



SSL 설치 가이드

APACHE HTTPD

OS	Windows Server / Linux
Server S/W Ver.	Apache 2.x
SSL type	Single / Multi / Wildcard
SSL CA	Symantec / Thawte / COMODO / Geo Trust / Entrust / KISA

2018.05.04.

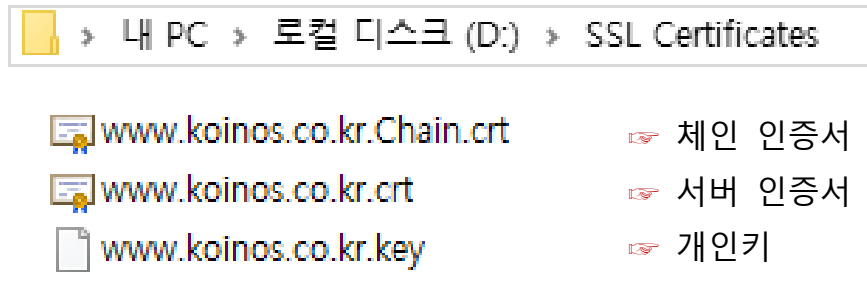
(주)코이노스



1. SSL 인증서 확인

① SSL 인증서 확인.

- 서버인증서, 체인인증서, 개인키를 확인합니다.
- 서버인증서와 체인인증서의 유효기간을 확인합니다.
- 개인키는 CSR을 저희가 직접 생성해드린 경우에 제공합니다.



② SSL 인증서 업로드.

- FTP 프로그램을 사용하여 서버인증서, 체인인증서, 개인키를 웹서버에 업로드 합니다.
- SSL 인증서가 위치한 디렉토리 경로를 확인합니다. 이 디렉토리 경로와 파일들을 잠시 후에 APACHE SSL 환경설정에서 지정해야 합니다.

로컬 사이트: D:\SSL Certificates\				리모트 사이트: /usr/local/apache/certificates			
<div>OneDriveTemp Outlook Pictures Program Files Recovery Share SSL Certificates System Volume Information Utilities Videos VM Workstaion</div>				<div>apache bin build certificates cgi-bin conf error htdocs icons include lib</div>			
파일명	크기	파일 유형		파일명	크기	파일 유형	최종 수정
..				..			
www.koinos.co.kr.Chain.crt	4,111	보안 인증서		www.koinos.co.kr.Chain.crt	4,111	보안 인증서	
www.koinos.co.kr.crt	2,277	보안 인증서		www.koinos.co.kr.crt	2,277	보안 인증서	
www.koinos.co.kr.key	1,675	KEY 파일		www.koinos.co.kr.key	1,675	KEY 파일	
3 파일. 총 크기: 8,063 바이트				3 파일. 총 크기: 8,063 바이트			
서버/로컬 파일	방향	리모트 파일		크기	우선	상태	

2. 암호화 모듈과 라이브러리 확인

① OPENSSL 라이브러리 확인.

- 명령어를 실행하여 openssl 라이브러리 설치 여부를 확인합니다.
#openssl version 또는
#find / -name openssl 또는
#rpm -qa openssl
- openssl 은 다양한 경로를 통해 설치되므로, 웹서버의 기능을 하는 컴퓨터에는 대부분 설치 되어있습니다.
- 리눅스의 경우 사용자의 권한(Permission)부족으로 실행되지 않을 수도 있습니다. 실행이 되지 않을 경우 root 계정으로 확인해 보시기 바랍니다.

```
localhost[root /usr/local/apache]#openssl version ; rpm -qa openssl ; find / -name openssl
OpenSSL 0.9.8e-fips-rhel5 01 Jul 2008
openssl-0.9.8e-12.el5_4.6
/usr/include/openssl
/usr/bin/openssl
/usr/local/ssl/include/openssl
/usr/local/ssl/bin/openssl
```

- 2012 년 이후 openssl 의 보안 취약점이 발견되어 낮은 버전을 사용하시면, 여러 가지의 보안 위협에 노출 될 수 있습니다. openssl 홈페이지에서 최근 버전으로 업데이트 하시는것을 권장 드립니다.
- openssl 홈페이지 : <https://www.openssl.org/>

② MOD_SSL 암호화 모듈 설치 확인.

