Securing your sites with LetsEncrypt

`Let's Encrypt` is the leading supplier of free & simple to obtain SSL certificates. It runs on the notion that obtaining an SSL certificate for your website should require minimal human interaction, with the goal to secure all sites on the internet. Over a billion certificates have already been issued using `Let's Encrypt`.

How it works

`Let's Encrypt` uses a 2 step process to issue a certificate - **domain validation** (to prove you own the domain) and then the ability to issue/renew/revoke certificates thereafter.

There a few distinct types of domain validation available, so you will need to first assess which method best suits your applicatio/ and needs.

Types of challenges

```
TO DO

note - ensure you have port 80 open, best practice

HTTP-01 -
What it is/How it works
Requirements
No more than 10 redirects deep
No redirects to IP addresses
Doesn't care about HTTPS validation (eg. if you have a self-signed)

Pros:

It’s easy to automate without extra knowledge about a domain’s configuration.
It allows hosting providers to issue certificates for domains CNAMEd to them.
It works with off-the-shelf web servers.
Cons:

It doesn’t work if your ISP blocks port 80 (this is rare, but some residential
ISPs do this).
Let’s Encrypt doesn’t let you use this challenge to issue wildcard certificates.
If you have multiple web servers, you have to make sure the file is available on
all of them.


DNS-01 - TXT value (link to SafeDNS)

If using an API, such as the Plesk Extension, this is quick and easy to add and
automate

Pros:
something something
```

```
You can use this challenge to issue certificates containing wildcard domain names.
It works well even if you have multiple web servers.
Cons:

Keeping API credentials on your web server is risky.
Your DNS provider might not offer an API.
Your DNS API may not provide information on propagation times.


TLS-ALPN-01 - least suitable
```

```
.. warning::
  Currently, if your sites/services  DDoSX/Webcel/WAF you will currently not be
able to use Let's Encrypt certificates
```

Linux

For Linux servers, the `certbot` tool is currently the most popular tool for issuing `Let's Encrypt` certificates in a hassle free way. Here, we will show you how to install `certbot` on **CentOS** & **Ubuntu** servers, but this will be available on most Linux distributions.

`Certbot` has additional plugins specifically for servers that run `apache` or `nginx` as the web service, so be sure to install the correct plugin for your needs.

```
.. warning::
  These plugins will amend your virtual host configurations, but may interfere
with any application rewrite rules you already have in place. Always ensure you
have backed up vital configuration files before use.
```

CentOS

Apache

Installation

You will need to have the **Epel** repository (or repo) enabled to install certbot. If not installed, run the following;

```
yum install epel-release
```

Next, install the following package from this repo. This will pull in additional packages automatically, such as mod_ssl if not already installed

```
yum install certbot-apache --enablerepo=epel
```

Issuing a certificate

As root (or using sudo if a sudo user), you can specify multiple domains/subdomains using the following syntax.

```
certbot --apache -d yourdomain.com -d www.youdomain.com
```

You can secure up to 100 domains using -d in the one command. The installation plugin will then ask you a few questions before proceding with the installation. This includes It will also ask if you want to add a redirect to https. If you select 'yes'. It will amend your apache vhost with a permanent redirect.

Nginx

Installation

```
yum install epel-release
```

Next, install the following package from this repo

```
yum install certbot-nginx --enablerepo=epel
```

Issuing a certificate

As root (or using sudo if a sudo user), you can specify multiple domains/subdomains using the following syntax.

```
certbot --nginx -d yourdomain.com -d www.youdomain.com
```

By default, this will append your nginx configuration file for the chosen domain a rewrite to https

```
      server {
    if ($host = shop.yourdomain.com) {
        return 301 https://$host$request_uri;
    } # managed by Certbot


        listen ip.ip.ip.ip:80;
```

```
        server_name shop.yourdomain.com;
    return 404; # managed by Certbot


}
```

If you wish to amend this yourself, you should chose the 'certonly' option, and manual specify the new certificates in your domain's nginx configuration file.

```
certbot certonly
```

Additional options

Here is a selection of additional flags/options that you can use, should you need a more granular installation.

