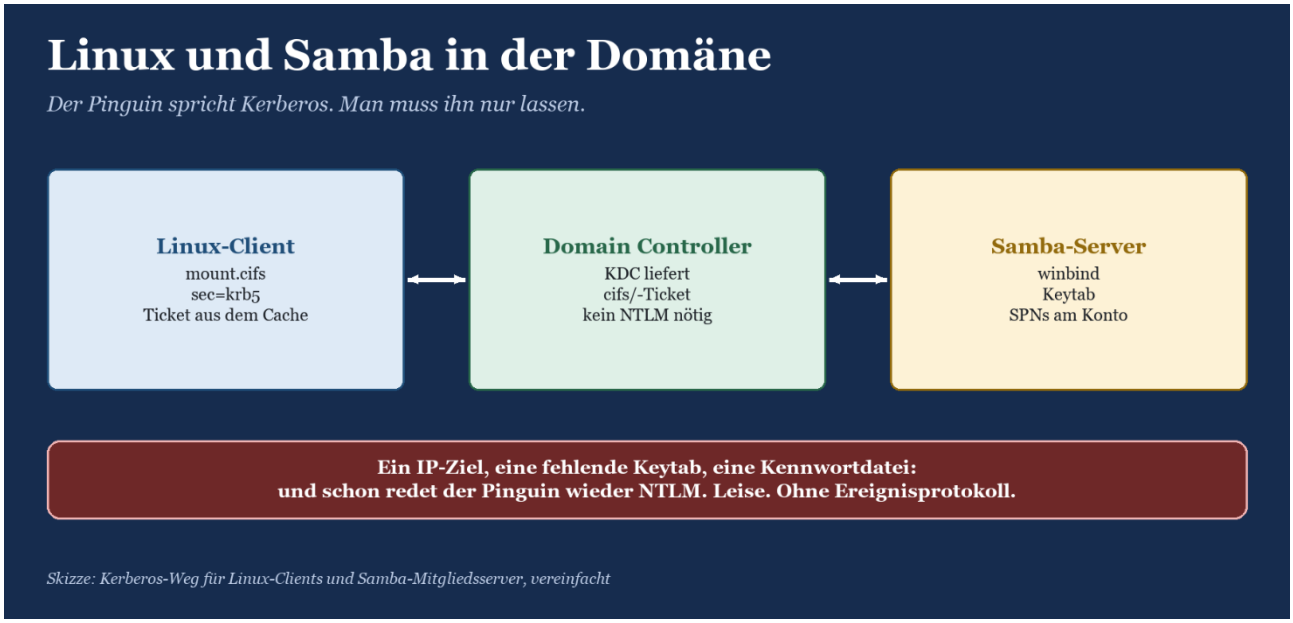
Linux und Samba in der Domäne: Kerberos statt NTLM



WISSEN
Was der NTLM Diagnostiker erfasst, wie er arbeitet und was er dir zeigt.
› **NTLM Diagnostiker**

BERATUNG
Linux, Samba und Windows unter einem Kerberos-Dach? Wir planen das mit dir.
› **Beratung zur NTLM-Ablösung**

SCHULUNG
Ereignisse lesen, Verursacher finden, Ausnahmen begründen: praxisnah für dein Team.
› **Schulung zum NTLM Diagnostiker**

Irgendwo in deinem Netz steht ein Linux-Server, den alle „den Pinguin“ nennen. Er hat die Fachanwendung überlebt, für die er gebaut wurde, zwei Admins und einen Serverraumumzug. Er hängt per fstab die Freigabe der Buchhaltung ein, mit einer Kennwortdatei, die 2019 jemand angelegt hat und die seitdem niemand mehr angefasst hat. Er spricht NTLM, er spricht es jeden Tag, und in deiner Ereignisanzeige taucht er trotzdem nirgends als Verursacher auf. Nicht, weil er brav wäre, sondern weil er gar kein Windows-Ereignisprotokoll hat.

Genau darum geht es in diesem Beitrag zum Thema samba ntlm kerberos: wie du Linux-Clients und Samba-Mitgliedsserver in der Active-Directory-Domäne sauber auf Kerberos bringst, warum sie im Windows-Audit nur von der Gegenseite sichtbar sind, und welche Konfigurationen sie still auf NTLM zurückwerfen. Wir schauen auf mount.cifs mit sec=krb5, auf Samba mit winbind, auf SSSD, auf Keytabs und auf die Klassiker unter den Rückfällen: IP-Ziele und fehlende Keytabs.

Der Zeitdruck ist bekannt: Seit dem Windows-Update vom 13.10.2026 blockieren Windows 11 ab 24H2 und Windows Server 2025 NTLMv1 für Single Sign-on standardmäßig, und mit der nächsten Windows-Server-Version soll NTLM standardmäßig abgeschaltet sein. Den ganzen Fahrplan findest du in NTLM-Abschaltung: Microsofts Fahrplan und was er für dich bedeutet. Für Linux heißt das: Was heute über NTLM an einen Windows-Server geht, geht morgen womöglich gar nicht mehr. Und der Pinguin wird es dir nicht vorher sagen.

FAKTEN · Das Wichtigste in Kürze

Linux-Systeme schreiben keine Windows-Ereignisse. NTLM von Linux siehst du nur auf dem Windows-Zielserver (4022/4023, 8002/8003) und auf dem Domain Controller (4030 bis 4033,

8004, 4776).

Ein Samba-Mitgliedsserver erzeugt keine eingehenden Windows-Ereignisse. Zugriffe darauf siehst du auf den Windows-Clients (4020/4021, 8001) und als Durchreichung auf dem Domain Controller.