- Apache 설치경로/bin/ 에서 mod_ssl 모듈이 설치 되었는지 확인합니다.
- %apache_HOME%/bin/httpd -l 또는 apachectl -l (소문자 l)

i) 정적 모듈 (Static) 방식

부팅시 설치된 모듈을 모두 호출하는 방식이며, 위의 명령어를 실행하였을 때, **mod_ssl.c** 파일이 있는지 확인합니다.

```
[root@ns bin]# /usr/local/apache/bin/httpd -l
Compiled-in modules:
  mod_ssl.c
[root@ns bin]#
```

위 그림과 같이 **mod_ssl.c** 항목이 나올경우 **MOD_SSL** 암호화 모듈이 정상적으로 설치된 상태입니다.

ii) 동적 모듈 (Dynamic Shared Object) 방식

동적 모듈 방식에서는 **mod_ssl.c** 항목 대신, **mod_so.c** 항목이 나옵니다.

```
localhost[root /usr/local/apache/bin]#./httpd -l
Compiled in modules:
  core.c
  prefork.c
  http_core.c
  mod_so.c
```

이것은 모듈을 필요에 따라 호출하여 사용하는 방식으로,
Apache 설치경로/modules/ 디렉토리에서 **mod_ssl.so** 항목을 확인하시면 됩니다.

```
localhost[root /usr/local/apache/modules]#ls mod_ssl.so
mod ssl.so
localhost[root /usr/local/apache/modules]#
```

동적 모듈 방식에서는 **httpd.conf** 설정에서 모듈을 호출하여 SSL 암호화 통신을 할 수 있게 합니다.

- 정적 모듈 방식에서의 **mod_ssl.c** 또는 동적 모듈 방식에서의 **mod_ssl.so** 항목을 찾으신 경우 **MOD_SSL** 암호화 모듈이 설치 되어있는 상태 입니다.
- 만약 위의 항목을 찾지 못하신 경우, SSL 적용을 위해 따로 **MOD_SSL** 모듈을 설치하시거나 경우에 따라서는 Apache 를 다시 설치 하셔야 합니다.
- **MOD_SSL** 설치 방법에 대한 것은 기술지원팀에 문의 주시기 바랍니다.

3. SSL 인증서 설치

- 아래 설정은 Apache 에서 HTTPS (HTTP over SSL) 암호화 구간을 사용하기 위해 필요한 기본 설정입니다.
- 이름 기반 가상호스트(NameVirtualHost)를 기준으로 설정된 예시입니다.
설정 항목을 분산(확장)하는 이유는 각 설정을 세부적으로 상세하게 적용할 수 있는 장점이 있습니다.
- 단일 호스트를 사용하는 경우, httpd-vhost.conf 항목의 내용을 httpd-ssl.conf 에 설정하시거나, httpd.conf 파일에 모든 설정을 적용 하셔도 됩니다.
- 소스 컴파일로 설치된 경우 Apache 의 설치 경로는 대략 /usr/local/ 에 위치합니다. 환경설정 파일은 /usr/local/apache/conf 경로에 있습니다.
- RPM 설치로 구성한 apache 는 /etc/httpd/conf/ 경로에서 확인하세요.

설정 파일	설정 항목	설정 예시
httpd.conf	SSL 모듈 호출	LoadModule ssl_module modules/mod_ssl.so
	설정파일 확장	Include conf/extra/httpd-ssl.conf
	설정파일 확장	Include conf/extra/httpd-vhost.conf
httpd-ssl.conf	SSL 통신 포트 지정	Listen 443
	SSL 가상 호스트 설정	NameVirtualHost *:443
	SSL 프로토콜 지정	SSLProtocol all -SSLv2 -SSLv3
	SSL 암호조합 지정	ALL:!ADH:!EXPORT56:RC4+RSA:!SSLv2:!SSLv3
httpd-vhosts.conf	<VirtualHost>	<VirtualHost *:443>
	ServerName	www.domain.tld
	ServerAlias	domain.tld *.domain.tld
	DocumentRoot	/www/domain
	SSL 엔진 스위치	SSLEngine on
	인증서 파일 지정	SSLCertificateFile "/경로/파일명.crt
	인증서 개인키 지정	SSLCertificateKeyFile "/경로/파일명.key
	인증서 체인 지정	SSLCertificateChainFile "/경로/파일명.Chain.crt
	</VirtualHost>	</VirtualHost>

① httpd.conf 환경설정

- Apache 의 httpd.conf 설정파일을 vi 에디터 등으로 수정합니다.

i) SSL 모듈 호출

· mod_ssl 모듈을 로딩하도록 설정합니다.

```
# Dynamic Shared Object (DSO) Support
LoadModule ssl module modules/mod_ssl.so
```

ii) 확장(Supplemental) 설정파일 추가

· httpd-ssl.conf (ssl.conf) 와 httpd-vhosts.conf (vhosts.conf) 설정을 활성화 합니다.

```
# Secure (SSL/TLS) connections
Include conf/extra/httpd-ssl.conf

# Virtual hosts
Include conf/extra/httpd-vhosts.conf
```

