



HTTPS 무료 사용을 위한 Let's Encrypt 사용자 가이드

시큐리티플러스 심홍록, 이용훈, 유성은, 김경민, 김현정 지음



HTTPS 무료 사용을 위한 Let's Encrypt 사용자 가이드

시큐리티플러스 심홍록, 이용훈, 유성은, 김경민, 김현정 지음





표지 사진 손경선

이 책의 표지는 손경선님이 보내 주신 풍경사진을 담았습니다.
리얼타임은 독자의 시선을 담은 풍경사진을 책 표지로 보여주고자 합니다.

사진 보내기 ebookwriter@hanbit.co.kr

HTTPS 무료 사용을 위한 Let's Encrypt 사용자 가이드

초판발행 2016년 8월 31일

지은이 시큐리티플러스 심홍록, 이용훈, 유성은, 김경민, 김현정 / 펴낸이 김태현
펴낸곳 한빛미디어(주) / 주소 서울시 마포구 양화로7길 83 한빛미디어(주) IT출판부
전화 02-325-5544 / 팩스 02-336-7124
등록 1999년 9월 30일 제10-1779호
ISBN 978-89-6848-840-5 95000 / 비매품

총괄 전태호 / 책임편집 김창수 / 기획·편집 정지연

디자인 표지 강은영, 내지 여동일, 조판 최승실

마케팅 박상용, 송경석, 변지영 / 영업 김형진, 김진불, 조유미

이 책에 대한 의견이나 오탈자 및 잘못된 내용에 대한 수정 정보는 [한빛미디어\(주\)](http://www.hanbit.co.kr)의 홈페이지나 아래 이메일로 알려주십시오.
한빛미디어 홈페이지 www.hanbit.co.kr / **이메일** ask@hanbit.co.kr

Published by HANBIT Media, Inc. Printed in Korea

Copyright © 2016 시큐리티플러스 & HANBIT Media, Inc.

이 책의 저작권은 시큐리티플러스와 한빛미디어(주)에 있습니다.

저작권법에 의해 보호를 받는 저작물이므로 무단 복제 및 무단 전재를 금합니다.

지금 하지 않으면 할 수 없는 일이 있습니다.

책으로 떠내고 싶은 아이디어나 원고를 메일(ebookwriter@hanbit.co.kr)로 보내주세요.

한빛미디어(주)는 여러분의 소중한 경험과 지식을 기다리고 있습니다.

작성자 소개

- 팀장** 심홍록(SecurityPlus Union Academy 서울경기지부, 한국산업기술대 컴퓨터 공학과, 정보보안동아리 MA 회장)
- 기술팀** 이용훈(SecurityPlus Union Academy 영남지부, 영진전문대 컴퓨터정보계열)
- 문서팀** 유성은(SecurityPlus Union Academy 영남지부, 영남대 컴퓨터공학과)
- 이전 멤버** 김경민(SecurityInsight Research Group)
김현정(세종대학교 컴퓨터공학과, 정보처리기사)
- 자문** 김기태(한국정보기술단 본부장)
- 감수** 박형근(시큐리티플러스 비영리단체 대표)
- 후원** 시큐리티플러스(<http://www.securityplus.or.kr>)

chapter 1 소개 —— 007

- 1.1 등장 배경 —— 007
- 1.2 Let's Encrypt란 —— 009
- 1.3 Let's Encrypt의 기능 —— 011

chapter 2 운영체제별 설치환경 및 요구사항 015

- 2.1 Debian 계열 —— 015
 - 2.1.1 기본 환경설정 —— 015
 - 2.1.2 추가 설정 —— 017
- 2.2 Redhat 계열 —— 019
- 2.3 Windows Server 2012 —— 021

chapter 3 Apache 웹 서버에서의 Let's Encrypt 적용 023

- 3.1 Debian 계열 —— 023
- 3.2 Redhat 계열 —— 028

chapter 4 Nginx 웹 서버에서의 Let's Encrypt 적용 035

- 4.1 Debian 계열 —— 035
- 4.2 Redhat 계열 —— 038



chapter 5 IIS 웹 서버에서의 Let's Encrypt 적용 043

chapter 6 Let's Encrypt 인증서 갱신 047

- 6.1 Apache 웹 서버 ————— 047
- 6.2 Nginx 웹 서버 ————— 048
- 6.3 IIS 웹 서버 ————— 050

chapter 7 마무리 ————— 057

소개

이 책은 Let's Encrypt에 관한 사용자 가이드로, Let's Encrypt의 기본개념과 해당 기술이 대두한 배경, 각종 서버 OS와 각 HTTP 웹 서버에서 Let's Encrypt 설치를 어떻게 진행하는지에 대한 내용을 전반적으로 기술하였다.

1.1 등장 배경

고속 인터넷망의 확산과 다양한 스마트기기의 빠른 보급으로 인터넷 사용이 폭발적으로 증가하였다. 그중 우리가 인터넷을 통하여 가장 많이 사용하는 서비스는 아마도 웹 서비스일 것이다. HTTP 프로토콜을 사용하여 HTML 문서를 주고받는 웹 서비스는 사용자의 플랫폼과 환경에 구애받지 않고 다양한 형태의 정보들을 손쉽게 주고받을 수 있어 매우 편리한 정보공유 서비스임이 틀림없다.

하지만 다양한 정보가 공유되고 쉽게 접근할 수 있을 뿐만 아니라 플랫폼에 구애받지 않는다는 장점은 그만큼 많은 악의적 해킹 공격에 노출될 수 있다는 단점이 있다. 실제로 대부분 공격이 웹 취약점을 이용하여 이루어지고 악성코드와 바이러스 대부분도 웹을 통하여 유포된다. 이러한 웹 서비스의 허점들은 안전한 정보공유활동을 침해하는 요소로 작용할 수 있으며, 더 나아가 사용자의 개인정보를 탈취하여 제2, 제3의 피해를 일으키는 위험으로도 발전할 수 있다.

그렇지만 여러 가지 요소를 통하여 웹 환경에서의 안전한 통신을 구축할 수 있다. 그중 하나인 HTTPS는 암호화되고 인증된 통신을 위하여 어느새 필수불가결한

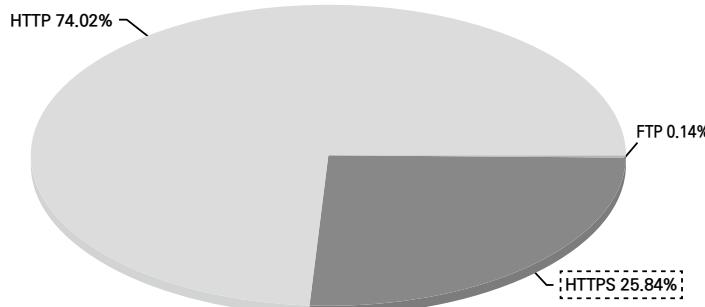
요소가 되었다. HTTPS는 HTTP의 보안 허점을 극복하기 위하여 1995년 ‘넷스 케이프 커뮤니케이션즈 코퍼레이션’에서 SSL^{Secure Socket Layer} 암호화를 적용한 프로토콜이다. HTTPS 프로토콜은 HTTP보다 보안이 한층 강화된 프로토콜로, 세션에서 주고받는 데이터를 암호화해 데이터의 적절한 보호를 보장한다.

HTTPS를 적용하여 얻을 수 있는 이점은 크게 다음 세 가지로 정의할 수 있다.

- 데이터 암호화 TLS 프로토콜이 적용된 HTTP 프로토콜을 사용함으로써 송수신하는 데이터를 암호화하여 전송한다. 공격자에 의하여 패킷이 도청/감청당하더라도 암호화되어 있으므로 아무 의미 없는 데이터를 보는 것과 다름없게 된다.
- 데이터 무결성 TLS 프로토콜의 알고리즘을 이용하여 무결성을 해칠 수 있는 요소들을 어느 정도 보완할 수 있다.
- 인증 송수신 측 모두 신뢰할 수 있는 통신을 하기 위하여 웹 페이지 제공자는 전자서명이 포함된 보안 인증서를 사용하여 자신의 사이트가 신뢰할 수 있는 연결을 맺을 수 있음을 증명한다.

하지만 HTTPS 프로토콜을 사용하려면 인증 기관^{CA, Certificate Authority}에서 인증서를 발급받아야 하며(과정 또한 복잡하다), 적지 않은 금액의 수수료도 지급해야 한다. 이러한 불편사항 때문에 많은 웹 서비스 공급자가 HTTPS 사용을 적극적으로 수용하지 않는 실정이다. 다음 그림은 2015년 Websense에서 발표한 보고서 일부를 발췌한 것으로, 그림을 보면 현재 HTTPS 프로토콜은 전체 웹 프로토콜의 25%만 차지하는 것을 볼 수 있다(HTTP 약 75%, HTTPS 약 25%).

그림 1-1 웹 프로토콜 분석 현황



이처럼 저조한 HTTPS의 보급률을 끌어올리고, 모든 웹 서비스가 안전한 HTTPS 프로토콜을 사용하는 환경을 조성하고자 Mozilla, Cisco, Akamai, IdenTrust, Facebook, Google 등 다양한 글로벌 IT기업들이 ISRG^{Internet} Security Research Group라는 인증기관을 만들어 ‘Let’s Encrypt’를 시작하였다. 여기서 Let’s Encrypt는 보안 인증서를 일반 사용자가 구축한 웹 사이트에 쉽게 적용할 수 있는 소프트웨어를 가리킨다.

1.2 Let’s Encrypt란

Let’s Encrypt 프로젝트는 누구든지 HTTPS 프로토콜을 적용하여 안전한 웹 서비스 사용환경을 조성하는 데 그 목적이 있다. 이러한 목적을 수행하기 위하여 Let’s Encrypt 프로젝트는 웹 서비스 공급자들이 더 쉽고 경제적으로 HTTPS 프로토콜을 공급할 수 있도록 인증서 관리 프로그램을 개발하였다(개발한 프로그램의 이름 역시 ‘Let’s Encrypt’라고 불린다). Let’s Encrypt를 왜 사용해야 하는지에 대한 이유를 먼저 설명하겠다.

1. 무료다

글로벌 루트 인증기관의 인증서를 발급받으려면 수수료를 지급해야 한다. 수수료는 2015년 1월을 기준으로 1년에 20~40만 원 정도다(회사와 지원 서비스에 따라 천차만별의 가격을 형성하고 있지만, 평균 금액은 이 정도다).

2. 안전하다

사람들은 값이 저렴하고 무료면 제품의 질이 떨어진다고 가끔 오해한다. 하지만 Let’s Encrypt는 그렇지 않다. Global Sign, VeriSign, Geo Trust 같은 글로벌 루트 인증기관과 동일한 강도의 안전성을 보장한다.

3. 적용하기 쉽다

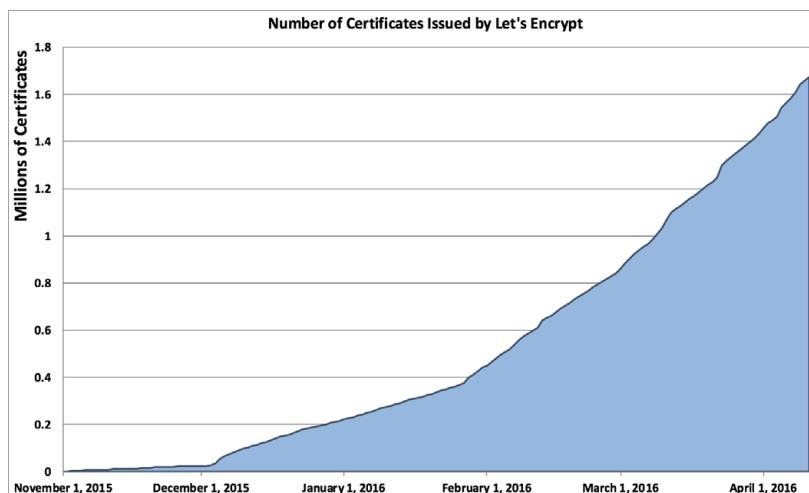
Let’s Encrypt는 인증서 발급 절차가 쉽도록 별도의 프로그램을 지원한다. 해당

프로그램을 이용하면 손쉽게 인증과정을 수행할 수 있으며, 지원하는 웹 서버에 한해 인증서 적용도 쉽게 진행할 수 있다. 또한, 무료 인증서의 사용 연장도 자체로 제공하는 도구를 이용하기 때문에 훨씬 쉽다.

아울러 Let's Encrypt는 2016년 4월을 기준으로 베타를 벗어났음을 공표하였고,⁰¹ Let's Encrypt 프로젝트가 시작되고 나서 HP Enterprise, Gemalto, ReliableSite.net, Fastly, Duda 등 많은 사용자가 Let's Encrypt를 사용하기 시작하였다. 현재 더욱 많은 기업이 지원을 자처하고 나섰다.

이토록 많은 지원과 관심을 받는 이유는 바로 Let's Encrypt가 이루어 낸 성과를 보면 바로 알 수 있다. 다음 그래프는 2015년 9월부터 베타 서비스를 진행한 후 Let's Encrypt에서 발급받은 인증서 수의 증가를 보여준다.

