Microsoft Active Directory Certificate Services (ADCS) – Grundlagen, Architektur und Best Practices

Wissen	Beratung	Schulung
Der Überblick zu ADCS und PKI auf der Themenseite.	PKI-Design, Audit und Betrieb mit Unterstützung.	Workshops zu Active Directory und Zertifikaten im Haus.
ADCS – Active Directory Certificate Services	Consulting zu PKI und Zertifikaten (ADCS)	Consulting zu PKI und Zertifikaten (ADCS)

Zertifikate sind die Art von Infrastruktur, die niemand bemerkt, solange sie funktioniert – und die alle gleichzeitig bemerken, wenn sie es nicht mehr tut. Ein abgelaufenes CA-Zertifikat, eine nicht mehr erreichbare Sperrliste oder eine zu großzügig freigegebene Vorlage: In jedem dieser Fälle stehst du mit einer Umgebung da, in der plötzlich niemand mehr vertraut. Active Directory Certificate Services ist Microsofts Antwort auf die Frage, wie man eine unternehmensweite Public-Key-Infrastruktur betreibt, ohne für jedes Zertifikat einen externen Dienstleister zu bezahlen. Die Rolle ist mächtig, kostenlos im Windows Server enthalten – und genau deshalb oft genug halbherzig aufgesetzt.

Dieser Beitrag zeigt dir, wie Active Directory Certificate Services wirklich aufgebaut wird: welche Hierarchie du brauchst, warum der Root-Schlüssel niemals ans Netz gehört, wie du Zertifikatvorlagen als das behandelst, was sie sind – Sicherheitsobjekte –, und wie du die PKI so betreibst, überwachst und wiederherstellst, dass sie dich auch im Ernstfall nicht im Stich lässt. Wir reden dabei Klartext, inklusive der Angriffsklassen, mit denen Penetration Tester seit Jahren fehlkonfigurierte ADCS-Umgebungen in wenigen Minuten übernehmen. Wer das Thema zum ersten Mal von oben sehen will, findet den Rahmen auf der Themenseite:

ADCS – Active Directory Certificate Services

Fakten

Eine CA ist kein Server, sondern ein Vertrauensanker. Fällt ein Webserver aus, ist eine Website offline. Wird ein CA-Schlüssel kompromittiert, ist jedes Zertifikat, das diese CA je ausgestellt hat, wertlos – rückwirkend und für alle Clients gleichzeitig. Dieser Unterschied erklärt praktisch jede Designentscheidung in diesem Artikel.

Architektur: Hierarchie, Offline-Root und wann drei Ebenen

Warum der Root-Schlüssel offline bleibt

Die bewährte Standardarchitektur für ADCS ist zweistufig: eine Stammzertifizierungsstelle (Root CA), die offline betrieben wird, und darunter eine oder mehrere ausstellende CAs (Issuing CAs), die online im Active Directory hängen und die eigentliche Arbeit machen. Die Root-CA signiert ausschließlich die Zertifikate ihrer untergeordneten CAs und gelegentlich ihre eigene Sperrliste – sonst nichts. Sie ist bewusst eine Standalone-CA ohne Domänenbeitritt, idealerweise eine virtuelle Maschine, die nach dem Aufsetzen heruntergefahren und deren virtuelle Festplatte verschlossen weggesperrt wird.

Der Grund ist simpel: Ein Schlüssel, der nicht am Netz hängt, kann nicht über das Netz gestohlen werden. Die Root-CA ist der Vertrauensanker der gesamten PKI. Wird ihr privater Schlüssel kompromittiert, musst du das Root-Zertifikat auf jedem Client, jedem Server und jedem Gerät als nicht vertrauenswürdig markieren und eine komplett neue PKI aufbauen. Das ist kein Wochenendprojekt, das ist ein Quartalsprojekt mit Beteiligung des Vorstands. Eine Issuing CA dagegen lässt sich im Kompromittierungsfall widerrufen und ersetzen, ohne dass das Fundament wackelt.