· RPM 버전의 경우 conf.d 의 모든 설정을 활성화 합니다.

```
# Load config files from the config directory "/etc/httpd/conf.d".
Include conf.d/*.conf
```

② httpd-ssl.conf 환경설정

/etc/httpd/conf/extra/httpd-ssl.conf 설정파일을 vi 에디터 등으로 수정합니다.

i) HTTPS port 설정

· 시스템에서 사용중인 port와 충돌이 없도록 비어 있는 port 를 할당합니다.

· 사용하는 호스트별로 443, 444, 446 ... 등으로 포트를 구분해야 합니다.

(445 는 시스템에서 사용합니다)

Wildcard , Multi Domain SSL 인증서의 경우는 포트를 공유할 수 있습니다.

Listen 443

```
#Listen 12.34.56.78:80
Listen 80
Listen 443
```

ii) 이름기반 가상 호스트 설정

· 이 설정은 Wildcard SSL, Multi Domain SSL 을 사용하여, 포트를 공유하는 경우 사용합니다. 위에서 지정한 포트를 지정해 줍니다. 단일 호스트를 사용하는 경우 이 부분은 생략합니다.

NameVirtualHost *:443

```
NameVirtualHost *:80
NameVirtualHost *:443
```

iii) SSL 프로토콜 및 암호조합(CipherSuite) 설정

- HTTPS 암호화 통신을 위하여 서버와 클라이언트는 암호화 통신을 위하여 사용할 알고리즘을 협상합니다.
SSL 프로토콜은 키교환 방식, 암호문 생성, 블록암호 운영방식 등 알고리즘을 조합하는 것을 말합니다.
- 이러한 조합 방법을 규정한 프로토콜을 각 버전으로 구분합니다.
SSLv2, SSLv3, TLSv1.0, TLSv1.1, TLSv1.2, TLSv1.3
- SSLv2, SSLv3 는 보안에 취약한 부분이 발견 되어 현재 사용하지 않습니다.
- Cipher Suite(암호조합) 설정을 통하여 보안 취약점을 제거한 강력한 암호 조합을 설정 할 수 있습니다.
※ Windows XP SP2 이하 OS에서는 TLS 1.1 이상 알고리즘조합이 호환 되지 않습니다.
- 아래 설정을 그대로 복사하셔서 사용 하세요.

SSL Protocol:

SSLProtocol ALL -SSLv2 -SSLv3

SSL Cipher Suite:

SSLCipherSuite ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-GCM-SHA384:DHE-RSA-AES128-GCM-SHA256:DHE-DSS-AES128-GCM-SHA256:kEDH+AESGCM:ECDHE-RSA-AES128-SHA256:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA:ECDHE-ECDSA-AES256-SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES128-SHA:DHE-DSS-AES128-SHA256:DHE-RSA-AES256-SHA256:DHE-DSS-AES256-SHA:DHE-RSA-AES256-SHA:AES128-GCM-SHA256:AES256-GCM-SHA384:AES128-SHA256:AES256-SHA256:AES128-SHA:AES256-SHA:AES:CAMELLIA:DES-CBC3-SHA:!aNULL:!eNULL:!EXPORT:!DES:!RC4:!MD5:!PSK:!aECDH:!EDH-DSS-DES-CBC3-SHA:!EDH-RSA-DES-CBC3-SHA:!KRB5-DES-CBC3-SHA

- SSLCipherSuite 설정에서 구분은 ":" 으로 입력하며, 사용하지 않는 알고리즘은 "!" 를 붙입니다.
- 아래와 같이 기본설정에서 SSLv2, SSLv3 을 제거 하셔도 됩니다.

#SSLCipherSuite

ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:+eNULL

SSLCipherSuite

ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:!SSLv2:!SSLv3:+EXP:+eNULL

```
# SSL Protocol:
SSLProtocol ALL -SSLv2 -SSLv3

# SSL Cipher Suite (recommended):
SSLCipherSuite ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES256
ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256
CM-SHA384:AES128-SHA256:AES256-SHA256:AES128-SHA:AES256-SHA:AES:CAMELLIA:DES-CBC3-SHA:!a

# SSL Cipher Suite (default):
#SSLCipherSuite ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:+eNULL
SSLCipherSuite ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:!SSLv2:!SSLv3:+EXP:+eNULL
```

③ httpd-vhosts.conf 환경설정

- 기존 설정되어있는 <VirtualHost :80> 설정과 별개로 <VirtualHost :443> 을 추가로 구성합니다. 이것은 독립적인 구성이며 기본 페이지를 :80 과 다르게 별도 구성할 수도 있습니다.
- :80 설정에 SSL Engine on, SSL 인증서, 개인키, 체인을 추가로 설정합니다.
- 호스트를 추가할 경우 아래 <VirtualHost> </VirtualHost> 구간을 추가합니다.

