3333333333

Let's Encrypt (SSL 인증서)

Certbot 사용매뉴얼

㈜에이클라우드

2019 May 3

기술팀

Let's Encrypt (SSL 인증서)

Certbot 사용매뉴얼

목차

a-Cloud

SSL(Secure Socket Layer)

1. SSL 과 TLS

SSL 또는 TLS 인증서는 클라이언트와 서버 간의 통신이 발생할 때 전송되는 모든 패킷 데이터를 암호화하여 감청이나 식별을 어렵게하여 보안에 있어 강력한 역할을 합니다.

웹 사이트 인증서를 발급해주는 **인증 기관(CA) 이나** 호스팅 업체에서 등록을 대행해주기도 합니다.

2. SSL 이점

1. 통신 내용이 공격자에게 노출되는 것을 막을 수 있습니다.
2. 클라이언트가 접속하려는 서버가 신뢰할 수 있는 서버인지를 판단할 수 있습니다.
3. 통신 내용의 악의적인 변경을 방지할 수 있습니다.

3. SSL 의 암호화

대칭키

대칭키는 동일한 키로 암호화와 복호화를 같이 할 수 있는 방식의 암호화 기법을 의미합니다

※ 대칭키 방식은 단점이 있습니다. 암호를 주고 받는 사람들 사이에 대칭키를 전달하는 것이 어렵다는 점입니다. 대칭키가 유출되면 키를 획득한 공격자는 암호의 내용을 복호화 할 수 있기 때문에 암호가 무용지물이 되기 때문입니다.

공개키

공개키 방식은 두개의 키를 갖게 되는 방식입니다.
두개의 키 중 하나를 개인키 나머지 하나를 공개키로 지정합니다. 개인키는 자신이 가지고 있고, 공개키는 타인에게 제공합니다. 공개키를 제공 받은 타인은 공개키를 이용해서 정보를 암호화합니다. 암호화한 정보를 개인키를 가지고 있는 사람에게 전송합니다. 개인키의 소유자는 개인키를 이용해서 암호화된 정보를 복호화합니다. 이 과정에서 공개키가 유출되어도 개인키를 모르면 정보를 복호화할 수 없기 때문에 안전합니다. 공개키로는 암호화할 수 있지만 복호화는 할 수 없기 때문입니다.

4. SSL 통신방식

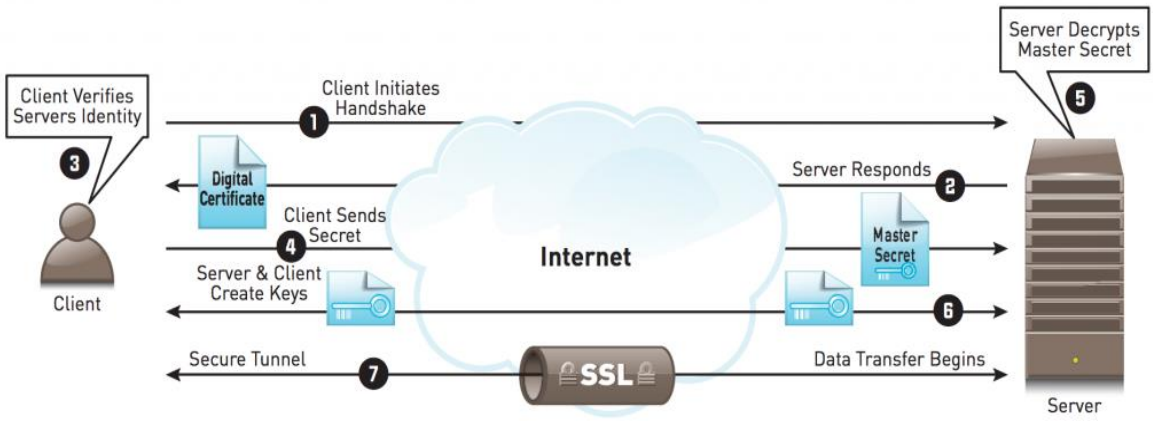


Figure 2 - SSL Transaction

[1] 클라이언트가 서버에 접속한다.

[2] 서버가 **보안인증서**를 제공한다.

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

[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

Lets Encrypt

Let's Encrypt 는 사용자에게 **무료**로 TLS 인증서를 발급해주는 비영리기관이다.

Let`s Encrypt 를 사용하는 이유

- **인증 절차가 단순화**되었을 뿐만 아니라 **발급 대기 시간이 없어** 빠르게 인증서를 적용할 수 있습니다.
- **와일드 카드 인증서**를 지원합니다
- 발급을 위한 정보는 발급자 **이메일**만 요구됩니다.

