Wildcard Certificates:

Pros, Cons, and Proper Usage

Wildcard certificates offer a tempting shortcut: one certificate covering an entire set of subdomains instead of issuing and managing a separate certificate for each one. That convenience comes with real tradeoffs that are worth understanding clearly before adopting wildcards as a default strategy. This article covers what wildcard certificates actually do, their genuine benefits, and the risks that come bundled with that convenience.

What a Wildcard Certificate Actually Covers

A wildcard certificate uses an asterisk in place of the leftmost subdomain label, such as an entry covering star dot example dot com, which validates any single-level subdomain of that domain, including mail, shop, or api, but importantly does not cover the bare root domain itself or any second-level subdomain beyond the single wildcard position, meaning a subdomain nested two levels deep would not be covered without a separate certificate or an additional wildcard entry.

The Genuine Benefits

Wildcard certificates meaningfully simplify certificate management for organizations running many subdomains under a single parent domain, replacing what could be dozens of individual certificates, each with its own renewal schedule, with a single certificate to issue, install, and renew. This reduces both the administrative overhead and the number of distinct renewal events an operations team has to track, which matters more, not less, as certificate lifespans continue shrinking industry-wide.

The Real Risks Worth Understanding

The core risk with wildcard certificates is that they concentrate trust: if a wildcard certificate's private key is ever compromised, every subdomain covered by that certificate is compromised simultaneously, a considerably larger blast radius than a single compromised certificate covering one specific subdomain. This risk is compounded when a wildcard certificate's private key is deployed across many different servers or services to cover all the subdomains it protects, since each additional deployment location is another potential point of exposure for the same shared key. Wildcards can also obscure visibility, since a single certificate covering many subdomains may make it harder for security monitoring to distinguish which specific subdomain is actually being accessed or targeted in an incident.

When Wildcards Are the Right Call

Wildcard certificates make the most sense when the subdomains they cover are operated by the same team, deployed with comparable levels of security rigor, and hosted in environments where the shared private key can be protected consistently across every deployment location. A marketing organization running several similarly low-risk marketing subdomains under one umbrella might reasonably use a wildcard. An organization with wildly different risk profiles across its subdomains, for instance a customer login portal alongside a low-stakes marketing microsite, is generally better served by separate, appropriately scoped certificates for each.

Combining Wildcards With Multi-Domain Certificates

Many CAs support combining wildcard entries with additional specific Subject Alternative Names in a single certificate, offering some of the consolidation benefit of a wildcard while still explicitly naming particular domains that fall outside the wildcard's single-level scope. This hybrid approach can strike a reasonable balance for organizations with a mix of subdomain patterns that a pure wildcard alone would not fully cover.

Wildcards for AI-Serving Subdomain Fleets

Organizations running many AI model-serving or agent-orchestration subdomains, such as separate endpoints per model version or per customer tenant, sometimes reach for a wildcard certificate to simplify what could otherwise be an unwieldy number of individual certificates. This can work well when every subdomain shares a consistent security posture, but teams should weigh it against the concentrated-key-compromise risk described above, particularly for AI infrastructure handling sensitive customer data across many tenant-specific subdomains, where a narrower, more segmented certificate strategy may be the more prudent choice despite the added management overhead.

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

Wildcard certificates issued by public CAs are fully subject to the shrinking lifetime schedule below, and because a single wildcard often protects many subdomains at once, automating its renewal correctly matters even more as that renewal cadence compresses toward every 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.