SSL FAQ and Troubleshooting

What is SSL/TLS?

SSL performs several functions to help secure your server.

SSL/TLS (Secure Sockets Layer/Transport Layer Security) encrypts information between a visitor's browser and a server. These protocols protect against electronic eavesdroppers. This also protects sensitive data (for example, credit card numbers, and login information) that you transmit over the Internet with SSL/TLS.

Both of these protocols initiate a "handshake", during which your server and the user's computer agree on specific conditions. These conditions include a set of public and private keys that the two computers use to encrypt and decrypt messages that they send during the secure session.

You can set up SSL/TLS for your server in cPanel's *SSL/TLS* interface *(cPanel >> Home >> Security >> SSL/TLS)*. This interface allows you to configure how SSL/TLS certificates run on your server.

> **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.

What is an SSL certificate?

An SSL certificate is an electronic document that uses the `.crt` file extension. This document binds a public key to an identity that consists of an email address, a company, and a location. The authentication process relies on this essential electronic document.

SSL certificates provide public information about the security of a domain, server, or service. The certificate consists of the following two parts to protect sensitive data:

- Encryption — Encodes data to ensure that if someone intercepts the transmission, that they cannot understand it.
- Identification verification — Ensures that you connect to the correct server.

What is a Certificate Authority (CA) bundle?

A Certificate Authority (CA) bundle file contains the following details about the SSL certificate:

- From whom it was issued.
- Any certificates of the authority that issued the CA bundle.
- The "chain of trust" for the issuer.

> **Note:**
> A CA can vouch for other CAs, which results in a "chain of trust." In order for a CA to sell certificates, another CA must vouch for them.

- Certificate revocation lists (CRLs).

Browsers have a built-in list of trusted certificate authorities, and they use the list to determine whether to trust an authority.

What is a CAA record?

A CAA (Certification Authority Authorization) record specifies which CAs may issue certificates for a domain. If no CAA records exist for a domain, all CAs can issue certificates for that domain. If conflicting CAA records already exist, remove the current CAA records or add one for the desired CAA.

For example, a CAA record for Comodo would resemble the following example, where `example.com` represents the domain name:

```
example.com. 86400 IN CAA 0 issue "comodoca.com"
```

Similarly, a CAA record for Let's Encrypt would resemble the following example, where `example.com` represents the domain name:

```
example.com. 86400 IN CAA 0 issue "letsencrypt.com"
```

You can manage CAA records through WHM's *Edit DNS Zone* interface (*WHM >> Home >> DNS Functions >> Edit DNS Zone*) or through cPanel's *Zone Editor* interface (*cPanel >> Home >> Domains >> Zone Editor*).

For more information about a CA's requirements, read their documentation.

What are limitations of SSL/TLS?

SSL certificates review domain names literally. For example, `www.example.com` and `example.com` are two different domains in relation to SSL.

How do I revoke an SSL certificate?

We do **not** support the revocation of certificates through cPanel & WHM at this time.

AutoSSL

Which domains does AutoSSL add to the certificate first?

AutoSSL uses a sort algorithm to establish which domains to add to the certificate first. This sort order ensures that the system adds the domains that customers will most likely visit to the certificate first. For example, customers most likely intend to navigate to `example.com` versus `www.subdomain.example.com`.

> **Note:**
> This function assumes that all of the fully qualified domain names (FQDNs) resolve to the same virtual host.

This sort order ensures that the system adds the domains that users will most likely visit to the certificate first. The default sort algorithm prioritizes

domains in the following order:

1. Any FQDNs that the virtual host's current SSL certificate secures.
2. The primary domain on the cPanel account and then its `www.` and `mail.` subdomains.
3. Each addon domain followed by its `www.` and `mail.` subdomains. For example: A cPanel user called `example` (whose primary domain is `example.com`), creates an addon domain called `foo.com`. This addon domain, like all cPanel addon domains, exists on a separate virtual host with a subdomain `foo.example.com`. In this case, the system prioritizes `foo.com` over `foo.example.com`.
4. Domains with fewer dots. For example, prioritize `foo.com` ahead of `www.foo.com`.
5. Subdomains: `www`, `mail`, `whm` (if reseller), `webmail`, `cpanel`, `autodiscover`, `webdisk`.
6. Shorter domains.
7. Apply lexicographical sort.

Does AutoSSL cover proxy subdomains?