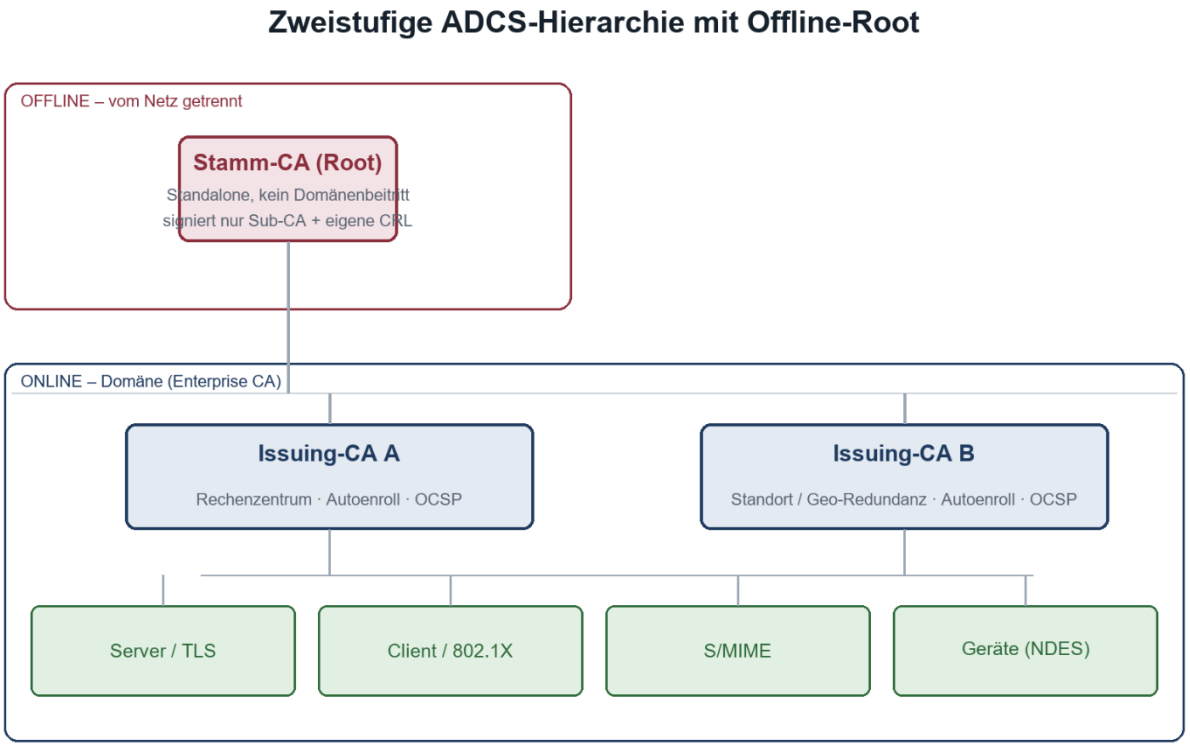


Abbildung: Zweistufige Hierarchie – die Offline-Root signiert nur die Issuing-CAs, die online die Endzertifikate ausstellen.

Tipp

Eine typische Consultant-Beobachtung: In kleineren Umgebungen wird die Root-CA gern „schnell mal" als Enterprise-CA im Domänenbetrieb aufgesetzt, weil das Wizard-gesteuert am einfachsten ist. Damit ist der wichtigste Schlüssel der Organisation ab Tag eins online und steht als Domänenmitglied im selben Verzeichnis wie alles andere. Nimm dir die zwei Stunden mehr und mach es richtig – eine nachträgliche Trennung ist ungleich teurer.

Was die Offline-Root trotzdem regelmäßig braucht

Offline heißt nicht vergessen. Die Root-CA hat auch im Ruhezustand wiederkehrende Aufgaben, und die musst du planen, weil die VM dafür jedes Mal kontrolliert hochgefahren werden muss. Erstens signiert sie ihre eigene Sperrliste. Diese CRL hat eine Gültigkeit, und läuft sie ab, gilt auch das Zertifikat der Issuing-CA als nicht mehr prüfbar – die gesamte PKI steht. Deshalb setzt man die CRL-Gültigkeit der Root bewusst lang (oft sechs oder zwölf Monate) und trägt den Termin zur Neuveröffentlichung fett in den Kalender ein. Zweitens signiert die Root neue oder erneuerte Sub-CA-Zertifikate. Drittens gehören Betriebssystem-Updates eingespielt, bevor die VM wieder schlafen gelegt wird.

Wichtig

Die häufigste selbstverschuldete PKI-Katastrophe ist eine abgelaufene Root-CRL. Niemand merkt es, bis sie abläuft – dann aber schlagartig alle. Lege die Root-CRL-Gültigkeit lang genug an und veröffentliche rechtzeitig neu. Ein einzelner nicht gepflegter Kalendereintrag kann hier eine ganze Organisation lahmlegen.

Wann eine dreistufige Hierarchie wirklich gerechtfertigt ist

Eine dreistufige PKI schiebt zwischen Root und Issuing-CAs eine Policy-CA. Sie klingt nach „mehr Sicherheit“ und ist in den allermeisten Fällen schlicht mehr Aufwand. Eine Policy-CA trägt dann ihren Namen zu Recht, wenn du tatsächlich getrennte Zertifikatsrichtlinien pro Geschäftsbereich kodieren willst – etwa dokumentiert über Certificate Policy OIDs – oder wenn regulatorische Vorgaben eine organisatorische Trennung einzelner Teil-PKIs erzwingen, sodass du einen ganzen Zweig sperren kannst, ohne die übrigen zu berühren. Konzerne mit mehreren rechtlich getrennten Einheiten oder sehr strenge Compliance-Umgebungen sind reale Anwendungsfälle.

Alle anderen fahren mit zwei Ebenen besser. Jede zusätzliche Ebene verlängert die Kette, die jeder Client validieren muss, verdoppelt die Zahl der offline zu pflegenden CAs und erhöht die Zahl der CRLs, die rechtzeitig veröffentlicht sein wollen. Die Faustregel lautet: Baue drei Ebenen nur, wenn du eine konkrete Richtlinie oder Compliance-Anforderung benennen kannst, die zwei Ebenen nicht erfüllen – nicht, weil es sicherer aussieht.

Architektur	Einsatzbereich	Vorteil	Preis
Einstufig (nur eine CA)	Labor, Test	Schnell aufgesetzt	Keine Sicherheitsreserve – in Produktion tabu
Zweistufig (Offline-Root + Issuing)	Standard für fast alle	Root-Schutz bei vertretbarem Aufwand	Root muss für Wartung aktiviert werden
Dreistufig (+ Policy-CA)	Konzern, strenge Compliance	Getrennte Richtlinien, Zweig sperrbar	Zwei Offline-CAs, längere Kette
Mehrstandort (mehrere Issuing)	Global verteilt	Geo-Redundanz, kurze Latenz	AIA/CDP weltweit konsistent halten

Zertifikatvorlagen sind Sicherheitsobjekte

Eine Zertifikatvorlage legt fest, wie ein Zertifikat aussieht und wer es bekommen darf. Genau deshalb ist jede Vorlage eine Berechtigungsentscheidung, keine kosmetische Einstellung. Behandle Änderungen an Vorlagen so sensibel wie Änderungen an Gruppenrichtlinien – nur PKI-Admins sollten Schreibrechte haben, denn wer eine Vorlage ändern kann, kann das Verhalten jedes künftig damit ausgestellten Zertifikats ändern.

