Tutorial

How To Secure Apache with Let's Encrypt on Ubuntu

Introduction

Let's Encrypt is a Certificate Authority (CA) that facilitates obtaining and installing free TLS/SSL certificates, thereby enabling encrypted HTTPS on web servers. It streamlines the process by providing a software client, Certbot, that attempts to automate most (if not all) of the required steps. Currently, the entire process of obtaining and installing a certificate is fully automated on both Apache and Nginx.

In this guide, you'll use Certbot to obtain a free SSL certificate for Apache on Ubuntu and make sure this certificate is set up to renew automatically.

This tutorial uses a separate virtual host file instead of Apache's default configuration file for setting up the website that will be secured by Let's Encrypt. We recommend creating new Apache virtual host files for each domain hosted in a server because it helps to avoid common mistakes and maintains the default configuration files as a fallback setup.

How To Secure Apache with Let's Encrypt on Ubuntu

1. Installing Certbot
2. Checking your Apache Virtual Host Configuration
3. Allowing HTTPS Through the Firewall
4. Obtaining an SSL Certificate
5. Verifying Certbot Auto-Renewal

Prerequisites

To follow this tutorial, you will need:

- One Ubuntu server set up with a non-*root* user with sudo administrative privileges and firewall enabled.
- A fully registered domain name. This tutorial will use **your_domain** as an example throughout. You can purchase a domain name on Namecheap, get one for free on Freenom, or use the domain registrar of your choice.
- Both of the following DNS records set up for your server.
    - An A record with your_domain pointing to your server's public IP address.
    - An A record with www.your_domain pointing to your server's public IP address.
- Apache installed by following How To Install Apache on Ubuntu. Be sure that you have a virtual host file for your domain. This tutorial will use /etc/apache2/sites-available/your_domain.conf as an example.

Step 1 — Installing Certbot

To obtain an SSL certificate with Let's Encrypt, you need to install the Certbot software on your server. You'll use the default Ubuntu package repositories for that.

First, update the local package index:

sudo apt update

You need two packages: certbot, and python3-certbot-apache. The latter is a plugin that integrates Certbot with Apache, making it possible to automate obtaining a certificate and configuring HTTPS within your web server with a single command:

sudo apt install certbot python3-certbot-apache

You will be prompted to confirm the installation by pressing Y, then ENTER.

Certbot is now installed on your server. In the next step, you'll verify Apache's configuration to make sure your virtual host is set appropriately. This will ensure that the certbot client script will be able to detect your domains and reconfigure your web server to use your newly generated SSL certificate automatically.

Step 2 — Checking your Apache Virtual Host Configuration

To automatically obtain and configure SSL for your web server, Certbot needs to find the correct virtual host within your Apache configuration files. Your server domain name(s) will be retrieved from the ServerName and ServerAlias directives defined within your VirtualHost configuration block.

You should have a VirtualHost block set up for your domain at /etc/apache2/sites-available/your_domain.conf with the ServerName and also the ServerAlias directives already set appropriately.

To confirm this is set up, open the virtual host file for your domain using nano or your preferred text editor:

sudo nano /etc/apache2/sites-available/your_domain.conf

Find the existing ServerName and ServerAlias lines. They should be listed as follows:

/etc/apache2/sites-available/your_domain.conf

...

ServerName your_domain

ServerAlias www.your_domain

...

If you already have your ServerName and ServerAlias set up like this, you can exit your text editor and move on to the next step. If your current virtual host configuration doesn't match the example, update it accordingly. If you're using nano, you can exit by pressing CTRL+X, then Y and ENTER to confirm your changes, if any. Then, run the following command to validate your changes:

sudo apache2ctl configtest

You should receive Syntax OK as a response. If you get an error, reopen the virtual host file and check for any typos or missing characters. Once your configuration file's syntax is correct, reload Apache so that the changes take effect:

sudo systemctl reload apache2

With these changes, Certbot will be able to find the correct VirtualHost block and update it.

Next, you'll update the firewall to allow HTTPS traffic.

Step 3 — Allowing HTTPS Through the Firewall

If you have the UFW firewall enabled, as recommended by the prerequisite guides, you'll need to adjust the settings to allow HTTPS traffic. Upon installation, Apache registers a few different UFW application profiles. You can leverage the **Apache Full** profile to allow both HTTP and HTTPS traffic on your server.

To verify what kind of traffic is currently allowed on your server, check the status:

sudo ufw status

If you followed one of our Apache installation guides, you will have output similar to the following, meaning that only HTTP traffic on port 80 is currently allowed:

Output

Status: active

To	Action	From
OpenSSH	ALLOW	Anywhere
Apache	ALLOW	Anywhere
OpenSSH (v6)	ALLOW	Anywhere (v6)
Apache (v6)	ALLOW	Anywhere (v6)

To allow for HTTPS traffic, allow the "Apache Full" profile:

sudo ufw allow 'Apache Full'

Then delete the redundant "Apache" profile:

sudo ufw delete allow 'Apache'

Your status will display as the following:

sudo ufw status

Output

Status: active

To Action From

-- ------ ----

OpenSSH ALLOW Anywhere

Apache Full ALLOW Anywhere

OpenSSH (v6) ALLOW Anywhere (v6)

Apache Full (v6) ALLOW Anywhere (v6)

You are now ready to run Certbot and obtain your certificates.