In cPanel & WHM version 64 and later, AutoSSL adds proxy subdomains to the SSL certificate in accordance with the sort algorithm. For more information about proxy subdomains, read our Service and Proxy Subdomains documentation.

> **Note:**
> AutoSSL only adds the `whm` proxy subdomain to the SSL certificate for reseller accounts.

When does an AutoSSL cPanel-issued certificates expire?

A cPanel-issued AutoSSL certificate expires after 90 days. However, AutoSSL attempts to automatically replace that certificate before it expires.

What rate limits does Let's Encrypt impose?

cPanel & WHM ships with the cPanel (powered by Comodo) provider. To install the Let's Encrypt™ AutoSSL provider plugin, read our Let's Encrypt Plugin documentation.

> **Warnings:**
> - Certificates that Let's Encrypt provides through AutoSSL can secure a **maximum** of 100 subdomains per domain (Apache® virtual host).
> - Let's Encrypt issues one certificate per domain, and issues a maximum of 20 certificates per week. Each certificate can secure up to 100 subdomains of the domain on the certificate.
> - Let's Encrypt continues to issue up to 20 certificates per week, if you request more than 20 domain certificates.
> - Let's Encrypt uses the domain's alias (parked domain), **not** the main domain, as the common name for AutoSSL. To use the main domain as the common name for AutoSSL, you **must** use cPanel or another AutoSSL provider. For more information, consult the Let's Encrypt Community Support page.

Why won't Let's Encrypt issue a certificate for a virtual host list (website)?

Let's Encrypt **only** issues a certificate five times per week to a specific set of domains before it blocks any further certificates for that set of domains.

To work around this rate limitation, create an alias to a domain in the virtual host list (website) so that Let's Encrypt interprets the virtual host as a new set of domains.

Does AutoSSL renew certificates for wildcard domains?

No. A wildcard domain appears with an asterisk (`*`) before the domain name (for example, `*.example.com`). AutoSSL does **not** renew certificates that contain wildcard domains.

Does AutoSSL issue certificates for suspended accounts?

No. AutoSSL does **not** issue certificates for websites on suspended accounts. You must first activate the account in order for AutoSSL to issue a certificate.

What is SNI support?

SNI (Server Name Indication) support allows you to host multiple SSL certificates for different domains on the same IP address. At the start of the

"handshake" process, SNI indicates the hostname to which the client connects. Users who are on shared servers that support SNI can install their own certificates without a dedicated IP address.

In order to experience the full benefit of SNI, your server **must** run an operating system that supports this functionality, (for example, CentOS 6).

What is a multi-domain or UC/SAN SSL certificate?

Multi-domain certificates allow you to secure multiple, potentially unrelated domains with a single SSL certificate. This includes UC/SAN certificates and wildcard certificates. Unified Communications/Subject Alternate Name (UC/SAN) certificates allow you to specify a list of hostnames that the same SSL certificate protects.

> **Note:**
> You must reissue these certificates each time that you add a new hostname.

What is a wildcard SSL certificate?

A wildcard certificate allows you to install the same certificate on any number of subdomains if they share an IP address. You can apply a wildcard certificate to services in WHM's *Manage Service SSL Certificates* interface *(WHM >> Home >> Service Configuration >> Manage Service SSL Certificates*).

- For example, you can use a wildcard certificate for `*.example.com` to securely connect to `mail.example.com` and `www.example.com`, but not to `example.com`.
- The `root` user may install a wildcard certificate on a collection of subdomains that are associated with a single root domain on multiple IP addresses. If this configuration uses multiple IP addresses, a user on the server must **not** own the `root` domain.

What is the difference between a wildcard and a webserver certificate?

Webserver certificates only allow you to secure a single domain. Wildcard certificates allow you to secure a domain and an unlimited number of subdomains. For example, if you wish to secure `store.example.com` and `blog.example.com`, you can use a single wildcard certificate to do so. However, each subdomain requires its own dedicated IP address.

What is a shared SSL certificate? How do I install one?

A Shared SSL Certificate is an SSL certificate that is installed on the server's hostname. If the server administrator enables the Apache mod_userdir Tweak setting, all of the users on that server can use a Shared SSL Certificate to access their sites securely via their user directories. For example: `https://hostname.example.com/~username`

After you install the certificate, set the certificate as *shared* in WHM's *Manage SSL Hosts* interface *(WHM >> Home >> SSL/TLS >> Manage SSL Hosts)*.