그림 1-2 Let's Encrypt에서 발급받은 인증서 수⁰²



01 출처: <https://letsencrypt.org/2016/04/12/leaving-beta-new-sponsors.html>

02 출처: <https://letsencrypt.org/2016/04/12/leaving-beta-new-sponsors.html>

1.3 Let's Encrypt의 기능

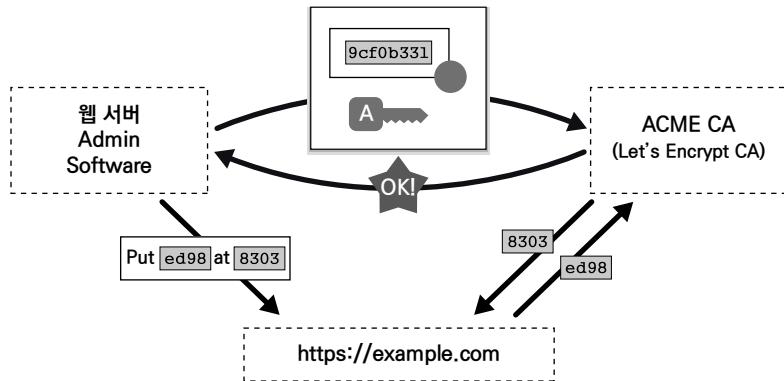
이처럼 무료고 적용하기 쉽다는 점이 Let's Encrypt의 가장 큰 장점이다. Let's Encrypt의 특징과 장점을 살펴보았으니 이제 지원하는 기능에 대해서 알아보자.

Let's Encrypt는 도메인에 따라 Let's Encrypt CA에서 부여받은 공개키로 구별되고, 처음 등록하는 과정에서 신뢰성 있는 연결을 보장함을 알려준다. 그러면 Let's Encrypt CA에서는 서버 측에 개인 키를 부여하고 이를 통하여 신뢰성 있는 연결을 구축할 수 있다. [그림 1-3]과 [그림 1-4]는 이 과정을 도식화한 것이다.⁰³

그림 1-3 도메인을 Let's Encrypt CA에 등록하는 과정 (1)



그림 1-4 도메인을 Let's Encrypt CA에 등록하는 과정 (2)



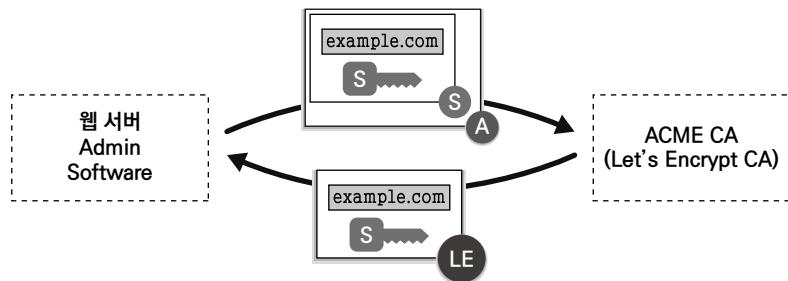
Let's Encrypt에 도메인을 등록하는 과정은 이와 같다. 다음으로 Let's Encrypt에서 제공하는 기능을 살펴보자. Let's Encrypt는 다음과 같은 기능을 제공한다.

03 출처: Let's Encrypt 공식 홈페이지, <https://letsencrypt.org/how-it-works/>(CC BY 3.0)(일부 수정)

인증서 발급

Let's Encrypt를 설치한 후 간단한 명령어만 입력하면 인증서를 쉽게 발급받을 수 있다. 즉, 웹 페이지의 도메인 주소를 입력하면 Let's Encrypt에서 키와 인증서를 받는다. 서버 설정에 따라 수동으로 발급받고 등록할 수도 있다. [그림 1-5]는 이 과정을 도식화한 것이다.

그림 1-5 인증서를 발급받고 보안 연결을 수립하는 과정⁰⁴



이처럼 웹 서버가 자신의 공개키와 도메인 정보를 포함하여 Let's Encrypt의 인증기관인 ISRG에 전송하면 ISRG는 이 두 정보를 기관의 개인키로 암호화하여 서버에 다시 전송한다. 이때 인증기관이 암호화하여 보내준 것이 최종으로 사이트의 인증서가 된다(각 기관의 공개키는 브라우저에 자체로 내장이 되어 있고, ISRG 기관의 공개키도 존재한다). 이로써 클라이언트는 웹 사이트에 접속할 때 자신이 접속하는 사이트가 위조된 사이트가 아닌 ISRG 기관으로부터 인증받은 정상적인 사이트라는 것을 확인하고 안전하게 연결을 수행할 수 있게 된다.

해당 작업은 웹 서비스 공급자가 사용하는 웹 서버에 따라 자동으로 수행하는 경우가 있고, 서비스 공급자가 수동으로 인증서를 발급받고 직접 적용해야 하는 경우도 있다. 발급받은 인증서는 90일에 한 번씩 재인증해야 사용할 수 있으므로 이 점을 유의하고 적용해야 한다.

04 출처: Let's Encrypt 공식 홈페이지, <https://letsencrypt.org/how-it-works/>(CC BY 3.0)(일부 수정)

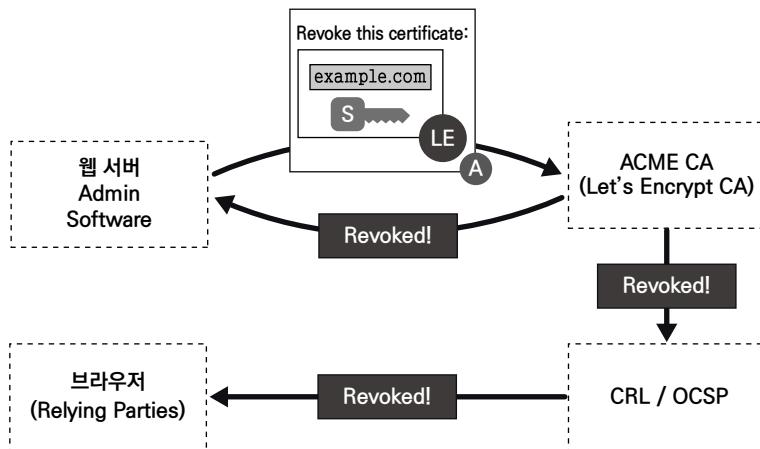
표 11 웹 서버별 Let's Encrypt 지원 가능 기능

Plug-in	Auth	Inst
Apache	Y	Y
Standalone	Y	N
Webroot	Y	N
Nginx	Y	Y
Manual	Y	N

인증서 삭제

인증서 삭제 또한 인증서 발급과 비슷한 과정을 거친다. 인증서를 해지하기 위한 도메인과 키를 가지고 Let's Encrypt 프로토콜 해지를 요청하면 CA에서 키와 도메인을 확인한 후 프로토콜을 통하여 해지한다. [그림 1-6]은 이 과정을 보여준다.

그림 1-6 Let's Encrypt CA에서 보안 연결을 해제하는 과정⁰⁵



05 출처: Let's Encrypt 공식 홈페이지 <https://letsencrypt.org/how-it-works/>(CC BY 3.0)(일부 수정)

인증서 갱신

Let's Encrypt는 인증서 유효기간이 90일이다. 이 때문에 인증서를 연속해서 사용하기 위한 자동 갱신 기능을 제공한다. 간단한 명령어만 입력하면 쉽게 적용할 수 있으며, 웹 서버의 종류에 맞게 갱신할 수 있다. 이 과정은 보통 서버 운영체제의 예약 작업을 통하여 인증서 갱신 기간 30일 전에 자동으로 갱신을 진행할 것을 권장한다. 인증서 갱신과정은 삭제과정과 유사하다.

운영체제별 설치환경 및 요구사항

Let's Encrypt는 기본적으로 Unix 계통의 운영체제만 지원한다. 하지만 사용자들이 직접 제작하고 수정한 프로그램 덕분에 Windows Server에서도 적용할 수 있게 되었다. 다만, Windows Server는 IIS 웹 서버에서만 작동함에 유의하길 바란다.

2.1 Debian 계열

Let's Encrypt를 설치하기 위한 Debian 계열 운영체제의 최소 사양은 Debian 7 이상 또는 Ubuntu 12.04 이상이다. 환경설정은 다음 순서대로 진행한다.

2.1.1 기본 환경설정

Debian 리눅스 apt 리스트 업데이트Linux apt list update

모든 패키지를 정상적으로 다운로드하려면 리포지터리 Repository 리스트를 업데이트하는 것이 좋다. 명령어는 다음과 같다.

```
$ sudo apt-get update
```

Git 패키지 설치

Let's Encrypt는 소프트웨어 배포 시 기본적으로 Git을 이용한다. 따라서 해당 패키지가 설치되어 있어야 정상적으로 Let's Encrypt를 설치할 수 있다.

Git은 다음 명령어로 설치한다(이미 Git이 설치되어 있으면 해당 과정을 건너뛰어도 된다).

```
$ sudo apt-get install git
```

Apache 2 웹 서버 버전 확인

현재 사용하는 Apache 2의 버전을 확인해야 한다. 안정적으로 설치하려면 2.x 버전의 아파치 사용을 추천한다. 사용 버전은 다음 명령으로 확인할 수 있다.(이 안내서는 Apache 2.4.18 버전을 기준으로 작성하였다).

```
$ apache2 -v
```

웹 서버와 데이터베이스 종료

Let's Encrypt는 웹 서버와 사용 중인 데이터베이스를 정상적으로 종료하고 설치하는 것이 좋다. 이는 나중에 발생할 수 있는 다양한 오류를 미리 방지하기 위해 서다. 웹 서버와 데이터베이스는 다음 명령으로 각각 종료할 수 있다.

```
$ /etc/init.d/apache2 stop  
$ /etc/init.d/mysql stop
```

Python 설치

Let's Encrypt를 설치하려면 반드시 Python을 설치해야 한다. Let's Encrypt에서 권장하는 Python 버전은 2.7이지만 3.x 버전도 사용할 수 있다. 다음 명령어를 입력하면 Python이 설치된다.

```
$ sudo add-apt-repository ppa:fkrull/deadsnakes  
$ sudo apt-get update  
$ sudo apt-get install python2.7
```

같은 Debian 계열의 운영체제를 사용하는 Raspberry Pi는 기존 환경과 다소 차이가 있기 때문에 고려해야 할 사항과 사전에 수행할 작업이 몇 가지 더 있다. 이번에는 이러한 추가 설정을 살펴보겠다.

2.1.2 추가 설정

SSH 환경설정

Raspberry Pi에서의 작업은 SSH를 통하여 원격으로 편하게 작업할 수 있다. 사용자 환경에 따라 설정된 네트워크 환경에서 SSH로 접속하면 다음과 같은 화면이 뜰 것이다.

그림 2-1 SSH로 Raspberry Pi 접속

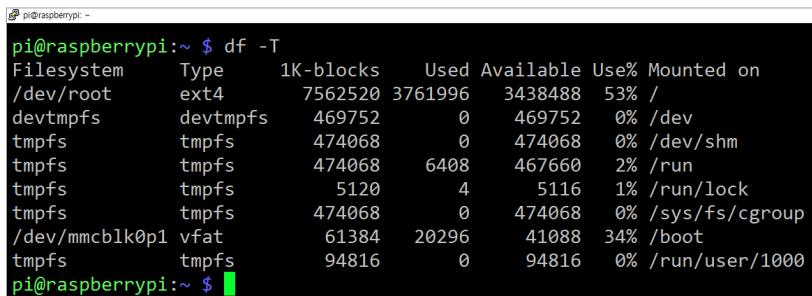


```
pi@raspberrypi: ~ $
```

윈도우 환경에서 Raspbian을 다운로드하고 SD 카드에 write해서 Raspberry pi를 이용 중인 사용자라면 작업에 앞서 용량을 확인해야 한다. 초기 Raspbian은 write하는 SD 카드에 대해 모든 용량을 사용하지 않고 default로 설정한 2GB만 사용한다(8GB SD 카드에 Raspbain을 설치한 경우 남은 6GB는 미사용 공간으로 남게 된다). 사용 중인 용량은 다음 명령으로 확인할 수 있다.

```
df -T
```

그림 2-2 라즈베리 파이의 초기용량 확인 결과



Filesystem	Type	1K-blocks	Used	Available	Use%	Mounted on
/dev/root	ext4	7562520	3761996	3438488	53%	/
devtmpfs	devtmpfs	469752	0	469752	0%	/dev
tmpfs	tmpfs	474068	0	474068	0%	/dev/shm
tmpfs	tmpfs	474068	6408	467660	2%	/run
tmpfs	tmpfs	5120	4	5116	1%	/run/lock
tmpfs	tmpfs	474068	0	474068	0%	/sys/fs/cgroup
/dev/mmcblk0p1	vfat	61384	20296	41088	34%	/boot
tmpfs	tmpfs	94816	0	94816	0%	/run/user/1000

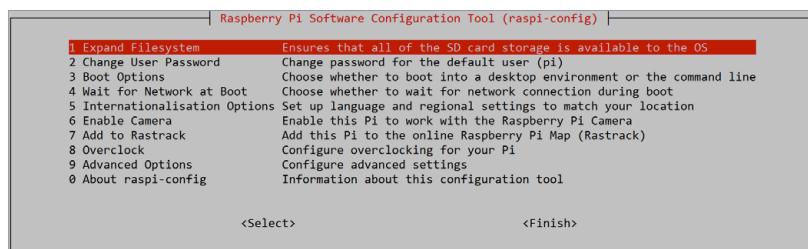
```
pi@raspberrypi: ~ $
```

Raspbian의 용량을 재설정(Re-size)하지 않고 웹 서버와 Let's Encrypt를 설치하면 용량이 부족하여 설치과정에서 오류가 발생할 수 있다. 설치 중 용량 부족으로 인한 오류를 방지하려면 먼저 Raspbian의 용량을 재설정한다. Raspbian의 용량을 재설정하는 방법에는 여러 가지가 있으나 여기서는 raspi-config를 이용하는 방법을 알아보겠다.