- `certonly` - If you wish to install the certificate manually, this will provide you with the SSL component files;
- `--webroot` - If you have a non-standard document root that perhaps is obfuscated in your application, this is useful so that the HTTP-01 challenge file can be placed correctly
- `-d` - For specifying up to 100 domains/subdomains in the same command.
- `standalone` - Runs a webserver that binds to port `80`, so you may need to stop your current webservice
- `--agree-tos` - Automatically agree to the terms of service
- `--email` - To specify an address for registration/correspondence
- `--uir` - This enables a Content-Security-Policy in every request to *upgrade-insecure-requests*

Auto-Renewing certificates

Due to the short lifespan of the certificate, it introduces the risk of your certificates expiring at an inopportune time. Therefore you should look towards scheduling in **automatic renewal**.

There are two methods to achieve this: With a scheduled task (a cron job) or using an additional utility that comes with certbot.

Cron Method

The `certbot` utility offers a *renew* option that will check your installed certificates and renew any that are within a 30 day expiration period.

You can test this feature using the 'dry-run' option

```
certbot renew --dry-run
```

As root, you can then add a cron task with either of the following commands

```
crontab -e
or
crontab -u root -e
```

In it you can then set your domains to be checked for renewal. In this example it checks twice a month and writes to a log

```
[root@server ~]# crontab -l
0 0  */15 * 6 /usr/bin/certbot renew >> /var/log/certbot.log
```

This outputs information like...

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Processing /etc/letsencrypt/renewal/docs.yourdomain.com.conf
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

Processing /etc/letsencrypt/renewal/p.yourdomain.com.conf
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

Processing /etc/letsencrypt/renewal/shop.yourdomain.com.conf
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

The following certs are not due for renewal yet:
  /etc/letsencrypt/live/docs.yourdomain.com/fullchain.pem expires on 2020-10-05
(skipped)
  /etc/letsencrypt/live/p.yourdomain.com/fullchain.pem expires on 2020-10-05
(skipped)
  /etc/letsencrypt/live/shop.yourdomain.com/fullchain.pem expires on 2020-10-05
(skipped)
No renewals were attempted.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Certbot timer methods

The `certbot` package comes with a **timer** service that you can leave to run and automatically update your certificates. This is a systemd service, and can be enabled with the following;

```
[root@ ~]# systemctl enable --now certbot-renew.timer
Created symlink from /etc/systemd/system/timers.target.wants/certbot-renew.timer
to /usr/lib/systemd/system/certbot-renew.timer.

[root ~]# systemctl status certbot-renew.timer
● certbot-renew.timer - This is the timer to set the schedule for automated
renewals
   Loaded: loaded (/usr/lib/systemd/system/certbot-renew.timer; enabled; vendor
preset: disabled)
```

```
    Active: active (waiting) since Thu 2020-07-09 08:56:24 BST; 12s ago
```

Revoking certificates

To revoke a LetsEncrypt certificate, use the following command

```
certbot revoke (supply --cert-name or --cert-path)
```

You can obtain the cert-name/path with the 'certbot certificates' command, but this will usually be the domain name itself.

Ubuntu

You can install the certbot utility in Ubuntu using the official PPA from certbot. First, install the **software-properties-common** package, if you don't already have this

```
apt install software-properties-common
```

Next, install the repo, update the apt db and install the module

```
add-apt-repository ppa:certbot/certbot
apt update
apt install certbot python3-certbot-apache
```

There after you can use the same methods to install a certificate as previously mentioned <link to above)>

cPanel

Overview

cPanel/WHM offers a feature called `AutoSSL` that integrates with both `LetsEncrypt` and their default provider (`Sectigo`). This allows you to install and automatically renew certificates for your domains.

Limitations

The `Let's Encrypt` plugin will supply certificates much faster than the default provider (`Sectigo`), but will **not** cover certificates for your `cPanel`services (eg. Mail, Hostname, FTP).

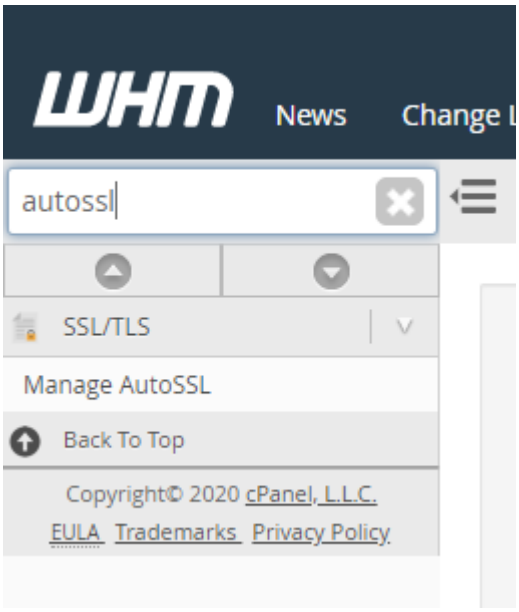
```
.. note::
  The plugin only allows for Wildcard certificates to be generated if you use WHM
as your DNS provider: If using SafeDNS or an external DNS provider then you will
need to do this manually using a tool like certbot.
```