Self-signed SSL certificates

What is a self-signed SSL certificate?

A self-signed SSL certificate does not verify the identity of the server. You can create your own self-signed SSL certificate in WHM's *Generate an SSL Certificate and Signing Request* interface *(WHM >> Home >> SSL/TLS >> Generate an SSL Certificate and Signing Request)*.

> **Notes:**
> - A self-signed certificate contains the label “self-signed”, which only self-identifies.
> - If you choose to use a self-signed SSL certificate, you can secure a connection to the site, but you **cannot** verify the identity of the site. As a result, browsers warn users about the authenticity of the server that they want to reach.

What is the difference between a self-signed certificate and a purchased SSL certificate?

Based on the needs of your website, you may decide to either create a self-signed certificate or purchase an SSL certificate. Browsers consider a a purchased SSL certificate to be more secure because they verify the identity of the server.

- If your site only handles minimally sensitive data, it may be appropriate to create your own self-signed certificate.
- If your site handles extremely sensitive data (such as credit card information), purchase an SSL certificate to create a more trustworthy connection for your customers.

Can I get a free cPanel-signed or self-signed SSL certificate for my cPanel DNSONLY server?

cPanel, Inc. does **not** offer free signed or self-signed hostname certificates for cPanel DNSONLY™ servers.

How to troubleshoot an SSL installation

The following sections describe some common certificate installation issues and how to fix them:

My certificate will not install — I receive a message about a certificate/key mismatch.

If you receive the `modulus mismatch` or `key file does not match the certificate` error messages, then the private key that you entered did not generate the certificate that you wish to install. The correct private key may exist in a different file.

WHM may automatically complete the *Private Key* text box when you attempt to install a certificate. To properly install the certificate, paste the private key that you generated in the *Private Key* text box in WHM's *Install an SSL Certificate on a Domain* interface (*WHM >> Home >> SSL/TLS >> Install an SSL Certificate on a Domain*).

My certificate will not install — I receive a message about a dedicated IP.

Without Server Name Indication (SNI) enabled, SSL only allows one certificate per IP address. Because each cPanel account uses a single IP ad dress, you can only assign one certificate per account. If you experience problems with a subdomain, assign a dedicated IP address to it, or enable SNI on the server.

For more information, read our Install an SSL Certificate on a Domain documentation.

My certificate installed, but my visitors receive warnings about a self-signed certificate.

Self-signed certificates typically cause the following behaviors:

- Most browsers do not trust self-signed certificates because the certificate only encrypts data, but does not verify identity.
- Because browsers do not trust these certificates, your visitors will see a warning message.
- If you do not want visitors to encounter this warning, purchase an SSL certificate from an SSL provider.
    - If you choose to do this, do **not** remove the installed self-signed certificate. Instead, purchase and install the additional certificate in WHM's *Install an SSL Certificate on a Domain* interface (*WHM >> Home >> SSL/TLS >> Install an SSL Certificate on a Domain*).

My certificate installed, but my visitors see a warning about a domain mismatch.

It is likely that your server contains a self-signed certificate or a signed certificate that does not match the domain name.

- This warning notifies visitors that the name on the certificate does not match the name of the domain that they tried to reach.
- This should not be a security issue when you log in to a site's cPanel interface.
- Before they proceed, visitors can check to ensure that the SSL certificate matches to the domain of the correct host.
- Visitors who are concerned about security should contact the host to make sure it is safe to proceed.

To identify your hosting provider, enter your domain name at `WhoIsHostingThis.com`

My certificate installed, but visitors who try to securely access other sites on the shared IP address can only see the site with an installed SSL certificate, not my default domain.

You may encounter this problem if your server hosts multiple sites that share an IP address but only one domain with an installed SSL certificate. Apache cannot serve unsecured websites through a secure protocol.

For example, your server uses the following setup:

IP address	Domain	SSL status
`1.2.3.4`	`example.com`	Insecure
`1.2.3.4`	`domain.com`	Secure
`9.8.7.6`	`example2.com`	Insecure

`9.8.7.6`	`domain2.com`	Insecure

If this setup resembles your shared IP address' domain structure, expect the following behavior:

> **Warning:**
> If you enter `https://` before a domain name, the browser uses the secure `HTTPS` protocol. If you enter `http://` before a domain name, the browser uses the **not** secure `HTTP` protocol.

