Lael's World

천재는 노력하는 사람을 이길 수 없고, 노력하는 사람은 즐기는 자를 이길 수 없다.

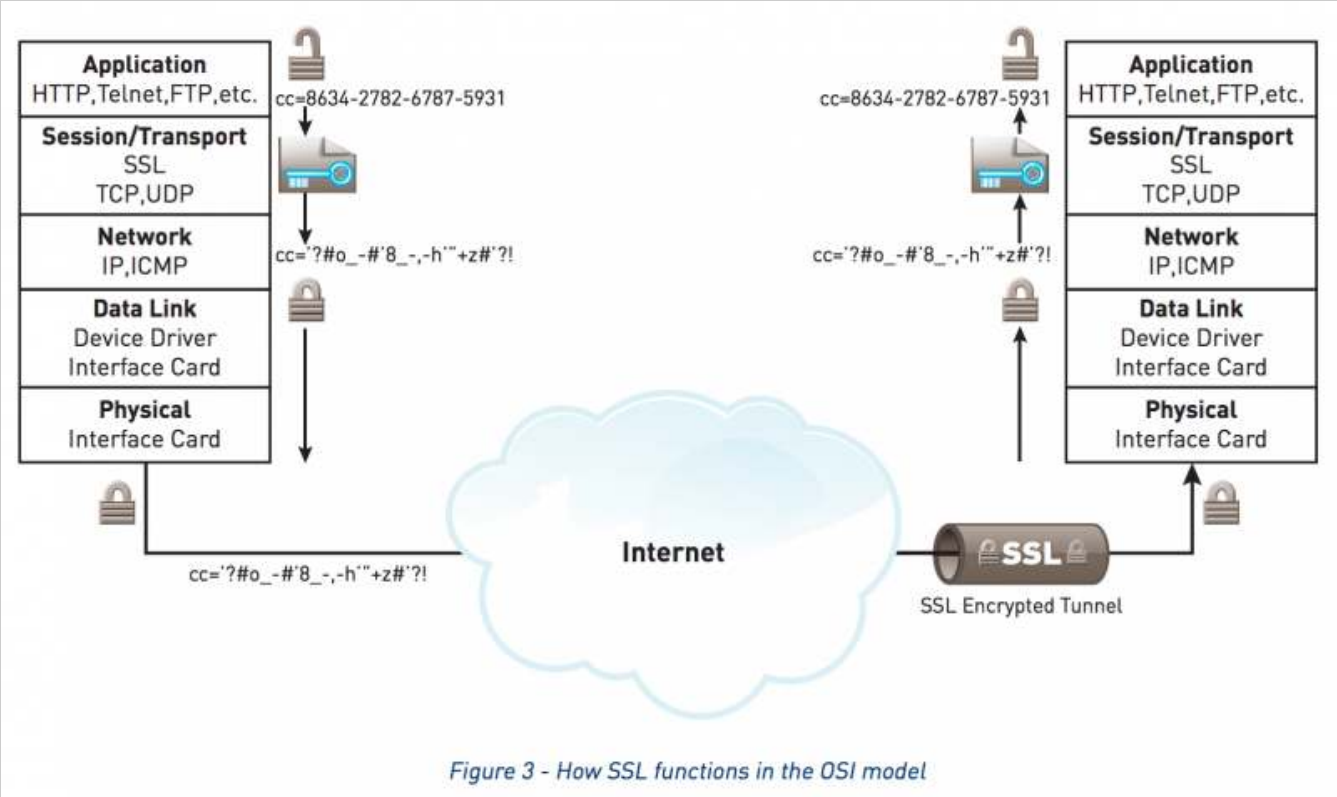
[Ubuntu] Let's Encrypt 를 사용하여 무료로 SSL 사이트를 구축하는 방법

HYEONG HWAN, MUN / 10월 2, 2016 / 미분류 / 148 comments

https://blog.lael.be/post/5107

웹은 계속 발전하고 있으며, 여러 새로운 기술이 끊임 없이 등장하고 있다.

이 글에서는 웹 기술 중 하나인 SSL (Secure Socket Layer) 에 대해서 이야기 해 보고자 한다.



< 그림 : OSI Model 에서 SSL 의 위치 >

정확히 말해서 SSL 은 전송계층과 (Transport Layer) 응용계층 (Application Layer) 사이에서 동작한다. Transport 에서 패킷을 받으면 -> SSL 에서 패킷의 암호를 해독하고 -> Application 에게 전달하는 것이다.

