CERTPANEL
powered by digicert

SSL / TLS Best Practices Checklist

The SSL/TLS Best Practices Checklist provides a comprehensive framework for implementing and maintaining secure TLS configurations across internet-facing environments such as websites and web applications. Organized into eight logical categories, the checklist offers specific, actionable guidance with clear importance ratings to help prioritize implementation efforts. This structured approach helps organizations systematically establish robust TLS implementations that protect against known vulnerabilities while maintaining operational efficiency.

Find more guides and information at https://certpanel.com/resources/ssl-tls-best-practices

Category: Private Keys

Use strong private keys

Importance: Critical **Applies To: All Websites**

Use at least 2048-bit RSA or 256-bit ECC/ECDSA keys to provide sufficient security. Consider ECC/ECDSA for better performance and security if supported by your website/server.

Protect private keys

Importance: Critical **Applies To: All Websites**

Restrict access to private keys to the smallest possible group of employees. Ensure keys cannot be accessed publicly (eg. exclude from code repositories or public web directories.) Use HSMs for critical systems.

Avoid reusing private keys

Importance: High **Applies To: All Websites**

Generate new private keys when renewing certificates to limit the exposure window if keys are compromised. To reduce risk, avoid using the same private key/certificate on multiple servers.

Category: Certificate Management

Use valid certificates from reliable CAs

Importance: Critical **Applies To: All Websites**

Select CAs with strong security posture and dependable support.

Use strong signature algorithms

Importance: Critical **Applies To: All Websites**

Ensure certificates use secure hash algorithms (SHA-256 or better). Avoid deprecated algorithms like SHA-1 or MD5.

Tips/More Info: Note: all publicly-trusted CAs already enforce these rules.

Install complete certificate chains

Importance: Critical **Applies To: All Websites**

Deploy complete certificate chains including all intermediate certificates (sometimes referred to as the "ca bundle") to avoid browser trust issues.

Automated certificate management

Importance: High **Applies To: All Websites**

Use automation for certificate issuance, installation, renewal, and monitoring to reduce risks of human error and expired certificates.

Tips/More Info: Tip: AutoInsta/1 SSL automates the entire certificate lifecycle.

Deploy certificates with proper hostname coverage

Importance: Critical **Applies To: All Websites**

Include all relevant domain names (including domains that redirect) in certificates (with and without www prefix). Include subdomains individually or as a wildcard.

Implement Certificate Transparency (CT) monitoring

Importance: High **Applies To: All Websites**

Monitor CT logs to detect unauthorized certificates issued for your domains.

Configure DNS CAA records

Importance: Medium **Applies To: All Websites**

Add Certificate Authority Authorization records to your DNS to control which CAs can issue certificates for your domains.

Automate certificate replacements for revocations

Importance: High **Applies To: All Websites**

In addition to automating renewals, ensure that your certificate automation can detect a revocation event and replace the certificate within 24 hours.

Tips/More Info: Tip: AutoInsta/1 SSL checks every 12 hours and replaces your certificate if there's a scheduled revocation.

Avoid HTTP public key pinning (HPKP)

Importance: High **Applies To: All Websites**

Key or certificate pinning is no longer recommended as it often results in security and availability problems.

Category: Protocol Configuration

Use only secure protocol versions (i.e., TLS 1.2 minimum)

Importance: Critical **Applies To: Websites w/Self-Managed Server**

Only support TLS 1.2 and TLS 1.3. Disable older protocols (SSL v2, SSL v3, TLS 1.0, TLS 1.1), which have known vulnerabilities.

Tips/More Info: Use Mozilla's SSL Configuration Generator: https://ssl-config.mozilla.org/ (for Windows Servers see https://learn.microsoft.com/en-us/windows/win32/secauthn/protocols-in-tls-ssl--schannel-ssp-

Use secure cipher suites

Importance: Critical **Applies To: Websites w/Self-Managed Server**

Prioritize Authenticated Encryption with Associated Data (AEAD) ciphers (CHACHA20_POLY1305, GCM, CCM). Remove weak, null, and export ciphers.

Tips/More Info: Use Mozilla's SSL Configuration Generator: https://ssl-config.mozilla.org/ or Windows Servers can use https://www.nartac.com/Products/IISCrypto

Prefer forward secrecy

Importance: High **Applies To: Websites w/Self-Managed Server**

Prioritize ciphers that enable forward secrecy (ECDHE, DHE, CECPQ1, all TLS 1.3 ciphers).

Tips/More Info: Use Mozilla's SSL Configuration Generator: https://ssl-config.mozilla.org/ or Windows Servers can use https://www.nartac.com/Products/IISCrypto

Configure secure key exchange

Importance: Critical **Applies To: Websites w/Self-Managed Server**

Use ECDHE with secp256r1 curve (p-256) or DHE (with at least 2048-bits of security) for forward secrecy.

Tips/More Info: Use Mozilla's SSL Configuration Generator: https://ssl-config.mozilla.org/ or Windows Servers can use https://www.nartac.com/Products/IISCrypto

Enable OCSP stapling

Importance: Medium **Applies To: Websites w/Self-Managed Server**

Configure Online Certificate Status Protocol (OCSP) stapling to deliver revocation information efficiently during the TLS handshake.