Protocol	IP address or domain	Apache will serve:
`https://`	`1.2.3.4`	`domain.com`
`http://`	`1.2.3.4`	The default page redirect, or `example.com`.
`https://`	`9.8.7.6`	An error message. **Note:** Because Apache cannot serve an unsecured website with a secure protocol and no secure sites exist on the shared IP address, Apache serves an error message.
`http://`	`9.8.7.6`	`domain2.com`
`https://`	`example.com`	`domain.com` **Note:** Because Apache cannot serve an unsecured site with a secure protocol, Apache defaults to the secure website on the shared IP address.
`http://`	`example.com`	`example.com`
`https://`	`domain.com`	`domain.com`
`http://`	`domain.com`	`domain.com`

To allow visitors to visit an unsecured domain regardless of which type of protocol they enter, perform the following steps:

Basic Users Advanced Users

1. Navigate to WHM's *Install an SSL Certificate on a Domain* interface *(WHM >> Home >> SSL/TLS >> Install an SSL Certificate on a Domain)*.
2. Click *Browse Certificates*.
3. In the *Browse Account* menu, select `root`.
4. In the *Certificate* list, select the option for the server's hostname certificate.
5. Click *Use Certificate*.
6. In the *IP Address (non-user domains only)* menu, select the server's shared IP address.
7. Click *Install*.
8. Navigate to WHM's *Manage SSL Hosts* interface *(WHM >> Home >> SSL/TLS >> Manage SSL Hosts)*.
9. In the *Installed SSL Hosts* table, click *Make Primary* in the appropriate row for the server's hostname.
1. Navigate to WHM's *Include Editor* interface *(WHM >> Home >> Service Configuration >> Apache Configuration >> Include Editor)*:
2. Select the *Pre Virtual Host Include* option.
3. Select the Apache version from the menu. We recommend that you select *All Versions*.
4. Enter the following text in the available text box:

```
<VirtualHost IPADDRESS:443>
  ServerName HOSTNAME
  DocumentRoot /usr/local/apache/htdocs
  ServerAdmin EMAIL
  <IfModule mod_suphp.c>
    suPHP_UserGroup nobody nobody
  </IfModule>
 SSLEngine on
 SSLCertificateFile SSLCERTIFICATEFILE
 SSLCertificateKeyFile YOUR-SSLCERTIFICATEKEYFILE
 </VirtualHost>
```

5. Click *Proceed*
6. Click *Update.*

Note:
This example uses the following values:

- `IPADDRESS` represents your server's IP address.
- `HOSTNAME` represents your server's hostname.
- `EMAIL` represents your contact email address.
- `SSLCERTIFICATEFILE` represents the full file path to your SSL certificate.
- `SSLCERTIFICATEKEYFILE` represents the full file path to your SSL certificate's key.

Visitors now can access unsecured sites, even if they use a secure protocol. For example, if `example.com` is your default website, the system redirects a visitor who enters `https://1.2.3.4` in their web browser to `example.com` .

When I log in with https, I get a certificate mismatch warning. Is it okay to ignore this and log in?

Your web host likely uses a self-signed certificate, or a signed certificate that does not match your domain name. This warning exists to notify you that the name on the certificate does not match the name of the domain that you wish to visit.

Ensure that the SSL certificate matches a domain that belongs to your web host before you proceed. If this still concerns you, contact your web hosting provider to confirm that you can safely proceed.

What do I do if my system fails and I do not have my Trustwave authentication data in WHM?

If you have suffered a serious drive failure, you may lose this data.

If you can access the old drive, the system stores your authentication data in the `/root/.trustwavereqs` file.

Additional documentation

Suggested documentation For cPanel users For WHM users For developers

- The set-tls-settings Script
- The checkallsslcerts Script
- Manage AutoSSL
- Manage Service SSL Certificates
- Purchase and Install an SSL Certificate

- SSL TLS Wizard
- Install and Manage SSL for your site HTTPS
- SSL TLS Status
- Private Keys - KEY
- Certificates - CRT

- The set-tls-settings Script
- The checkallsslcerts Script
- Manage AutoSSL
- Manage Service SSL Certificates
- Purchase and Install an SSL Certificate

- UAPI Functions - WebVhosts::list_ssl_capable_domains
- cPanel API 1 Functions - SSL::deletecsr
- cPanel API 1 Functions - SSL::gencsr
- cPanel API 1 Functions - SSL::showcsr
- UAPI Functions - SSL::can_ssl_redirect