Self Signed & Free Certificates

==========

What are self-signed certificates (OpenSSL)?

- **Generated using OpenSSL:** You can generate these certificates yourself without any cost.
- **Not Trusted by Browsers:** Browsers and operating systems do not recognize self-signed certificates as trusted because they are not signed by a recognized Certificate Authority (CA). This results in security warnings when users visit your site.
- **Use Cases:** Self-signed certificates are typically used for internal testing, development environments, or intranets where trust can be manually configured.

What are Let'sEncrypt Certificates?

- **Generated using Let's Encrypt:** Let's Encrypt is a free, automated, and open CA that provides SSL/TLS certificates.
- **Trusted by Browsers:** Certificates from Let's Encrypt are recognized and trusted by all major browsers, ensuring that users won't see security warnings when visiting your site.
- **Automation:** The process can be automated using tools like Certbot, which handles the issuance and renewal of certificates.
- **Free:** These certificates are provided at no cost.

==========

OpenSSL

using OpenSSL, you can generate a private key and a CSR to either:

1. **Send to a Certificate Authority (CA)** to obtain a certificate that will be trusted by browsers and other clients.
2. **Generate a self-signed certificate** for your own use, which will not be trusted by browsers by default but can be useful in certain scenarios.

/etc/ssl

perms for SSL files needs to be 600

Generating a private key and CSR

Generate a private key:

generating a private key is a prerequisite for creating a Certificate Signing Request (CSR). The private key is essential because it is used to sign the CSR and is part of the SSL/TLS certificate generation process.

```
openssl genpkey -algorithm RSA -pkeyopt rsa_keygen_bits:2048 -out keyfilename.key
```

Generate a CSR:

A Certificate Signing Request (CSR) is a block of encoded text that is given to a Certificate Authority (CA) when applying for an SSL/TLS certificate. The CSR contains information about the organization and the public key that will be included in the certificate.

```
openssl req -new -key keyfilename.key -out csrfilename.csr
```

Generating a certificate

Using the steps above, you will generate a private key and CSR file. We can then use these files to generate a self-signed certificate.

```
openssl x509 -req -days 365 -in csrfilename.csr -signkey keyfilename.key -out crtfilename.crt
```

==========

LetsEncrypt

Apache

https://www.digitalocean.com/community/tutorials/how-to-secure-apache-with-let-s-encrypt-on-centos-7

Running certbot for a single domain

```
sudo certbot --apache -d example.com
```

Running certbot for multiple domains (or subdomains)

```
sudo certbot --apache -d example.com -d www.example.com
```

Auto Renewal

Nginx

https://www.digitalocean.com/community/tutorials/how-to-secure-nginx-with-let-s-encrypt-on-ubuntu-20-04

```
sudo apt install certbot python3-certbot-nginx
```

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

Generate a certificate to be manually installed

```
certbot certonly -manual -d example.com -d example.com --webroot -w /path/to/doc/root
```

Auto Renewal

```
systemctl status certbot.timer
```

test renewal:

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

Revision #6
Created 2024-06-21 15:18:23 UTC by Daniel
Updated 2024-07-07 00:11:54 UTC by Daniel