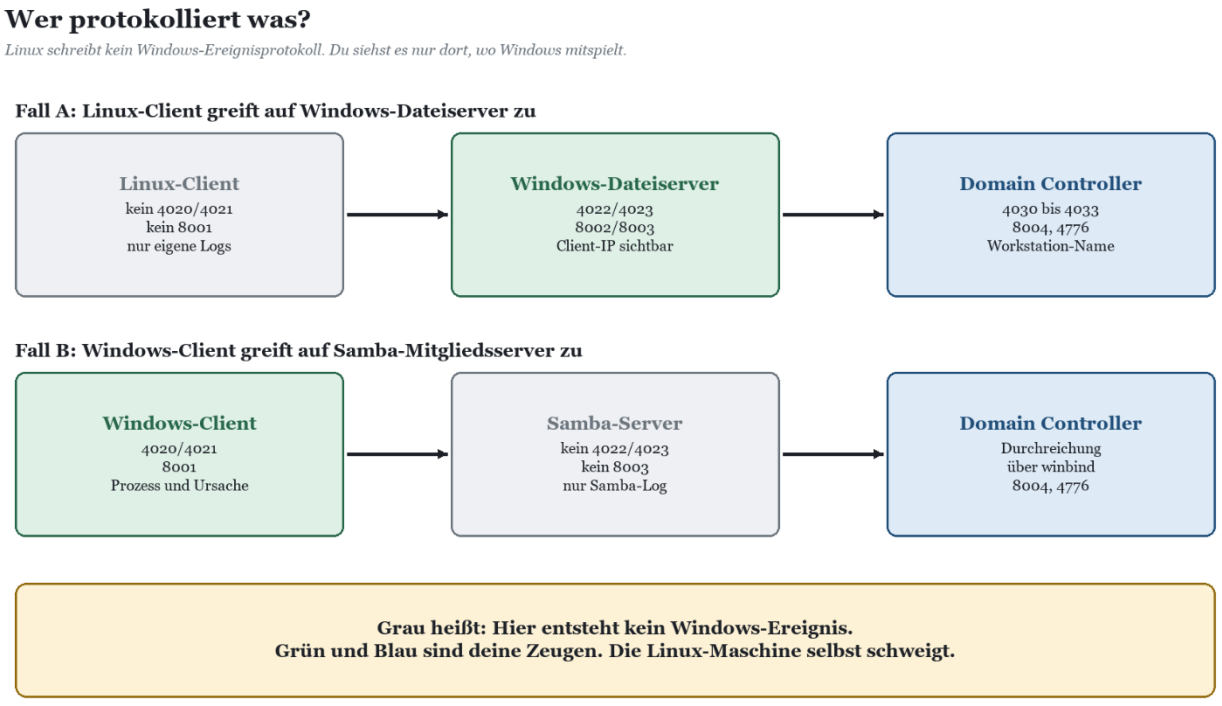
mount.cifs mit sec=krb5 holt sich über cifs.upcall ein Ticket für cifs/<servername>. Ohne Namen, also mit IP-Adresse im Pfad, klappt das nicht.

Samba als Mitgliedsserver braucht winbind und ein funktionierendes Maschinenkonto mit passender Keytab beziehungsweise passendem Maschinenkennwort.

Die häufigsten Rückfälle: IP-Ziele, Kennwortdateien statt Keytab, veraltete Keytabs nach Kennwortwechsel, Aliasnamen ohne SPN und Systeme, die nie in die Domäne aufgenommen wurden.

Warum du Linux im Windows-Audit nur von der Gegenseite siehst

Das NTLM-Auditing von Windows ist eine sehr ehrliche Einrichtung, aber sie hat einen blinden Fleck: Sie protokolliert nur dort, wo Windows-Code läuft. Die ausgehenden Ereignisse 4020 und 4021 (erweitertes Auditing) beziehungsweise 8001 (klassische Überwachung) schreibt der Windows-Client, der NTLM benutzt. Ein Linux-Client hat keinen Windows-Client-Code, also auch kein 4020, kein ProcessName, keine NtlmUsageReason. Die schönste Information, nämlich „welcher Prozess hat warum NTLM gewählt“, gibt es für Linux schlicht nicht. Wie das erweiterte Auditing grundsätzlich funktioniert und wie du es einschaltest, steht in Erweitertes NTLM-Auditing einschalten: Schritt für Schritt.



Skizze: Ereignisse laut Microsoft-Dokumentation zum NTLM-Auditing; Fall C (Linux zu Samba) sieht nur der DC

Was bleibt, ist die Gegenseite. Greift ein Linux-Client per SMB auf einen Windows-Dateiserver zu, protokolliert der Dateiserver die eingehende NTLM-Anmeldung: mit erweitertem Auditing als 4022 oder 4023, klassisch als 8002 und 8003. Im 8003 steht im Feld „Workstation“ der anfragende Client, und zwar das, was der Linux-Client als Arbeitsstationsnamen mitschickt. Das kann der Hostname sein, ein kurzer NetBIOS-Name, ein Name aus der Samba-Konfiguration oder gar nichts. Bei 4022/4023 helfen dir RemoteClientMachine und ClientIP weiter. Leere Werte stehen dort als „Null“ oder „(NULL)“. Auf dem Domain Controller landet die Prüfung des Kontos

zusätzlich in 4030 bis 4033, 8004 oder im Sicherheitsereignis 4776, wie in Ereignis 4776 auf dem Domain Controller: NTLM-Anmeldungen im Sicherheitsprotokoll beschrieben.

Szenario	Linux-Seite	Windows-Gegenseite	Was dir fehlt
Linux-Client auf Windows-Dateiserver	keine Windows-Ereignisse	Server: 4022/4023, 8002/8003; DC: 4030 bis 4033, 8004, 4776	Prozess und Grund auf dem Client
Windows-Client auf Samba-Server	Samba-Log, sofern konfiguriert	Client: 4020/4021, 8001 mit TargetMachine = Samba-Server; DC: Durchreichung	eingehende Sicht am Ziel
Linux-Client auf Samba-Server	Samba-Log, sofern konfiguriert	nur DC, wenn NTLM per winbind durchgereicht wird	fast alles
Linux-Werkzeug per LDAP am DC	keine Windows-Ereignisse	DC: NTLM-Ereignisse, bei unsignierten Bindungen 2889	Anwendung und Konfiguration

WICHTIG · Auswerten über die Felder, nicht über den Meldungstext

Sammelst du die Ereignisse per Windows Event Forwarding, kann der Meldungstext weitergeleiteter Ereignisse leer sein. Dann zählen nur die EventData-Felder. Filtere deshalb auf ClientIP, RemoteClientMachine, TargetMachine und NtlmVersion, nicht auf Textfragmente. Rezepte dafür findest du in NTLM-Ereignisse mit PowerShell auswerten: Get-WinEvent-Rezepte.