Installation

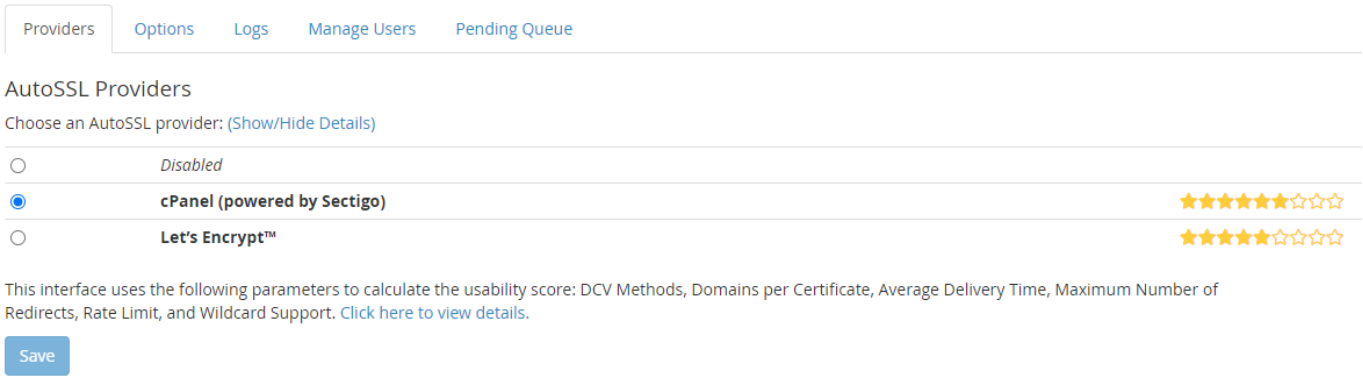
If not installed already, then you will need to ssh onto the server as root and run the following command

```
/usr/local/cpanel/scripts/install_lets_encrypt_autossl_provider
```

Next, open WHM and search for 'Manage AutoSSL'.



Here, you will have a list of providers, and `Let's Encrypt` will be one of them.



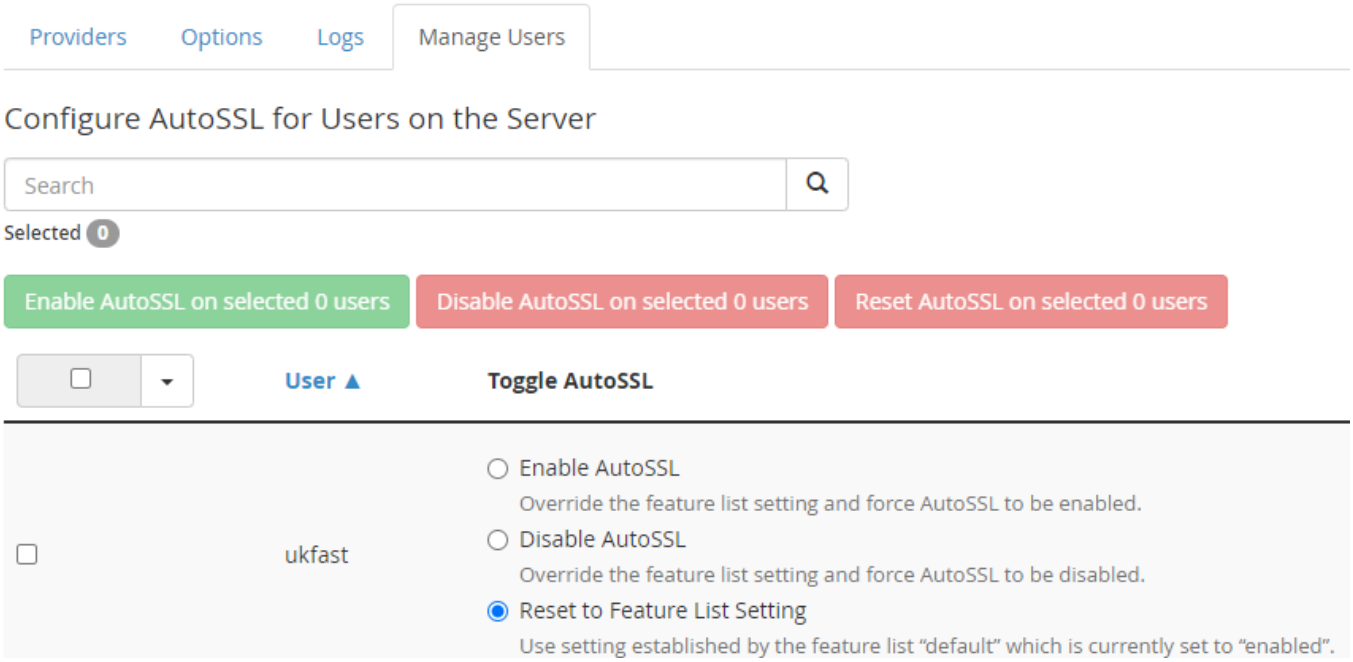
Select this provider, and after agreeing to the terms of service this will be available for you to use.

```
.. note::
  More information on this plugin is available in the WHM plugin documentation -
https://docs.cpanel.net/knowledge-base/third-party/the-lets-encrypt-plugin/86/
```