창에 다음 명령을 입력하면 raspi-config를 설정하는 화면으로 이동한다.

```
sudo raspi-config
```

그림 2-3 SD 카드 용량 설정 화면



[그림 2-3]의 화면이 보이면 [Expand Filesystem]을 선택하고 적절한 드라이브에서 [Select]를 누른 후 재부팅하면 Raspbian의 용량이 재설정된다. 앞에서 사용한 df -T 명령을 다시 입력하여 늘어난 용량을 확인한다.

리포지터리 업데이트와 Git 및 Python 설치

apt 리포지터리를 업데이트하여 패키지를 업데이트하는 과정도 필요하다. 다음 명령을 입력하면 해당 작업을 수행한다.

```
$ sudo apt-get update  
$ sudo apt-get upgrade
```

리포지터리 업데이트와 패키지 업그레이드를 종료한 후 Let's Encrypt를 다운로드하기 위하여 Git을 설치한다. Git을 설치하는 명령어는 다음과 같다.

```
$ sudo apt-get install git
```

또한, Let's Encrypt 설치를 위해서는 Python 사용이 필수이므로 Python이 설치되어 있는지 확인한다. 다음 명령을 실행하면 설치되어 있는지 확인할 수 있다.

```
$ sudo add-apt-repository ppa:fkrull/deadsnakes  
$ sudo apt-get update  
$ sudo apt-get install python2.7
```

2.2 Redhat 계열

이번에는 Redhat 계열 운영체제에서 Let's Encrypt를 설치하기 위한 환경설정을 해보자. 특히 Redhat 계열 운영체제에서는 Python 2.7.x 버전으로 업데이트해야 정상 작동하므로 이 점을 유의한다.

DNF 리스트 업데이트 또는 YUM 패키지 관리자 업데이트

별다른 오류 없이 설치하려면 DNF 리포지터리를 업데이트한다. 업데이트 명령어는 다음과 같다. 상황에 따라 YUM 패키지 관리자를 통하여 리포지터리 업데이트를 사용해도 무방하다.

```
$ sudo dnf update
```

Git 패키지 설치

Let's Encrypt는 기본적으로 소프트웨어 배포를 Git으로 진행한다. 따라서 해당 패키지를 이용해야 정상적으로 Let's Encrypt를 설치할 수 있다. 다음 명령어로 Git를 설치한다(이미 Git이 설치되어 있으면 해당 과정을 건너뛰어도 된다).

```
$ sudo dnf install git
```

Apache 2 웹 서버 버전 확인

현재 사용하는 Apache 2의 버전을 확인한다. 안정적으로 설치하려면 2.x 버전 사용을 추천한다. 버전 확인에는 다음 명령을 이용한다(이 안내서는 Apache 2.4.18 버전을 기준으로 작성하였다).

```
$ apache2 --version
```

웹 서버와 데이터베이스 서비스 데몬 종료

Let's Encrypt는 웹 서버와 사용 중인 데이터베이스를 정상적으로 종료하고 설치해야 나중에 발생할 다양한 오류를 미리 방지할 수 있다. 다음은 각각 Apache 와 MySQL을 종료하는 명령다.

```
$ systemctl stop httpd.service  
$ systemctl stop mysqld.service
```

Python 설치

Let's Encrypt를 설치하려면 반드시 Python을 설치해야 한다. 페도라에는 기본으로 Python이 설치되어 있지만 업데이트를 진행한다. Let's Encrypt에서 권장하는 Python 버전은 2.7이지만 3.x 버전도 사용할 수 있다.

```
$ sudo dnf check-update python  
$ sudo dnf upgrade python
```

2.3 Windows Server 2012

이번에는 Windows Server 2012에서 Let's Encrypt를 설치하기 위한 환경설정을 해보겠다. Windows Server 2012 이전 버전은 IIS에서 SNI를 지원하지 않으므로 Windows Server 2012부터 Let's Encrypt를 사용할 수 있다.⁰¹

Python 설치

Let's Encrypt는 Python으로 만들어진 도구이므로 Python을 필수로 설치하여야 한다. Windows 환경에서는 설치 패키지 파일로 설치하는데, Python 다운로드 페이지⁰²에서 본인의 운영체제에 맞는 MSI 파일을 다운로드하여 설치하면 된다.

그림 2-4 Python 다운로드

Python 2.7.12					
Version	Operating System	Description	MD5 Sum	File Size	PGP
Gzipped source tarball	Source release		88d61f82e3616a4be952828b3694109d	16935960	SIG
XZ compressed source tarball	Source release		57dffce9ceeb8b2ab5f82af1d8e9a69	12390820	SIG
Mac OS X 32-bit i386/PPC installer	Mac OS X	for Mac OS X 10.5 and later	3adbedcc935a0db1ab08aa41fec4e33	24214628	SIG
Mac OS X 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	86bedde2bedc37335d27aa9df84952e1	22355024	SIG
Windows debug information files	Windows		1751598e16431be04ef4f24ca52b53a	24678566	SIG
Windows debug information files for 64-bit binaries	Windows		c5433a7fca9ede6e52835bd40e40aa8d	25481382	SIG
Windows help file	Windows		7bc4e15ecae8ede7c85e122f0a6d5f27	6224175	SIG
Windows x86-64 MSI installer	Windows	for AMD64/EM64T/x64, not Itanium processors	8fa13925db87638aa472a3e794ca4ee3	19820544	SIG
Windows x86 MSI installer	Windows		fe0ef5b6fd02722f32f7284324934f9d	18907136	SIG

01 <http://stackoverflow.com/questions/36274691/how-to-install-lets-encrypt-on-windows-server-2008>(단축 URL <http://goo.gl/Quvuf5>), https://en.wikipedia.org/wiki/Server_Name_Indication

02 <https://www.python.org/downloads/release/python-2712/>

Let's Encrypt는 Python 2.7 버전을 권장하지만, 3.x 버전도 사용할 수 있으므로 서버 운영자의 편의에 맞게 설치한다.

Apache 웹 서버에서의 Let's Encrypt 적용

이번에는 Apache 웹 서버에 Let's Encrypt를 적용하는 과정을 살펴보겠다. 이 안내서에서는 Apache 2.4.18 버전을 사용하므로 적용 시 참고하기 바란다.

3.1 Debian 계열

먼저 Debian 기반의 운영체제에서 Let's Encrypt를 설치하는 과정을 살펴보자. 먼저 Let's Encrypt를 Git으로 다운로드한다.

```
$ git clone https://github.com/letsencrypt/letsencrypt
```

해당 명령을 입력하면 현재 셸에서 작업 중인 디렉터리에 프로그램을 다운로드하는데, 사용자가 쉽게 접근할 수 있는 디렉터리에 다운로드하기를 바란다.

그림 3-1 Git을 이용한 Let's Encrypt 프로그램 다운로드

```
root@debian:/home/sts# git clone https://github.com/letsencrypt/letsencrypt
Cloning into 'letsencrypt'...
remote: Counting objects: 32003, done.
remote: Compressing objects: 100% (258/258), done.
remote: Total 32003 (delta 136), reused 0 (delta 0), pack-reused 31745
Receiving objects: 100% (32003/32003), 8.34 MiB | 1.73 MiB/s, done.
Resolving deltas: 100% (22640/22640), done.
Checking connectivity... done.
root@debian:/home/sts#
```

다운로드한 Let's Encrypt를 설치할 차례다. 다음 명령을 실행하면 Let's Encrypt를 설치할 때 프로그램 구동에 필요한 별도의 라이브러리도 함께 설치하는 의존성 설치를 진행하게 된다. 환경에 따라 수분의 시간이 요구될 수 있다.

```
$ ./letsencrypt-auto --help
```

설치가 완료되면 다음과 같은 화면을 볼 수 있다. 이미 관련 의존성 라이브러리가 모두 설치되어 있어 별도로 다운로드를 진행한 구문이 보이지는 않지만, 처음 설치할 때는 상당량의 패키지를 설치하게 된다.

그림 3-2 Let's Encrypt 프로그램 설치 화면

```
root@debian:/home/sts/letsencrypt# ./letsencrypt-auto --help
Checking for new version...
Requesting root privileges to run letsencrypt...
  /root/.local/share/letsencrypt/bin/letsencrypt --no-self-upgrade --help

letsencrypt-auto [SUBCOMMAND] [options] [-d domain] [-d domain] ...

The Let's Encrypt agent can obtain and install HTTPS/TLS/SSL certificates. By
default, it will attempt to use a webserver both for obtaining and installing
the cert. Major SUBCOMMANDS are:

(default) run      Obtain & install a cert in your current webserver
certonly          Obtain cert, but do not install it (aka "auth")
install           Install a previously obtained cert in a server
renew             Renew previously obtained certs that are near expiry
revoke            Revoke a previously obtained certificate
rollback          Rollback server configuration changes made during install
config_changes   Show changes made to server config during installation
plugins           Display information about installed plugins

Choice of server plugins for obtaining and installing cert:

--apache          Use the Apache plugin for authentication & installation
--standalone       Run a standalone webserver for authentication
(nginx support is experimental, buggy, and not installed by default)
--webroot          Place files in a server's webroot folder for authentication

OR use different plugins to obtain (authenticate) the cert and then install it:

--authenticator standalone --installer apache

More detailed help:

-h, --help [topic]  print this message, or detailed help on a topic;
                   the available topics are:

all, automation, paths, security, testing, or any of the subcommands or
plugins (certonly, install, nginx, apache, standalone, webroot, etc)

root@debian:/home/sts/letsencrypt#
```

이 프로그램은 인증서의 발급과 적용, 해제까지 인증서를 통합 관리하는 데 필요한 모든 기능을 수행 할 수 있다. Let's Encrypt가 설치된 폴더에서 'letsencrypt

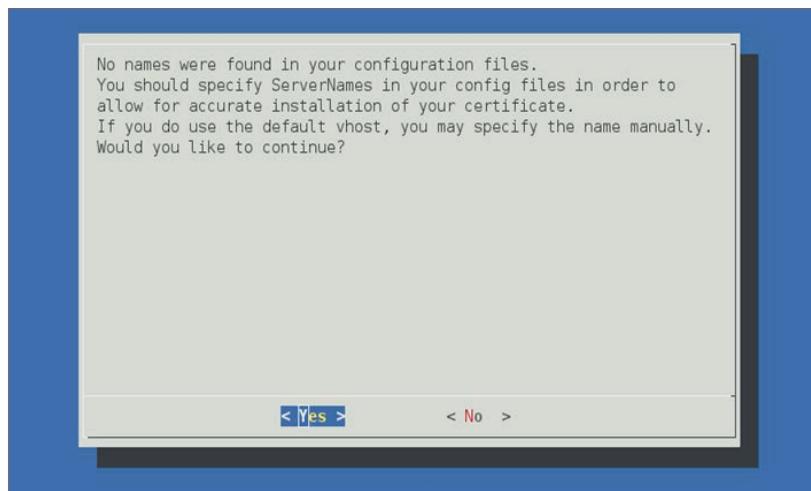
-auto'라는 실행파일을 관리자가 자주 이용하는 디렉터리에 링크하면 인증서와 관련된 작업을 수행할 때 편리하게 이용할 수 있다.

인증서 관리에 필요한 Let's Encrypt를 성공적으로 설치하였으니 이제 인증서를 발급받을 일만 남았다. Apache 2일 경우 Let's Encrypt가 지원하는 플러그인을 활용하면 설치보다 쉽게 인증서 발급과 적용이 가능하므로 너무 걱정하지 않아도 된다. 우선 Let's Encrypt가 설치된 디렉터리에서 다음 명령을 입력한다.

```
$ ./letsencrypt-auto --apache
```

앞의 명령을 실행하면 다음과 같은 화면이 나온다. 'Apache 2 웹 서버의 config 파일을 조사하여 보니 서버 이름이 설정되어 있지 않다'는 뜻이다. [YES]를 누르면 'Server Name'을 설정하는 창으로 이동한다.

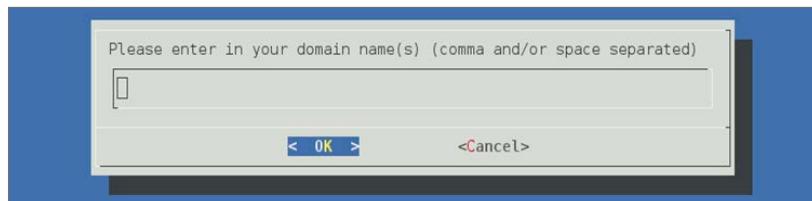
그림 3-3 프로그램 실행 시 관리자 이메일 확인



[YES]를 누르면 다음과 같이 'Server Name'을 설정하는 화면이 보인다. 이곳에 HTTPS를 적용할 도메인 이름을 입력한다(현재 웹 서버에 물려 있는 도메인을 입력하면

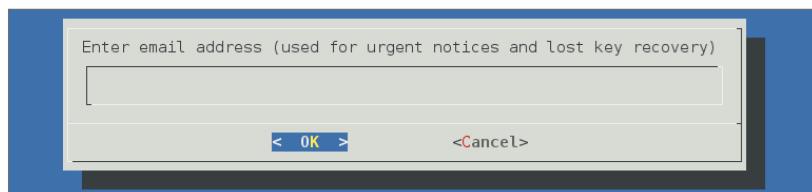
된다). 여러 개의 도메인이 물려 있는 웹 서버라면 스페이스 바로 구분하여 도메인을 입력하면 된다.