SSL 을 세션계층(Layer 5), 표현계층(Layer 6)으로 분류하는 사람도 있고, 응용계층(Layer 7)으로 분류하는 사람도 있다. 나름대로 주장에 대한 근거가 일리 있으니 여기서는 따로 정확히 분류하지는 않겠다. 확실한건 Transport(Layer 4) 와 Application(Layer 7) 의 사이라는 것.

I) 어떻게 SSL 은 암호화 통신을 하는가?

- 쉽게 말하자면 SSL 은 "보안인증서" 라고 말할 수 있다.

- 사람으로 비유하자면 보안인증서는 암호화코드가 내장된 주민등록증이라고 볼 수 있다.
요즘 대부분의 편의점이나 술집에서는 주민등록증 진위여부 판독기를 가지고 있다.

- 암호학에서 "비대칭키 암호화 방식" 이라는 것이 있다. 어떤 문자를 A 로 암호화하면 B 로만 해독할 수 있는 것이다.
일반적으로 암호화에 사용하는 것을 "암호화키" 라고 부르며 이것은 암호통신에서 공개되기 때문에 "공개키" 또는 "공개 암호화키", "Public Key" 라고도 부른다.

참조 (RSA_암호) : https://ko.wikipedia.org/wiki/RSA_%EC%95%94%ED%98%B8

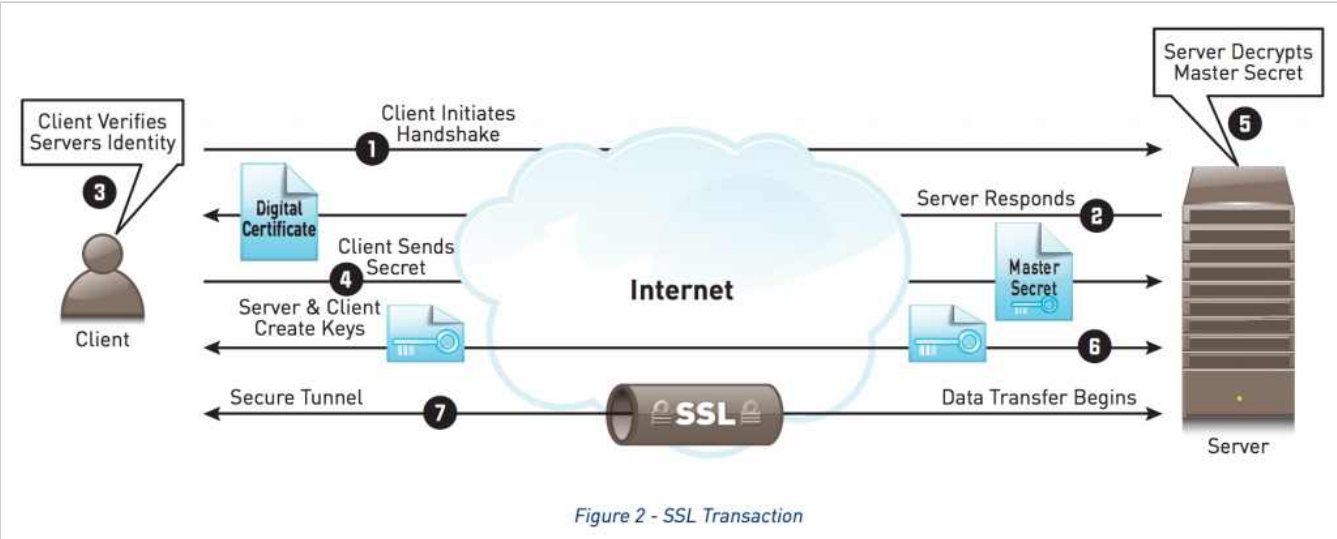


Figure 2 - SSL Transaction

[1] 클라이언트가 서버에 접속한다.

[2] 서버가 보안인증서를 제공한다.

[3] 클라이언트가 서버가 제출한 보안인증서의 유효성을 파악한다. 최상위 발급 기관과 통신하여 유효성을 확인한다. (최상위 발급기관은 운영체제 또는 웹브라우저에 미리 정의되어 있다.)

< 그림 : 서버에서 제공한 인증서가 유효하지 않다면 클라이언트는 검증오류로 인해 통신을 중단한다 >

[4] 보안인증서가 유효하면 인증서에 쓰여져 있는 공개 암호화키 A 를 사용하여 클라이언트 자신의 공개 암호화키 C 를 암호화 하여 전송한다.