Die kritischen Stellschrauben

Vier Eigenschaften entscheiden darüber, ob eine Vorlage ein Werkzeug oder ein Einfallstor ist. Erstens die Quelle von Subject und SAN: Baut die CA den Namen aus dem Active Directory, ist alles gut. Erlaubt die Vorlage dagegen, dass der Antragsteller Subject und SAN frei in die Anforderung schreibt (das Flag ENROLLEE_SUPPLIES_SUBJECT), kann jeder Berechtigte ein Zertifikat auf einen beliebigen Namen ausstellen – inklusive dem eines Domänenadministrators. Zweitens die EKU: Ein Zertifikat sollte genau den einen Zweck tragen, für den es gedacht ist, und nicht „Any Purpose“ oder gleich Client- und Serverauthentifizierung in einem. Drittens die Schlüsselgröße und der nicht exportierbare Schlüssel: mindestens RSA 3072 oder ECC P-256, und

der private Schlüssel bleibt auf dem System. Viertens die Berechtigungen: Wer darf anfordern (Enroll), wer automatisch (Autoenroll)?

Warnung

Die Kombination aus „Subject aus der Anforderung zulassen“, der EKU Clientauthentifizierung und Enroll-Recht für eine breite Gruppe wie „Domänen-Benutzer“ ist kein theoretisches Risiko. Sie ist exakt die Vorlage, nach der jeder Angreifer als Erstes sucht – sie erlaubt die Ausstellung eines Anmeldezertifikats auf einen fremden Namen und damit die Übernahme beliebiger Konten. Audite jede Vorlage auf diese Kombination.

Autoenrollment sauber einsetzen

Autoenrollment ist der richtige Weg, um Computer- und Benutzerzertifikate ohne manuelle Anträge auszurollen und rechtzeitig zu erneuern – gesteuert per Gruppenrichtlinie. Die Regel ist dieselbe wie überall: so eng wie möglich. Aktiviere Autoenroll nur für die konkret benötigten Vorlagen und gezielte Sicherheitsgruppen, nie pauschal für alle. Eine Vorlage mit Autoenroll-Recht für „Authentifizierte Benutzer“ verteilt ihr Zertifikat an praktisch jedes Objekt der Domäne – das willst du bei einem Smartcard-Logon- oder Code-Signing-Zertifikat garantiert nicht.

ESC: Wie fehlkonfigurierte ADCS-Umgebungen übernommen werden

2021 hat SpecterOps mit dem Papier „Certified Pre-Owned“ eine Reihe von Eskalationspfaden über ADCS katalogisiert und von ESC1 aufwärts durchnummeriert. Seitdem ist ADCS fester Bestandteil jedes Penetrationstests in einer Windows-Umgebung, weil die Fehlkonfigurationen weit verbreitet und die Folgen maximal sind: In vielen Fällen führt ein einziges falsch gesetztes Vorlagen-Flag direkt zur Domänenübernahme. Du musst diese Klassen kennen, nicht um anzugreifen, sondern um zu auditieren.

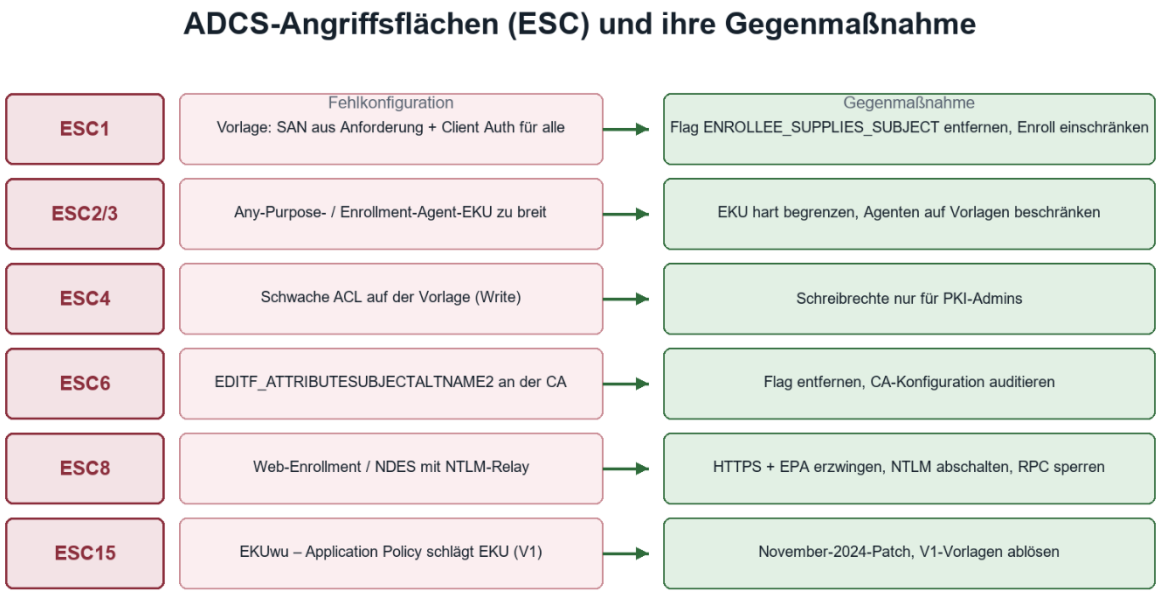


Abbildung: Die wichtigsten ESC-Klassen – jede Fehlkonfiguration hat eine konkrete, meist einfache Gegenmaßnahme.

