Best Practices for Using SSL (Publicly Trusted TLS) Certificates

Contents

Sources

[1] https://www.mozilla.org/en-US/about/governance/policies/security-group/certs/policy/#613-delayed-revocation
[2] https://googlechrome.github.io/chromerootprogram/

Automation

Automating SSL certificate management through protocols such as ACME has become both a technical and operational necessity for any organisation. It is not merely a recommended best practice, but the only sustainable approach given the ongoing regulatory developments within the public PKI ecosystem.

The CA/B Forum (Certification Authority/Browser Forum) has formalised a ***validity reduction schedule*** that progressively shortens the maximum lifespan of publicly trusted SSL certificates. Only a few years ago, certificates could remain valid for up to three years; today, that period is limited to 398 days. From mid-March 2026, the maximum validity will drop to 200 days. The roadmap also foresees further reductions – to 100 days on 15 March 2027 and to 47 days on 15 March 2029. This decision addresses growing **security requirements**: shorter certificate lifespans reduce the window of opportunity for attackers to exploit compromised keys and encourage organisations to maintain up-to-date certificate inventories.

Browser vendors' Root Store Programs have adopted increasingly stringent requirements. No trusted Certification Authority can disregard these directives without risking removal from browser Root Stores, which means that **every organisation relying on SSL certificates** must **adapt to the evolving** certificate management **policies**.

In the emerging regulatory landscape, manual certificate management will soon become unsustainable. Consider an organisation with 100 websites that currently renews annual certificates: it must perform 100 renewals per year. With 100-day certificates, that number rises to 400 renewals; with 45-day certificates, to 800. Each manual renewal requires CSR generation, domain validation, download, installation and certificate verification – all tasks that significantly increase the likelihood of human error and service downtime.

Automation with ACME

Automation via **ACME** already resolves these operational challenges by **managing the entire certificate lifecycle** – from request and domain validation to issuance, installation and renewal. Validation is scheduled automatically, eliminating the need for manual intervention. Automatic renewal ensures that certificates are replaced before they expire, typically maintaining a 30-day safety margin. Finally, automatic certificate installation – supported by several ACME clients – allows organisations to eliminate manual steps entirely.

The **operational benefits are substantial** and measurable. Automation drastically reduces workload, eliminates human error, ensures continuous compliance and enhances security through shorter certificate lifespans.
From an economic perspective, the initial investment required to adopt and configure ACME quickly pays for itself. The often hidden cost of manual processes – in terms of staff time and downtime risk – far outweighs the expense of implementing ACME. A single incident caused by an expired certificate can easily cost more than the entire automation project.

The ecosystem of ACME tools has now reached maturity. Solutions such as Certbot, Win-ACME, Posh-ACME, acme.sh and many other open-source clients – all ready to use and free of charge – greatly simplify integration. Organisations can begin with pilot projects on non-critical systems, gain experience and then progressively extend the technology to production environments.

Recognising that this transition is neither optional nor deferrable – but rather a **defined regulatory obligation** – is essential. The validity reduction schedule has already been established and will be enforced, regardless of whether organisations using SSL certificates are prepared or not. Certification Authorities will no longer be able to issue certificates that exceed the validity limits set by regulation, and browsers will reject any certificates that do not comply. Organisations that fail to adapt risk facing operational emergencies, potentially serious service disruptions and increased costs.

We recommend launching a project without delay that includes the following steps:

- Preparing a complete inventory of all certificates in use within the organisation
- Identifying systems that would benefit from automation
- Selecting suitable tools and training the technical team
- Gradually implementing ACME

The time invested in automation today will prevent operational crises and service disruptions tomorrow.

Critical infrastructure

Actalis strongly advises against using SSL (Publicly Trusted TLS) certificates for critical infrastructure in sectors such as healthcare, energy, transport, financial services, telecommunications, water management and public administration.