<VirtualHost *:443>

ServerAdmin 관리자 메일주소

DocumentRoot 웹문서의 경로

ServerName koinos.co.kr 서버이름(도메인이름)

ServerAlias 함께 사용하는 도메인이름

SSL Engine on

SSLCertificateFile "서버 인증서 저장 경로/서버인증서.crt"

SSLCertificateKeyFile "개인키 저장 경로/서버인증서.key"

SSLCertificateChainFile "인증서 체인 저장 경로/인증서체인.Chain.crt"

ErrorLog "logs/error.log" 로그 파일 저장 위치

CustomLog "logs/access.log" 로그 파일 저장 위치

</VirtualHost>

```
<VirtualHost *:443>
ServerAdmin admin@koinos.co.kr
DocumentRoot /usr/local/apache/docs/
ServerName koinos.co.kr
ServerAlias www.koinos.co.kr *.koinos.co.kr
SSL Engine on
SSLCertificateFile "/usr/local/apache/certificates/www.koinos.co.kr.crt"
SSLCertificateKeyFile "/usr/local/apache/certificates/www.koinos.co.kr.key"
SSLCertificateChainFile "/usr/local/apache/certificates/www.koinos.co.kr.Chain.crt"
ErrorLog "logs/error.log"
CustomLog "logs/access.log"
</VirtualHost>
```


4. SSL 적용 확인

① 환경설정 오류 체크

- httpd.conf , httpd-ssl.conf 설정의 오류를 확인합니다.
- /%home%/bin 경로에서 ./apachectl configtest 또는 ./apachectl -t 를 입력

apachectl -t

```
localhost[root /usr/local/apache/bin]#./apachectl -t
Syntax OK
```

② SSL 설정 적용

- apachectl 또는 httpd 명령어를 통해 위에서 설정한 SSL 을 적용합니다.
- graceful 옵션은 Apache 를 재시작 하지 않고, 설정을 적용하는 옵션입니다.

linux) apachectl graceful 또는 httpd graceful

- graceful 옵션이 적용되지 않는 경우는 apache 를 재시작 합니다.

linux) apachectl restart 또는 httpd restart

windows) httpd -k restart

- APACHE 2.0 버전의 경우 startssl 옵션을 사용해야 합니다.

linux) apachectl startssl

```
localhost[root /usr/local/apache/bin]#./apachectl graceful
localhost[root /usr/local/apache/bin]#
localhost[root /usr/local/apache/bin]#./apachectl restart
localhost[root /usr/local/apache/bin]#
localhost[root /usr/local/apache/bin]#./apachectl startssl
```

③ http 프로세스 확인

- APACHE daemon 이 정상적으로 실행되었는지 확인합니다.

linux) ps -ef | grep httpd

```
localhost[root /usr/local/apache/bin]#ps -ef | grep httpd
root      21297      1   0 17:09 ?        00:00:00 /usr/local/apache/bin/httpd
daemon    21373  21297   0 17:16 ?        00:00:00 /usr/local/apache/bin/httpd
daemon    21374  21297   0 17:16 ?        00:00:00 /usr/local/apache/bin/httpd
daemon    21375  21297   0 17:16 ?        00:00:00 /usr/local/apache/bin/httpd
daemon    21376  21297   0 17:16 ?        00:00:00 /usr/local/apache/bin/httpd
daemon    21377  21297   0 17:16 ?        00:00:00 /usr/local/apache/bin/httpd
root      21391  20869   0 17:23 pts/2    00:00:00 grep httpd
```

④ SSL 포트 확인

- 위 단계에서 설정한 SSL 포트가 설정되었는지 확인합니다.