그림 3-4 HTTPS 프로토콜을 적용할 도메인 이름 입력



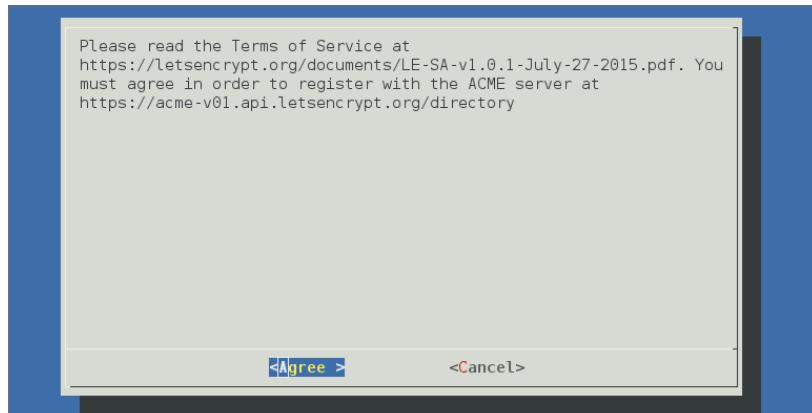
도메인 등록이 끝나면 공지사항을 받을 이메일을 입력하라고 나오는데, 각자의 이메일 주소를 입력한 후 [OK]를 누르면 된다.

그림 3-5 이메일 주소 입력



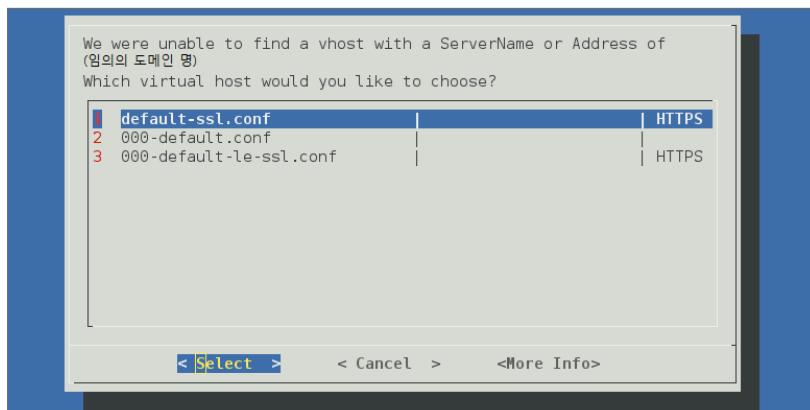
이메일을 등록하면 약관 동의 화면이 나온다. [Agree]를 눌러 약관에 동의한 후 계속 진행한다.

그림 3-6 Let's Encrypt 약관 동의



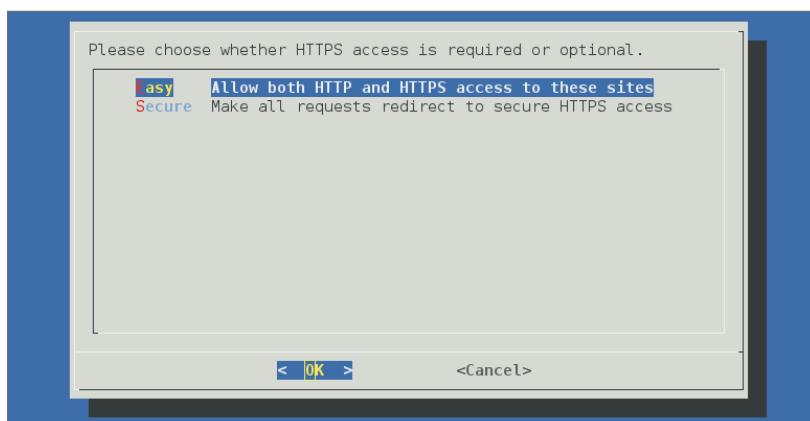
약관에 동의한 후에는 현재 설정된 가상 호스트 목록이 표시되며, HTTPS를 사용 하려는 가상 호스트 주소 또는 관련 HTTP 구성 파일을 선택한다.

그림 3-7 가상 호스트 옵션 설정



그다음 HTTPS 접속 옵션이 나오는데, 2가지 옵션 중 고를 수 있다. 첫 번째 'Easy'는 'HTTPS://~'로 들어오는 요청만 HTTPS를 제공하는 옵션이고, 두 번째 'Secure'는 해당 도메인으로 들어오는 모든 요청을 HTTPS로 만들어 주는 옵션이다. 적용하려는 서버의 상황에 따라 모든 연결에 HTTPS를 적용할 필요가 없다면 [Easy]를 선택하면 된다.

그림 3-8 HTTPS 접속 옵션 설정



여기까지 모두 마치면 HTTPS가 정상적으로 적용된다. 정상적으로 HTTPS 요청이 이루어지는지 확인하려면 도메인으로 이동하여 URL 옆에 녹색 자물쇠 모양이 있는지 확인한다(진행 중 오류가 발생하지 않았다면 정상적으로 적용되어 있을 것이다).

Secure 옵션을 선택해도 URL Redirecting이 이루어지지 않는 경우가 있다. 이럴 때는 직접 Apache 2의 설정 config 파일을 수정해야 한다. 보통 /etc/apache2/apache2.conf 파일에서 가장 아래줄에 다음 구문을 추가하면 성공적으로 작동하는 것을 볼 수 있다.

```
NameVirtualHost *:80
<VirtualHost *:80>
ServerName YourServerName.org
Redirect permanent / https://YourServerName.org/
</VirtualHost>
```

3.2 Redhat 계열

이번에는 Redhat 계열 운영체제에 Let's Encrypt를 설치하는 과정을 살펴보자. 먼저 Let's Encrypt를 DNF 패키지 관리자로 설치하기 위하여 다음 명령을 입력한다.

```
$ sudo dnf install letsencrypt
```

해당 명령을 실행하면 [그림 3-9]와 같이 설치할 패키지들에 대하여 사용자에게 묻는 화면이 나온다. 패키지 내용이 제대로 나오는지 확인한 후 설치를 진행한다.

그림 3-9 DNF 패키지 매니저로 설치하는 과정 (1)

```
[root@gregmid ~]# dnf install letsencrypt
Last metadata expiration check performed 0:08:51 ago on Tue
Dependencies resolved.
=====
           Package                     Arch
=====
Installing:
  dialog                         x86_64
  letsencrypt                    noarch
  python-chardet                  noarch
  python-configureobj              noarch
  python-mock                     noarch
  python-parsedatetime            noarch
  python-requests                 noarch
  python-urllib3                  noarch
  python-werkzeug                 noarch
  python-zope-component           noarch
  python-zope-event               noarch
  python-zope-interface           x86_64
  python2-acme                    noarch
  python2-configargparse           noarch
  python2-dialog                  noarch
  python2-letsencrypt              noarch
  python2-ndg_httpsclient          noarch
  python2-psutil                   x86_64
  python2-pyrfc3339                noarch
  pytz                            noarch

Transaction Summary
=====
Install 20 Packages

Total download size: 3.2 M
Installed size: 15 M
Is this ok [y/N]: _
```

그림 3-10 DNF 패키지 매니저로 설치하는 과정 (2)

```
Transaction Summary
=====
Install 20 Packages

Total download size: 3.2 M
Installed size: 15 M
Is this ok [y/N]: y
Downloading Packages:
(1/20): python-configureobj-5.0.5-3.fc23.noarch.rpm
(2/20): python-parsedatetime-1.5-1.fc23.noarch.rpm
(3/20): letsencrypt-0.1.1-2.fc23.noarch.rpm
(4/20): python-zope-component-4.2.1-2.fc23.noarch.rpm
(5/20): python2-letsencrypt-0.1.1-2.fc23.noarch.rpm
(6/20): python2-psutil-3.2.1-2.fc23.x86_64.rpm
(7/20): python2-configurationargparse-0.9.3-3.fc23.noarch.rpm
(8/20): python-zope-interface-4.1.2-2.fc23.x86_64.rpm
(9/20): python2-acme-0.1.1-1.fc23.noarch.rpm
(10/20): python2-dialog-3.3.0-7.fc23.noarch.rpm
(11/20): python-zope-event-4.0.3-3.fc23.noarch.rpm
(12/20): pytz-2015.4-1.fc23.noarch.rpm
(13/20): python2-pyrfc339-1.0-1.fc23.noarch.rpm
(14/20): python-mock-1.0.1-7.fc23.noarch.rpm
(15/20): python-werkzeug-0.9.6-1.fc22.noarch.rpm
(16/20): dialog-1.2-16.20150528.fc23.x86_64.rpm
(17/20): python2-ndg_httpsclient-0.4.0-2.fc23.noarch.rpm
(18/20): python-requests-2.9.1-1.fc23.noarch.rpm
(19/20): python-chardet-2.2.1-3.fc23.noarch.rpm
(20/20): python-urllib3-1.13.1-1.fc23.noarch.rpm

Total
Running transaction check
Transaction check succeeded.
Running transaction test
```

그림 3-11 DNF 패키지 매니저로 설치하는 과정 (3)

```
Installing : dialog-1.2-16.20150528.fc23.x86_64
Installing : python2-dialog-3.3.0-7.fc23.noarch
Installing : python-mock-1.0.1-7.fc23.noarch
Installing : python2-pyrfc3339-1.0-1.fc23.noarch
Installing : pytz-2015.4-1.fc23.noarch
Installing : python-werkzeug-0.9.6-1.fc22.noarch
Installing : python2-acme-0.1.1-1.fc23.noarch
Installing : python2-conf igargparse-0.9.3-3.fc23.noarch
Installing : python2-psutil-3.2.1-2.fc23.x86_64
Installing : python-parsedatetime-1.5-1.fc23.noarch
Installing : python-conf igobj-5.0.5-3.fc23.noarch
Installing : python2-letsencrypt-0.1.1-2.fc23.noarch
Installing : letsencrypt-0.1.1-2.fc23.noarch
Verifying  : letsencrypt-0.1.1-2.fc23.noarch
Verifying  : python2-letsencrypt-0.1.1-2.fc23.noarch
Verifying  : python-conf igobj-5.0.5-3.fc23.noarch
Verifying  : python-parsedatetime-1.5-1.fc23.noarch
Verifying  : python-zope-component-4.2.1-2.fc23.noarch
Verifying  : python-zope-interface-4.1.2-2.fc23.x86_64
Verifying  : python2-psutil-3.2.1-2.fc23.x86_64
Verifying  : python2-acme-0.1.1-1.fc23.noarch
Verifying  : python2-conf igargparse-0.9.3-3.fc23.noarch
Verifying  : python2-dialog-3.3.0-7.fc23.noarch
Verifying  : python-zope-event-4.0.3-3.fc23.noarch
Verifying  : python-werkzeug-0.9.6-1.fc22.noarch
Verifying  : pytz-2015.4-1.fc23.noarch
Verifying  : python2-pyrfc3339-1.0-1.fc23.noarch
Verifying  : python-mock-1.0.1-7.fc23.noarch
Verifying  : dialog-1.2-16.20150528.fc23.x86_64
Verifying  : python2-ndg_httpsclient-0.4.0-2.fc23.noarch
Verifying  : python-requests-2.9.1-1.fc23.noarch
Verifying  : python-chardet-2.2.1-3.fc23.noarch
Verifying  : python-urllib3-1.13.1-1.fc23.noarch

Installed:
dialog.x86_64 1.2-16.20150528.fc23                               letsencrypt
python-conf igobj.noarch 5.0.5-3.fc23                            python-mock
python-requests.noarch 2.9.1-1.fc23                                python-urllib3
python-zope-component.noarch 4.2.1-2.fc23                           python-zope
python2-acme.noarch 0.1.1-1.fc23                                 python2-conf ig
python2-letsencrypt.noarch 0.1.1-2.fc23                           python2-ndg
python2-pyrfc3339.noarch 1.0-1.fc23                                pytz.noarch
```

설치가 끝나면 --help 명령을 입력하여 정상적으로 설치되었는지 확인한다.

```
letsencrypt -- help
```

다음 결과 화면이 나온다면 정상적으로 설치된 것이다.