Die IP-Adresse ist dein einziger fester Anker

Weil der Arbeitsstationsname vom Client kommt und der Client lügen, schweigen oder einfach „localhost“ schicken kann, ist die IP-Adresse aus ClientIP die verlässlichste Spur. Die löst du per DNS rückwärts auf, gleichst sie mit DHCP und Inventar ab, und dann gehst du mit ss oder smbstatus auf dem Linux-System nachsehen, welche Verbindung wirklich offen ist. Auf einem Linux-Client zeigt dir außerdem die Datei /proc/fs/cifs/DebugData die eingehängten SMB-Sitzungen. Das ist Handarbeit, aber ehrliche.

```
# Windows-Dateiserver: eingehende NTLM-Anmeldungen der letzten 7 Tage nach Client-IP zählen
Get-WinEvent -FilterHashtable @{ LogName = 'Microsoft-Windows-NTLM/Operational';
    Id = 4022,4023; StartTime = (Get-Date).AddDays(-7) } |
  ForEach-Object { $x = [xml]$_.ToXml(); $f = @{}
    $x.Event.EventData.Data | ForEach-Object { $f[$_.Name] = $_.'#text' }
    [pscustomobject]@{ IP = $f['ClientIP']; Client = $f['RemoteClientMachine'];
                       Version = $f['NtlmVersion'] } } |
  Group-Object IP, Client, Version | Sort-Object Count -Descending | Select-Object -First 20

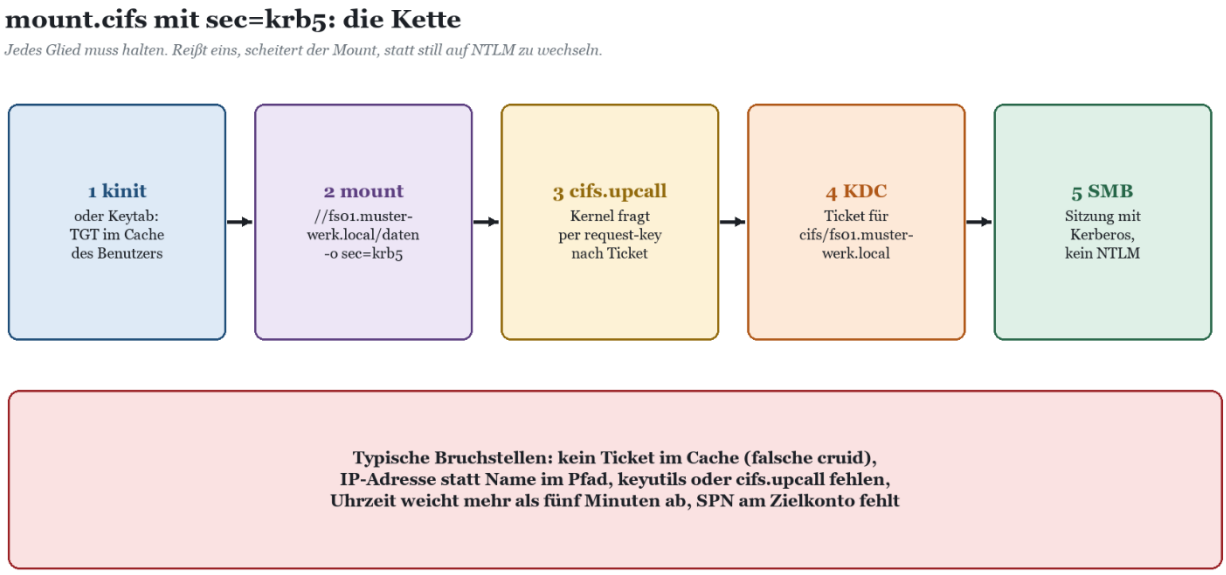
# Linux-Client: welche SMB-Sitzungen sind offen, mit welchem Verfahren?
cat /proc/fs/cifs/DebugData
mount -t cifs
```

Prüfe den Namen des Protokolls und der Felder einmal an einem echten Ereignis auf deinem Server, bevor du das Skript breit verteilst. Und: Die erweiterten Ereignisse gibt es ab Windows 11 24H2 und Windows Server 2025 sowie auf Server 2022 und 2019 mit den Updates ab dem 13.10.2026. Auf älteren Ständen bleibt dir 8002 und 8003; das Einrichten nimmt dir das NTLM-Audit-GPO-Skript ab.

SMB von Linux: mount.cifs mit sec=krb5

Der Linux-Kernel bringt einen eigenen SMB-Client mit, eingebunden über mount.cifs aus dem Paket cifs-utils. Ohne Angabe verhandelt er die Anmeldung per NTLMSSP mit Benutzername und Kennwort, gern aus einer Datei, die über die Option credentials= eingebunden wird. Das ist der

bequeme Weg, und er ist NTLM. Der Kerberos-Weg heißt sec=krb5. Dann fragt der Kernel über den Hilfsdienst cifs.upcall nach einem Kerberos-Ticket für den Dienst cifs/<servername>, das aus dem Ticket-Cache eines Benutzers oder aus einer Keytab stammt.



Skizze: Ablauf beispielhaft für die Musterwerk GmbH, vereinfacht

```
# Voraussetzungen (Paketnamen je nach Distribution): cifs-utils, keyutils, krb5-user bzw. krb5-
workstation
kinit hans.muster@MUSTERWERK.LOCAL
klist                                   # TGT vorhanden?

sudo mount -t cifs //fs01.musterwerk.local/daten /mnt/daten \
     -o sec=krb5,cruid=$(id -u hans.muster),uid=hans.muster,vers=3.1.1

klist                                   # jetzt sollte cifs/fs01.musterwerk.local auftauchen

# /etc/fstab, mehrere Benutzer mit eigenem Ticket
//fs01.musterwerk.local/daten  /mnt/daten  cifs  sec=krb5,multiuser,_netdev,vers=3.1.1  0 0
```

Zwei Optionen verdienen einen zweiten Blick. cruid= sagt cifs.upcall, in wessen Ticket-Cache es suchen soll. Ohne die Angabe sucht es beim Benutzer, der den Mount ausführt, und das ist bei sudo eben root, der meist kein Ticket hat. Und multiuser sorgt dafür, dass jeder Benutzer, der auf den Mountpunkt zugreift, mit seinem eigenen Ticket am Server erscheint, statt alle unter einer Identität. Für Rechte auf dem Windows-Server ist das genau das, was du willst.

TIPP · sec=krb5 ist ehrlich, und das ist gut so

Mit sec=krb5 fällt mount.cifs nicht still auf NTLM zurück. Gibt es kein Ticket, scheitert der Mount mit einer Fehlermeldung, meist „Required key not available“ oder einem Zugriffsfehler. Das ist lästig, aber es ist das Verhalten, das du für die NTLM-Ablösung brauchst: lieber ein lauter Fehler heute als eine leise NTLM-Verbindung, die am Stichtag bricht.

Der umgekehrte Effekt ist die eigentliche Gefahr: Wer bei einem Fehler einfach sec=krb5 aus der fstab löscht, bekommt sofort einen funktionierenden Mount. Per NTLM.

Mounts ohne Benutzer: Maschinenkonto und Keytab

Viele Linux-Mounts gehören keinem Menschen, sondern einem Dienst: das Backup-Ziel, der Export der Fachanwendung, der nächtliche Datenabgleich. Hier ist die Versuchung am größten, eine Kennwortdatei zu hinterlegen. Die saubere Alternative ist das Maschinenkonto: Ist der Linux-Rechner der Domäne beigetreten, liegt sein Schlüssel in /etc/krb5.keytab. cifs.upcall kann bei Root-Mounts ohne Benutzer-Ticket auf diese Keytab zurückgreifen und sich als Computerkonto anmelden. Auf dem Windows-Server berechtigst du dann das Computerkonto, etwa MUSTERWERK\LNXAPP01$, oder eine Gruppe, in der es Mitglied ist.