Step 4 — Obtaining an SSL Certificate

Certbot provides a variety of ways to obtain SSL certificates through plugins. The Apache plugin will take care of reconfiguring Apache and reloading the configuration whenever necessary. To use this plugin, run the following:

sudo certbot --apache

This script will prompt you to answer a series of questions in order to configure your SSL certificate. First, it will ask you for a valid email address. This email will be used for renewal notifications and security notices:

Output

Saving debug log to /var/log/letsencrypt/letsencrypt.log

Enter email address (used for urgent renewal and security notices)

(Enter 'c' to cancel): you@your_domain

After providing a valid email address, press ENTER to proceed to the next step. You will then be prompted to confirm if you agree to Let's Encrypt terms of service. You can confirm by pressing Y and then ENTER:

- -

Please read the Terms of Service at

https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf. You must

agree in order to register with the ACME server at

https://acme-v02.api.letsencrypt.org/directory

- -

(Y)es/(N)o: Y

Next, you'll be asked if you would like to share your email with the Electronic Frontier Foundation to receive news and other information. If you do not want to subscribe to their content, write N. Otherwise, write Y then press ENTER to proceed to the next step:

- -

Would you be willing to share your email address with the Electronic Frontier

Foundation, a founding partner of the Let's Encrypt project and the non-profit

organization that develops Certbot? We'd like to send you email about our work

encrypting the web, EFF news, campaigns, and ways to support digital freedom.

- -

(Y)es/(N)o: N

The next step will prompt you to inform Certbot of which domains you'd like to activate HTTPS for. The listed domain names are automatically obtained from your Apache virtual host configuration, so it's important to make sure you have the correct ServerName and ServerAlias settings configured in your virtual host. If you'd like to enable HTTPS for all listed domain names (recommended), you can leave the prompt blank and press ENTER to proceed. Otherwise, select the domains you want to enable HTTPS for by listing each appropriate number, separated by commas and/ or spaces, then press ENTER:

Which names would you like to activate HTTPS for?

- -

1: your_domain

2: www.your_domain

- -

Select the appropriate numbers separated by commas and/or spaces, or leave input

blank to select all options shown (Enter 'c' to cancel):

After this step, Certbot's configuration is finished, and you will be presented with the final remarks about your new certificate and where to locate the generated files:

Output

Successfully received certificate.

Certificate is saved at: /etc/letsencrypt/live/your_domain/fullchain.pem

Key is saved at: /etc/letsencrypt/live/your_domain/privkey.pem

This certificate expires on 2022-07-10.

These files will be updated when the certificate renews.

Certbot has set up a scheduled task to automatically renew this certificate in the background.

Deploying certificate

Successfully deployed certificate for your_domain to /etc/apache2/sites-available/your_domain-le-ssl.conf

Successfully deployed certificate for www.your_domain.com to /etc/apache2/sites-available/your_domain-le-ssl.conf

Congratulations! You have successfully enabled HTTPS on https:/your_domain and https://www.your_domain.com

- -

If you like Certbot, please consider supporting our work by:

** Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate*

** Donating to EFF: https://eff.org/donate-le*

- -

Your certificate is now installed and loaded into Apache's configuration. Try reloading your website using https:// and notice your browser's security indicator. It should indicate that your site is properly secured, typically by a lock icon in the address bar.

You can use the SSL Labs Server Test to verify your certificate's grade and obtain detailed information about it, from the perspective of an external service.

In the next and final step, you'll test the auto-renewal feature of Certbot, which guarantees that your certificate will be renewed automatically before the expiration date.

Step 5 — Verifying Certbot Auto-Renewal

Let's Encrypt's certificates are only valid for ninety days. This is to encourage users to automate their certificate renewal process, as well as to ensure that misused certificates or stolen keys will expire sooner rather than later.

The certbot package you installed takes care of renewals by including a renew script to /etc/cron.d, which is managed by a systemctl service called certbot.timer. This script runs twice a day and will automatically renew any certificate that's within thirty days of expiration.

To check the status of this service and make sure it's active, run the following:

sudo systemctl status certbot.timer

Your output will be similar to the following:

Output

● certbot.timer - Run certbot twice daily

Loaded: loaded (/lib/systemd/system/certbot.timer; enabled; vendor preset:>

Active: active (waiting) since Mon 2022-04-11 20:52:46 UTC; 4min 3s ago

Trigger: Tue 2022-04-12 00:56:55 UTC; 4h 0min left

Triggers: ● certbot.service

Apr 11 20:52:46 jammy-encrypt systemd[1]: Started Run certbot twice daily.

To test the renewal process, you can do a dry run with certbot:

sudo certbot renew --dry-run

Output

Saving debug log to /var/log/letsencrypt/letsencrypt.log

- -

Processing /etc/letsencrypt/renewal/your_domain.conf

- -

Account registered.

Simulating renewal of an existing certificate for your_domain and www.your_domain.com

- -

Congratulations, all simulated renewals succeeded:

/etc/letsencrypt/live/your_domain/fullchain.pem (success)

- -

If you don’t receive any errors, you’re all set. When necessary, Certbot will renew your certificates and reload Apache to pick up the changes. If the automated renewal process ever fails, Let’s Encrypt will send a message to the email you specified, warning you when your certificate is about to expire.