그림 3-12 정상 설치 여부 확인

```
[root@regnid ~]# letsencrypt --help
letsencrypt [SUBCOMMAND] [options] [-d domain] [-d domain] ...
The Let's Encrypt agent can obtain and install HTTPS/TLS/SSL certificates. By
default, it will attempt to use a webserver both for obtaining and installing
the cert. Major SUBCOMMANDS are:
(default) run      Obtain & install a cert in your current webserver
certonly          Obtain cert, but do not install it (aka "auth")
install           Install a previously obtained cert in a server
revoke            Revoke a previously obtained certificate
rollback          Rollback server configuration changes made during install
config_changes   Show changes made to server config during installation
plugins           Display information about installed plugins

Choice of server plugins for obtaining and installing cert:
(the apache plugin is not installed)
--standalone      Run a standalone webserver for authentication
(nginx support is experimental, buggy, and not installed by default)
--webroot         Place files in a server's webroot folder for authentication

OR use different plugins to obtain (authenticate) the cert and then install it:
--authenticator standalone --installer apache

More detailed help:
-h, --help [topic]  print this message, or detailed help on a topic;
                   the available topics are:
all, automation, paths, security, testing, or any of the subcommands or
plugins (certonly, install, nginx, apache, standalone, webroot, etc)

[root@regnid ~]# _
```

그리고 다음 명령을 입력하여 서버 도메인을 CA에 등록한다. 명령어 구성은 다음과 같다.

```
letsencrypt --text --email (user-email-address) --renew-by-default -d (domain-name) --agree-tos --webroot-path (letsencrypt-main-directory) certonly
```

그림 3-13 Let's Encrypt CA에 서버 도메인 등록

```
[root@regnid ~]# letsencrypt --text --email rudals531@gmail.com --renew-by-default -d regnid.ao.to --agree-to-s --webroot --webroot-path /var/www/html certonly

IMPORTANT NOTES:
- If you lose your account credentials, you can recover through
  e-mails sent to rudals531@gmail.com.
- Congratulations! Your certificate and chain have been saved at
  /etc/letsencrypt/live/regnid.ao.to/fullchain.pem. Your cert will
  expire on 2016-04-11. To obtain a new version of the certificate in
  the future, simply run Let's Encrypt again.
- Your account credentials have been saved in your Let's Encrypt
  configuration directory at /etc/letsencrypt. You should make a
  secure backup of this folder now. This configuration directory will
  also contain certificates and private keys obtained by Let's
  Encrypt so making regular backups of this folder is ideal.
- If you like Let's Encrypt, please consider supporting our work by:

  Donating to ISRG / Let's Encrypt:  https://letsencrypt.org/donate
  Donating to EFF:                  https://eff.org/donate-le

[root@regnid ~]# _
```

이어서 유효한 설정 값이 복사된 파일에 대하여 심볼릭 symbolic 링크를 만들고 설정 파일을 복사한다. 해당 명령어는 다음과 같이 구성된다.

```
$ ln -s /etc/letsencrypt/live/www.example.com/cert.pem /etc/pki/tls/certs/www.example.com.crt
$ ln -s /etc/letsencrypt/live/www.example.com/chain.pem /etc/pki/tls/certs/www.example.com.chain.crt
$ ln -s /etc/letsencrypt/live/www.example.com/privkey.pem /etc/pki/tls/private/www.example.com.key
$ cp /etc/httpd/conf.d/ssl.conf{,.backup}
$ sed -i 's@(\$SSLCertificateFile) .*\@1 /etc/pki/tls/certs/www.example.com.crt@' /etc/httpd/conf.d/ssl.conf
$ sed -i 's@(\$SSLCertificateKeyFile) .*\@1 /etc/pki/tls/private/www.example.com.key@' /etc/httpd/conf.d/ssl.conf
$ sed -i 's@(\$SSLCertificateChainFile) .*\@1 /etc/pki/tls/certs/www.example.com.chain.crt@' /etc/httpd/conf.d/ssl.conf
```

그림 3-14 추가 설정 적용

```
[root@regnid ~]# ln -s /etc/letsencrypt/live/regnid.ao.to/cert.pem /etc/pki/tls/certs/regnid.ao.to.crt
[root@regnid ~]# ln -s /etc/letsencrypt/live/regnid.ao.to/chain.pem /etc/pki/tls/certs/regnid.ao.to.chain.crt
[root@regnid ~]# ln -s /etc/letsencrypt/live/regnid.ao.to/privkey.pem /etc/pki/tls/private/regnid.ao.to.key
[root@regnid ~]# cp /etc/httpd/conf.d/ssl.conf{,.backup}
[root@regnid ~]# sed -i 's@(\$SSLCertificateFile) .*\@1 /etc/pki/tls/certs/regnid.ao.to.crt@' /etc/httpd/conf.d/ssl.conf
[root@regnid ~]# sed -i 's@(\$SSLCertificateKeyFile) .*\@1 /etc/pki/tls/private/regnid.ao.to.key@' /etc/httpd/conf.d/ssl.conf
[root@regnid ~]# sed -i 's@(\$SSLCertificateChainFile) .*\@1 /etc/pki/tls/certs/regnid.ao.to.chain.crt@' /etc/httpd/conf.d/ssl.conf
[root@regnid ~]# _
```

앞의 명령어에 적용된 SELinux 정책에 대한 익스플로잇을 방지하기 위하여 다음 명령을 추가로 입력한다.

```
$ semanage fcontext -a -t cert_t '/etc/letsencrypt/(archive|live)(/.*)?'
$ restorecon -Rv /etc/letsencrypt
```

앞의 명령을 실행하면 다음과 같이 적용되는 것을 볼 수 있다.

그림 3-15 SELinux 관련 이슈 해결 명령 실행

```
[root@regnid ~]# ls -z /etc/letsencrypt/live/regnid.os_to-cert.pem /etc/pki/tls/certs/regnid.os_to.crt
[root@regnid ~]# ls -z /etc/letsencrypt/live/regnid.os_to-chain.pem /etc/pki/tls/certs/regnid.os_to-chain.crt
[root@regnid ~]# cp /etc/httpd/conf.d/ssl.conf.bakup
[root@regnid ~]# sed -i '$a \SSLCertificateFile /etc/pki/tls/certs/regnid.os_to.crt' /etc/httpd/conf.d/ssl.conf
[root@regnid ~]# sed -i '$a \SSLCertificateKeyFile /etc/pki/tls/private/regnid.os_to.key' /etc/httpd/conf.d/ssl.conf
[root@regnid ~]# sed -i '$a \SSLCertificateChainFile /etc/pki/tls/certs/regnid.os_to-chain.crt' /etc/httpd/conf.d/ssl.conf
[root@regnid ~]# semanage fcontext -a -t cert_t '/etc/letsencrypt/(archive|live)(/.*)?'
[root@regnid ~]# restorecon -Rv /etc/letsencrypt
restorecon reset /etc/letsencrypt/archive context unconfined_u:object_r:cert_t:s0
restorecon reset /etc/letsencrypt/archive/regnid.os_to-cert.pem context unconfined_u:object_r:cert_t:s0
restorecon reset /etc/letsencrypt/archive/regnid.os_to-chain.pem context unconfined_u:object_r:cert_t:s0
restorecon reset /etc/letsencrypt/archive/regnid.os_to-chain1.pem context unconfined_u:object_r:cert_t:s0
restorecon reset /etc/letsencrypt/archive/regnid.os_to-fullchain.pem context unconfined_u:object_r:cert_t:s0
restorecon reset /etc/letsencrypt/live context unconfined_u:object_r:cert_t:s0
restorecon reset /etc/letsencrypt/live/regnid.os_to-cert.pem context unconfined_u:object_r:cert_t:s0
restorecon reset /etc/letsencrypt/live/regnid.os_to-chain.pem context unconfined_u:object_r:cert_t:s0
restorecon reset /etc/letsencrypt/live/regnid.os_to-chain1.pem context unconfined_u:object_r:cert_t:s0
[root@regnid ~]# systemctl restart httpd.service
[root@regnid ~]
```

추가로 HTTP 주소로 접근하더라도 HTTPS 프로토콜이 적용된 웹 서버로 리다이렉트하도록 다음 설정 값을 httpd.conf에 추가한다.

```
<VirtualHost *:80>
    RewriteEngine On
    RewriteCond %{HTTPS} off
    RewriteRule (.*) https:// %{HTTP_HOST}%{REQUEST_URI}
</VirtualHost>
```

Nginx 웹 서버에서의 Let's Encrypt 적용

Nginx 웹 서버에서 Let's Encrypt를 적용하는 방법에 대하여 살펴보겠다. 이 안내서에는 nginx-1.9.13 버전을 사용하므로 적용 시 참고하기 바란다.

4.1 Debian 계열

먼저 Debian 계열 운영체제를 살펴보자. 서버가 정상 작동하는지를 확인한 후 Nginx 관리 설정을 별도로 생성하여야 한다. 이는 Nginx의 독특한 특성으로, 마스터 프로세스와 워커 프로세스를 두어 웹 서버 관리와 관련 작업(예를 들어, 서버 사이드에서 실행되는 여러 가지 데몬)들을 별도로 관리하기 때문이다. 다음 명령으로 Nginx 관리 설정과 Nginx HTML 폴더에 접근 권한을 부여한다.

```
$ useradd --shell /usr/sbin/nologin www-data  
$ chown -R dev:dev /var/log/nginx /usr/share/nginx/html
```

Nginx 설정을 마친 후 다른 환경에서와 마찬가지로 Let's Encrypt를 설치하기 위하여 Git을 이용한다. Git으로 Let's Encrypt를 다운로드한 후 자동화 스크립트로 설치한다.

다만 Raspberry Pi 같은 경우 ARM 프로세서를 사용하기 때문에 별도의 Python 설정을 통하여 Let's Encrypt 설치를 진행하여야 한다. 따라서 설치 전

다음 명령을 입력하여 Python 가상환경 설정 작업을 한다. 해당 명령어는 Let's Encrypt를 다운로드한 경로에서 수행하여야 정상적으로 작동하므로 이에 유의 한다.

```
$ virtualenv --no-site-packages -p python2 venv
$ ./venv/bin/pip install -r readthedocs.org.requirements.txt acme
```

앞의 명령을 모두 수행한 후 설치 스크립트로 Let's Encrypt를 설치한다. 다음 명령을 실행하면 설치된다.

```
$ git clone https://github.com/letsencrypt/letsencrypt
$ ./letsencrypt-auto -help
```

설치를 마친 후 인증서를 발급한다. 인증서를 발급받기 전에 웹 서버를 종료하고
발급받는 것을 잊지 않도록 한다. 웹 서버 종료 및 인증서 발급 명령어는 다음과
같다.

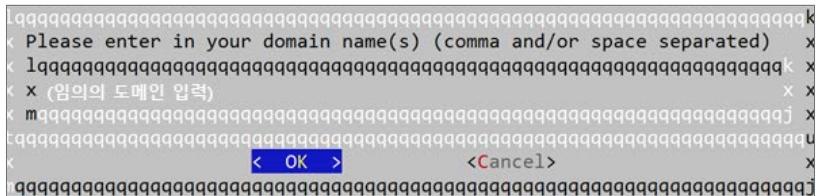
```
$ sudo /etc/init.d/nginx stop  
$ ./letsencrypt-auto certonly
```

다음으로 관리용 이메일 계정과 HTTPS 프로토콜을 적용하기 위한 도메인을 입력한다. 다음과 같은 화면이 나오면 이메일 주소를 입력하고 [OK]를 누른다.

그림 4-1 이메일 주소 입력

그다음 도메인을 입력하고 [OK]를 누른다.

그림 4-2 도메인 주소 입력



정상적으로 발급되면 만들어진 인증서가 위치한 디렉터리 경로를 알 수 있다.

그림 4-3 인증서 주소 확인

IMPORTANT NOTES:

- Congratulations! Your certificate and chain have been saved at
(인증서가 저장될 디렉토리 경로) Your
cert will expire on (만료일자) . To obtain a new version of the
certificate in the future, simply run Let's Encrypt again.
- If you like Let's Encrypt, please consider supporting our work by:

Donating to ISRG / Let's Encrypt: <https://letsencrypt.org/donate>
Donating to EFF: <https://eff.org/donate-le>

이제 발급받은 인증서를 Nginx에 적용하자. 인증서를 적용하여 HTTPS 서비스를 제공하려면 Nginx 설정 파일을 수정하여 인증서의 위치를 대입하면 된다. 다음 명령을 입력하여 Nginx 설정파일이 위치한 곳으로 이동한다.

```
$ cd /etc/nginx/sites-available/
```

해당 위치에 있는 default 파일에 다음 내용을 추가하여 인증서의 위치를 입력한다. 해당 작업은 반드시 root 권한으로 수행하여야 한다는 점을 유의하자. [Your domain] 부분에는 각자의 도메인을 입력한다.

```
listen 443 ssl;
    ssl_certificate /etc/letsencrypt/live/[Your domain]/cert.pem;
```

```
ssl_certificate_key /etc/letsencrypt/live/[Your domain]/privkey.pem;
ssl_stapling on;
ssl_stapling_verify on;
ssl_trusted_certificate /etc/letsencrypt/live/[Your domain]/fullchain.pem;

(종략)

Server_name [Your Domain];
```

파일을 수정한 후 Nginx 설정이 정상적으로 적용되는지 다음 명령으로 확인한다.

```
$ sudo nginx -t
```

성공적으로 설정 파일이 적용되면 다음과 같은 화면을 볼 수 있다.

그림 4-4 Nginx 설정 적용

```
pi@raspberrypi:/etc/nginx/sites-available $ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
pi@raspberrypi:/etc/nginx/sites-available $ █
```

Nginx 설정을 완료하고 웹 페이지에 접속하면 성공적으로 HTTPS가 적용된 것을 확인할 수 있다.

4.2 Redhat 계열

이번에는 Redhat 계열 운영체제에서의 설치 방법을 알아보자. 먼저 다음 명령으로 Fedora에서 Git을 이용하여 Let's Encrypt를 다운로드한다.

```
$ git clone https://github.com/letsencrypt/letsencrypt /opt/letsencrypt
```

다운로드한 후 설치할 패키지에 대하여 묻는 화면이 나오면 Let's Encrypt에 대한 패키지를 확인 후 다음 명령을 입력하여 설치를 진행한다.