Braucht der Dienst eine eigene Identität, legst du für ein Dienstkonto eine eigene Keytab an und holst das Ticket per kinit -k -t vor dem Mount, etwa über einen systemd-Timer. Wie du Dienstkonten grundsätzlich findest und umstellst, steht in Dienstkonten als NTLM-Quelle: finden, prüfen, umstellen.

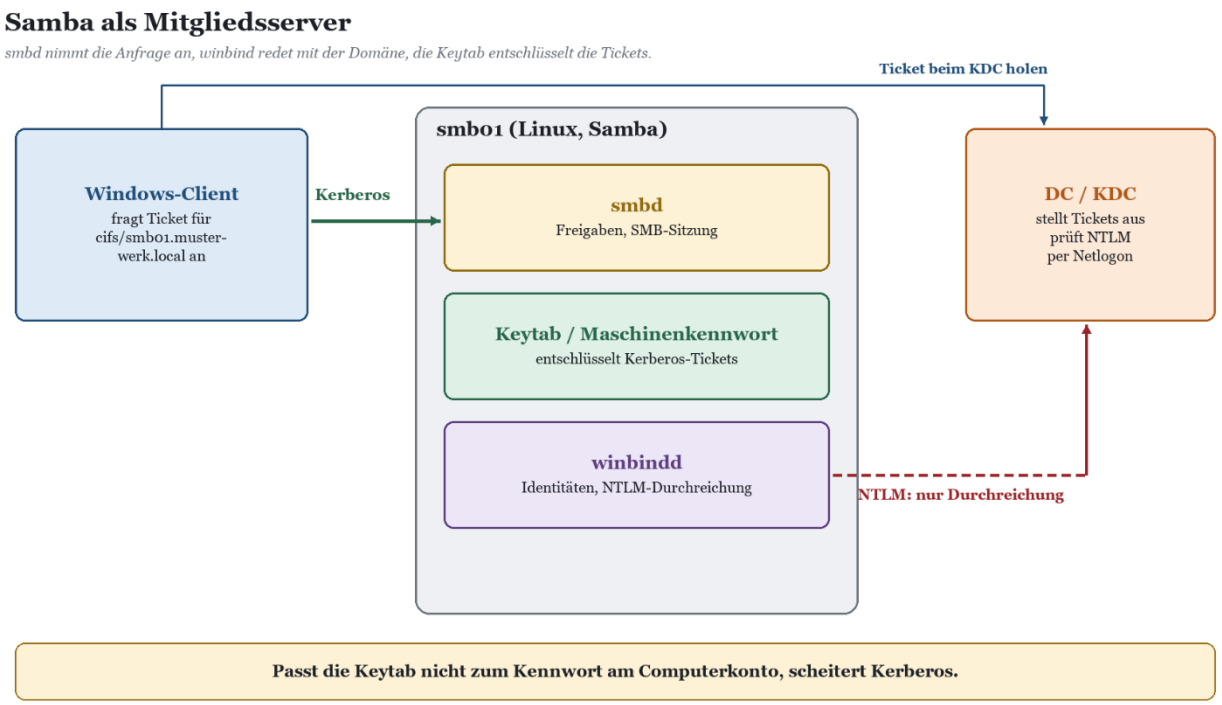
```
sudo klist -ke /etc/krb5.keytab          # welche Prinzipale und Schlüsseltypen stehen drin?
sudo kinit -k 'LNXAPP01$@MUSTERWERK.LOCAL'  # funktioniert die Keytab überhaupt?
sudo klist
```

Variante	Verfahren	Bewertung
credentials=/etc/cifs-cred, ohne sec	NTLMSSP mit gespeichertem Kennwort	NTLM, Kennwort im Klartext auf der Platte; bricht mit der Abschaltung
sec=ntlmssp ausdrücklich gesetzt	NTLMSSP	dasselbe, nur mit Absicht; Kandidat für die Maßnahmenliste
sec=krb5 mit Benutzer-Ticket (cruid)	Kerberos	gut für interaktive Benutzer
sec=krb5,multiuser	Kerberos je Benutzer	gut für Mehrbenutzersysteme und Terminalserver
sec=krb5 mit Maschinen-Keytab	Kerberos als Computerkonto	gut für Dienste ohne Benutzer
sec=krb5i	Kerberos mit Paketsignierung	wie krb5, zusätzlich signiert

Ob Kerberos wirklich greift, siehst du auf Linux mit klist: Nach dem ersten Zugriff muss ein Ticket für cifs/fs01.musterwerk.local im Cache stehen. Auf Windows-Seite ist die Gegenprobe, dass für diese Client-IP keine neuen 4022/4023 oder 8003 mehr auftauchen. Die Windows-Variante dieser Prüfung beschreibt Mit klist prüfen, ob Kerberos wirklich greift.

Samba als Mitgliedsserver: winbind, SSSD und die Keytab

Jetzt drehen wir die Richtung um. Ein Samba-Server, der Freigaben für Windows-Clients anbietet, ist im Active Directory ein Computerkonto wie jedes andere. Er braucht SPNs wie HOST/smb01.musterwerk.local und cifs/smb01.musterwerk.local (beziehungsweise HOST, das cifs abdeckt), einen Schlüssel, mit dem er die Tickets der Windows-Clients entschlüsseln kann, und winbind, um Identitäten aufzulösen. Seit Samba 4.8 setzt ein Mitgliedsserver mit security = ADS einen laufenden winbindd voraus.



Skizze: Samba-Mitgliedsserver mit security = ADS, vereinfacht

```
# /etc/samba/smb.conf, Auszug für smb01.musterwerk.local
[global]
    security = ADS
    workgroup = MUSTERWERK
    realm = MUSTERWERK.LOCAL
    kerberos method = secrets and keytab
    client use kerberos = required      # Samba-eigene Clientwerkzeuge ohne NTLM-Rückfall
    idmap config * : backend = tdb
    idmap config * : range = 3000-7999
    idmap config MUSTERWERK : backend = rid
    idmap config MUSTERWERK : range = 10000-999999

# Beitritt und Prüfung
sudo net ads join -U administrator
sudo net ads testjoin
sudo klist -ke /etc/krb5.keytab
wbinfo -t                           # Vertrauensstellung des Maschinenkontos prüfen
```

Mit kerberos method = secrets and keytab pflegt Samba neben seiner internen Datenbank auch die System-Keytab, sodass andere Dienste auf dem Server, etwa SSH mit GSSAPI, denselben Schlüssel nutzen können. Der Parameter client use kerberos betrifft nur Samba-eigene Clientwerkzeuge wie smbclient oder net, also ausgehende Verbindungen. Ob Windows-Clients mit Kerberos auf den Samba-Server kommen, entscheidet sich auf der Windows-Seite: an Name, SPN und gültigem Ticket. Prüfe die genaue Wirkung der Parameter in der Dokumentation deiner Samba-Version, denn Namen und Standardwerte haben sich über die Versionen verändert.

