Common Pitfalls When Implementing Private CAs

Standing up a private Certificate Authority is a well-documented process, discussed throughout this series, but the gap between a technically functioning private CA and a genuinely well-run one is where most organizations actually struggle. This article covers the specific pitfalls that recur most often during private CA implementation, drawn from patterns discussed across this series' coverage of internal PKI.

Pitfall: Treating the Root Key as an Afterthought

Organizations frequently generate a root CA key with the same casual process used for any other key, storing it on a general-purpose server rather than in an HSM or a properly secured offline location, discussed elsewhere in this series. This pitfall often only becomes apparent well after the fact, when an organization realizes its entire private PKI's trust rests on a key that was never given the protection its actual importance warranted, at which point remediation requires a considerably more disruptive re-key and redistribution process than getting it right from the start would have required.

Pitfall: No Documented Certificate Policy

Many organizations stand up a working private CA technically but skip writing any actual Certificate Policy or Certification Practice Statement, discussed in the private CA design article elsewhere in this series, resulting in inconsistent issuance practices that drift over time as different teams request certificates for different purposes with no consistent standard behind any of them. This pitfall tends to compound silently for years before surfacing as a genuine problem, typically during an audit or a security incident that reveals just how inconsistent actual practice has become relative to what anyone assumed the policy was.

Pitfall: Distributing Trust Incompletely

A private CA's root certificate needs to be distributed to and trusted by every system that will ever need to validate a certificate it issues, and organizations frequently discover gaps in this distribution only when a specific device or application unexpectedly fails validation, often well after the private CA has already been in production use for other systems. Building a comprehensive, tracked distribution process, rather than assuming trust distribution happened correctly everywhere, prevents this from becoming a recurring, one-off troubleshooting exercise every time a new system needs to trust the private CA.

Pitfall: No Real Automation Behind the Private CA

Organizations sometimes replicate the exact manual issuance workflow they were trying to escape by building a technically modern private CA but still requiring manual requests and manual certificate installation for every internal service, discussed as a specific anti-pattern in the legacy PKI migration article elsewhere in this series. A private CA without genuine ACME or API-driven automation behind it recreates the same scaling and reliability problems public certificate management has, just with an internally hosted CA instead of an external one.

Pitfall: Underestimating Certificate Template Governance

Skipping the template-based issuance approach discussed in the internal CA article elsewhere in this series, and instead issuing certificates with ad hoc, manually specified parameters each time, tends to produce inconsistent key usage restrictions, validity periods, and extensions across an organization's certificate estate, making it considerably harder to reason about or audit the estate's overall security posture later.

Pitfall: No Plan for Intermediate or Root Rotation

Organizations frequently stand up a private CA without a documented plan for what happens when the intermediate or root certificate itself eventually needs to be rotated or replaced, an event that, while infrequent, requires careful planning to avoid a disruptive, organization-wide trust re-establishment process, discussed in the legacy PKI migration article elsewhere in this series. Building this rotation plan proactively, even though it may not be needed for years, avoids scrambling to design a safe migration path only once the root's own expiration or a security concern makes rotation urgent.

Avoiding These Pitfalls When Provisioning for AI and Machine Identities

Organizations building or expanding a private CA specifically to serve the growing population of AI agent and machine identities, discussed throughout this series, should apply every lesson above with particular discipline, since the sheer volume and velocity of this identity category tends to expose weak governance, missing automation, or incomplete trust distribution considerably faster than a smaller, slower-growing traditional server certificate population would.

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

Every pitfall described above becomes more costly to fix retroactively as the schedule below compresses renewal cycles toward every 47 days, since a private CA implementation still working through these gaps will face a considerably higher volume of issuance and renewal events surfacing each unresolved weakness far more often than it would have in a slower, less automated era.

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.