

What is Domain TLS

[Overview](#)

[How Domain TLS works](#)

[Difference with Apache SSL certificate storage](#)

[Certificate maintenance](#)

Notes:

- We introduced the Domain TLS feature in cPanel & WHM version 60.
 - On cPanel & WHM version 60.0.X and earlier, this feature stores and manages **all** of the server's certificates.
 - On cPanel & WHM version 60.0.X+1 and newer, this feature stores and manages **only** the server's verified certificates.
- The Domain TLS file structure may change in future versions.

Overview

The Domain TLS system stores and manages the server's verified certificates in a domain-indexed repository. This system also allows faster and more efficient management of SNI services for a user's domains. The system performs the following actions:

- Looks up the domain
- Finds the necessary certificate.
- Retrieves that certificate, key, and CA bundle for that domain name.

Warning:

As of cPanel & WHM version 68, we only support Transport Layer Security (TLS) protocol [version 1.2](#).

- We will only support applications that use [TLSv1.2](#).
- We **strongly** recommend that you enable [TLSv1.2](#) on your server.

How Domain TLS works

When you install a certificate for Apache, the system also copies the certificate into the Domain TLS for each domain of the Apache virtual host that the certificate secures. If the certificate secures an Apache virtual host with five domains, Domain TLS contains five copies of the certificate.

Domain TLS handles SNI functionality for the following services:

- `cpsrvd` — cPanel, WHM, and Webmail logins and interfaces.
- `cpdabd` — Calendar, Contacts, and Web Disk services.
- `exim` — Mail transfer and receiving services.
- `dovecot` — Mailbox service.

Difference with Apache SSL certificate storage

Apache's SSL certificate storage groups domains into virtual hosts, which the cPanel interface refers to as websites.

Domain TLS uses the domain name as a key and the certificate the domain uses as a value.

Also, most of cPanel & WHM classifies the `www.` subdomain as functionally equivalent to its parent domain. For example, the cPanel Store issues certificates for `example.com` that automatically include the `www.example.com` subdomain. Because TLS classifies every domain as a separate entity, Domain TLS classifies the `www.` subdomain and parent domain as separate items. This action causes Domain TLS to store each as a separate entry on the index.

Finally, Domain TLS does not contain any expired or invalid certificates that the Apache SSL certificate storage contains.

Certificate maintenance

During the upgrade to cPanel & WHM version 60, servers automatically copied current and valid certificates from the Apache SSL certificate storage to Domain TLS storage. Domain TLS does **not** copy expired or invalid certificates from Apache's SSL storage. As you install, manage, and delete certificates through cPanel & WHM user interfaces or API calls, the system automatically performs the necessary updates to the

Domain TLS index and certificate storage.

We do not currently provide a user interface to manage Domain TLS. However, as more services use this feature for SNI, we may investigate the need for and value of such an interface.

Additional documentation

Suggested documentation [For cPanel users](#) [For WHM users](#) [For developers](#)

- [What is Domain TLS](#)
- [Service Subdomains Use the cPanel Service SSL](#)
- [How to Prevent cPanel Users from Creating Certain Domains](#)
- [Guide to SSL](#)
- [How to Copy Addon Domains to a New Account](#)

- [Guide to SSL](#)
- [How Your Server Handles Domains and Virtual Hosts](#)
- [Troubleshoot SSL-Related Issues](#)
- [More about TLS and SSL](#)

- [What is Domain TLS](#)
- [Service Subdomains Use the cPanel Service SSL](#)
- [How to Prevent cPanel Users from Creating Certain Domains](#)
- [Guide to SSL](#)
- [How to Copy Addon Domains to a New Account](#)

- [UAPI Functions - SSL::rebuild_mail_sni_config](#)
- [UAPI Functions - SSL::disable_mail_sni](#)
- [UAPI Functions - SSL::enable_mail_sni](#)
- [UAPI Functions - SSL::is_mail_sni_supported](#)
- [UAPI Functions - SSL::mail_sni_status](#)