WARNUNG · „ntlm auth“ schaltet NTLM für Domänenkonten nicht ab

Der Samba-Parameter ntlm auth regelt, ob Samba NTLM gegen seine eigene lokale Kontendatenbank akzeptiert. Auf einem Mitgliedsserver prüft aber nicht Samba die Domänenkonten, sondern der Domain Controller, an den winbind die NTLM-Anfrage durchreicht.

Ob NTLM für Domänenkonten erlaubt ist, entscheiden also die Windows-Richtlinien am DC, nicht die smb.conf. Die Details stehen in Restrict-NTLM-Richtlinien erklärt: Überwachen, Ausnahmen, Verweigern.

SSSD oder winbind?

Auf Linux-Clients und Servern ohne Freigaben hat sich SSSD durchgesetzt, meist eingerichtet über realm join. SSSD erledigt Anmeldung und Identitäten über Kerberos und LDAP und benutzt dafür selbst kein NTLM. Für einen Samba-Dateiserver ist winbind der vorgesehene Weg. Beide auf demselben System gleichzeitig für dieselbe Domäne zu betreiben, ist eine bekannte Fehlerquelle, denn beide wollen das Maschinenkennwort verwalten.

Kriterium	SSSD (z. B. per realm join)	winbind (Samba)
Typischer Einsatz	Linux-Clients, Anwendungsserver, SSH-Anmeldung	Samba-Dateiserver, Druckserver
Anmeldung von Benutzern	Kerberos	Kerberos, NTLM-Durchreichung möglich
Keytab	/etc/krb5.keytab, gepflegt von adcli/SSSD	Samba-Datenbank und optional /etc/krb5.keytab
Kennwortwechsel des Computerkontos	durch SSSD/adcli	durch winbind
NTLM-Risiko	gering; meist nur bei Zusatzwerkzeugen	Durchreichung, wenn Clients kein Ticket haben
Typischer Stolperstein	Samba zusätzlich installiert, Kennwort zweimal verwaltet	Keytab veraltet, SPN für Alias fehlt

AUS DER PRAXIS · Der Pinguin der Musterwerk GmbH

lnxapp01 wurde per realm join mit SSSD in musterwerk.local aufgenommen. Später installierte jemand Samba, um eine Exportfreigabe anzubieten, und trat mit net ads join noch einmal bei. Beide Werkzeuge verwalteten fortan das Maschinenkennwort. Nach dreißig Tagen rotierte SSSD das Kennwort, Samba wusste davon nichts. Ergebnis: Die Windows-Clients bekamen weiterhin Tickets für cifs/lnxapp01.musterwerk.local, die der Server mit seinem alten Schlüssel nicht mehr lesen konnte. Zugriffe per Name scheiterten, der „Workaround“ des Kollegen war der Pfad über die IP-Adresse. Seitdem lief alles. Über NTLM.

Keytab prüfen: kvno, Schlüsseltypen, SPNs

Eine Keytab ist nur so gut wie ihr Abgleich mit dem Computerkonto. Drei Dinge müssen passen: Die Schlüsselversionsnummer (kvno) in der Keytab muss zur aktuellen Version am Konto passen, die Schlüsseltypen müssen zu dem passen, was der DC ausstellt (heute AES), und das Konto muss die SPNs tragen, unter denen Clients den Server ansprechen. Die Windows-Seite prüfst du mit Bordmitteln:

```
# Windows: SPNs und Verschlüsselungstypen des Samba-Computerkontos
setspn -L SMB01
Get-ADComputer SMB01 -Properties msDS-KeyVersionNumber, msDS-SupportedEncryptionTypes,
servicePrincipalName

# Linux: kvno in der Keytab und aktuelle kvno am KDC vergleichen
sudo klist -ke /etc/krb5.keytab
kvno cifs/smb01.musterwerk.local
```

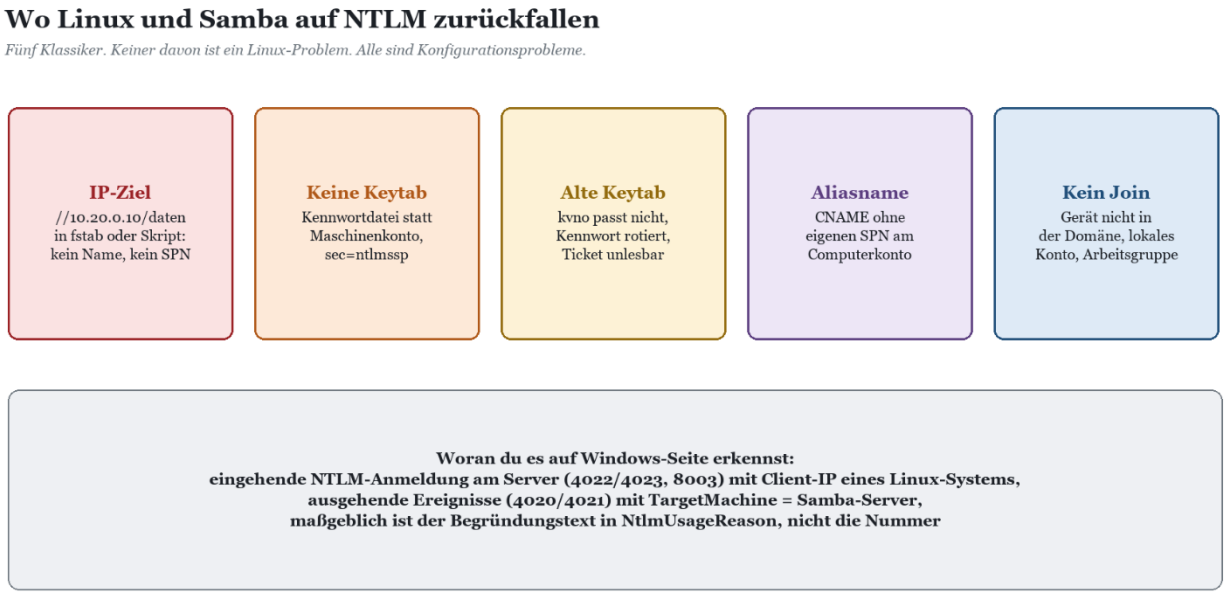
Sprechen Clients den Samba-Server unter einem Aliasnamen an, etwa daten.musterwerk.local als CNAME, braucht das Computerkonto einen SPN für genau diesen Namen, und die Keytab muss danach neu erzeugt werden. Fehlt der SPN, kann der Windows-Client kein Ticket holen und fällt

auf NTLM zurück. Mehr dazu in CNAME, Aliasnamen und hosts-Datei: warum Kerberos den Namen nicht findet und allgemein in Reason-ID 6: wenn der Zielname nicht auflösbar ist.

```
setspn -S cifs/daten.musterwerk.local MUSTERWERK\SMB01$
# danach auf smb01: Keytab neu erzeugen und mit klist -ke prüfen
```

Typische NTLM-Rückfälle und wie du sie findest

Linux ist in Sachen Kerberos kein Problemkind. Die Werkzeuge können das seit vielen Jahren. Die Rückfälle entstehen fast immer durch Konfiguration, durch Bequemlichkeit oder durch einen Workaround, der irgendwann zur Dauerlösung wurde.



