Revoking Compromised Certificates:

Incident Response Playbook

Knowing conceptually when and why to revoke a certificate, covered earlier in this series, is different from actually executing that revocation correctly and quickly during a genuine, active incident, under time pressure, with real stakes attached to getting it right the first time. This article lays out a practical incident response playbook specifically for compromised certificate scenarios.

Phase One: Rapid Confirmation and Scoping

The moment a potential key compromise is suspected, the first priority is rapid, focused confirmation: has the private key actually been exposed, and if so, exactly which certificate or certificates does that key back. This scoping step matters because acting on incomplete information, either revoking too narrowly and leaving a genuinely compromised certificate active, or revoking too broadly and causing unnecessary service disruption, both carry real costs, and a few minutes spent confirming scope accurately is almost always worth the delay compared to acting on an incomplete picture.

Phase Two: Immediate Containment Before Formal Revocation

While formal revocation with the issuing CA is being initiated, immediate containment steps can limit damage in parallel: rotating any credentials or access this certificate's identity granted, alerting monitoring systems to flag any continued use of the compromised certificate as suspicious, and if the compromised certificate is protecting a specific service that can be safely taken offline temporarily, doing so while the incident is actively being resolved rather than leaving it exposed during the revocation process itself.

Phase Three: Executing Revocation With the Issuing CA

The specific revocation process varies depending on whether the certificate came from a public CA, discussed throughout this series, or an internal private CA, but every organization should have this process, including exact contact points, API calls, or console steps required, documented and readily accessible before an incident occurs, not something a responder has to figure out for the first time while already under pressure. Public CAs typically offer expedited revocation specifically for confirmed key compromise scenarios, and knowing that expedited path in advance can meaningfully reduce response time.

Phase Four: Reissuing and Redeploying Cleanly

Once the compromised certificate is revoked, a new certificate needs to be issued and deployed to replace it, ideally generated with a fresh key pair rather than reusing anything from the compromised key generation process, and this reissuance should go through the same verification steps discussed in the chain-of-trust debugging article elsewhere in this series to confirm the replacement is working correctly before considering the incident closed.

Phase Five: Root Cause Analysis Without Blame

After immediate containment, a genuine root cause investigation should determine exactly how the key was exposed, whether through a code repository leak, a compromised server, a misconfigured storage location, or another vector entirely, and this investigation should follow the systemic, non-blame-oriented framing discussed in the human cost article elsewhere in this series, focusing on what process gap allowed the exposure rather than which individual is at fault.

Building and Testing the Playbook Before It Is Needed

A revocation incident response playbook is only as good as the organization's actual ability to execute it under pressure, which means this playbook should be tested through tabletop exercises or simulated incidents periodically, discussed in the resilience testing context elsewhere in this series, rather than existing purely as an untested document that might reveal critical gaps only during a genuine emergency.

Incident Response for Compromised AI Agent Certificates

AI agent certificate compromise deserves particular urgency in this playbook, given how quickly an autonomous agent with a compromised but still-valid identity could take unauthorized, potentially irreversible action compared to a more passive service credential. Organizations should build AI agent certificates specifically into their incident response playbook with defined, rapid revocation paths and pre-planned fallback behavior, such as an agent's actions requiring additional human confirmation the moment its certificate is flagged as potentially compromised, rather than treating agent certificate incidents identically to a standard service certificate compromise.

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 well-rehearsed revocation playbook matters increasingly as the schedule below compresses certificate lifetimes toward 47 days, since natural expiration alone will handle less of the risk reduction burden going forward, making the ability to execute rapid, deliberate revocation during a genuine compromise more important, not less.

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.