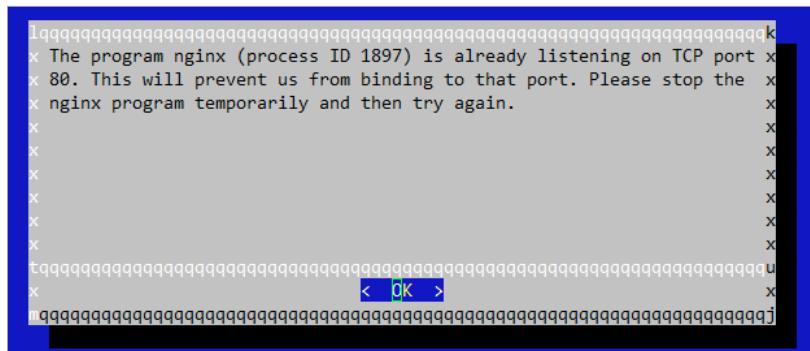
```
$ ./letsencrypt-auto --help
```

이때 Nginx 서비스를 중지한 후 Let's Encrypt를 실행해야 한다. Nginx 서비스를 중지하려면 다음과 같은 명령을 입력한다.

```
$ service nginx stop
```

Nginx 서비스를 중지하지 않고 Let's Encrypt를 설치하면 다음과 같은 오류가 발생한다.

그림 4-5 서비스를 중지하지 않았을 때 발생하는 오류 메시지



Let's Encrypt 설치가 완료되면 다음 명령을 입력하여 해당 디렉토리로 이동하고, 해당 디렉터리에서 Let's Encrypt를 실행한다.

```
$ cd /opt/letsencrypt  
$ ./letsencrypt-auto certonly --standalone
```

Let's Encrypt를 실행하면 [그림 4-6]과 같이 관리용 이메일 계정을 입력하는 화면이 나오는데, 각자의 이메일 주소를 입력한다.

그림 4-6 이메일 주소 입력

Enter email address (used for urgent notices and lost key recovery)

< OK > **<Cancel>**

그 다음 HTTPS 프로토콜을 적용할 도메인을 입력하는 화면이 나오는데, Let's Encrypt를 여러 도메인에 적용할지 단일 도메인에 적용할지를 정할 수 있다. 사용자가 원하는 대로 콤마나 스페이스로 분리한 후 [OK]를 누르면 등록된다.

그림 4-7 적용할 도메인명 입력

앞의 과정 사이에 Let's Encrypt 약관 동의를 위한 창이 나오는데, 화면에 나오는 문서 내용을 참고하여 설치하도록 한다.

그림 4-8 약관 동의 안내창

정상적으로 발급되면 인증서가 위치한 디렉터리 경로를 알 수 있다.

그림 4-9 입력한 도메인 주소와 이메일 확인

```
IMPORTANT NOTES:
- If you lose your account credentials, you can recover through
  e-mails sent to (입력한 이메일 주소)
- Congratulations! Your certificate and chain have been saved at
  /etc/letsencrypt/live/ (입력한 도메인 주소) /fullchain.pem.
  Your cert will expire on 2016-05-25. To obtain a new version of the
  certificate in the future, simply run Let's Encrypt again.
- Your account credentials have been saved in your Let's Encrypt
  configuration directory at /etc/letsencrypt. You should make a
  secure backup of this folder now. This configuration directory will
  also contain certificates and private keys obtained by Let's
  Encrypt so making regular backups of this folder is ideal.
- If you like Let's Encrypt, please consider supporting our work by:

  Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate
  Donating to EFF: https://eff.org/donate-le

[root@localhost letsencrypt]#
```

이어서 nginx.conf 파일을 수정하여 Nginx 웹 서버에 HTTPS를 적용한다.
nginx.conf 파일의 36~37번째 줄을 주석 처리하거나 삭제하면 된다.

그림 4-10 nginx.conf 수정 (1)

```
31      # See http://nginx.org/en/docs/ngx_core_module.html#include
32      # for more information.
33      include /etc/nginx/conf.d/*.conf;
34
35      server {
36          #listen      80 default_server; //remove
37          #listen      [::]:80 default_server; //remove
38
39          # new
40
41          listen 443 ssl;
42
43          server_name (입력한 도메인명);
44
45          ssl_certificate /etc/letsencrypt/live/ (입력한 도메인명)
  /fullchain.pem;
```

그다음 Server 블록에 다음 구문을 추가한다. 구문 추가 시 도메인명에는 앞에서 HTTPS를 적용하려고 입력한 도메인 명으로 수정한다.

```
listen 443 ssl;
server_name 도메인명;
```

```
ssl_certificate /etc/letsencrypt/live/도메인명/fullchain.pem;
ssl_certificate_key /etc/letsencrypt/live/도메인명/privkey.pem;

ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
ssl_prefer_server_ciphers on;
ssl_ciphers 'EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH';
```

그림 4-11 nginx.conf 수정 (2)

```
35     server {
36         #listen      80 default_server; //remove
37         #listen      [::]:80 default_server; //remove
38
39         # new
40
41         listen 443 ssl;
42
43         server_name (입력한 도메인명);
44
45         ssl_certificate /etc/letsencrypt/live/ (입력한 도메인명) /fullchain.pem;
46         ssl_certificate_key /etc/letsencrypt/live/ (입력한 도메인명) /privkey.pem;
47
48         ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
49         ssl_prefer_server_ciphers on;
50         ssl_ciphers 'EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH';
51
52
53         server_name _;
54         root      /usr/share/nginx/html;
55
56         # Load configuration files for the default server block.
57         include /etc/nginx/default.d/*.conf;
58
59         location / {
60     }
```

추가로 HTTP 주소에 접근하더라도 HTTPS 프로토콜이 적용된 웹 서버로 리다이렉트되게 다음 설정 값을 추가한다. 해당 설정 값은 반드시 HTTP 블록 안에 추가해야 한다.

```
server {
    listen 80;
    server_name 도메인명;
    return 301 https://$host$request_uri;
}
```

설정한 값을 적용하려면 다음 명령을 입력하여 Nginx를 재시작한다.

```
$ service nginx start
```

IIS 웹 서버에서의 Let's Encrypt 적용

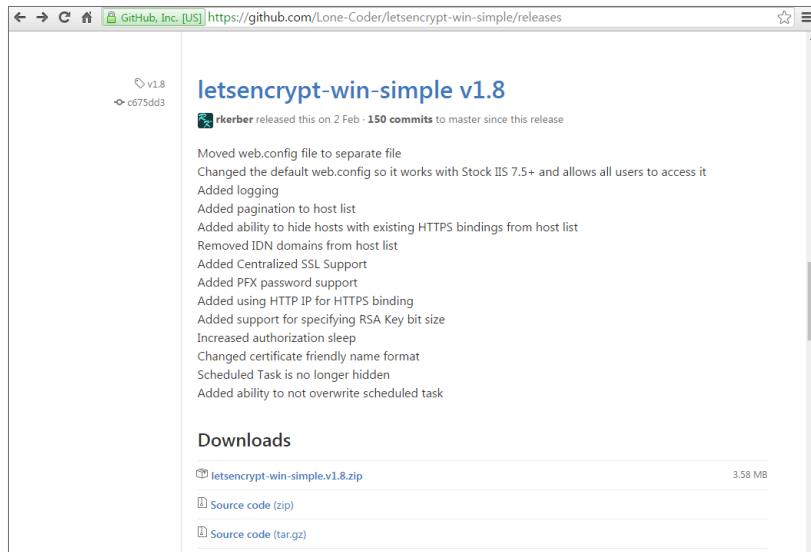
Windows Server 2012 R2의 IIS에서 Let's Encrypt를 설치하는 방법을 살펴보자. 이 안내서에서는 Windows Server 2012 R2(6.3) 버전을 사용하므로 적용 시 참조하기 바란다. 또한, 2016년 4월 기준으로 IIS 웹 서버에서의 Let's Encrypt는 최소 Windows Server 2012부터 작동한다.

IIS 환경에서는 여러 사용자가 제작한 다양한 자동화 도구를 이용하여 편리하게 Let's Encrypt를 설치할 수 있다. 이 안내서에서는 그중 하나를 선택하였는데, 여기서 소개하는 스크립트는 .NET Framework에서 ACME 프로토콜을 사용하기 위하여 수정한 도구다. Let's Encrypt 프로젝트에서 제공하는 ACME 클라이언트 저장 위치는 <https://github.com/letsencrypt/letsencrypt>고, Let's Encrypt 프로젝트를 수정하여 .NET Framework 용으로 수정한 클라이언트를 살펴보려면 <https://github.com/ebekker/ACMESharp>를 참조하길 바란다.⁰¹

먼저 <https://github.com/Lone-Coder/letsencrypt-win-simple/releases>에서 letsencrypt-win-simple 버전을 다운로드한다(이 안내서에서는 1.8 버전을 사용한다).

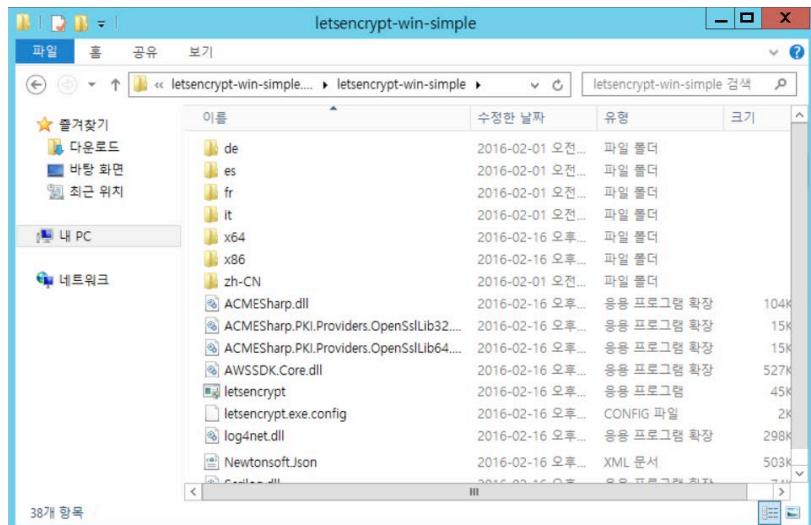
01 더 자세한 사용법 등은 해당 제작자의 GitHub(<https://github.com/Lone-Coder/letsencrypt-win-simple/wiki>)을 참조하길 바란다.

그림 5-1 letsencrypt-win-simple 다운로드



다운로드한 파일의 압축을 해제하고 폴더에 있는 letsencrypt.exe 파일을 관리자 권한으로 실행한다.

그림 5-2 관리자 권한으로 letsencrypt.exe 실행



letsencrypt.exe 파일을 실행하면 다음 그림처럼 서버 관리자가 Let's Encrypt 관리에 사용할 이메일 주소 요구하는데, 관리자 계정으로 사용할 이메일을 입력한 후 Enter를 누른다.

그림 5-3 관리자 이메일 주소 입력

```
Let's Encrypt <Simple Windows ACME Client>

ACME Server: https://acme-v01.api.letsencrypt.org/
Config Folder: C:\Users\He\AppData\Roaming\letsencrypt-win-simple\httpsacme-v01.
api.letsencrypt.org

Getting AcmeServerDirectory
Enter an email address <not public, used for renewal fail notices>: nono4227@gmail.com
```

그다음으로 약관 동의 창이 나오면 해당 링크의 문서를 검토한 후 동의하면 'Y'를 눌러 설치를 진행한다.

그림 5-4 약관 동의

```
Let's Encrypt <Simple Windows ACME Client>

ACME Server: https://acme-v01.api.letsencrypt.org/
Config Folder: C:\Users\He\AppData\Roaming\letsencrypt-win-simple\httpsacme-v01.
api.letsencrypt.org

Getting AcmeServerDirectory
Enter an email address <not public, used for renewal fail notices>: nono4227@gmail.com
Calling Register
Do you agree to https://letsencrypt.org/documents/LE-SA-v1.0.1-July-27-2015.pdf?
<Y/N>
-
```

약관에 동의하면 자동으로 IIS에 등록된 호스트를 스캔한 후 리스트를 나열한다. IIS에 대하여 인증서를 제작할 것인지 모든 호스트에 대하여 인증서를 제작할 것인지를 묻는데, 필요에 따라 번호를 입력하고 진행한다.

그림 5-5 IIS에 등록된 호스트를 스캔

```
Scanning IIS Site Bindings for Hosts
1: IIS sparr0w7.securityplus.or.kr (<SystemDrive\inetpub\wwwroot>)
```

선택한 호스트에 대하여 자동으로 인증서를 등록한 것을 볼 수 있다. 이제 프롬프트에 제시된 날짜에 따라 인증서를 갱신하면 된다. 인증서 갱신과 관련한 이슈는 다음 장에서 설명한다.