Skizze: typische Rückfallursachen, beispielhaft für die Musterwerk GmbH

Rückfall	Woran du ihn auf Windows-Seite erkennst	Lösung
IP-Adresse im UNC-Pfad (fstab, Skript, Anwendung)	4020/4021 auf Windows-Clients oder 4022/4023 auf dem Server; TargetMachine oder Ziel ist eine IP-Adresse	FQDN statt IP in fstab, Skripten und Anwendungskonfiguration
Kennwortdatei statt Keytab	regelmäßige 8003 bzw. 4022/4023 mit immer demselben Konto, oft nachts	Maschinenkonto oder Dienstkonto mit Keytab, sec=krb5
Veraltete Keytab auf Samba-Server	Windows-Clients greifen per IP zu oder melden Zugriffsfehler per Name	Kennwortverwaltung auf ein Werkzeug reduzieren, Keytab neu erzeugen
Aliasname ohne SPN	4020/4021 mit TargetMachine = Aliasname, Begründungstext zu nicht auflösbarem Ziel	SPN für den Alias am Computerkonto, Keytab erneuern
Linux-System nicht in der Domäne	Workstation-Feld leer oder lokal, Konto aus Domäne oder lokal	Domänenbeitritt oder Systemersatz; siehe Arbeitsgruppen-Beitrag
Uhrzeit weicht ab	Kerberos scheitert, Anwendung fällt auf NTLM zurück, wo sie es darf	chrony oder systemd-timesyncd auf die Domain Controller zeigen lassen

Bei Windows-Clients, die auf einen Samba-Server zugreifen, liefert das erweiterte Auditing die Ursache gleich mit, im Feld NtlmUsageReason. Maßgeblich ist der Begründungstext, die Nummer im Feld NtlmUsageId ist nur ein Hinweis: Microsoft dokumentiert in KB5064479 die Reason-IDs 0

bis 11, in der Praxis tauchen aber weitere Werte auf, und Nummer und Text passen nicht immer zur Liste. Mehr dazu in NTLM-Reason-IDs erklärt: warum Windows nicht Kerberos benutzt hat. Taucht ein IP-Ziel auf, lohnt sich der Blick in Gruppenrichtlinien und Anmeldeskripte, denn Laufwerkszuordnungen auf \\10.20.0.10\daten sind auch auf Windows-Seite ein Klassiker, siehe Laufwerkszuordnungen und Anmeldeskripte: NTLM-Fallen per GPO finden.

WARNUNG · Nicht vergessen: Linux spricht auch LDAP

Viele Linux-Anwendungen sprechen den Domain Controller direkt per LDAP an, mit einfacher Bindung oder SASL ohne Signierung. Das ist kein SMB, taucht aber genauso in deiner Härtungsliste auf. SSSD signiert seine LDAP-Verbindungen über GSSAPI. Ältere Anwendungen und Skripte tun das oft nicht.

Wie du solche Clients über die Ereignisse 2889 und 3039 findest, steht in LDAP-Signierung und Kanalbindung: Ereignisse 2889 und 3039 auswerten. Für Java-basierte Anwendungen auf Linux-Servern hilft Java-Anwendungen und NTLM: Kerberos mit SPNEGO.

Wo Bordmittel an ihre Grenzen kommen

Mit zwei Linux-Servern und einem Windows-Dateiserver ist das alles an einem Nachmittag erledigt. Realistisch sieht die Lage im Mittelstand, in der Klinik oder in der Verwaltung anders aus: ein paar Dutzend Linux-Systeme, verteilt auf Fachabteilungen und Dienstleister, dazu Appliances, die intern Linux mit Samba-Client fahren und sich nie als Linux zu erkennen geben, und Laborgeräte, die nicht einmal in der Domäne sind. Die Windows-Ereignisse der Gegenseite verteilen sich auf Dutzende Server. Die Linux-Systeme mit Computerkonto tauchen im Active Directory auf, liefern aber naturgemäß nie selbst ein Ereignis. Und Zugriffe von Linux auf Samba siehst du in Windows überhaupt nicht.

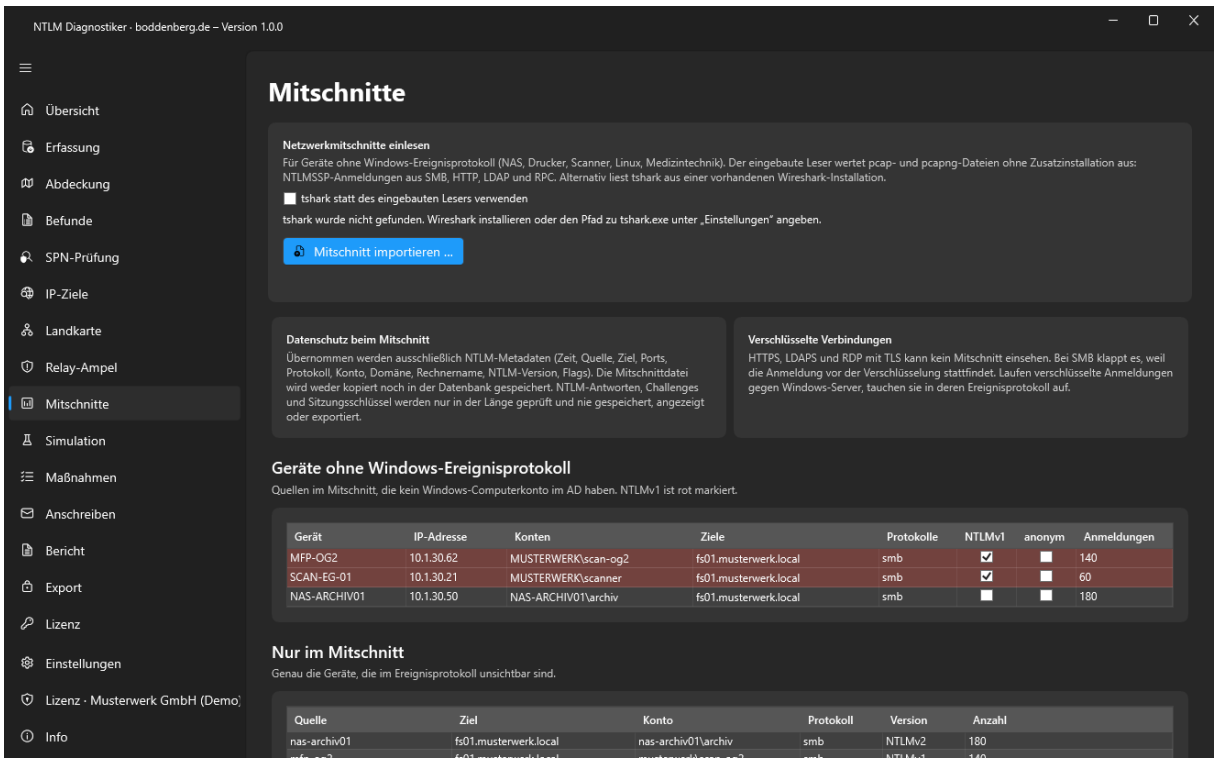
An dieser Stelle hilft ein Werkzeug, das beide Lücken ausdrücklich benennt, statt sie stillschweigend zu übergehen. Der NTLM Diagnostiker gleicht in seiner Abdeckungskarte aktive Computerkonten mit den Rechnern ab, die tatsächlich Ereignisse liefern. Ein Linux-Server mit Computerkonto erscheint dort folgerichtig als „schweigt trotz aktivem Konto“. Das ist kein Fehler, sondern der Hinweis, dass du ihn von der Gegenseite oder per Mitschnitt betrachten musst. Für diesen zweiten Weg gibt es den Mitschnitt-Import.