Die wichtigsten Klassen von ESC1 bis ESC8

ESC1 ist der Klassiker: eine Vorlage, die den Subject Alternative Name aus der Anforderung übernimmt, Clientauthentifizierung erlaubt und von einer breiten Gruppe angefordert werden darf. ESC2 und ESC3 drehen sich um zu großzügige EKUs – eine Any-Purpose-EKU beziehungsweise ein missbrauchbares Enrollment-Agent-Zertifikat, mit dem man im Namen anderer anfordert. ESC4 ist eine schwache ACL auf der Vorlage selbst: Wer Schreibrecht hat, macht aus einer harmlosen Vorlage eine ESC1-Vorlage. ESC6 ist das CA-weite Flag EDITF_ATTRIBUTESUBJECTALTNAME2, das SAN-Angaben aus Anforderungen generell zulässt. ESC7 betrifft zu weitreichende Rechte auf der CA selbst, ESC5 die Objekte im AD rundherum.

Web-Enrollment, NDES und NTLM-Relay (ESC8)

ESC8 verdient besondere Aufmerksamkeit, weil es nicht an einer Vorlage hängt, sondern an den HTTP-Endpunkten: der Web-Enrollment-Oberfläche und dem Certificate Enrollment Web Service. Akzeptieren diese NTLM-Authentifizierung, lässt sich ein NTLM-Relay fahren – ein Angreifer zwingt etwa einen Domänencontroller zur Authentifizierung und leitet diese an den HTTP-Endpunkt weiter, um in dessen Namen ein Zertifikat zu bekommen. Die Gegenmaßnahme: HTTPS mit Extended Protection for Authentication erzwingen, NTLM auf diesen Endpunkten abschalten und, wo das Web-Enrollment nicht gebraucht wird, es schlicht nicht installieren.

ESC15 und der Patch-Stand

Neuer in der Reihe ist ESC15, auch „EKUwu“ genannt (CVE-2024-49019). Bei alten Version-1-Vorlagen wurde die Application-Policy-Erweiterung gegenüber der EKU bevorzugt behandelt, sodass sich ein Zertifikat mit einem anderen Verwendungszweck ausstellen ließ, als die Vorlage vorsah. Microsoft hat das im November 2024 gepatcht – wer seine CAs aktuell hält, ist gegen den Kern geschützt, aber viele Forests schleppen noch die alten V1-Vorlagen mit sich herum. Löse V1-Vorlagen ab und ziehe auf Version 2 oder höher um.

Fakten

Seit dem Windows-Update vom 9. September 2025 verlangen Domänencontroller zwingend eine starke Zertifikatszuordnung bei der zertifikatbasierten Anmeldung (KB5014754). Der zuvor verfügbare Kompatibilitätsmodus, der sich per Registry erzwingen ließ, ist damit endgültig abgeschaltet – ein gepatchter DC ignoriert den Override. Das ist der Abschluss eines Rollouts, der 2022 begann. Zertifikate ohne starke Zuordnung (etwa über die SID-Erweiterung) werden zur Anmeldung nicht mehr akzeptiert. Prüfe vor diesem Hintergrund alle Anmeldezertifikate und deren Vorlagen.

Auditieren statt hoffen

Du musst diese Fehlkonfigurationen nicht von Hand suchen. Es gibt etablierte Werkzeuge, die eine ADCS-Umgebung gegen alle bekannten ESC-Klassen prüfen und dir eine priorisierte Liste liefern. Nimm ein solches Audit in den regelmäßigen Betrieb auf – mindestens jährlich, besser nach jeder größeren Änderung an Vorlagen oder CA-Konfiguration. Ein PKI-Audit, das nur Schlüssellängen und Gültigkeiten prüft, aber die ESC-Klassen auslässt, prüft am eigentlichen Risiko vorbei.

Sperrprüfung: CRL, Delta-CRL und OCSP

Ein Zertifikat sagt „vertrau mir bis Datum X“. Die Sperrprüfung ist der Mechanismus, mit dem du dieses Versprechen vorzeitig zurücknehmen kannst – etwa wenn ein privater Schlüssel abhandenkommt. Ohne funktionierende Sperrprüfung ist ein widerrufenes Zertifikat so gültig wie jedes andere. Es gibt drei Verfahren, und du kombinierst sie je nach Bedarf.