※제한사항

- 퍼블릭 도메인이 할당된 서버에서만 발급이 가능합니다.
- 인증서를 설치할 서버의 터미널에 접속할 수 있어야 하며 **root 권한**을 사용할 수 있어야합니다.(ex 카페 24,가비아 웹 호스팅 서비스)
- 내부 테스트용으로 구성된 서버이거나 공개 서버가 아닌 등의 이유로 자신의 서버에 IP 만 할당되어 있는 경우에는 인증서 발급과 설치에 어려움이 있으므로 반드시 **도메인을 할당**해주어야 합니다.
- Let's Encrypt SSL 인증서는 **5 회 발급으로 제한**되어 있습니다.
- 인증서 유효기간은 **90 일**이므로 지속적인 갱신 과정이 필요합니다.

※주의사항

- 인증서로 인해 발생한 피해에 대해서는 해당 기관으로부터 보상 받을 수 없다.

- 일부 오래된 운영체제나 브라우저에서 인증서로인한 오작동이 발생할 수 있다.

1.1 도메인 소유자 인증방법

1.도메인이 연결된 서버에서 일련의 랜덤 문자열인 Challenge Seed 를 생성하고, 이를 외부에서 HTTP(S)로 접근하여 확인할 수 있도록 약속된 URI 에 위치시킵니다.

2.Let's Encrypt 서버로 인증받으려는 도메인과 Challenge Seed 를 함께 전송합니다.

3.Let's Encrypt 서버에서 인증받으려는 도메인으로 접속하여 사전에 약속된 URI 에 위치한 Challenge Seed 값을 확보합니다.

4.두 개의 경로로 확보한 Challenge Seed 가 일치하면 인증서 발급 절차를 계속 진행합니다.

1.2 인증서 발급 방식

Standalone

Certbot 에 내장된 웹 서버를 돌려서 Let's Encrypt 서버로부터 오는 도메인 인증 요청을 직접 받아서 처리하는 방식입니다.

Webroot

인증을 위한 Challenge Seed 를 외부에서 접근 가능한 미리 약속된 경로에 위치시킨 뒤, Let's Encrypt 서버가 해당 경로로 접속해 인증에 필요한 정보를 읽어갑니다

DNS

도메인이 연결된 DNS 에 TXT 레코드를 생성해서 인증서를 발급하는 방식으로, 인증서 발급 과정에 웹 서버가 필요 없다

※Webroot 방식은 직접 운영하는 도메인을 확인하므로 여러개의 도메인을 한 서버, 사이트에서 등록시키기는 어렵습니다.
※standalone 방식은 갱신 시 웹서버를 작동 중지 시켜야 하지만 여러개의 도메인(100 개까지가능)을 한꺼번에 인증 받을 수 있습니다.

Certbot

1.1 Certbot 설치

1.패키지관리도구 다운

```
Sudo apt-get update
Sudo add-apt-repository ppa:certbot/certbot
※위의 명령이 안될시 sudo apt-get install software-properties-common
sudo apt-get update
sudo apt-get install certbot python-certbot-apache
```

2.스크립트 다운

```
wget https://dl.eff.org/certbot-auto
chmod +x certbot-auto
./certbot-auto
```

[options]
Certonly : 인증서만 발급받음
-webroot : webroot 플러그인을 사용하겠음
-w /var/www/challenge : 챌린지 파일을 생성할 디렉토리 설정
-d example.com : 인증서를 생성할 도메인 지정

Ex) /var/www/certbot 디렉토리를 webroot 로 설정

(1)디렉터리 생성후 웹서버만 접근가능 하도록 권한 설정

```
mkdir /var/www/certbot
chown root:root /var/www/cert/bot
chmod 700 /var/www/certbot
```

(2) SELinux 가 돌고있는 경우 웹서버가 내용을 볼 수 있게 보안 문맥을 변경

```
chcon -t httpd_sys_content_t /var/www/certbot
```

(3) 웹 서버설정 파일을 수정해 가상 호스트의 URL 이 위의 Webroot 를 가리키도록 설정

```
...
location /.well-known {
        root /var/www/certbot-webroot/;
}
...
```

※주의사항

HTTPS 가 아닌, HTTP 포트 설정에 추가해줄 것.

