BLOG TECH

Amir Rawdat of F5 January 28, 2021

Update: Using Free Let's Encrypt SSL/TLS Certificates with NGINX

security certificate, *Let's Encrypt*, *SSL/TLS*, *Certificate Authority (CA)*

Editor – The blog post detailing the original procedure for using Let's Encrypt with NGINX (from February 2016) redirects here. The instructions in that post are deprecated.

This post has been updated to eliminate reliance on `certbot-auto`*, which the Electronic Frontier Federation (EFF) deprecated in Certbot 1.10.0 for Debian and Ubuntu and in Cerbot 1.11.0 for all other operating systems. For additional details and alternate installation methods, see this post from the EFF.*

Also see our blog post from nginx.conf 2015, in which Peter Eckersley and Yan Zhu of the Electronic Frontier Foundation introduce the then-new Let's Encrypt certificate authority.

It's well known that SSL/TLS encryption of your website leads to higher search rankings and better security for your users. However, there are a number of barriers that have prevented website owners from adopting SSL.

Two of the biggest barriers have been the cost and the manual processes involved in getting a certificate. But now, with Let's Encrypt, they are no longer a concern. Let's Encrypt makes SSL/TLS encryption freely available to everyone.

Let's Encrypt is a free, automated, and open certificate authority (CA). Yes, that's right: SSL/TLS certificates for free. Certificates issued by Let's Encrypt are trusted by most browsers today, including older browsers such as Internet Explorer on Windows XP SP3. In addition, Let's Encrypt fully automates both issuing and renewing of certificates.

In this blog post, we cover how to use the Let's Encrypt client to generate certificates and how to automatically configure NGINX Open Source and NGINX Plus to use them.

How Let's Encrypt Works

Before issuing a certificate, Let's Encrypt validates ownership of your domain. The Let's Encrypt client, running on your host, creates a temporary file (a token) with the required information in it. The Let's Encrypt validation server then makes an HTTP request to retrieve the file and validates the token, which verifies that the DNS record for your domain resolves to the server running the Let's Encrypt client.

Prerequisites

Before starting with Let's Encrypt, you need to:

- Have NGINX or NGINX Plus installed.
- Own or control the registered domain name for the certificate. If you don't have a registered domain name, you can use a domain name registrar, such as GoDaddy or dnsexit.
- Create a DNS record that associates your domain name and your server's public IP address.

Note: We tested the procedure outlined in this blog post on Ubuntu 16.04 (Xenial).

1. Download the Let's Encrypt Client

First, download the Let's Encrypt client, `certbot`.

As mentioned just above, we tested the instructions on Ubuntu 16.04, and these are the appropriate commands on that platform:

```
$ apt-get update
$ sudo apt-get install certbot
$ apt-get install python-certbot-nginx
```

With Ubuntu 18.04 and later, substitute the Python 3 version:

```
$ apt-get update
$ sudo apt-get install certbot
$ apt-get install python3-certbot-nginx
```

2. Set Up NGINX

`certbot` can automatically configure NGINX for SSL/TLS. It looks for and modifies the `server` block in your NGINX configuration that contains a `server_name` directive with the domain name you're requesting a certificate for. In our example, the domain is **www.example.com**.

1. Assuming you're starting with a fresh NGINX install, use a text editor to create a file in the **/etc/nginx/conf.d** directory named ***domain-name*.conf** (so in our example, **www.example.com.conf**).

2. Specify your domain name (and variants, if any) with the `server_name` directive:

```
server {
    listen 80 default_server;
    listen [::]:80 default_server;
    root /var/www/html;
    server_name example.com www.example.com;
}
```

3. Save the file, then run this command to verify the syntax of your configuration and restart NGINX:

```
$ nginx -t && nginx -s reload
```

3. Obtain the SSL/TLS Certificate

1. Run the following command to generate certificates with the NGINX plug-in:

```
$ sudo certbot --nginx -d example.com -d www.example.com
```

2. Respond to prompts from `certbot` to configure your HTTPS settings, which involves entering your email address and agreeing to the Let's Encrypt terms of service.

   When certificate generation completes, NGINX reloads with the new settings. `certbot` generates a message indicating that certificate generation was successful and specifying the location of the certificate on your server.

```
Congratulations! You have successfully enabled https://example.com and
https://www.example.com

---------------------------------------------------------------------------
---------------
IMPORTANT NOTES:

Congratulations! Your certificate and chain have been saved at:
/etc/letsencrypt/live/example.com/fullchain.pem
Your key file has been saved at:
/etc/letsencrypt/live/example.com//privkey.pem
Your cert will expire on 2017-12-12.
```

   Note: Let's Encrypt certificates expire after 90 days (on 2017-12-12 in the example). For information about automatically renenwing certificates, see Automatic Renewal of Let's Encrypt Certificates below.

If you look at ***domain-name*.conf**, you see that `certbot` has modified it:

```
server {
    listen 80 default_server;
    listen [::]:80 default_server;
    root /var/www/html;
    server_name  example.com www.example.com;

    listen 443 ssl; # managed by Certbot

    # RSA certificate
    ssl_certificate /etc/letsencrypt/live/example.com/fullchain.pem; # managed by
Certbot
    ssl_certificate_key /etc/letsencrypt/live/example.com/privkey.pem; # managed
by Certbot

    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot

    # Redirect non-https traffic to https
    if ($scheme != "https") {
        return 301 https://$host$request_uri;
    } # managed by Certbot
}
```

Let's Encrypt certificates expire after 90 days. We encourage you to renew your certificates automatically. Here we add a `cron` job to an existing **crontab** file to do this.

