Migrating from Legacy PKI to Modern
Certificate Management

Plenty of organizations are running PKI infrastructure that was designed and deployed a decade or more ago, back when certificates lasted years, renewal happened manually a few times annually, and nobody had heard of an AI agent needing its own certificate. Migrating that legacy PKI to a modern, automation-first model is a substantial project, but it is also an increasingly unavoidable one. This article covers how to approach that migration without breaking everything currently depending on the old system.

Recognizing the Signs of a Legacy PKI

Legacy PKI environments share recognizable characteristics: certificate issuance and renewal handled manually or through custom scripts nobody fully documents anymore, a root CA that may be running on aging hardware or software no longer receiving vendor support, certificate templates and policies that have drifted informally over years of ad hoc exceptions, and often no complete, accurate inventory of every certificate currently in production. None of this necessarily means the legacy PKI is insecure today, but it does mean it is poorly positioned for the automation and shorter lifespans the rest of the industry is moving toward.

Starting With a Genuine Inventory

Before touching anything, a migration needs an accurate picture of what currently exists: every certificate in production, what it protects, who owns it, how it was issued, and when it expires. This step alone often surfaces certificates nobody remembered existed, expired certificates still technically installed and unnoticed because nothing depends on them anymore, and dependencies that were never properly documented. Certificate discovery tools that scan networks and cloud environments can accelerate this considerably compared to a purely manual audit, and this inventory becomes the foundation every subsequent migration decision is based on.

Choosing a Parallel-Run Migration Strategy

Rather than attempting a single cutover from the legacy CA to a new one, most successful migrations run the old and new systems in parallel for a defined transition period. New certificates are issued from the modern infrastructure going forward, while existing certificates continue running under the legacy CA until they naturally approach renewal, at which point they are migrated to the new system as part of that renewal rather than through a separate, disruptive migration event. This spreads the actual cutover work out over the natural renewal cycle rather than concentrating it into one high-risk event.

Preserving Trust During the Transition

A critical detail often overlooked is ensuring that clients trusting the legacy CA's root also trust the new CA's root well before any certificate migrates, since a client encountering a certificate from a root it does not yet trust will fail validation regardless of how correctly everything else is configured. This means distributing the new root, or ideally cross-signing the new intermediate with the legacy root during the transition period, well ahead of migrating any actual production certificates, giving every downstream system time to update its trust store before it actually needs to.

Retiring the Legacy System Deliberately

Decommissioning the legacy CA should happen only after a defined grace period with no remaining certificates depending on it, confirmed against the same inventory built at the start of the migration, and the retirement itself should be documented and communicated clearly, since some overlooked internal system depending on the old root's continued trust can otherwise cause a surprising and hard-to-diagnose outage months after everyone assumed the migration was complete.

Building Automation Into the New System From Day One

The entire point of this kind of migration is largely undermined if the new PKI simply replicates the old manual processes with newer software. Modern PKI deployments should build ACME-based or API-driven automated issuance in as the default path from the very start, rather than treating automation as a future enhancement to be added once the migration itself is considered complete.

Planning Explicitly for Machine and AI Identity Growth

Legacy PKI systems were almost universally designed around a much smaller, slower-growing population of human-facing certificates, with no consideration for the scale of machine and AI identities many organizations now need to support. A migration to modern certificate management is the natural opportunity to build in the identity-type segmentation and high-volume, short-lived issuance patterns discussed elsewhere in this series, rather than migrating to a modern platform only to immediately hit the same structural limitations the legacy system had once AI-driven certificate demand starts climbing.

The Countdown Is Already Running: 200 Days, 100 Days, 47 Days

Every certificate conversation in 2026 eventually arrives at the same clock, and it is worth closing on it here. The CA/Browser Forum's Ballot SC-081v3 is not a proposal under discussion; it is an approved, already-in-motion schedule. Maximum public TLS certificate lifetimes fall from 398 days to 200 days on March 15, 2026. They fall again to 100 days on March 15, 2027. By March 15, 2029, they drop to just 47 days, with domain validation itself needing to be re-proven roughly every 10 days.

01

Translate that into operational terms and the picture gets stark quickly. An organization currently renewing certificates a few times a year will be handling renewal events on the order of every couple of weeks by the end of this countdown, across every endpoint it operates. Manual tracking, calendar reminders, and a spreadsheet somebody checks once a month will not survive contact with that cadence. What has always been an occasional chore is becoming a continuous, automated operation, whether an organization plans for it or not.

02

A legacy PKI migration is exactly the moment to build in full support for the schedule below from the start, since a newly modernized system still built around annual manual renewal habits will need to be revisited again almost immediately as maximum lifetimes fall toward 47 days.

03

The 200-day, 100-day, and 47-day milestones are not distant hypotheticals; the first has already arrived. Organizations that build the automation loop now, generating keys, vaulting them securely, brokering issuance across Certificate Authorities through APIs, and rebinding certificates to live endpoints without manual intervention, will meet each deadline without disruption. Organizations that wait will be rebuilding their certificate operations under deadline pressure, with far less room for error and far less time to get it right. The countdown is the call to action. The only real decision left is whether to automate on your own schedule, or on the CA/Browser Forum's.