Issuing a certificate with AutoSSL

Once you have selected `Let'sEncrypt` as a provider, it's time to generate certificates for your domains.

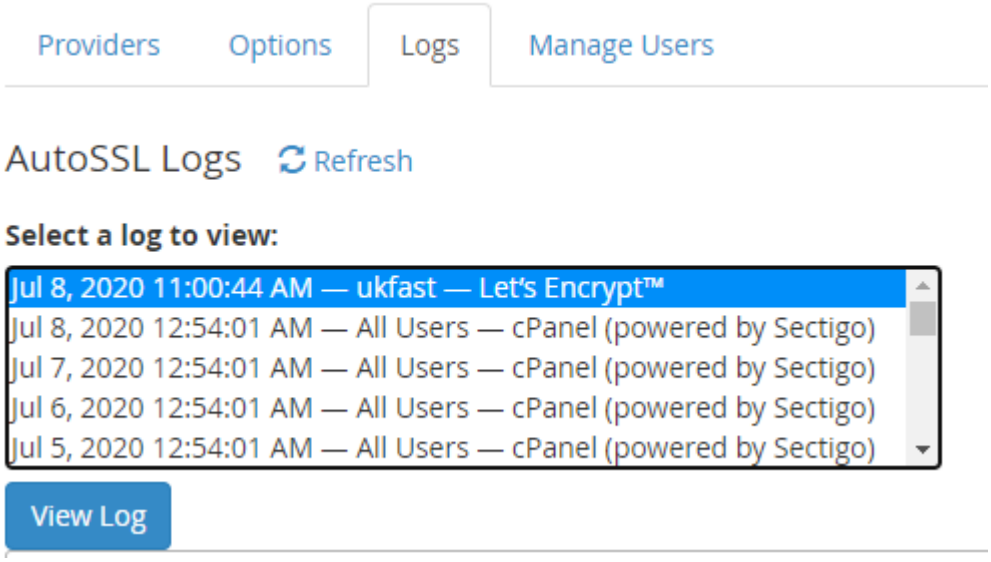
In the AutoSSL section of WHM, click on the Manage Users tab. Here you will have both global and per account options for enabling/disabling AutoSSL.



Once enabled, certificates will be automatically renewed close to the expiration date of the SSL

Troubleshooting

If you are having issues with generating a certificate, the first place you should check is the **logs** tab in AutoSSL



This should highlight any Let's Encrypt challenge issues you may have. Beyond this, you can raise a support ticket and we can help identify the underlying issues with you.

