Certificate Revocation Lists (CRLs):
What They Are and How They Work

Issuing a certificate is only half of the trust equation. The other half is being able to un-trust it before its natural expiration, if something goes wrong. Certificate Revocation Lists are one of the two primary mechanisms, alongside OCSP covered elsewhere in this series, that make revocation actually enforceable rather than merely theoretical. This article explains what a CRL is, how it works, and where it still fits into a modern PKI.

The Problem Revocation Solves

A certificate's expiration date is a known, predictable event, but plenty of situations demand invalidating a certificate long before that date arrives: a private key gets exposed, a certificate was issued in error, an employee holding a client certificate leaves the organization, or a Certificate Authority itself is compromised. Without a revocation mechanism, any of these situations would leave a valid-looking, cryptographically sound certificate usable by an attacker or an unauthorized party until its original expiration date, regardless of how much time remained.

What a CRL Actually Contains

A Certificate Revocation List is a file, signed by the issuing CA, listing the serial numbers of every certificate that CA has revoked, along with the date and time of revocation and, often, a stated reason such as key compromise, superseded, or cessation of operation. The CRL itself carries an expiration and is reissued periodically, sometimes daily, sometimes more frequently, and clients checking certificate validity are expected to fetch the current CRL and check whether the certificate's serial number appears on it.

How Clients Use CRLs in Practice

Each certificate typically includes a CRL Distribution Point extension, a URL pointing to where the current CRL for that issuing CA can be downloaded. A client performing full certificate validation is expected to fetch this list and check the certificate's serial number against it before trusting the connection. In practice, full CRL checking by every client for every connection is relatively rare in consumer browsers today, partly because CRLs for large CAs can grow substantial in size, and partly because OCSP and OCSP stapling have become the more commonly used real-time alternative for many use cases.

The Practical Limitations of CRLs

CRLs come with real tradeoffs. They can grow large for CAs that issue and revoke certificates at high volume, making them slow to download and parse. They are only as fresh as their last publication, meaning a certificate revoked minutes after a CRL was published will not appear as revoked until the next CRL is issued, creating a window where a just-revoked certificate could still be treated as valid. And because fetching a CRL is itself a separate network request, distinct from the connection being validated, it introduces both latency and a privacy consideration, since the CA can observe which CRLs are being requested and roughly infer which certificates are being checked.

Where CRLs Still Make Sense

Despite these limitations, CRLs remain widely used, particularly within enterprise and government PKI deployments, code signing infrastructure, and situations where a complete, verifiable, offline-checkable list of revocations is preferable to a real-time query-based system. Some environments specifically require CRL support for compliance reasons, and CRLs remain useful as a fallback mechanism even in systems that primarily rely on OCSP, since a client unable to reach an OCSP responder can sometimes fall back to a cached or downloaded CRL.

Revocation in Automated, AI-Driven Environments

As certificate issuance and consumption increasingly happen through automated pipelines and AI-driven services rather than manual human validation, the mechanics of revocation checking need to keep pace with that automation. A CRL-based approach that assumes occasional, human-paced validation checks does not map cleanly onto an environment where thousands of short-lived machine and AI agent certificates are being issued and potentially revoked continuously. This is a meaningful part of why many high-velocity, automation-heavy environments lean more heavily on OCSP or on issuing certificates with such short lifespans that revocation becomes less critical, since the certificate will expire naturally within days regardless.

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

Shorter certificate lifespans, driven by the schedule below, reduce how much any single organization has to lean on revocation in the first place, since a compromised certificate with a 47-day maximum lifetime is a smaller window of exposure than one that could otherwise remain valid for a year.

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.