How To Secure Apache with Let's Encrypt on Ubuntu 20.04

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 simplifies 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, we'll use Certbot to obtain a free SSL certificate for Apache on Ubuntu 20.04, 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. I 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.

Prerequisites

To follow this tutorial, you will need:

- One Ubuntu 20.04 server set up by following this initial server setup for Ubuntu 20.04 tutorial, including a sudo non-root user and a firewall.

- 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. You can follow this introduction to DigitalOcean DNS for details on how to add them.

    - 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 20.04. 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

In order to obtain an SSL certificate with Let's Encrypt, we'll first need to install the Certbot software on your server. We'll use the default Ubuntu package repositories for that.

We 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, we'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

In order to be able 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.

If you followed the virtual host setup tutorial, 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 check this 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 look like this:

```
...
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 you're using **nano**, you can exit by typing `CTRL+X`, then `Y` and `ENTER` to confirm.

If your current virtual host configuration doesn't match the example, update it accordingly. When you're done, save the file and quit the editor. Then, run the following command to validate your changes:

```
sudo apache2ctl configtest
```

You should get a **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, we'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. We 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, you can use:

```
sudo ufw status
```

If you have followed my Apache installation guide, your output should look something like this, meaning that only HTTP traffic on port **80** is currently allowed:

```
Status: active
```

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

To additionally let in HTTPS traffic, allow the **Apache Full** profile and delete the redundant **Apache** profile:

```
sudo allow "Apache Full"
sudo delete allow "Apache"
```

Your status will now look like this:

```
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 - Obtain 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, type 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 advisories:

```
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Plugins selected: Authenticator apache, Installer apache
Enter email address (used for urgent renewal and security notices) (Enter "c" to cancel): your@your_domain
```

After providing a valid email address, press ENTER to go to the next step. You will then be asked to confirm whether you agree to the Let's Encrypt terms of service. You can confirm by pressing A 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
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(A)gree/(C)ancel: A
```

Next, you will be asked if you would like to share your email with the Electronic Frontier Foundation to receive news and other information. If you don't want to subscribe to their content, type N. Otherwise, enter Y. Then, press ENTER to go 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 the domains for which you want to enable HTTPS. The domain names listed are automatically obtained from your Apache virtual host configuration, which is why it is important to ensure that you have the **ServerName** and **ServerAlias** directives configured in your virtual host. If you want to enable HTTPS for all listed domain names (recommended), you can leave the prompt blank and press ENTER. Otherwise, select the domains for which you want to enable HTTPS 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):
```

You'll see output like this:

```
Obtaining a new certificate
Performing the following challenges:
http-01 challenge for your_domain
http-01 challenge for www.your_domain
Enabled Apache rewrite module
Waiting for verification...
Cleaning up challenges
Created an SSL vhost at /etc/apache2/sites-available/your_domain-le-ssl.conf
Enabled Apache socache_shmcb module
Enabled Apache ssl module
Deploying Certificate to VirtualHost /etc/apache2/sites-available/your_domain-le-ssl.conf
Enabling available site: /etc/apache2/sites-available/your_domain-le-ssl.conf
Deploying Certificate to VirtualHost /etc/apache2/sites-available/your_domain-le-ssl.conf
```

Then you will be prompted to indicate whether or not you want HTTP traffic to be redirected to HTTPS. In practice, this means that if someone visits your website through unencrypted channels (HTTP), they will automatically be redirected to your website's HTTPS address. Choose `2` to enable redirect, or `1` if you want to keep HTTP and HTTPS as separate methods of accessing your website.

```
Please choose whether or not to redirect HTTP traffic to HTTPS, removing HTTP access.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1: No redirect - Make no further changes to the webserver configuration.
2: Redirect - Make all requests redirect to secure HTTPS access. Choose this for
new sites, or if you're confident your site works on HTTPS. You can undo this
change by editing your web server's configuration.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Select the appropriate number [1-2] then [enter] (press "c" to cancel): 2
```

After this step, the configuration of Certbot is complete and you will be presented with the last remarks on your new certificate, where to locate the generated files and how to test your configuration using an external tool that analyzes the authenticity of your certificate.:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Congratulations! You have successfully enabled https://your_domain and
https://www.your_domain

You should test your configuration at:
https://www.ssllabs.com/ssltest/analyze.html?d=your_domain
https://www.ssllabs.com/ssltest/analyze.html?d=www.your_domain
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

IMPORTANT NOTES:
- Congratulations! Your certificate and chain have been saved at:
/etc/letsencrypt/live/your_domain/fullchain.pem
Your key file has been saved at:
/etc/letsencrypt/live/your_domain/privkey.pem
Your cert will expire on 2020-07-27. To obtain a new or tweaked
version of this certificate in the future, simply run certbot again
with the "certonly" option. To non-interactively renew *all* of
your certificates, run "certbot renew"
- Your account credentials have been saved in your Certbot
configuration directory at /etc/letsencrypt. You should make a
secure backup of this folder now. This configuration directory will
also contain certificates and private keys obtained by Certbot so
making regular backups of this folder is ideal.
- 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 the Apache configuration. Try reloading your website using **https://** and notice your browser's security flag. It should tell you that your site is properly secured, usually by including a lock icon in the address bar.

You can use SSL Labs Server Test to check your certificate score and get detailed information about it, from the point of view an external service.

In the next and final step, we will test Certbot's auto-renew feature, which ensures that your certificate is automatically renewed before the expiration date.

Step 5 - Verifying the automatic renewal of Certbot

Let's Encrypt 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 expire as soon as possible.

The **certbot** package we installed supports renewals by including a renewal script for **/etc/cron.d**, which is managed by **systemctl** and the service called **certbot.timer**. This script runs twice a day and will automatically renew any certificate within thirty days of expiration.

To check the status of this service and make sure it is active and running, you can use:

```
sudo systemctl status certbot.timer
```

You will get output similar to this:

```
sudo systemctl status nginx
```

```
● certbot.timer - Run certbot twice daily
     Loaded: loaded (/lib/systemd/system/certbot.timer; enabled; vendor preset: enabled)
     Active: active (waiting) since Tue 2020-04-28 17:57:48 UTC; 17h ago
     Trigger:       Wed 2020-04-29 23:50:31 UTC; 12h left
     Triggers: ● certbot.service

Apr 28 17:57:48 fine-turtle systemd[1]: Started Run certbot twice daily.
```

To test the renewal process, you can give it a try with **certbot** :

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

If you don't see any errors, you're good to go. If necessary, Certbot will renew your certificates and reload Apache to pick up the changes. If the automated renewal process fails, Let's Encrypt will send a message to the email you specified, notifying you when your certificate is about to expire.