그림 5-6 성공적인 인증서 CA 등록

```
Scanning IIS Site Bindings for Hosts
  i: IIS sparr0w7.securityplus.or.kr <SystemDrive>\inetpub\wwwroot>

  M: Generate a certificate manually.
  A: Get certificates for all hosts
  Q: Quit

Which host do you want to get a certificate for: i

Authorizing Identifier sparr0w7.securityplus.or.kr Using Challenge Type http-01
  Writing challenge answer to C:\inetpub\wwwroot\well-known\acme-challenge\wres49_UF4XrIgbqa6JC98ZfM0ScdBDCjedksbI8N@o
  Writing web.config to add extensionless mime type to C:\inetpub\wwwroot\well-known\acme-challenge\web.config
    Answer should now be browsable at http://sparr0w7.securityplus.or.kr/.well-known/acme-challenge/wres49_UF4XrIgbqa6JC98ZfM0ScdBDCjedksbI8N@o
  Submitting answer
  Refreshing authorization
  Authorization Result: valid
  Deleting answer

Requesting Certificate
  Request Status: Created
  Saving Certificate to C:\Users\He\AppData\Roaming\letsencrypt-win-simple\httpsacme-v01.api.letsencrypt.org\sparr0w7.securityplus.or.kr-crt.der
  Saving Issuer Certificate to C:\Users\He\AppData\Roaming\letsencrypt-win-simple\httpsacme-v01.api.letsencrypt.org\ca-009813F47513E5750B43E7431E971E44BD-crt.pem

  Saving Certificate to C:\Users\He\AppData\Roaming\letsencrypt-win-simple\httpsacme-v01.api.letsencrypt.org\sparr0w7.securityplus.or.kr-all.pfx <with no password set>
  Opened Certificate Store "WebHosting"
  Adding Certificate to Store
  Closing Certificate Store
  Adding https Binding
  Committing binding changes to IIS
  Creating Task letsencrypt-win-simple httpsacme-v01.api.letsencrypt.org with Windows Task Scheduler at 9am every day.
  Renewal Scheduled IIS sparr0w7.securityplus.or.kr <SystemDrive>\inetpub\wwwroot> Renew After 2016-04-16
Press enter to continue.
```

Let's Encrypt 인증서 갱신

Let's Encrypt로 발급받은 인증서의 유효기간은 90일이고, `renew` 명령어를 사용하면 자동으로 인증서의 기간을 확인해준다. 하지만 유효기간 만료 30일 이내 일 때 `renew` 명령어를 실행하면 갱신이 진행되므로 인증서 갱신은 특별한 사유가 없는 한 발급일 또는 갱신일로부터 60일이 지난 후부터 진행하길 권장한다.

6.1 Apache 웹 서버

Apache 웹 서버에서는 `renew` 명령어를 사용거나 수동으로 인증서를 갱신할 수 있다. `renew` 명령어는 다음과 같이 사용하는데, 이 명령어는 Let's Encrypt가 설치된 디렉터리에서 실행해야 한다.

```
$ ./letsencrypt-auto renew
```

유효기간 안에 해당 명령어를 사용하면 다음과 같은 형태의 메시지를 볼 수 있다.

```
Checking for new version...
Requesting root privileges to run letsencrypt...
/root/.local/share/letsencrypt/bin/letsencrypt renew
Processing /etc/letsencrypt/renewal/[도메인명].conf

The following certs are not due for renewal yet:
  /etc/letsencrypt/live/[도메인명]/fullchain.pem (skipped)
No renewals were attempted.
```

갱신에 성공하면 다음과 같은 메시지를 볼 수 있다.

IMPORTANT NOTES:

- Congratulations! Your certificate and chain have been saved at /etc/letsencrypt/live/[도메인 명]/fullchain.pem. Your cert will expire on (yyyy-mm-dd).

To obtain a new version of the certificate in the future, simply run Let's Encrypt again.

일반적으로 서버를 운영할 때 이러한 명령은 crontab⁰¹을 이용하여 자동으로 갱신을 확인하고, 갱신기간이면 자동으로 갱신하게 하는 것이 훨씬 효율적이다.

이번에는 crontab 예약에서 인증서 유효기간이 30일 이내면 자동으로 갱신하는 명령어를 작성해보자. 해당 동작은 root 계정으로 작업해야 함에 유의한다. vi 편집기를 이용하여 crontab 예약을 수정한다.

```
$ sudo crontab -e
```

다음은 매주 월요일 오전 05시 00분에 ‘letsencrypt-auto renew’ 명령을 실행하고 해당 명령에 대한 결과는 ‘/var/log/le-renew.log’에 저장하는 내용으로, 이 내용을 추가하여 자동 갱신되게 한다. 입력을 완료하면 저장한 후 종료한다.

```
0 5 * * 1 /etc/letsencrypt/letsencrypt-auto renew >> /var/log/le-renew.log
```

6.2 Nginx 웹 서버

Nginx 웹 서버 환경에서 인증서 갱신은 Apache 웹 서버와 유사하다. renew 명령어로 인증서의 기간을 확인하고 연장하면 된다. 해당 명령어 또한 Let's

01 cron에 대한 상세한 설명은 <https://en.wikipedia.org/wiki/Cron>을 참조하라.

Encrypt가 설치된 디렉터리에서 실행해야 한다. 명령어는 다음과 같다.

```
$ ./letsencrypt-auto renew
```

유효기간 안에 해당 명령어를 사용하면 다음과 같은 메시지를 볼 수 있다.

```
Checking for new version...
Requesting root privileges to run letsencrypt...
/root/.local/share/letsencrypt/bin/letsencrypt renew
Processing /etc/letsencrypt/renewal/[도메인명].conf

The following certs are not due for renewal yet:
  /etc/letsencrypt/live/[도메인명]/fullchain.pem (skipped)
No renewals were attempted.
```

갱신에 성공하면 다음과 유사한 내용의 메시지를 볼 수 있다.

그림 6-1 인증서 갱신 성공 후 메시지

```
IMPORTANT NOTES:
- Congratulations! Your certificate and chain have been saved at
  /etc/letsencrypt/live/[도메인명]/fullchain.pem. Your cert
  will expire on (yyy-mm-dd). To obtain a new version of the
  certificate in the future, simply run Let's Encrypt again.
nginx: the configuration file /usr/local/nginx/conf/nginx.conf syntax is ok
nginx: configuration file /usr/local/nginx/conf/nginx.conf test is successful
```

Nginx 웹 서버에서도 crontab을 이용하여 자동으로 갱신하는 것이 효율적이다. 이 동작은 root 계정으로 작업해야 하며 vi 편집기로 cron 작업을 수정한다.

```
$ sudo crontab -e
```

'매주 월요일 오전 05시 00분에 'letsencrypt-auto renew' 명령을 실행하고 오전 5시 5분에 Nginx를 reload하여 새로 갱신된 인증서를 사용하게 한 후, 해당 명령에 대한 결과는 '/var/log/le-renew.log'에 저장하는 내용을 추가하여 자동 갱신하게 해보자.'

Debian 계열의 리눅스에서는 다음과 같이 입력한다.

```
0 5 * * 1 /etc/letsencrypt/letsencrypt-auto renew >> /var/log/le-renew.log  
5 5 * * 1 /etc/init.d/nginx reload
```

Redhat 계열 리눅스의 입력 내용은 Debian 계열의 입력 내용과 거의 동일하고, 다만 서비스 reload 명령만 다르다.

```
0 5 * * 1 /etc/letsencrypt/letsencrypt-auto renew >> /var/log/le-renew.log  
5 5 * * 1 /usr/bin/systemctl reload
```

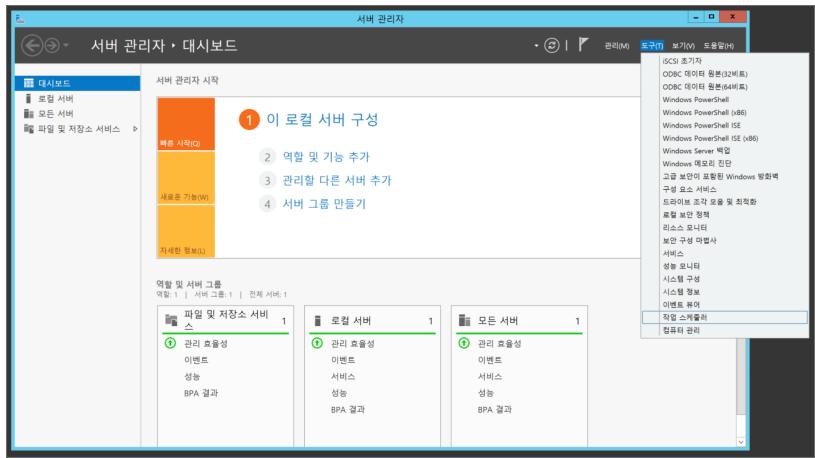
6.3 IIS 웹 서버

IIS는 앞서 소개한 유ти리티 중 갱신을 위한 명령어를 실행하여 인증서를 쉽게 갱신할 수 있다. 폴더의 데이터 중 letsencrypt.exe 파일을 실행할 때 다음 파라미터를 함께 입력한다. 이 명령어는 BaseUri를 거쳐서 인증서를 갱신하는 것으로, 이는 ACME 서버로 사용할 주소를 의미한다.

```
letsencrypt.exe --renew --baseuri https://acme-v01.api.letsencrypt.org/
```

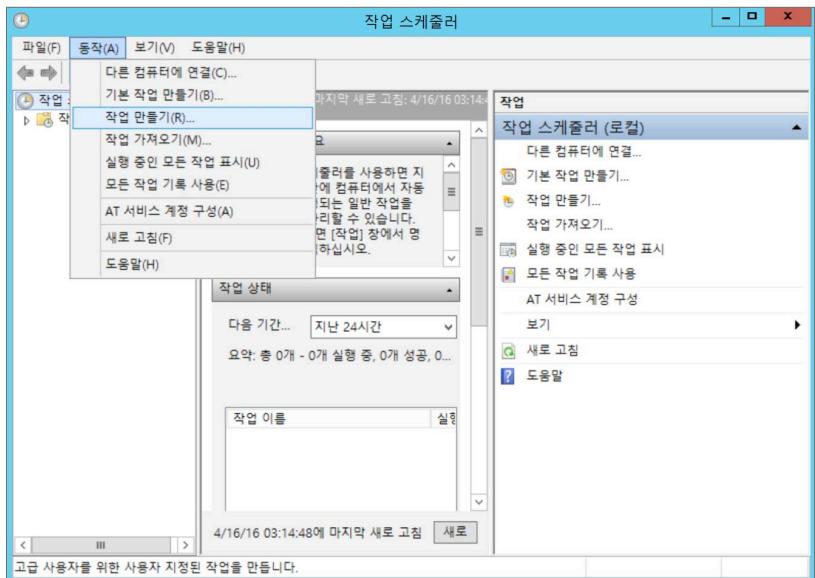
그리고 Windows Server 2012 R2에서는 작업 스케줄러를 제공한다. 이는 리눅스 서버의 cron에 해당하는 자동 반복 작업실행 도구의 Windows 버전으로, 이를 이용하여 인증서를 갱신하는 방법을 소개하겠다. 해당 작업 스케줄러는 [서버관리자 → 도구 → 작업 스케줄러]에 있다.

그림 6-2 Windows Server 2012 R2에서 제공하는 작업 스케줄러



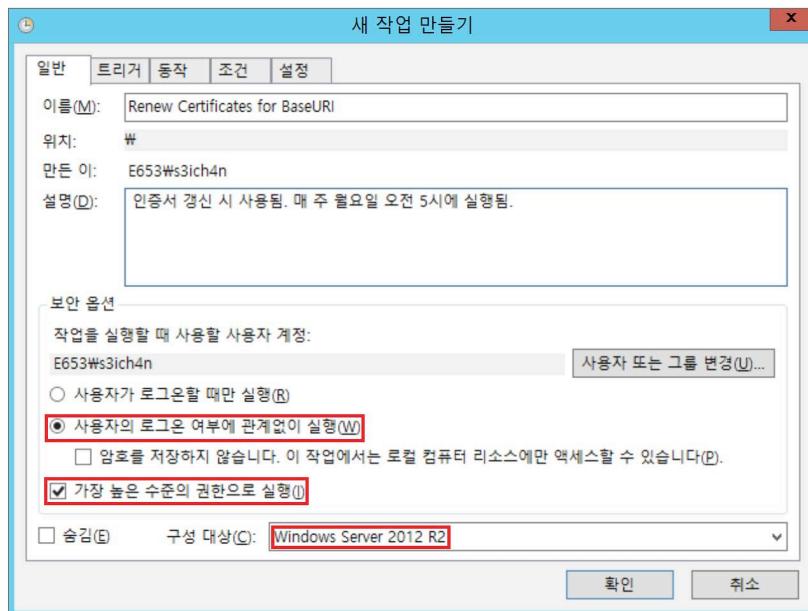
먼저 반복해서 수행할 작업을 만든다. 이 작업은 [작업 스케줄러 → 동작 → 작업 만들기]로 생성할 수 있다.