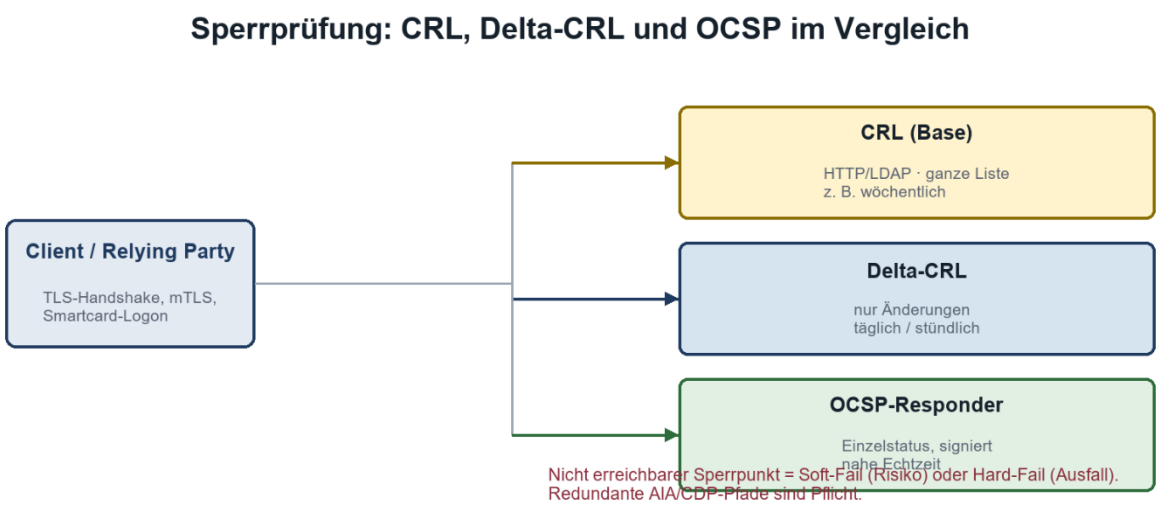


Abbildung: Die drei Sperrverfahren und die Folge eines nicht erreichbaren Sperrpunkts.

CRL gegen OCSP abwägen

Die klassische Sperrliste (CRL) ist eine von der CA signierte Liste aller widerrufenen Zertifikate. Clients laden sie herunter und prüfen lokal. Das funktioniert offline und ist robust, aber die Liste wächst mit jedem Widerruf, und zwischen zwei Veröffentlichungen ist ein frisch gesperrtes Zertifikat noch gültig. Die Delta-CRL mildert das, indem sie nur die Änderungen seit der letzten Base-CRL enthält und häufiger erscheint. OCSP dagegen fragt den Status eines einzelnen Zertifikats in nahezu Echtzeit bei einem Online-Responder ab – schlank und aktuell, aber abhängig von der Verfügbarkeit dieses Responders.

Kriterium	CRL (Base)	Delta-CRL	OCSP
Aktualität	mittel (Intervall)	besser	nahe Echtzeit
Datenmenge am Client	ganze Liste	nur Änderungen	Einzelantwort
Offline nutzbar	ja	ja	nein
Abhängigkeit	CDP erreichbar	CDP erreichbar	Responder online
Typischer Einsatz	Basis, Fallback	Ergänzung zur Base	mTLS, hohe Last, schnelle Prüfung

AIA und CDP redundant auslegen

Damit ein Client die Kette prüfen und die Sperrliste finden kann, trägt jedes Zertifikat die Pfade dafür: Authority Information Access (AIA) verweist auf das CA-Zertifikat und den OCSP-Responder, der CRL Distribution Point (CDP) auf die Sperrliste. Diese Pfade gehören redundant ausgelegt, typischerweise über LDAP für domänengebundene interne Clients und zusätzlich über HTTP für alles andere – externe Geräte, Partner, Maschinen ohne AD-Anbindung. HTTP ist dabei flexibler, lässt sich hinter einen Load Balancer legen und einfacher cachen.

Warnung

Entscheide dich bewusst, ob deine Clients bei nicht erreichbarem Sperrpunkt „Soft-Fail“ (Zertifikat trotzdem akzeptieren) oder „Hard-Fail“ (ablehnen) machen. Soft-Fail macht eine Sperrung wirkungslos, sobald der CDP weg ist – ein Angreifer muss die CRL nur unerreichbar machen. Hard-

Fail dagegen legt bei einem CDP-Ausfall den kompletten Dienst lahm, gerade bei mTLS zwischen Diensten. Beide Varianten haben einen Preis; die Antwort heißt hochverfügbare, mehrfach ausgelegte Sperrpunkte, nicht die Wahl des kleineren Übels.

NDES/SCEP, Schlüsselmanagement und Monitoring

NDES und SCEP sicher betreiben

Der Network Device Enrollment Service stellt über SCEP Zertifikate an Geräte aus, die nicht in der Domäne hängen – Router, Switches, vor allem aber per Intune oder einem anderen MDM verwaltete Mobilgeräte. NDES läuft auf einem IIS und ist damit ein HTTP-Dienst im Netz, der im Namen der CA Zertifikate besorgt. Das macht ihn zu einem beliebten Angriffsvektor. Betreibe ihn auf einem dedizierten Server, beschränke sein Dienstkonto auf genau die Vorlagen, die es ausstellen darf, und sichere den IIS ab (HTTPS, Extended Protection, keine überflüssigen Module).

Das Challenge-Passwort für SCEP ist der Schwachpunkt: Ein statisches, dauerhaft gleiches Secret, das einmal durchsickert, erlaubt fremden Geräten Zertifikatsanträge. In der Intune-Anbindung übernimmt der Certificate Connector die dynamische Vergabe der Challenge, sodass du kein statisches Passwort verteilen musst – nutze diesen Weg und verlasse dich nicht auf ein handverteiltes Secret.

Tipp

Setze den NDES-Server möglichst in ein abgesichertes Netzsegment und gib seinem Dienstkonto nur die ein oder zwei konkret benötigten Vorlagen frei. Ein NDES, der jede Vorlage ausstellen darf, ist ein Generalschlüssel mit Weboberfläche.

Schlüssel der CA schützen

Der private Schlüssel einer CA ist das wertvollste Objekt der ganzen Infrastruktur. Für die Root-CA ist ein Hardware Security Module (HSM) praktisch Pflicht – der Schlüssel wird darin erzeugt und verlässt das Modul nie, auch nicht beim Backup. Für online betriebene Issuing-CAs erhöht ein HSM die Sicherheit ebenfalls deutlich und wird bei Audits oft verlangt. Wo kein HSM möglich ist, bleibt wenigstens ein abgesicherter Key Storage Provider und ein verschlüsseltes, offline verwahrtes Backup. Das TPM eines Servers ist für die Schlüssel von Clientzertifikaten sinnvoll, ersetzt aber kein HSM für den CA-Schlüssel selbst.

