Configuring SSL with Apache and Let's Encrypt (Part 1 - Let's Encrypt and CertBot)

Author	josh
Date	2017-10-19 22:47:46

Overview

With the robustness of modern webpages, HTTPS is the way to go for almost any website. Most of the time an individual is online, they're exchanging sensitive information like passwords and personal information. If you're logging into a site with a password, a securely encrypted connection is a must.

However, even informational sites should use encryption! If you're running a site for a small business, a topic that interests you, or even your own portfolio, your users can benefit from increased privacy and security when they connect using HTTPS rather than plain text HTTP.

Configuring SSL for a Website

Why SSL?

It's important to roughly understand how HTTPS works, and why using it will improve the privacy and security of your users. From the 10,000 foot view, what SSL does is encrypt all the traffic between the client (the user's browser) and the server (your web server). Everything beyond the domain name is encrypted, and therefore hidden from a man-in-the middle attacker.

From a security perspective, this prevents an adversary on the network from modifying content en-route to the user. Imagine you're an application developer and you have software available for download. You even go so far as to supply a SHA-256 fingerprint of your application so that users can verify its contents. If you're transmitting your application over HTTP, a malicious man-in-the-middle could modify the application *and* fingerprint in transit, giving your end user a nice dose of malware. But if you allows your users to download over a secure connection, the attacker could not see or change either component on its way to your (happy, malware free) consumer.

From a privacy perspective, HTTPS prevents a man-in-the-middle from snooping on what your user is viewing on your website. The only thing someone on the network could see is that they are connecting to your particular domain, no specific URLs or content. Imagine your user is doing research on a sensitive topic, or trying to download a piece of software that someone doesn't want them to have. To some extent, HTTPS protects that person's privacy by preventing a snooper from understanding what they see going over the wire.

Obtaining an SSL Certificate with Let's Encrypt

If you're sufficiently convinced that your website should use SSL, the good news is that you can set that up for free! The awesome folks over at Let's Encrypt have developed a service that allows you to obtain your very own, CA signed SSL certificate from the command line.

To actually get the certificate for your system, you'll want to use an ACME client called CertBot from the Electronic Frontier Foundation. CertBot uses the ACME protocol to automatically verify your ownership of the domain you want a certificate for and fetch that certificate from the Let's Encrypt service.

The CertBot website has instructions for all kinds of web servers and server operating systems. You can visit that site if you need further help for a particular combination, but in this article I'll show you how to get a certificate on Ubuntu Server for Apache (that's what I use). To install the `CertBot` application, run: `sudo apt-get update sudo apt-get install software-properties-common sudo add-apt-repository ppa:certbot/certbot sudo apt-get update sudo apt-get install python-certbot-apache` Once certbot is installed, you have two options for fetching a certificate. If you only have one website with a basic configuration, you can have CertBot automatically configure Apache to use the SSL certificate it will generate. This is done using: `sudo certbot --apache` That's all you need! You can now visit your website and you should see a little green lock in your browser indicating that you're connected via HTTPS with a valid certificate.

Using HTTPS to Keep Your Users Secure

It's important to encrypt traffic between your server and the clients browsing your site so that sensitive information cannot be snooped or manipulated by someone on the network. This article discusses *why* you should be using HTTPS for your websites, and how to get a free certificate with EFF's CertBot and Let's Encrypt. The Apache configuration is done automatically using the commands shown here.

However, if you're like me, you may prefer a little more fine-tuned control over your system configuration. The next article will go more in depth on hand-tuning Apache for your needs - we'll cover certificate-only CertBot usage, making SSL virtual hosts, and forcing secure connections.