여러 가상 호스트를 사용하고 있다면 각각을 별도로 추가해 줄 것

1.2 인증서 발급

1. Standalone 방식

(1) sudo service nginx/apache stop

(2) certbot certonly –-cert-name <인증서이름> --standalone -d <도메인 1>,<도메인 2>,....,<도메인 n>

(3) sudo service nginx/apache start

2. Webroot 방식

(1) certbot certonly –cert-name <인증서이름> --webroot -w <webroot 디렉토리> -d <도메인 1>,...,<도메인 n>

```
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Plugins selected: Authenticator standalone, Installer None
Enter email address (used for urgent renewal and security notices) (Enter 'c' to
cancel):
```

E-mail 인증

```
Starting new HTTPS connection (1): acme-v02.api.letsencrypt.org

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
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:
```

약관동의 (동의해야함)

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
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:
```

수신정보동의 (거부해도 무관)

```
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Plugins selected: Authenticator standalone, Installer None
Starting new HTTPS connection (1): acme-v02.api.letsencrypt.org
Cert is due for renewal, auto-renewing...
Renewing an existing certificate
Performing the following challenges:
http-01 challenge for example.com
Waiting for verification...
Cleaning up challenges
Resetting dropped connection: acme-v02.api.letsencrypt.org

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 2019-04-06. To obtain a new or tweaked
   version of this certificate in the future, simply run certbot
   again. To non-interactively renew *all* of your certificates, run
   "certbot renew"
 - 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
```

인증서발급완료

1.3 웹 서비스 파일 설정

(1) 인증서 확인

발급된 인증서는 /etc/letsencrypt/live/<domainname> 디렉터리에 생성됩니다.

```
/etc/letsencrypt/live/some.site.com/fullchain.pem
/etc/letsencrypt/live/some.site.com/privkey.pem
```

Certbot cerificates

```
-------------------------------------------------------------------------------
Found the following certs:
  Certificate Name: yourdomain.com
    Domains: yourdomain.com www.yourdomain.com
    Expiry Date: YYYY-MM-DD mm:ii:ss+00:00 (VALID: NN days)
    Certificate Path: /path/to/fullchain.pem
    Private Key Path: /path/to/privkey.pem
-------------------------------------------------------------------------------
```

인증서의 실제 파일은 다른 위치에 있고, 위 경로는 심볼릭 링크 파일입니다.

1.4 conf 파일설정

1. Nginx

기본위치 : /usr/local/nginx/conf

```
server
{
    listen       443;
    server_name {서버 도메인};

    ssl on;
    ssl_certificate      {Certificate Path};
    ssl_certificate_key {Private Key Path};

    ssl_protocols    SSLv2 SSLv3 TLSv1 TLSv1.1 TLSv1.2;
    ssl_ciphers
ALL:!aNULL:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP;
    ssl_prefer_server_ciphers    on;

    ...
}
```

2. Apache

기본경로

Debian 계열 : /etc/apache2/apache2.conf

RedHat 계열 : /etc/httpd/conf/httpd.conf

```
<VirtualHost *:443>
DocumentRoot /var/www/html
ServerName <설정한도메인.com>:443
</VirtualHost>
SSLEngine on
SSLCertificateFile /etc/letsencrypt/live/example.com/cert.pem
SSLCertificateKeyFile /etc/letsencrypt/live/example.com/privkey.pem
```

```
SSLCertificateChainFile /etc/letsencrypt/live/example.com/fullchain.pem

Sudo service httpd start
```

[openSSL]사용

1. 모듈 로드 명령어를 사용하여 Apache2 SSL 모듈을 로드합니다.

[openSSL 이 설치가 안되어있다면 (sudo apt-get install openssl)로 설치합니다.]

```
sudo a2enmod ssl
```

2. apache2/site-available/default-ssl 파일을 설정합니다.

(필요시 default-ssl 이 아닌 다른 파일명을 사용해도 됩니다.)

```
1  <IfModule mod_ssl.c>
2  <VirtualHost *:443>
3
4  ...
5
6  # SSL 활성화 및 인증서 정보
7  SSLEngine on
8  SSLCertificateFile      (*.crt 파일의 경로)
9  SSLCertificateKeyFile   (*.key 파일의 경로)
10 #SSLCertificateChainFile (Chain 인증서 경로)
11 #SSLCACertificateFile    (Root 인증서 경로)
12
13 ...
14
15 </VirtualHost>
16 </IfModule>
```

#2 : SSL/HTTPS 연결을 사용할 포트번호를 지정합니다. 기본값은 443 입니다.

#10 ~ 11 : 해당사항이 있는 경우만 입력합니다. (일부 인증기관의 경우 이 두 줄을 추가해야 정상 동작)

3. default-ssl 사이트를 활성화 시킵니다.