[5] 서버는 전송된 암호화 구문을 자신만 가지고 있는 해독키(개인 비밀키) B 를 통해서 해독한다.

[6] 해독한 메세지가 유효한 요청이고 클라이언트의 공개 암호화키 C 를 포함하고 있다면 암호화키 C 를 사용하여 잘 받았다는 메세지를 암호화해서 응답한다.

[7] 클라이언트는 자신만 가지고 있는 해독키(개인 비밀키) D 를 통해서 해독한다. 서버에서 받은 응답 메시지가 유효하다면 클라이언트는 A 를 통해 암호화해서 메세지를 보내고, 서버는 C 를 통해 암호화해서 메세지를 보낸다. A키로 암호화된 메세지는 B키로만 해독이 가능하고, C키로 암호화된 메세지는 D키로만 해독이 가능하므로 서로 종단간(End-to-End) 암호화 통신이 성립하는 것이다.

SSL이 적용된 사이트는 "종단간 암호화" 가 적용되기 때문에 중간 패킷 감청으로부터 안전하다. (해독키가 없으므로 메세지 해석이 불가능.)

보안인증서(SSL) 최상위 발급기관은 매우 많으며, 전세계 점유율은 다음 표와 같다.

참조 : https://en.wikipedia.org/wiki/Certificate_authority

A W3Techs survey from April 2016 shows:[11]

Rank	Issuer	Usage	Market share
1	Comodo	8.1%	40.6%
2	Symantec	5.2%	26.0%
3	GoDaddy	2.4%	11.8%
4	GlobalSign	1.9%	9.7%
5	IdenTrust	0.7%	3.5%
6	DigiCert	0.6%	3.0%
7	StartCom	0.4%	2.1%
8	Entrust	0.1%	0.7%
9	Trustwave	0.1%	0.5%
10	Verizon	0.1%	0.5%
11	Secom	0.1%	0.5%
12	Unizeto	0.1%	0.4%
13	QuoVadis	< 0.1%	0.1%
14	Deutsche Telekom	< 0.1%	0.1%
15	Network Solutions	< 0.1%	0.1%
16	TWCA	< 0.1%	0.1%

< 그림 : World SSL Issuer Market Share >

최상위 발급기관중 하나인 Let's Encrypt 를 사용하여 SSL 을 발급받아 사이트에 적용하는 방법을 설명하겠다.

Let's Encrypt 는 점유율 0.1% 미만의 인증기관이다.(점유율이 매우 낮음) 하지만 발급절차가 간단하고, 무료이기 때문에 점유율이 점차 늘어나지 않을까 싶다.

Let's Encrypt의 점유율 (Market Share) : https://w3techs.com/technologies/details/sc-letsencrypt/all/all

II) SSL 발급 및 적용 방법

암호화 통신 이론에 대해서 알아보았으니 이제 실제로 적용을 해보도록 하자.

인증서 발급 프로그램을 서버에 설치해야한다.

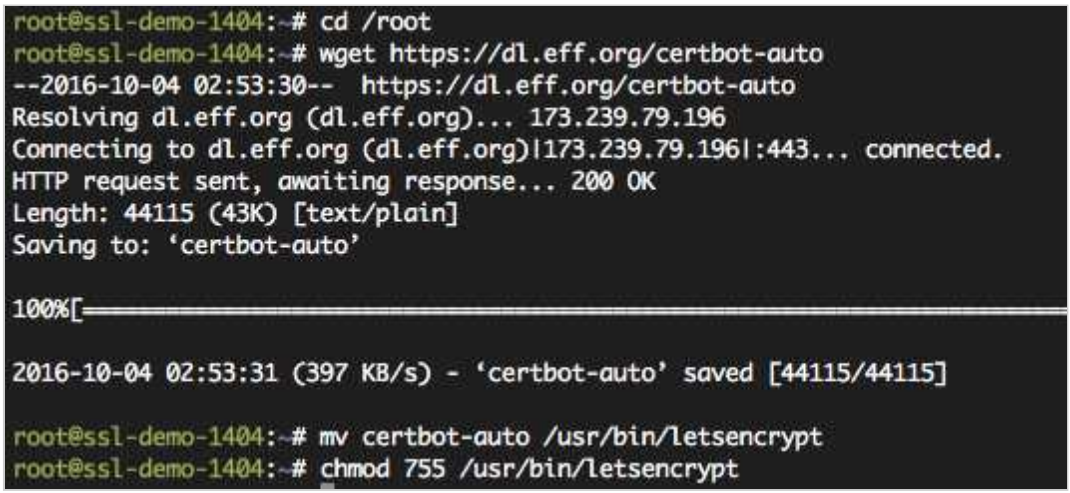
Let's Encrypt 는 Ubuntu 16.04 LTS 에서 기본패키지로 추가되었다. 따라서 쉽게 설치할 수 있다.
반면 Ubuntu 14.04 LTS 에서는 기본패키지가 아니기 때문에 몇 줄 더 입력해야 한다.