Schlüsselarchivierung – Hilfe oder Lücke

Für S/MIME-Verschlüsselung willst du den privaten Schlüssel der Nutzer archivieren können, denn verschlüsselte Altmails sind ohne den Schlüssel unwiederbringlich verloren, wenn jemand das Unternehmen verlässt oder sein Gerät kaputtgeht. Dafür richtest du einen Key Recovery Agent (KRA) ein und aktivierst die Archivierung in der Vorlage. Der entscheidende Punkt: Das ist genau für Verschlüsselungsschlüssel richtig und für Signaturschlüssel falsch. Einen Signaturschlüssel, der archiviert wird, kann jemand anderes zur Signatur missbrauchen – die Nichtabstreitbarkeit ist dahin. Archiviere nur, was du zur Datenwiederherstellung brauchst, und schütze den KRA-Zugang streng mit Protokollierung und Vier-Augen-Prinzip.

Warnung

Schlüsselarchivierung ist eine zentrale Sammelstelle privater Schlüssel. Wer den KRA kontrolliert, kann jede damit archivierte verschlüsselte Kommunikation lesen. Das ist bei Verschlüsselungszertifikaten ein akzeptabler, kontrollierter Kompromiss – bei Authentifizierungs- oder Signaturzertifikaten eine Hintertür. Trenne die Vorlagen entsprechend.

Monitoring und Auditing

Eine CA, die niemand beobachtet, meldet ihren Ausfall erst, wenn die Anrufe kommen. Aktiviere das Audit-Logging der Zertifizierungsstelle für Ausstellung, Ablehnung, Widerruf und Konfigurationsänderungen und leite die relevanten Ereignis-IDs ins SIEM weiter. Überwache dort nicht nur Fehler, sondern auch Muster – ein plötzlicher Ausstellungs-Peak kann auf Missbrauch hindeuten. Jenseits der Ereignisprotokolle brauchst du drei Kennzahlen im Dauerblick: das Alter der CRLs (Alarm, bevor eine abläuft), die Antwortzeiten der OCSP-Responder und den anstehenden Ablauf wichtiger Zertifikate.

Was überwachen	Woran erkennbar	Warum kritisch
CRL-Alter	Verbleibende Gültigkeit der veröffentlichten CRL	Abgelaufene CRL stoppt die gesamte Prüfung
OCSP-Antwortzeit	Latenz des Responders	Hohe Latenz bremst jeden Handshake
Zertifikatsablauf	Restlaufzeit kritischer Zertifikate	Ungeplanter Ausfall von Diensten
Ausstellungs-Peaks	Zahl ausgestellter Zertifikate pro Zeit	Hinweis auf Missbrauch oder Fehlkonfiguration
Konfigurationsänderung	Ereignis-ID bei CA- oder Vorlagenänderung	Unautorisierte Änderung = möglicher Angriff

Backup, Wiederherstellung und Lebenszyklus

Was du sichern musst – und in welcher Reihenfolge du zurückspielst

Eine CA besteht aus mehreren Teilen, und ein Backup ist nur dann eines, wenn es alle erfasst: den privaten Schlüssel und das CA-Zertifikat, die CA-Datenbank samt Transaktionslogs sowie die Konfiguration aus der Registry. Die Zertifikatvorlagen liegen dagegen im Active Directory, nicht in der CA-Datenbank – sie werden über das AD-Backup gesichert. Für das eigentliche CA-Backup nutzt du certutil mit den Verben -backupkey (Schlüssel und Zertifikat) und -backupDB (Datenbank mit Logs); die Registry-Konfiguration liest du zusätzlich per certutil -getreg aus und sicherst sie weg.

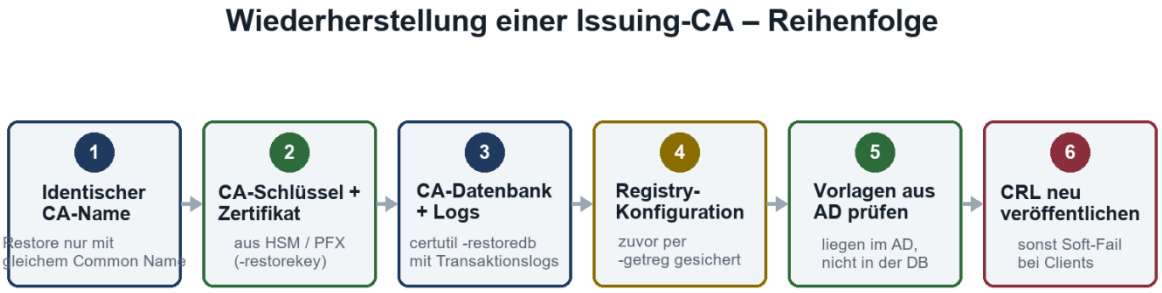


Abbildung: Die Reihenfolge bei der Wiederherstellung einer Issuing-CA – der identische CA-Name steht am Anfang.

Wichtig

Ein Restore funktioniert nur mit identischem CA-Namen. Der Common Name der CA ist in jedes ausgestellte Zertifikat und jede CRL eingebacken. Spielst du die Datenbank auf einer CA mit abweichendem Namen ein, passt nichts mehr zusammen – die Kette bricht. Dokumentiere den exakten CA-Namen im Notfallplan, nicht aus dem Gedächtnis.