SO GEHT'S MIT DEM NTLM DIAGNOSTIKER · Geräte ohne Ereignisprotokoll sichtbar machen

Für Systeme, die kein Windows-Ereignisprotokoll schreiben, importierst du im NTLM Diagnostiker einen Netzwerkmitschnitt im Format pcap oder pcapng, zum Beispiel aufgenommen mit tcpdump auf dem Linux-Server selbst oder an einem Spiegelport vor dem Samba-Server. Der Diagnostiker liest daraus die NTLM-Anmeldungen heraus.

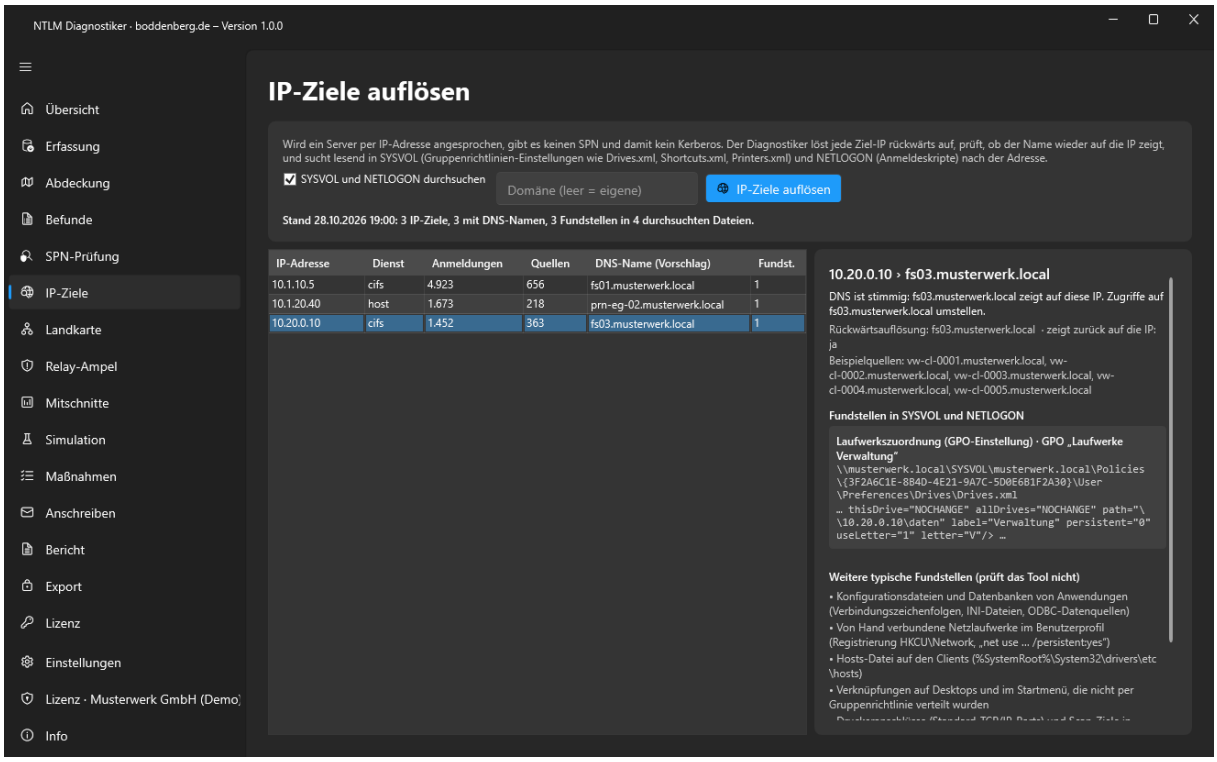
Im Bereich Mitschnitte listet der Abschnitt „Geräte ohne Windows-Ereignisprotokoll“ jedes Gerät mit IP-Adresse, verwendeten Konten, Zielen und Protokollen, mit einer Kennzeichnung für NTLMv1 und anonyme Anmeldungen sowie der Anzahl der Anmeldungen. NTLMv1-Zeilen sind rot markiert. Darunter zeigt „Nur im Mitschnitt“ die Verbindungen, die in keinem Windows-Ereignis auftauchen, mit Quelle, Ziel, Konto, Protokoll und NTLM-Version: genau die Zugriffe von Linux auf Samba, die dir sonst entgehen.

Wie alles im Diagnostiker geschieht das ausschließlich lesend. Der Diagnostiker ändert weder an Linux noch an Windows etwas; welche Mounts du umstellst, entscheidest und erledigst du.



So siehst du Geräte aus dem Netzwerkmitschnitt, die kein Windows-Ereignisprotokoll schreiben, mit Konten, Zielen und rot markiertem NTLMv1 (Demodaten der Musterwerk GmbH).

Die IP-Ziele aus den Windows-Ereignissen der Gegenseite, also Zugriffe auf \\10.20.0.10\daten statt auf einen Namen, ordnet der Diagnostiker ebenfalls ein: mit Rückwärtsauflösung und den Fundstellen in Gruppenrichtlinien und Anmeldeskripten in SYSVOL und NETLOGON. Damit trennst du schnell, ob ein IP-Ziel aus einer Windows-Laufwerkszuordnung stammt oder von einem Linux-System, das du dann per Mitschnitt oder direkt in der fstab suchst.