Plesk

https://www.plesk.com/extensions/letsencrypt/

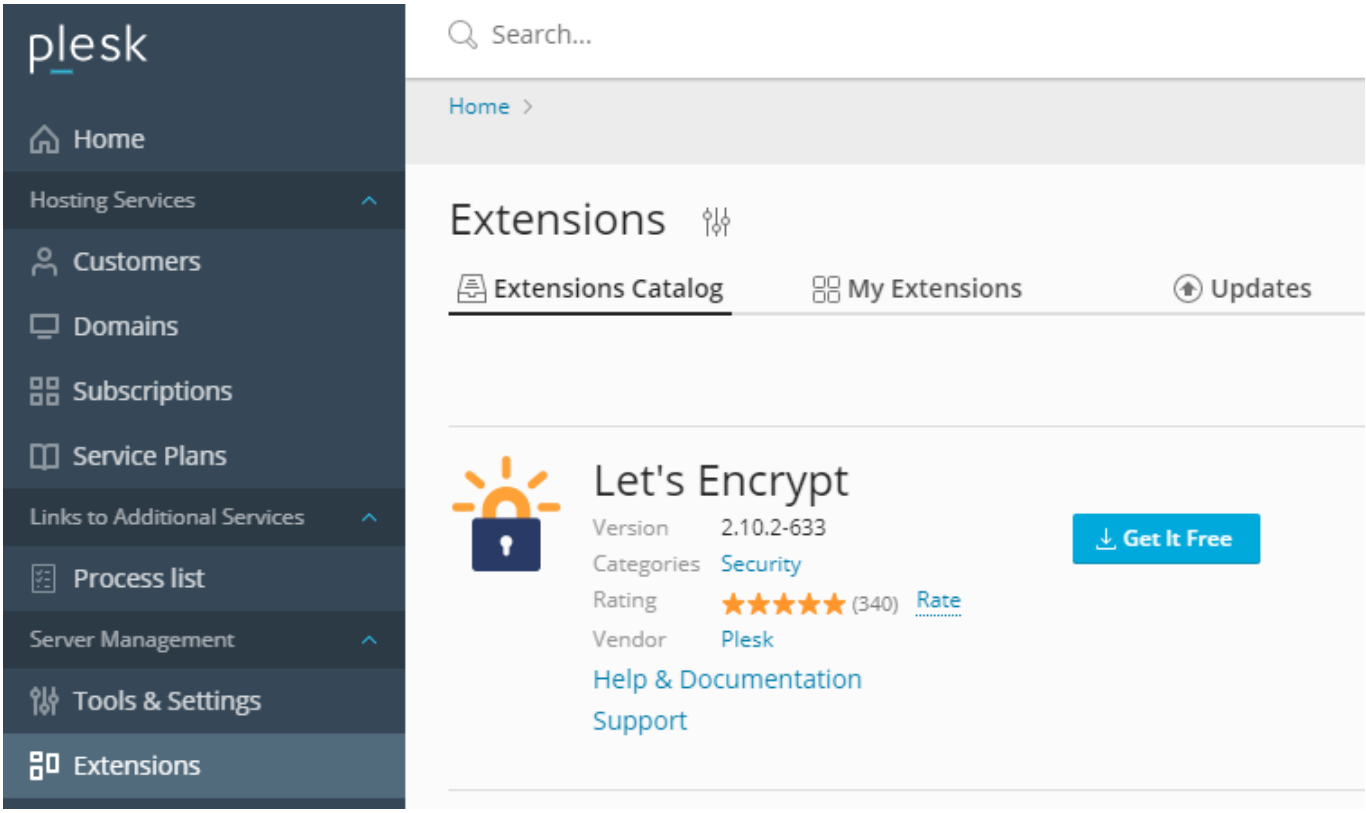
Overview

`Plesk` offers `Let's Encrypt` as an extension and makes it easy to obtain a certificate for your website.

Installation

Within `Plesk`, you can install this via the **Plesk Extensions** utility.