Tips/More Info: Use Mozilla's SSL Configuration Generator: https://ssl-config.mozilla.org/

Disable compression

Importance: Critical **Applies To: Websites w/Self-Managed Server**

Disable TLS compression to mitigate CRIME attacks. Address HTTP compression vulnerabilities (e.g., TIME and BREACH) at the application level.

Tips/More Info: Use Mozilla's SSL Configuration Generator: https://ssl-config.mozilla.org/

Configure proper Diffie-Hellman groups

Importance: High **Applies To: Websites w/Self-Managed Server**

Use standardized Diffie-Hellman groups with sufficient strength (2,048-bit) to prevent downgrade attacks.

Tips/More Info: Use Mozilla's SSL Configuration Generator: https://ssl-config.mozilla.org/

Implement post-quantum cryptography (PQC)

Importance: High **Applies To: All Websites**

Implement PQC algorithms to defend against harvest now decrypt later (HNDL) attacks. At present, this means enabling the X25519MLKEM768 hybrid key agreement algorithm.

Tips/More Info: See https://pq.cloudflareresearch.com/

Category: Monitoring & Maintenance

Implement continuous monitoring

Importance: Critical **Applies To: All Websites**

Monitor certificate expiration and configuration to prevent outages and security issues.

Tips/More Info: See https://certpanel.com/ssl-monitor

Be proactive with certificate renewals

Importance: High **Applies To: All Websites**

Renew certificates at least 30 days before expiration to allow time for testing and troubleshooting.

Implement crypto-agility

Importance: High **Applies To: All Websites**

Maintain ability to rapidly replace certificates and keys in response to cryptographic incidents like CA compromise or algorithm vulnerabilities.

Perform regular security assessments

Importance: High **Applies To: All Websites**

Regularily test your SSL/TLS configuration to identify vulnerabilities or misconfigurations.

Tips/More Info: Tip: SSL Monitor does this automatically: https://certpanel.com/ssl-monitor

Apply security patches promptly

Importance: Critical **Applies To: Websites w/Self-Managed Server**

Keep cryptographic libraries and server software up-to-date with security patches.

Category: Application Security

Deploy HTTP Strict Transport Security (HSTS)

Importance: High **Applies To: All Websites**

Implement HSTS headers to ensure browsers always use HTTPS connections to your site and prevent downgrade attacks.

Tips/More Info: Don't use HSTS if you have subdomains or applications that use HTTP.

Secure cookies

Importance: Critical **Applies To: All Websites**

Set cookies as secure and use HTTP-only flags when appropriate. Consider adding cryptographic validation for sensitive use cases.

Tips/More Info: See https://developer.mozilla.org/en-US/docs/Web/Security/Practical_implementation_guides/Cookies

Load all pages over HTTPS

Importance: Critical **Applies To: All Websites**

Configure your website/web application to load all pages over HTTPS. Redirect all HTTP URLs to HTTPS.

Eliminate mixed content

Importance: Critical **Applies To: All Websites**

Ensure all resources (scripts, images, stylesheets) are loaded over HTTPS to prevent man-in-the-middle (MitM) attacks.

Tips/More Info: SSL Monitor alerts you if your site has insecure content.

Deploy Content Security Policies

Importance: High **Applies To: All Websites**

Use Content Security Policy (CSP) response headers to restrict third-party content, mitigate cross-site scripting (XSS), and prevent mixed-content vulnerabilities.

Tips/More Info: Check your headers at https://securityheaders.com

Category: Incident Response

Implement timely certificate revocation

Importance: High **Applies To: All Websites**

Have clear procedures for revoking certificates in case of key compromise.

Define a crypto-incident response plan

Importance: High **Applies To: All Websites**

Document procedures for handling certificate and key compromises. (If a private key may have been compromised or exposed, you'll need to revoke all associated certificates immediately.)

Maintain backup CAs

Importance: Medium **Applies To: All Websites**

Use a multi-CA certificate provider or establish relationships with backup CAs to enable rapid transition in case of primary CA compromise.

Tips/More Info: Tip: CertPanel is a multi-CA provider!

Category: TLS Traffic Management

Secure private key transport for inspection

Importance: Optional **Applies To: Websites w/Self-Managed Server**

If TLS traffic inspection is required, establish secure methods to transport private keys to decryption devices.

Category: Performance Optimization

Use session resumption

Importance: Medium **Applies To: Websites w/Self-Managed Server**

Carefully implement session resumption to reduce handshake overhead and improve performance. (Misconfigurations affect performance and security.)

Optimize for network latency

Importance: Medium **Applies To: Websites w/Self-Managed Server**

Use HTTP/2, keep-alives, and Content Delivery Networks (CDNs) to minimize the latency impact of TLS handshakes.

Consider hardware acceleration

Importance: Optional **Applies To: Websites w/Self-Managed Server**

High traffic websites may want to use CPUs with hardware-accelerated AES support for better TLS performance.

Source: https://certpanel.com/resources/ssl-tls-best-practices