발급, 설치, 적용 방법은 다음과 같다.
모든 단계는 Linux root 계정으로 진행한다.

1. 인증서 발급 프로그램 설치하기

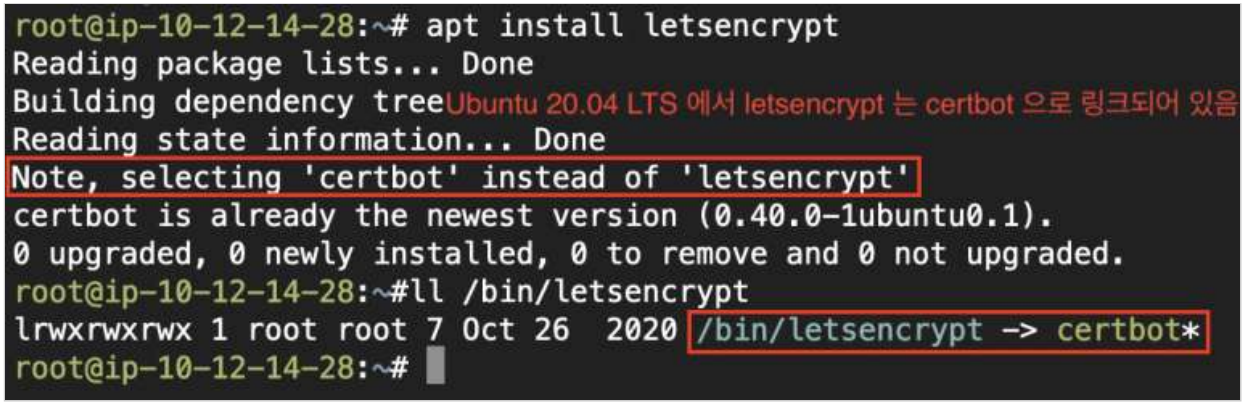
- Ubuntu 14.04 일 경우

```
# cd /root
# wget https://dl.eff.org/certbot-auto
# mv certbot-auto /usr/bin/letsencrypt
# chmod 755 /usr/bin/letsencrypt
```



- Ubuntu 16.04 일 경우 (또는 그 이후 버전, 18.04, 20.04 등)

letsencrypt 는 최근에 certbot 으로 패키지 이름이 변경되었습니다. Ubuntu 내부적으로 letsencrypt -> certbot 으로 설정되어 있으니, 둘 중 아무 단어나 사용하셔도 됩니다. (동작이 완전히 동일함)




```
# apt-get install letsencrypt
```

2. 인증서 발급 프로그램을 사용하여 인증서 발급받기

Gandi.net: Domain Names, Web Hosting, SSL Certificates and Emails - https://www.gandi.net/

인증서는 Domain Control Validation 을 거쳐야 발급받을 수 있다. 당신이 도메인이 없다면 인증서를 발급받을 수 없다.
Domain Control Validation 은 주로 다음의 3가지 방법으로 이루어진다.

```
참고 : Domain Control Validation 의 3가지 방법
(https://docs.gandi.net/en/ssl/common_operations/dcv.html)
```

Let's Encrypt 는 위의 도메인 인증방법 중 3번째인 Validation by file 를 사용하여 인증한다. (최근 Let's Encrypt 에서 Validation by DNS 방식도 지원하도록 변경되었다. 하지만 이 방법은 이 글에서 다루지 않겠다.)

따라서 먼저 해당 도메인에 대한 HTTP 접속이 가능해야 한다.

서버를 처음 세팅하는 경우 HTTP 를 먼저 설정한 다음에 -> 인증서를 발급받고 -> HTTPS 를 추가 설정하는 단계를 거쳐야 할 것이다.

이미 HTTPS 를 사용하고 있을 경우 -> 인증서를 발급받고 -> HTTPS 인증서 교체 하는 단계를 거쳐야 할 것이다.