그림 6-3 반복할 동작 생성



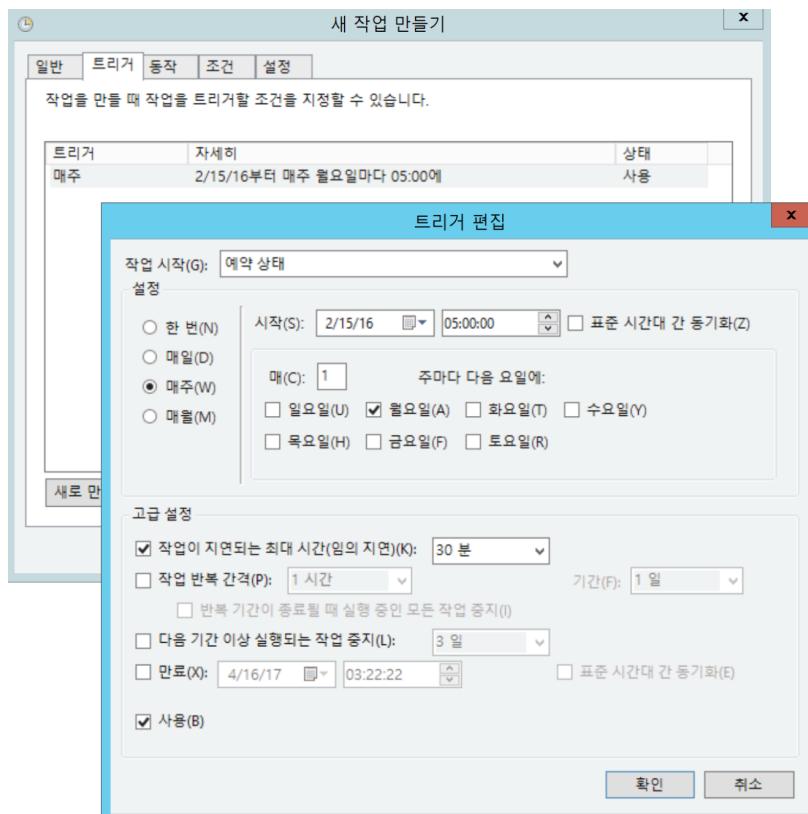
새 작업에 대한 [이름]과 [설명]을 적는다. 이때 ‘사용자의 로그온 여부와 관계없이 실행’과 ‘가장 높은 수준의 권한으로 실행’을 반드시 체크하고, [구성 대상]은 ‘Windows Server 2012 R2’로 선택한다.

그림 6-4 새 작업에 대한 이름, 설명, 권한 부여



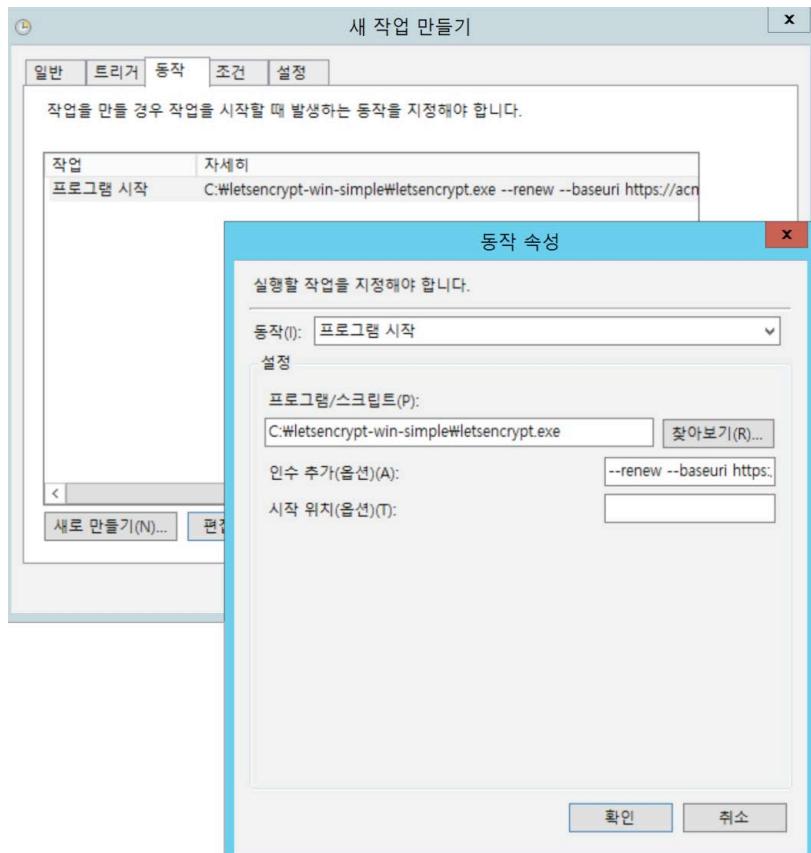
그다음 [트리거] 탭을 클릭하여 예약 실행 조건을 설정한다. 예를 들어, ‘매주 월요일 오전 05시 00분’에 일정을 진행한다면 [그림 6-5]와 같이 설정한다. [작업이 지연되는 최대 시간] 등은 임의로 설정하여도 무방하다.

그림 6-5 예약 작업 실행 조건 설정



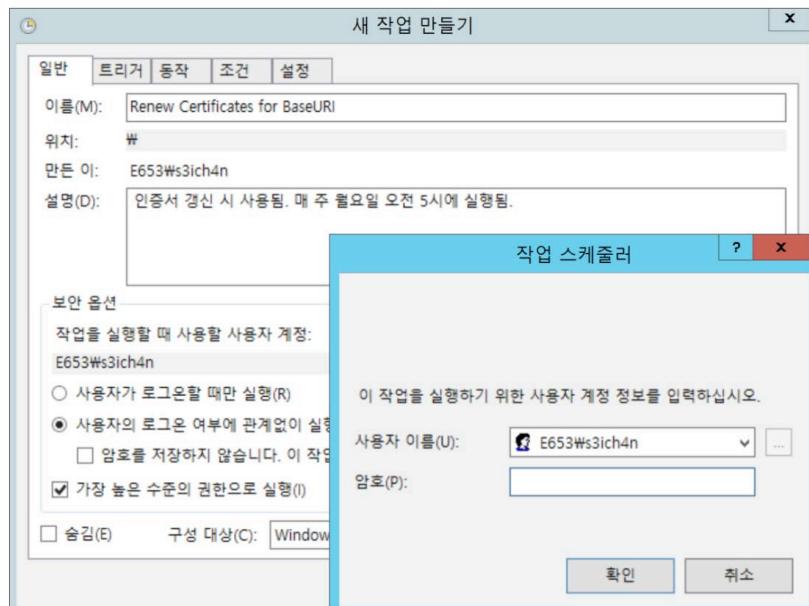
이어서 실행할 프로그램 설정과 파라미터 입력을 진행한다. Let's Encrypt가 설치된 디렉토리와 파라미터 입력을 확실하게 하였는지 확인한다. 나머지 조건과 설정은 자세히 읽어보고 웹 서버의 환경에 맞게 정하면 된다.

그림 6-6 인증서 갱신에 필요한 프로그램과 파라미터 설정



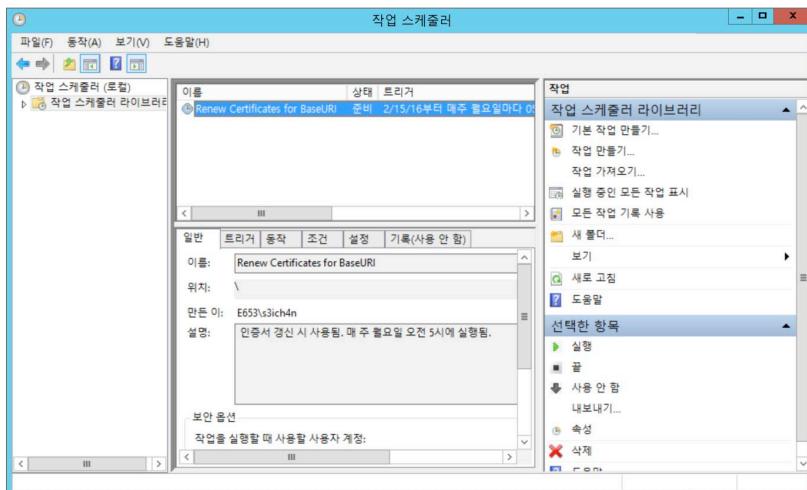
끝으로 새 작업을 설정하기 전 사용자 계정정보를 묻는데, 사용자의 이름에 맞는 비밀번호를 입력하면 설정이 완료된다.

그림 6-7 사용자 이름과 비밀번호 확인



설정이 완료되면 [그림 6-8]과 같은 화면이 보인다. 이 그림을 보면 매주 월요일 오전 05시 00분에 예약된 작업을 수행하도록 설정한 것을 알 수 있다.

그림 6-8 간단 작업 설정 완료



마무리

이 안내서에 제시되지 않은 내용은 Let's Encrypt에서 운영하는 공식 포럼 (<https://community.letsencrypt.org/>)에 있는 많은 질의와 글을 참고하길 바란다. 해당 커뮤니티에 올라온 글을 읽고 운영하는 서버에 맞게 작업하면 상당히 도움이 될 것이다.

이 안내서를 꼼꼼히 읽었다면 웹 서버에 HTTPS 프로토콜을 적용하는 것이 어렵게만 느껴지지 않으며 Let's Encrypt를 사용할 수 있게 되었을 것이다. 전 세계에 이미 많은 사람이 앞다투어 사용할 정도로 Let's Encrypt의 규모가 커졌다. 이제는 플랫폼이나 OS에 구애받지 않으며, 기존의 웹 서버와는 달리 누구나 쉽게 보안 연결된 웹 서버를 운영할 수 있게 되었다. 아무쪼록 이 책을 통하여 많은 사람이 자신의 웹 서버에 HTTPS 프로토콜을 적용하게 되어 보안에 취약하지 않은 웹 서버가 늘어나는 계기가 되었으면 하는 바람이다.

한빛 리얼타임

한빛 리얼타임은 IT 개발자를 위한 전자책입니다.

요즘 IT 업계에는 하루가 멀다 하고 수많은 기술이 나타나고 사라져 갑니다. 인터넷을 아무리 뒤져도 조금이나마 정리된 정보를 찾기도 쉽지 않습니다. 또한, 잘 정리되어 책으로 나오기까지는 오랜 시간이 걸립니다. 어떻게 하면 조금이라도 더 유용한 정보를 빠르게 얻을 수 있을까요? 어떻게 하면 남보다 조금 더 빨리 경험하고 습득한 지식을 공유하고 발전시켜 나갈 수 있을까요? 세상에는 수많은 종이책이 있습니다. 그리고 그 종이책을 그대로 옮긴 전자책도 많습니다. 전자책에는 전자책에 적합한 콘텐츠와 전자책의 특성을 살린 형식이 있다고 생각합니다.

한빛이 지금 생각하고 추구하는, 개발자를 위한 리얼타임 전자책은 이렇습니다.

1 eBook First –

빠르게 변화하는 IT 기술에 대해 핵심적인 정보를 신속하게 제공합니다

500페이지 가까운 분량의 잘 정리된 도서(종이책)가 아니라, 핵심적인 내용을 빠르게 전달하기 위해 조금은 거칠지만 100페이지 내외의 전자책 전용으로 개발한 서비스입니다. 독자에게는 새로운 정보를 빨리 얻을 기회가 되고, 자신이 먼저 경험한 지식과 정보를 책으로 펴내고 싶지만 너무 바빠서 엄두를 못 내는 선배, 전문가, 고수 분에게는 좀 더 쉽게 집필할 수 있는 기회가 될 수 있으리라 생각합니다. 또한, 새로운 정보와 지식을 빠르게 전달하기 위해 O'Reilly의 전자책 번역 서비스도 하고 있습니다.

2 무료로 업데이트되는 전자책 전용 서비스입니다

종이책으로는 기술의 변화 속도를 따라잡기가 쉽지 않습니다. 책이 일정 분량 이상으로 집필되고 정리되어 나오는 동안 기술은 이미 변해 있습니다. 전자책으로 출간된 이후에도 버전 업을 통해 중요한 기술적 변화가 있거나 저자(역자)와 독자가 소통하면서 보완하여 발전된 노하우가 정리되면 구매하신 분께 무료로 업데이트해 드립니다.

3 독자의 편의를 위해 DRM-Free로 제공합니다

구매한 전자책을 다양한 IT 기기에서 자유롭게 활용할 수 있도록 DRM-Free PDF 포맷으로 제공합니다. 이는 독자 여러분과 한빛이 생각하고 추구하는 전자책을 만들어 나가기 위해 독자 여러분이 언제 어디서 어떤 기기를 사용하더라도 편리하게 전자책을 볼 수 있도록 하기 위함입니다.

4 전자책 환경을 고려한 최적의 형태와 디자인에 담고자 노력했습니다

종이책을 그대로 옮겨 놓아 가독성이 떨어지고 읽기 어려운 전자책이 아니라, 전자책의 환경에 가능한 한 최적화하여 쾌적한 경험을 드리고자 합니다. 링크 등의 기능을 적극적으로 이용할 수 있음은 물론이고 글자 크기나 행간, 여백 등을 전자책에 가장 최적화된 형태로 새롭게 디자인하였습니다.

앞으로도 독자 여러분의 총고에 귀 기울이며 지속해서 발전시켜 나가겠습니다.

지금 보시는 전자책에 소유 권한을 표시한 문구가 없거나 타인의 소유권한을 표시한 문구가 있다면 위법하게 사용하고 있을 가능성이 큽니다. 이 경우 저작권법에 따라 불이익을 받으실 수 있습니다.

다양한 기기에 사용할 수 있습니다. 또한, [한빛미디어 사이트](#)에서 구매하신 후에는 횟수와 관계없이 내려받으실 수 있습니다.

한빛미디어 전자책은 인쇄, 검색, 복사하여 불이기가 가능합니다.

전자책은 오픈자 교정이나 내용의 수정·보완이 이뤄지면 업데이트 관련 공지를 이메일로 알려 드리며, 구매하신 전자책의 수정본은 무료로 내려받으실 수 있습니다.

이런 특별한 권한은 [한빛미디어 사이트](#)에서 구매하신 독자에게만 제공되며, 다른 사람에게 양도나 이전은 허락되지 않습니다.