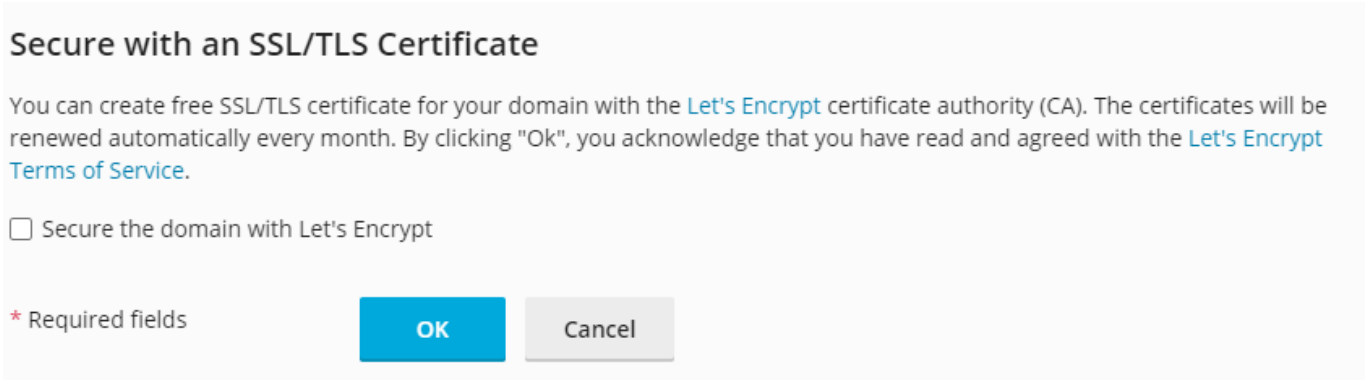
Simply search within their extension store for `Let's Encrypt` and click **Get it Free** to install.



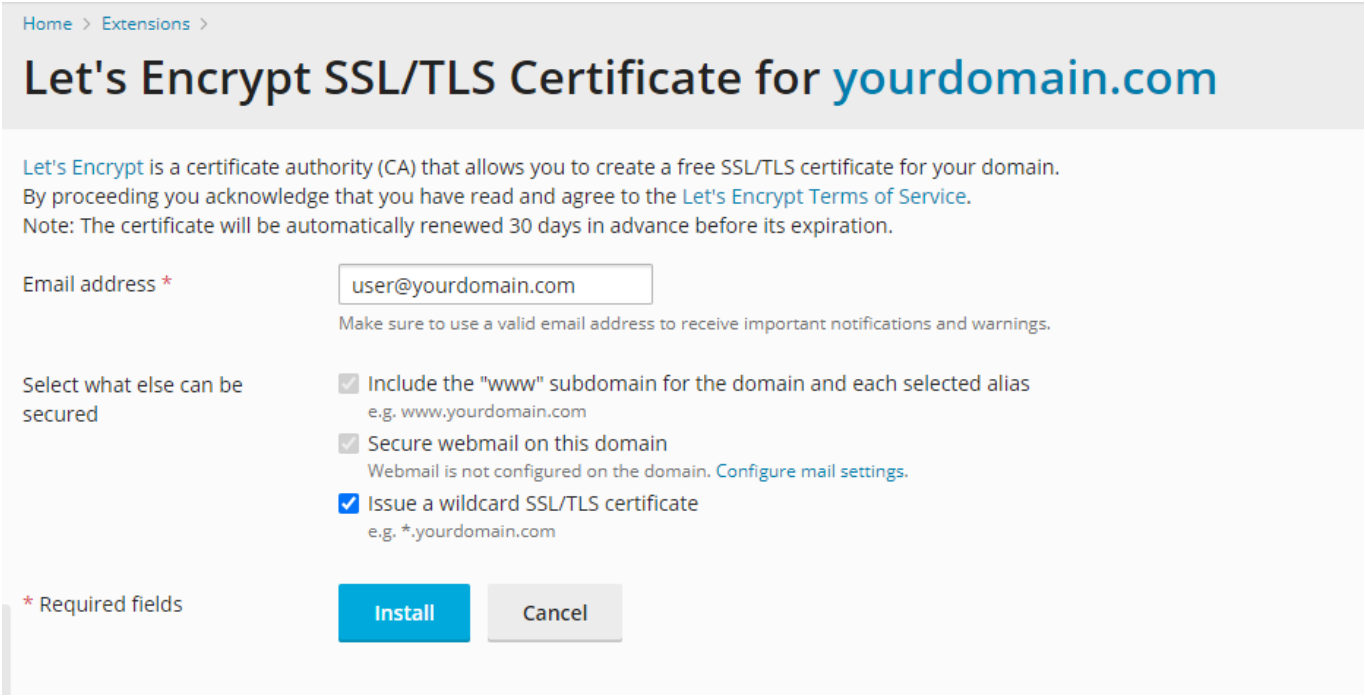
Once installed, it will appear within each domain's configuration page.