예제

명령 실행시 1회에 한해 인증관련 프로그램이 추가설치될 수 있다.
*예제 1 :

```
letsencrypt certonly --webroot --webroot-path=/home/myuser1/www -d myuser1.com
```

*예제 2 :

```
letsencrypt certonly --webroot --webroot-path=/home/myuser2/www -d myuser2.com
```

*예제 3 : * 널리 사용됨 (원본 도메인과 www 도메인 동시인증)

```
letsencrypt certonly --webroot --webroot-path=/home/myuser3/www -d myuser3.com -d
www.myuser3.com
```

*예제 4 :

```
letsencrypt certonly --webroot --webroot-path=/home/myuser4/www -d myuser4.com -d
www.myuser4.com -d blog.myuser4.com -d myhome.myuser4.com
```

*** 명령어 설명**

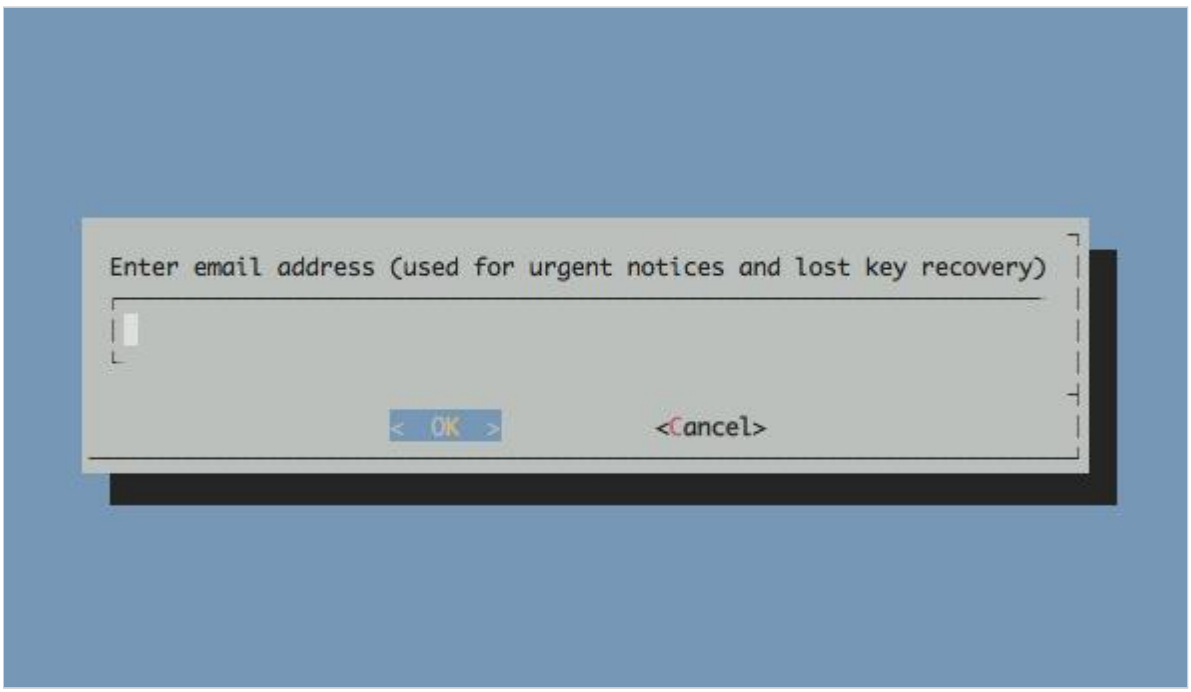
-d 는 도메인명을 지정하면 된다. 최대 100개의 도메인 이름을 지정할 수 있다. (한 인증서가 서로다른 100개의 도메인 인증을 할 수 있음) 일반적으로는 기본도메인과 www 도메인 두개를 지정한다.

-webroot 는 웹인증을 받을 것이라는 것이다. 외부 인증프로그램이 -d 에 지정된 도메인 사이트에 접속한다.

-webroot-path 는 웹루트의 경로이다. 보통 index 페이지가 위치하는 경로이다. 인증 프로그램이 이 경로에 임시 랜덤 파일을 생성하고, 외부 인증프로그램이 이 파일을 접근할 수 있다면 Domain Validation 이 되는 것이다.

명령이 실행되면 다음 화면이 표시된다.

비상연락처 : 인증에 문제가 생겼거나, 인증서 만료기간이 가까웠을때 갱신 알림 메일을 수신할 주소이다.



약관에 동의하면 즉시 인증서가 발급된다.