Bei einer zweistufigen Hierarchie gilt für den Wiederaufbau: Die Root-CA kommt zuerst, weil ihr Zertifikat der Anker ist, auf dem die Issuing-CA ruht. Erst wenn der Vertrauensanker wieder steht, baust du die Issuing-CA auf, spielst Schlüssel, Datenbank und Konfiguration ein und veröffentlichst zum Schluss eine frische CRL – sonst laufen deine Clients in den Soft- oder Hard-Fail.

Erneuerung und Root Key Rollover

Zertifikate erneuert man nicht auf den letzten Drücker, sondern bei etwa 80 Prozent der Laufzeit – so bleibt Puffer, falls die Erneuerung schiefgeht. Für CAs selbst planst du den Schlüsselwechsel langfristig: Eine Root-CA läuft lang (oft zehn, teils bis zwanzig Jahre), eine Issuing-CA kürzer (um die fünf Jahre), Endzertifikate möglichst kurz (heute meist ein Jahr). Steht die Erneuerung der Root an, ist das ein Projekt mit Vorlauf von ein bis zwei Jahren: neues Root-Zertifikat erzeugen, an jeden Client, Server und jedes Gerät verteilen und eine Übergangsphase einplanen, in der altes und neues Root gleichzeitig vertraut werden.

Notfallplan und jährliche Übung

Für den Kompromittierungsfall brauchst du einen geschriebenen, geprobten Plan. Wird eine Issuing-CA kompromittiert, widerruft die übergeordnete Root deren Zertifikat, womit alle davon ausgestellten Zertifikate fallen; danach baust du eine neue Issuing-CA auf und stellst neu aus. Wird der Root-Schlüssel kompromittiert, bleibt nur der harte Weg: Root überall als nicht vertrauenswürdig markieren und eine neue PKI etablieren. Genau dieser Albtraum ist der Grund, warum die Root offline und im HSM liegt. Teste die Desaster-Wiederherstellung mindestens jährlich in einer isolierten Umgebung – inklusive der Frage, ob die Passwörter für die Backup-Schlüssel überhaupt noch jemand kennt.

Warnung

Ein Backup, das nie zurückgespielt wurde, ist eine Vermutung, keine Sicherheit. Die jährliche Desaster-Übung ist der einzige Moment, in dem du herausfindest, ob dein Notfallplan funktioniert – bevor es der Ernstfall für dich tut. Wer die Übung aus Zeitgründen streicht, spart an der einen Stelle, an der Sparen am teuersten wird.

ADCS, Intune und Cloud PKI abgrenzen

ADCS ist nicht die einzige Art, in einer Microsoft-Welt Zertifikate auszustellen – und nicht für jeden Fall die beste. Wer heute vor allem Cloud-verwaltete Geräte über Intune mit Zertifikaten versorgt, muss nicht zwingend eine komplette On-Prem-PKI mit NDES-Server und Certificate Connector betreiben.

Microsoft Cloud PKI

Microsoft Cloud PKI ist ein in Intune integrierter Dienst, der CAs in der Cloud betreibt und Zertifikate an Intune-verwaltete Geräte ausstellt – ohne lokalen CA-Server, ohne NDES, ohne Connector. Für reine Cloud- und BYOD-Szenarien ist das oft der schlankere Weg. Lizenzseitig ist Cloud PKI seit Juli 2026 Bestandteil von Microsoft 365 E5; für E3 und andere Pläne ist es ein kostenpflichtiges Add-on beziehungsweise Teil der Intune Suite. Prüfe den aktuellen Lizenzstand im Zweifel direkt bei Microsoft, denn die SKUs ändern sich.

Szenario	Besser mit	Warum
Domänengebundene Server, S/MIME, Code Signing	ADCS On-Prem	Volle Kontrolle, AD-Integration, Autoenroll
Reine Intune-/BYOD-Geräte, WLAN/VPN	Microsoft Cloud PKI	Kein NDES/Connector, cloud-nativ
Öffentlich vertraute TLS-Zertifikate	Externe/öffentliche CA	Browser vertrauen der internen Root nicht
Hybrid – bestehende On-Prem-PKI + Cloud-Geräte	ADCS + Intune-Connector oder Cloud PKI	Vorhandene Investition weiternutzen

Hybride Betriebsmodelle

In der Praxis läuft es meist auf einen Mix hinaus. Eine bestehende ADCS-PKI bleibt für domänengebundene Server, S/MIME und Code Signing zuständig, während Cloud-verwaltete Geräte ihre WLAN- und VPN-Zertifikate entweder über den Intune-Connector aus genau dieser PKI oder über Cloud PKI beziehen. Wichtig ist dabei, keine „Schatten-PKI“ entstehen zu lassen: Egal wo ein Zertifikat ausgestellt wird, es gehört in ein zentrales Inventar und unter dieselben Richtlinien zu Algorithmen und Laufzeiten. Für die Planung eines solchen hybriden Modells – oder eines sauberen Neuaufbaus – lohnt der Blick in die begleitende Unterstützung:

Consulting zu PKI und Zertifikaten (ADCS)

Fazit

Active Directory Certificate Services ist kein Dienst, den man installiert und vergisst – es ist der Vertrauensanker einer Organisation, und als solcher verdient es dieselbe Sorgfalt wie das Active Directory selbst. Die wichtigsten Entscheidungen sind erstaunlich einfach und erstaunlich oft falsch getroffen: eine zweistufige Hierarchie mit konsequent offline gehaltener Root, Zertifikatvorlagen, die als Sicherheitsobjekte behandelt und gegen die ESC-Klassen auditiert werden, redundant ausgelegte Sperrpunkte und ein Backup, das in der jährlichen Übung auch wirklich zurückgespielt wird.