So findest du zu einem IP-Ziel die Rückwärtsauflösung und die Gruppenrichtlinie, die den Server per IP-Adresse einbindet (Demodaten der Musterwerk GmbH).

Die kostenlose Express-Edition liefert dir den Fünf-Minuten-Befund direkt vom DC mit Readiness-Ampel, Abdeckungskarte und Top-10-Verursachern. Den Mitschnitt-Import und die übrigen Funktionen kannst du sieben Tage lang je Domäne testen. Wenn du Linux-, Samba- und Windows-Seite in einem Projekt zusammenführen willst, unterstützt dich unsere Beratung zur NTLM-Ablösung.

Ausnahmen für Linux und Samba sauber begründen

Nicht jedes System lässt sich bis zum Stichtag umstellen. Eine Appliance mit festem Samba-Client, ein Laborrechner ohne Domänenbeitritt, ein Altsystem, dessen Hersteller nicht mehr antwortet: Für solche Fälle brauchst du Ausnahmen in den Restrict-NTLM-Richtlinien. Jede Ausnahme braucht einen Grund, einen Verantwortlichen und ein Ablaufdatum. Sonst ist sie in drei Jahren noch da und keiner weiß, warum. Für Geräte außerhalb der Domäne lies Rechner ohne Domäne: Arbeitsgruppen, Laborgeräte und NTLM, für Geräte ohne Windows generell NAS, Drucker und Scanner: NTLM bei Geräten ohne Windows.

TIPP · Checkliste für jeden Linux- und Samba-Rechner

1. Ist das System in der Domäne, und verwaltet genau ein Werkzeug das Maschinenkennwort?
2. Enthält /etc/krb5.keytab aktuelle AES-Schlüssel mit der richtigen kvno?
3. Stehen alle Namen, unter denen das System angesprochen wird, als SPN am Computerkonto?
4. Verwenden alle cifs-Mounts in fstab, systemd-Units und Skripten einen FQDN und sec=krb5?
5. Gibt es noch credentials=-Dateien mit Kennwörtern? Wenn ja: wofür?
6. Läuft die Zeitsynchronisierung gegen die Domain Controller?
7. Zeigt die Windows-Gegenseite für die IP-Adresse dieses Systems noch NTLM-Anmeldungen?

FAQ zu Samba, NTLM und Kerberos unter Linux

Warum taucht mein Linux-Server nicht als NTLM-Verursacher in der Ereignisanzeige auf?

Weil Linux keine Windows-Ereignisse schreibt. Die ausgehenden Ereignisse 4020/4021 und 8001 entstehen nur auf Windows-Clients. Du siehst den Linux-Server nur auf der Gegenseite: auf dem Windows-Zielserver und auf dem Domain Controller.

Fällt mount.cifs mit sec=krb5 auf NTLM zurück, wenn Kerberos nicht klappt?

Nein. Mit sec=krb5 scheitert der Mount, wenn kein Ticket zu bekommen ist. Zu NTLM kommt es erst, wenn jemand die Option entfernt oder sec=ntlmssp setzt.

Kann ich mit sec=krb5 einen Server über seine IP-Adresse einhängen?

In der Regel nicht, denn das Ticket wird für cifs/<servername> angefordert, und für IP-Adressen gibt es standardmäßig keinen SPN. Verwende den vollqualifizierten Namen.

Brauche ich auf einem Samba-Dateiserver SSSD oder winbind?

Für einen Samba-Mitgliedsserver mit security = ADS ist winbind erforderlich. SSSD eignet sich für Linux-Systeme ohne Samba-Freigaben. Beide parallel für dieselbe Domäne führen leicht zu Konflikten beim Maschinenkennwort.

Schaltet „ntlm auth = disabled“ in der smb.conf NTLM für meine Domäne ab?

Nein. Der Parameter betrifft die lokale Kontendatenbank von Samba. Domänenkonten prüft der Domain Controller, und dort regeln die Windows-Richtlinien, ob NTLM erlaubt ist.

Wie erkenne ich, ob ein Windows-Client per Kerberos auf meinen Samba-Server zugreift?

Auf dem Windows-Client zeigt klist ein Ticket für cifs/<samba-server>. Gleichzeitig sollten für dieses Ziel keine Ereignisse 4020/4021 oder 8001 mehr auftauchen.

Was mache ich mit Linux-Geräten, die gar nicht in der Domäne sind?

Entweder in die Domäne aufnehmen, dann Kerberos, oder bewusst als Ausnahme mit Ablaufdatum führen. Sichtbar machst du sie über die Gegenseite oder über einen Netzwerkmitschnitt.

Fazit: Der Pinguin kann Kerberos, er muss nur dürfen

Linux und Samba sind bei der NTLM-Ablösung selten das technische Problem. Kerberos funktioniert dort seit Jahren zuverlässig, mit mount.cifs und sec=krb5, mit SSSD, mit winbind und einer sauberen Keytab. Das eigentliche Problem ist die Sichtbarkeit: Linux schweigt im Windows-Audit, und Samba schweigt noch etwas lauter. Wer nur auf die Ereignisse seiner Windows-Clients schaut, übersieht genau die Verbindungen, die am Stichtag still brechen.

Also: Gegenseite auswerten, IP-Ziele in Namen verwandeln, Kennwortdateien gegen Keytabs tauschen, Kennwortverwaltung auf ein Werkzeug reduzieren, SPNs für Aliasnamen nachtragen und den Rest per Mitschnitt sichtbar machen. Danach bleibt vielleicht ein Pinguin übrig, der NTLM wirklich braucht. Der bekommt dann eine begründete Ausnahme und ein Ablaufdatum, wie jeder andere auch. Wie du nach der Umstellung erkennst, ob jemand heimlich zurückfällt, steht in Nach der Abschaltung: Rückfälle erkennen und NTLM-frei bleiben. Und wenn dein Team die Auswertung selbst sicher beherrschen will, schau in die Schulung zum NTLM Diagnostiker.

WEITERLESEN · Weiter in dieser Serie

› NTLM-Nutzung ermitteln: der komplette Leitfaden
› Erweitertes NTLM-Auditing einschalten: Schritt für Schritt
› Reason-ID 6: wenn der Zielname nicht auflösbar ist
› CNAME, Aliasnamen und hosts-Datei: warum Kerberos den Namen nicht findet
› Rechner ohne Domäne: Arbeitsgruppen, Laborgeräte und NTLM
› NAS, Drucker und Scanner: NTLM bei Geräten ohne Windows
› LDAP-Signierung und Kanalbindung: Ereignisse 2889 und 3039 auswerten
› Mit klist prüfen, ob Kerberos wirklich greift
› Microsoft Kerberos