발급에 실패했다면 외부 접속이 문제가 있거나 -webroot-path 를 올바르게 입력하지 않았을 것이다.

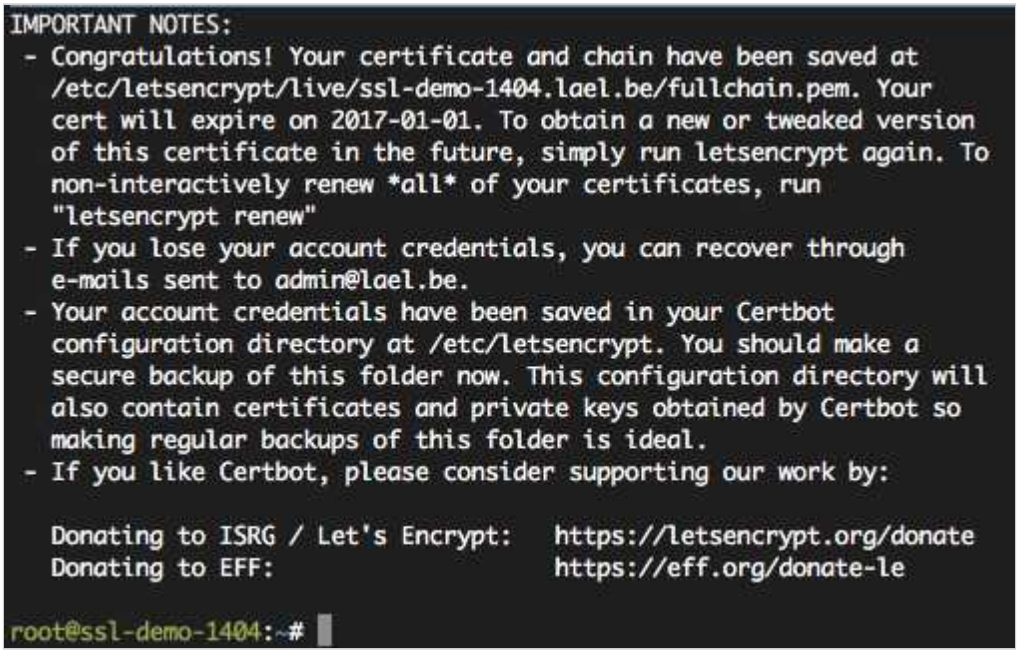
Let's Encrypt 인증 오류 발생시 다음의 사항을 확인한다.

다음의 문서로 이동해서
Allow Lets Encrypt Domain Validation Program
설정이 되어있는지 확인하여라.

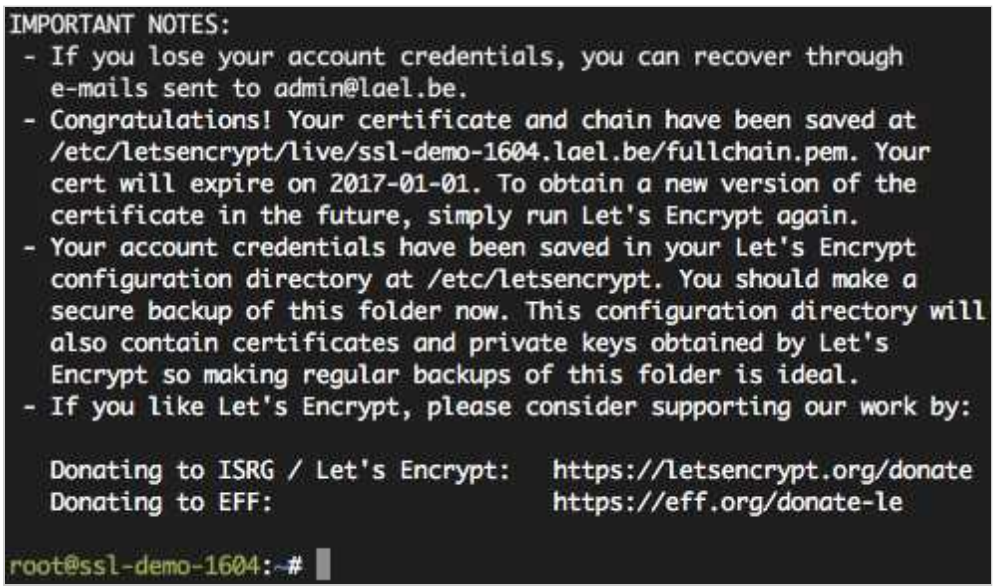
Apache 사용자의 경우 : https://blog.lael.be/post/73#apache2.conf