Issuing a certificate

For **new domains**, you can include `Let's Encrypt` functionality when creating the domain itself by ticking the *Secure the domain with Let's Encrypt* option

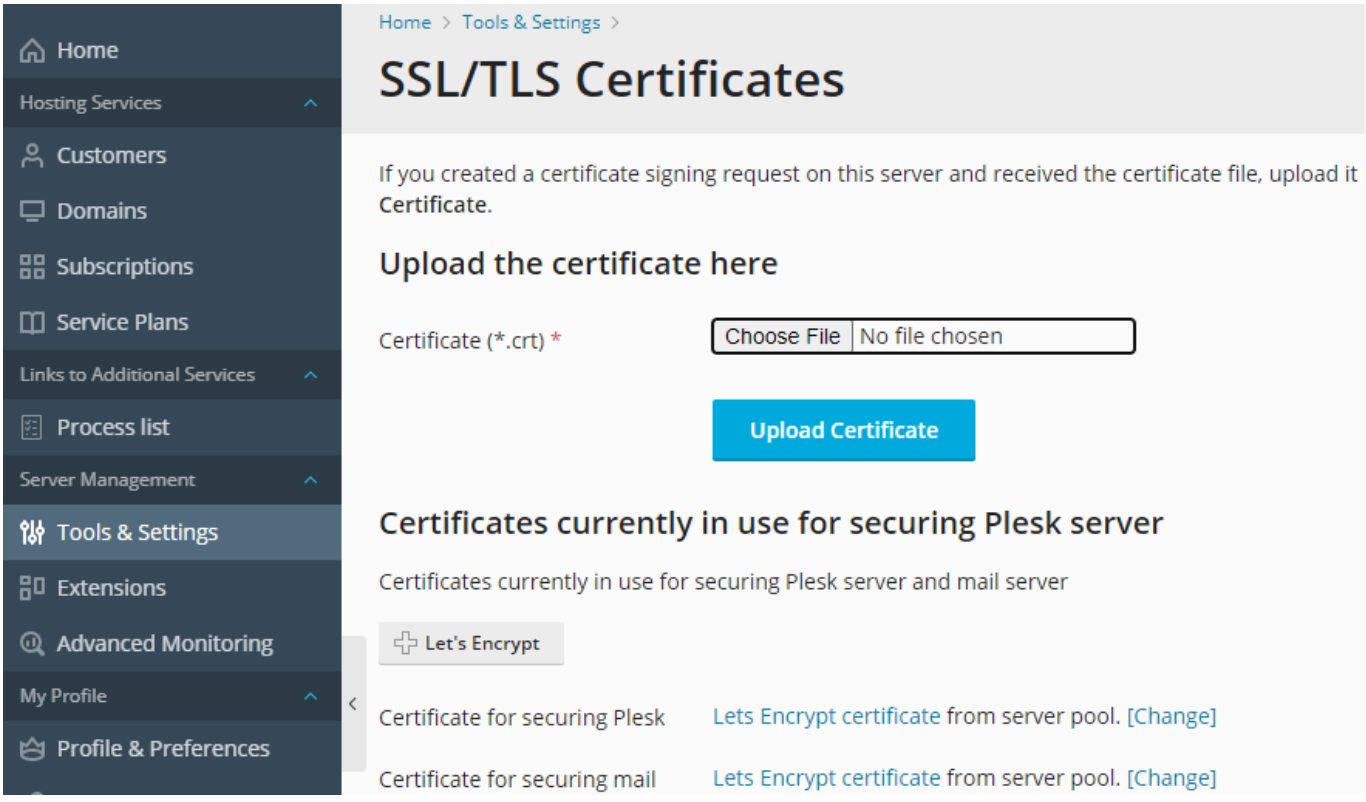


For **existing domains**, you can select your domain (or mutliple domains) within the `Let's Encrypt` extension itself. Here it provides a few options for what you would like to cover, such as 'www' and wildcard subdomaisn.



Once enabled, certificates will be automatically renewed close to the expiration date of the SSL

You can also secure your `Plesk` Panel and mail services using Let's Encrypt by selecting this in the SSL/TLS Certificates section of 'Tools & Settings'



Troubleshooting

Windows https://certbot.eff.org/lets-encrypt/windows-other https://weblog.west-wind.com/posts/2016/feb/22/using-lets-encrypt-with-iis-on-windows Overview Installation Getting a certificate Renewing a certificate Troubleshooting