Wer diese vier Punkte ernst nimmt, hat mehr für die Sicherheit seiner PKI getan als mit jeder zusätzlichen Hierarchieebene. Und wer heute ohnehin stärker in Richtung Cloud und Intune geht, trifft die Abgrenzung zwischen On-Prem-ADCS und Cloud PKI am besten bewusst statt nebenbei – denn eine halb migrierte, undokumentierte PKI ist gefährlicher als eine klar umrissene alte.

FAQ – Häufige Fragen zu ADCS und PKI

Was ist der Unterschied zwischen CRL und OCSP, und wann nutzt man welches?

Die CRL ist eine signierte Liste widerrufener Zertifikate, die der Client herunterlädt und lokal prüft – robust und offline nutzbar, aber potenziell groß und nur so aktuell wie das Veröffentlichungsintervall. OCSP fragt den Status eines einzelnen Zertifikats in nahezu Echtzeit bei einem Online-Responder ab – schlank und aktuell, aber abhängig von dessen Verfügbarkeit. In der Praxis setzt man OCSP für schnelle Prüfungen und hohe Last ein und behält die CRL als Fallback.

Welche Gültigkeitsdauern sind für Root, Issuing-CA und Endzertifikate sinnvoll?

Die Root läuft lang, oft zehn, teils bis zwanzig Jahre, weil ihr Wechsel aufwendig ist. Issuing-CAs liegen meist bei etwa fünf Jahren. Endzertifikate hält man kurz – heute überwiegend ein Jahr –,

weil kurze Laufzeiten das Risiko veralteter Algorithmen und vergessener Sperrungen senken. Kurze Laufzeiten setzen funktionierende Erneuerung per Autoenrollment voraus.

SAN oder Subject-CN – was prüfen moderne Anwendungen?

Den Subject Alternative Name. Ist ein SAN vorhanden, ignorieren aktuelle Clients den Common Name. Der erwartete Name – DNS, E-Mail oder UPN – muss also im SAN stehen, sonst scheitert die Prüfung. Reine CN-Zertifikate sind nur noch eine Übergangslösung für Altsysteme, die man besser ablöst.

Warum gilt eine Vorlage mit „Subject aus der Anforderung“ als gefährlich?

Weil der Antragsteller dann frei bestimmt, auf welchen Namen das Zertifikat lautet. In Kombination mit Clientauthentifizierung und breitem Enroll-Recht (ESC1) kann jemand ein Anmeldezertifikat auf den Namen eines Domänenadministrators ausstellen und dessen Konto übernehmen. Diese Kombination gehört in jedem Audit zuerst gesucht und entschärft.

Ist ein HSM für die CA Pflicht?

Vorgeschrieben nicht, aber für die Root-CA praktisch unverzichtbar und für produktive Issuing-CAs dringend empfohlen. Der Schlüssel wird im HSM erzeugt und verlässt es nie – auch nicht beim Backup. Kleine Testumgebungen laufen ohne, eine produktive PKI ab einer gewissen Größe sollte es nicht.

Wann NDES/SCEP, wann Microsoft Cloud PKI?

NDES/SCEP ist der Weg, um Geräte ohne Domänenbeitritt aus einer bestehenden On-Prem-PKI zu versorgen – inklusive Intune über den Certificate Connector. Microsoft Cloud PKI stellt dagegen Zertifikate direkt aus der Cloud an Intune-Geräte aus, ohne NDES und Connector. Für reine Cloud- und BYOD-Szenarien ist Cloud PKI der schlankere Weg; wo eine gepflegte On-Prem-PKI existiert, nutzt man sie oft weiter.

Was tun bei einem kompromittierten CA-Schlüssel?

Bei einer Issuing-CA widerruft die übergeordnete Root deren Zertifikat – damit fallen alle davon ausgestellten Zertifikate –, dann baust du eine neue Issuing-CA auf. Ist der Root-Schlüssel betroffen, musst du das Root-Zertifikat überall als nicht vertrauenswürdig markieren und eine neue PKI etablieren. Genau deshalb liegt die Root offline und im HSM.

Was bedeutet die erzwungene starke Zertifikatszuordnung für mich?

Seit dem Update vom 9. September 2025 akzeptieren gepatchte Domänencontroller zur zertifikatbasierten Anmeldung nur noch Zertifikate mit starker Zuordnung, etwa über die SID-Sicherheitserweiterung; der frühere Kompatibilitätsmodus ist abgeschaltet. Du musst sicherstellen, dass deine CA diese Erweiterung einträgt und deine Anmeldezertifikate entsprechend ausgestellt sind, sonst scheitert das Logon.

Wie läuft ein Root Key Rollover ab?

Lange vor Ablauf – ein bis zwei Jahre vorher – erzeugst du ein neues Root-Zertifikat und verteilst es an alle Vertrauensstellungen. Danach stellen die Issuing-CAs unter der neuen Root aus. Es gibt eine Übergangsphase, in der beide Root-Zertifikate vertraut werden, bis alle alten Zertifikate abgelaufen oder ersetzt sind. Sorgfältige Planung verhindert, dass ein System den neuen Anker verpasst.

Wie vermeide ich eine „Schatten-PKI“ in hybriden Umgebungen?

Indem jedes Zertifikat – egal ob aus ADCS, Cloud PKI oder einer externen CA – in ein zentrales Inventar wandert und unter dieselben Richtlinien zu Algorithmen, Schlüssellängen und Laufzeiten fällt. Uneinheitliche, nirgends erfasste Zertifikate sind der Stoff, aus dem die unangenehmen

Vorfälle sind. Einheitliche Governance über alle Ausstellungswege ist wichtiger als die Frage, welcher Dienst ausstellt.