Nginx 사용자의 경우 : https://blog.lael.be/post/2600#myuser1.conf

- Ubuntu 14.04 발급성공



- Ubuntu 16.04 발급성공



/etc/letsencrypt/live/[인증서명]/ 위치에 발급된다.

cert.pem - 인증서 파일

chain.pem - 인증서 발급자 파일

fullchain.pem - cert.pem 과 chain.pen 을 하나로 합쳐놓은 파일
privkey.pem - 인증암호를 해독하는 개인키

#TIP : 이때 발급된 인증서 파일은 **웹서버 인증서**(HTTPS), **메일서버 인증서**(IMAPS, POP3S, SMTPS), **VPN서버 인증서**(SSTP), **윈도우 원격데스크톱 인증서**(MSTSC), **Gitlab**, **OwnCloud** 등의 용도로 사용 할 수 있다.
(인증서 만료시 갱신해야하는 번거로움이 생기겠지만..)

Apache2 서버에서는 cert.pem, chain.pem, privkey.pem 을 사용합니다.
Nginx 서버에서는 fullchain.pem, privkey.pem 을 사용합니다.

Apache2 적용방법 : https://blog.lael.be/post/73 13번 항목을 참조한다.

```
SSLEngine on
SSLProtocol all -SSLv2 -SSLv3
SSLCipherSuite ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES

SSLHonorCipherOrder on

SSLCertificateFile "/etc/letsencrypt/live/mysitename.com/cert.pem"
SSLCertificateKeyFile "/etc/letsencrypt/live/mysitename.com/privkey.pem"
SSLCertificateChainFile "/etc/letsencrypt/live/mysitename.com/chain.pem"
```

Nginx 적용방법 : https://blog.lael.be/post/2600 16번 항목을 참조한다.

dhparam.pem 파일은 한번만 생성하면 된다. (중복하여 생성해도 문제가 발생하지는 않음)

```
# openssl dhparam -out /etc/ssl/certs/dhparam.pem 2048
```

```
listen       443 ssl http2;
server_name  ssl-demo-1604.lael.be;
root   /home/ssl-demo-1604/www;
client_max_body_size 10M;

ssl_certificate "/etc/letsencrypt/live/mysitename.com/fullchain.pem";
ssl_certificate_key "/etc/letsencrypt/live/mysitename.com/privkey.pem";
ssl_dhparam "/etc/ssl/certs/dhparam.pem";

# Enable HSTS. This forces SSL on clients that respect it, most modern browsers. The
add_header Strict-Transport-Security "max-age=31536000";
```

3. 인증서 발급 프로그램을 통해 인증서 갱신하기

Let's Encrypt 는 3개월짜리 인증서를 발급해준다.

즉 인증서를 3개월마다 주기적으로 갱신(renewal)해 주어야한다. 갱신은 만료일 기준 1개월전부터 할 수 있다.

```
# letsencrypt renew
```

갱신할 인증서가 있다면 자동으로 갱신 작업을 진행한다.

갱신 작업에는 인증서의 재발급이 이루어진다. (renew = reissue)
동일한 도메인에 대해서, 동일한 인증기관이 여러개의 인증서를 발급할 수도 있다. (모두 유효)

4. 인증서 갱신 프로그램 주기적으로 실행하기

큰 부하가 일어나는 것이 아니기 때문에 아무때나 실행해 주어도 큰 문제가 없다.

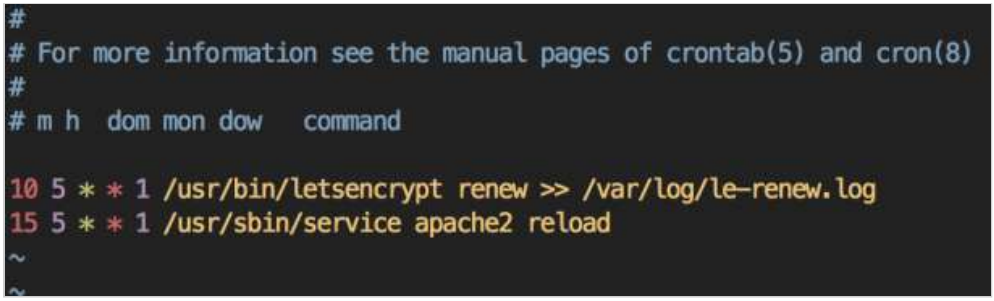
매주 월요일 새벽 5시 10분에 인증서 갱신을, 매주 월요일 새벽 5시 15분에 웹서버 프로그램 변경사항 적용을 실행하도록 설정한다.

```
# crontab -e
```