1. Open the **crontab** file.

```
$ crontab -e
```

2. Add the `certbot` command to run daily. In this example, we run the command every day at noon. The command checks to see if the certificate on the server will expire within the next 30 days, and renews it if so. The `--quiet` directive tells `certbot` not to generate output.

```
0 12 * * * /usr/bin/certbot renew --quiet
```

3. Save and close the file. All installed certificates will be automatically renewed and reloaded.

Summary

We've installed the Let's Encrypt agent to generate SSL/TLS certificates for a registered domain name. We've configured NGINX to use the certificates and set up automatic certificate renewals. With Let's Encrypt certificates for NGINX and NGINX Plus, you can have a simple, secure website up and running within minutes.

To try out Let's Encrypt with NGINX Plus yourself, start your free 30-day trial today or contact us to discuss your use cases.

Magento Developer • 3 years ago

Can anyone make sure we need to check n renew via cron?
I have seen articles which says it's automatic renewal, please confirm.

20 • Reply • Share ›

Vaibhavraj R • 3 years ago

Awesome....
I added ssl to my site within 2 mins...

19 • Reply • Share ›

Sourav Ghosh • 2 years ago

Followed these steps but getting this error. Any idea what are we doing wrong?
Both @ & www A records are pointed to Digital Ocean droplet IP.

17 2 • Reply • Share ›

Faisal Memon ➚ Sourav Ghosh • 2 years ago

Is there anything in your NGINX error or access logs?

1 • Reply • Share ›

sunil951 • 3 years ago

nginx -t && nginx -s reload
nginx: [emerg] a duplicate default server for 0.0.0.0:80 in /etc/nginx/sites-enabled/default:22
nginx: configuration file /etc/nginx/nginx.conf test failed

This error coming

6 • Reply • Share ›

Owen Garrett Mod ➚ sunil951 • 3 years ago

Check your configuration... you may have two separate server blocks with `listen 80 default_server;`. You can only have one default server for requests to :80.

5 • Reply • Share ›

CharlES ➚ Owen Garrett • 2 months ago

Henrique Silvério • 3 years ago

Great! Thanks.

3 ^ | v • Reply • Share ›

İsmail Şener • a year ago

Awesome, it was really easy to implement. Thanks a lot!!

1 ^ | v • Reply • Share ›

Sylvain Max • 4 years ago

Should the --nginx option have to be added in the crontab renew command ? Else the challenge might not work ?

1 ^ | v • Reply • Share ›

Amir Rawdat Mod ➚ Sylvain Max • 4 years ago

The command in the crontab should work without the --nginx command. The crontab will renew the certificates if they are close to expiry and NGINX is already properly configured.

^ | v • Reply • Share ›

許震緯 ➚ Amir Rawdat • 3 years ago

Do you have to use post-hook to reload nginx yourself? Or does the renew command somehow reload nginx itself?

^ | v • Reply • Share ›

Amir Rawdat Mod ➚ 許震緯 • 3 years ago

NGINX will reload when you run the "sudo certbot --nginx -d example.com -d www.example.com" command. After that, renewing the SSL certificates will do the job, you dont need to reload NGINX again.

^ | v • Reply • Share ›

許震緯 ➚ Amir Rawdat • 3 years ago

great, thanks

^ | v • Reply • Share ›

Alexander MarThius • 20 days ago

Thanks Amir, you saved my day

^ | v • Reply • Share ›

Mohamed Aimen Hassen • 6 months ago

Your system is not supported by certbot-auto anymore. Certbot cannot be installed. Please visit https://certbot.eff.org/ to check for other alternatives. hello as you see it asks me to find an alternative I do not know what to do I delete the old site because it does not renew the certificate I thought it was a bad formatting ubunto 16

^ | v • Reply • Share ›

Tony Mauro Mod ➚ Mohamed Aimen Hassen • 4 months ago

Hi Mohamed -- I apologize for the delay in responding. It took a while to investigate the issue. We discovered that the author of `certbot-auto`, the Electronic Frontier Foundation, deprecated it for Debian-based systems in December 2020 and for all other OSs in January 2021. We have updated the installation instructions to eliminate reliance on `certbot-auto`. See also the editor's note at the start of this post.

^ | v • Reply • Share ›

Alaa • 9 months ago

Thank you very much sir !! it works !!

^ | v • Reply • Share ›

Sean Reifschneider • a year ago

This article adds the certbot PPA to your system, but be very, very careful with that. The certbot PPA includes incompatible versions of

around that. This has been open since Nov 2017, and it has caused issues with other software for me. https://github.com/certbot/...

▲ | ▼ • Reply • Share ›

Tony Mauro Mod ➦ Sean Reifschneider • 4 months ago

Hi Sean: Thanks for this warning, and sorry for not responding to it when you posted. We have just revised the instructions to eliminate use of certbot PPA.

▲ | ▼ • Reply • Share ›

Manish Maurya • 2 years ago

Awesome....
thank you

▲ | ▼ • Reply • Share ›

Francis Rodrigues • 3 years ago • edited

Could you please explain to me how can I validate/renew a Let's encrypt certificate if traffic is already on HTTPS in "options-ssl-nginx.conf" file?

▲ | ▼ • Reply • Share ›

Amir Rawdat Mod ➦ Francis Rodrigues • 3 years ago

Hello Francis,

I am not sure I understand your question. The options-ssl-nginx.conf file contains security parameters that you can change (for example what ssl protocol to use) and is not

STAY IN THE LOOP