The reason is operational and concerns service continuity. Public Certification Authorities (CAs) are subject to strict regulations requiring them, in certain circumstances, to revoke certificates within five working days – or even within 24 hours. Any sudden or unexpected revocation would immediately render all services depending on those certificates inaccessible, potentially leading to:

- **Critical service interruptions** with possible consequences for citizens' health and safety
- **Significant financial losses** for organisations and users, including potential penalties
- **Severe operational disruption** to the delivery of essential services
- **Risks to public safety** in sensitive sectors such as healthcare and transport
- **Serious reputational damage** to third parties

For all these reasons, critical infrastructures should instead use private certificates, issued by private Certification Authorities, which provide full control over timing and certificate lifecycle management.

It is important to note that, where required by regulation, a **Certification Authority is obliged** to revoke certificates regardless of whether they are used within critical infrastructure.

Certificates for internal use

Using publicly trusted SSL certificates for internal services – such as intranet sites or applications that do not interact directly with external users on the public internet – is strongly discouraged, as it presents several significant issues:

- **Domain validation requirements**: public Certification Authorities must verify domain ownership and control. For internal services using names such as intranet.local or private IP addresses (192.168.x.x, 10.x.x.x), this validation is impossible. You would therefore need to use publicly registered domains, potentially exposing sensitive information about your infrastructure.
- **Exposure in Certificate Transparency Logs**: all certificates issued by public Certification Authorities are mandatorily recorded in public CT Logs. This exposes the names of your internal servers, reveals details about your network architecture and potential attack targets, and remains permanently accessible.

- **Operational constraints**: public Certification Authorities enforce a progressively shorter maximum validity period (currently 398 days, decreasing to 200 days from March 2026, with further reductions to follow). This results in frequent renewals. Certificate revocation is mandatory in the event of private key compromise or any "defect" – as defined by CA/B Forum requirements – in the certificate profile. Additional restrictions apply to eligible domain names.
- **Security and segmentation risks**: mixing public and private certificates weakens the separation between external and internal perimeters. Internal security incidents can become publicly visible through CT Logs, and revocation mechanisms such as CRL or OCSP may fail on internal networks isolated from the public internet.

The need to enable SSL on internal sites can be easily met through a private Certification Authority, which can be implemented using a range of available technologies – including free solutions.

Certificate pinning

Certificate Pinning is a security technique in which a specific TLS certificate (known as a pin) belonging to the destination web server is hard-coded into a client application (typically an Android or iOS app). In some cases, the certificate of the issuing CA itself is hard-coded instead. Rather than accepting any certificate issued under a trusted Root CA, the client proceeds with the SSL/TLS connection only if the server's TLS certificate (or the issuing CA's certificate) matches the pinned value. This mechanism was originally designed to defend against Man-in-the-Middle (MITM) attacks, particularly those arising from the compromise of a Certification Authority – as occurred in the well-known DigiNotar [SD1] incident.

When properly implemented and managed, Certificate Pinning can be effective; however, it also carries disadvantages that must be carefully weighed – and these ultimately outweigh its benefits. In short, Certificate Pinning **should generally be avoided,** as it introduces considerable complexity and operational risk.

Implementing Certificate Pinning is notoriously risky and error-prone. It also entails substantial **maintenance overhead and reduced operational flexibility**. Any change to the server certificate (for example, during renewal, which is required periodically) or to the issuing Certification Authority (whose certificate also has a limited lifespan and may be subject to revocation) requires updating the client application with the new pin. Failure to update, or delays in doing so, can lead to service interruptions or connection failures for users of the affected applications.

Moreover, Certificate Pinning **undermines an organisation's ability to respond swiftly to certificate-related issues**, particularly in light of the CA/B Forum requirements that all trusted Certification Authorities must strictly observe.

These requirements oblige Certification Authorities to **revoke certificates within very short time frames** under certain circumstances. For instance, if a certificate is found to contain incorrect identification data, the issuing Authority must revoke it **within five days**. In other cases, such as a compromised private key or other security-related incidents, revocation must occur **within 24 hours.**

If an organisation has pinned a certificate that becomes subject to such a revocation, it must distribute updates to all affected applications within the same narrow timeframe to include the replacement certificate. This inherent conflict – where mandatory, rapid certificate changes collide with the complex process of updating tightly controlled client applications – is the primary technical reason for which Certificate Pinning **presents more risks than benefits** in most real-world scenarios.