혹시나 에디터 선택문구가 출력된다면 vim.basic 을 선택하세요. (재선택 명령어 : select-editor)

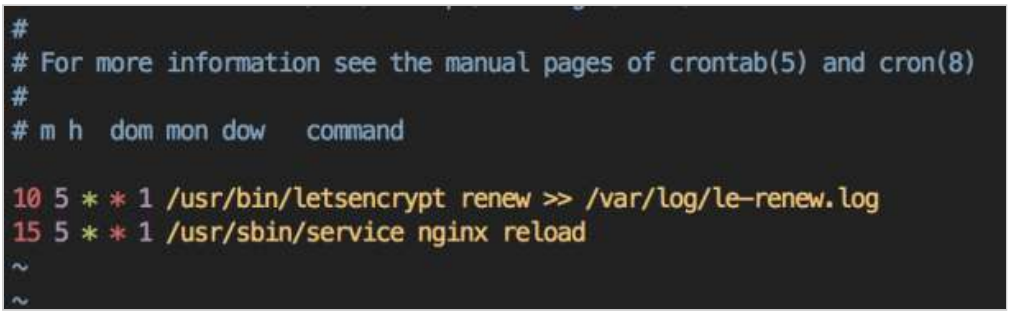
- Apache2 웹서버를 사용하는 경우

```
10 5 * * 1 /usr/bin/letsencrypt renew >> /var/log/le-renew.log
15 5 * * 1 /usr/sbin/service apache2 reload
```



- Nginx 웹서버를 사용하는 경우

```
10 5 * * 1 /usr/bin/letsencrypt renew >> /var/log/le-renew.log
15 5 * * 1 /usr/sbin/service nginx reload
```



참조 : https://www.ssl.com/article/dv-ov-and-ev-certificates/

참조 : https://community.letsencrypt.org/t/which-browsers-and-operating-systems-support-lets-encrypt/4394

참조 : https://www.bluecoat.com/documents/download/0485e335-7437-4c4e-bfc0-ca5ffc5bfd4d/16f27cf7-5d59-44b4-b17f-fb04acea369f

참조 : https://www.digitalocean.com/community/tutorials/how-to-secure-nginx-with-let-s-encrypt-on-ubuntu-16-04

인증서 관련 상황 및 대처 방법 예시

현재 서버에서 사용중인 인증서 목록 보기

```
# cd /etc/letsencrypt/live
```

```
# ls
```

일부 서버 인증서 삭제

반드시 위의 인증서 목록 보기를 통해 인증서 이름을 알아두어야 한다.

현재 사용중인 인증서 취소(revoke)

```
# letsencrypt revoke --cert-path /etc/letsencrypt/archive/my-
examplesite.com/cert1.pem
```

보통 revoke 는 거의 하지 않는다. 사용 안하는 인증서는 그냥 그대로 두어서 만료 시킨다.

다만, 인증서 파일 및 인증서 키가 유출되었고, 도메인 제어 권한에 문제가 있을때, 특수한 상황일 때, revoke를 진행하곤 한다.

갱신 목록에서 제거

이 목록에서 지우면 renew 를 하지 않는다.

```
# cd /etc/letsencrypt/renewal/
```

```
# ls
```

설정파일을 수동으로 선택해서 지운다. (항상 지우는 명령을 실행할 땐 5초동안 고민한 뒤에 엔터를 눌러라.)

기존에 발급받았었던 파일 삭제 (옵션)

찌꺼기 파일인데, 지워도 되고, 안지워도 되고. 아래 위치에서 도메인 폴더를 선택하여, 삭제하면 된다.

```
# /etc/letsencrypt/live/
```

```
# /etc/letsencrypt/renewal/
```

apache | Let's Encrypt | nginx | SSL | StartCom | TCP/IP | ubuntu | 무료인증서

148 Comments

안녕하세요

2016년 10월 9일 at 오전 2:02

우분투 16.04 lts에서의 http2 적용방법도 혹시 포스팅해주실 수 있나요?

항상 도움 많이 받고 있습니다

HYEONG HWAN, MUN author

2016년 10월 11일 at 오후 1:05

안녕하세요.
https://blog.lael.be/post/2600 글을 따라하시면 https2 가 구축됩니다.