```
sudo a2ensite default-ssl
```

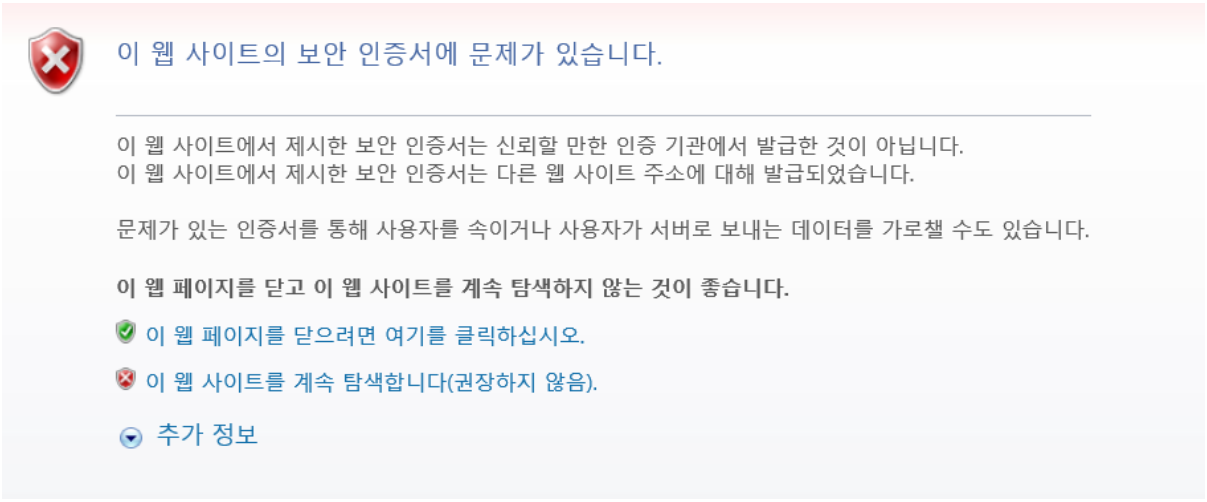
4. apache 서버 재부팅 합니다.

```
Sudo service apache2 restart
```

1.5 인증서 발급완료

1.6 인증서 갱신

인증서 갱신은 최소 만료기간 30 일전부터 가능하다



인증서가 만료되어 없을경우 보안경고가 뜨게된다.

인증서 갱신 command

```
certbot renew
```

자동갱신을 위한 cronjob 사용

※crontab 사용법 : * * * * * [명령어] # 분 시 월 요 일

```
0 8 * * *certbot-auto renew --pre-hook "service nginx stop" --post-hook "service nginx start"

0 8 * * * certbot-auto renew –renew-hook “service reload nginx”

0 8 * * * /home/bitnami/certbot-auto renew –renew-hook “sudo/opt/bitnami/ctlscript.sh restart apache”
```

--pre-hook “인증서 발행전에 일어나는 프리후크”

--post-hook “인증서 발행하려고 시도한 후에 발생하는 포스트후크”

--renew-hook “인증서가 성공적으로 갱신될때 호출되는 후크”

AWS 클라우드 솔루션

Amazon Web Services(AWS)는 믿을 수 있는 클라우드 기반 솔루션을 제공하여 비즈니스 요구 사항을 충족할 수 있게 지원합니다. AWS 클라우드에서 솔루션을 실행하면 애플리케이션을 더욱 빠르게 시작 및 실행할 수 있으며 Pfizer, Intuit 및 미 해군 등의 조직이 사용하는 것과 동일한 수준의 보안이 제공됩니다. 또한, AWS 는 전 세계적으로 리소스를 제공하기 때문에 고객의 지역에 상관없이 솔루션을 배포할 수 있습니다. AWS 클라우드에서는 광범위한 서비스 세트, 파트너, 지원 옵션을 손쉽게 사용할 수 있기 때문에 사용자는 솔루션의 성공에만 집중할 수 있습니다.

AWS는 "클라우드 기반 솔루션을 위한 믿을 수 있는 선택"입니다.

자세한 정보는 https://aws.amazon.com/ko/solutions/을 참조하십시오.

Amazon Web Service, Inc.

Last Updated: December 23, 2011.

P.O. Box 81226
Seattle, WA 98108-1226
USA

http://aws.amazon.com

Acloud Co., Ltd.

06159

9F, Hyundai Tower, 423,
Teheran-ro, Gangnam-gu,
Seoul, Korea

http://www.a-cloud.co.kr