linux) netstat -nap | grep LISTEN

windows) netstat -nap | findstr LISTEN

```
localhost[root /usr/local/apache/bin]#netstat -nap | grep LISTEN
tcp        0      0 127.0.0.1:2208        0.0.0.0:*               LISTEN      2884/hpid
tcp        0      0 0.0.0.0:718           0.0.0.0:*               LISTEN      2656/rpc.statd
tcp        0      0 0.0.0.0:111           0.0.0.0:*               LISTEN      2621/portmap
tcp        0      0 0.0.0.0:21            0.0.0.0:*               LISTEN      3582/vsftpd
tcp        0      0 127.0.0.1:25          0.0.0.0:*               LISTEN      2955/sendmail: acce
tcp        0      0 127.0.0.1:2207        0.0.0.0:*               LISTEN      2889/python
tcp        0      0 :::80                 :::*                    LISTEN      21297/httpd
tcp        0      0 :::22                 :::*                    LISTEN      2902/sshd
tcp        0      0 :::443                :::*                    LISTEN      21297/httpd
```

⑤ SSL 암호화 통신 확인

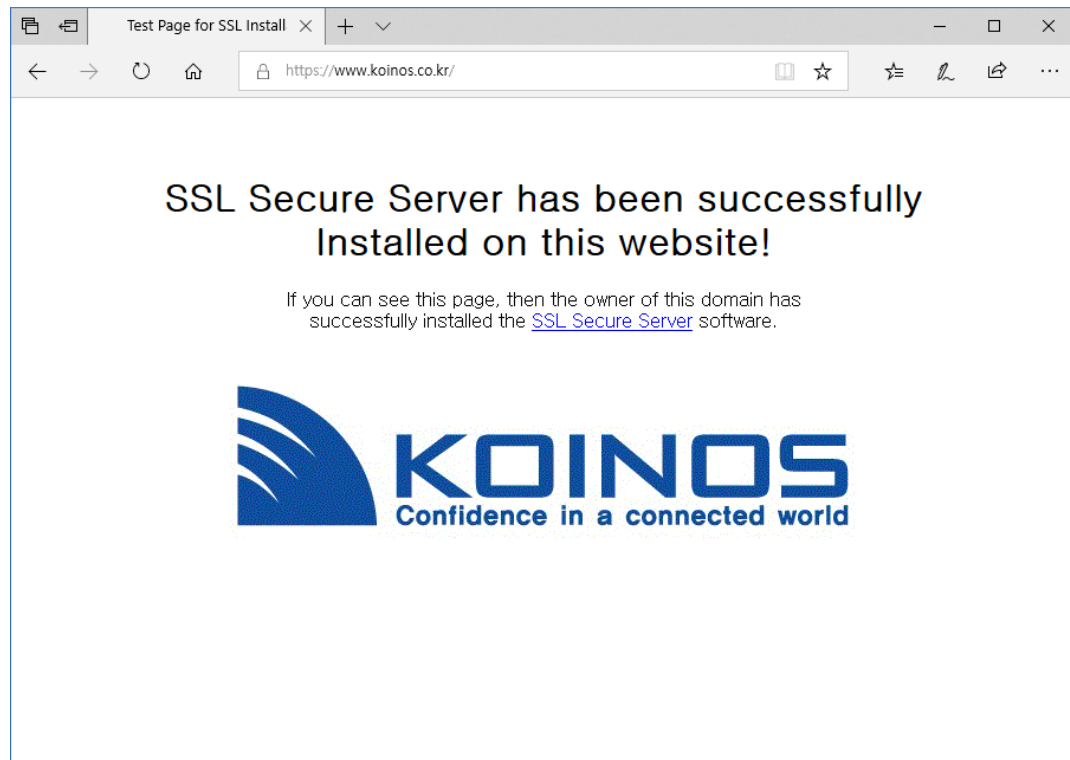
https 세션 확인 : openssl s_client -connect 127.0.0.1:443 -state -tls1

유효기간 확인 : openssl s_client -connect 127.0.0.1:443 | openssl x509 -dates

```
localhost[root /usr/local/apache/bin]#openssl s_client -connect 127.0.0.1:443 -state -tls1
CONNECTED(00000003)
SSL_connect:before/connect initialization
SSL_connect:SSLv3 write client hello A
SSL_connect:SSLv3 read server hello A
depth=3 /C=SE/O=AddTrust AB/OU=AddTrust External TTP Network/CN=AddTrust External CA Root
verify return:1
depth=2 /C=GB/ST=Greater Manchester/L=Salford/O=COMODO CA Limited/CN=COMODO RSA Certification Authority
verify return:1
depth=1 /C=GB/ST=Greater Manchester/L=Salford/O=COMODO CA Limited/CN=COMODO RSA Domain Validation Secure Server CA
verify return:1
depth=0 /OU=Domain Control Validated/OU=PositiveSSL Multi-Domain/CN=www.koinos.co.kr
verify return:1
SSL_connect:SSLv3 read server certificate A
SSL_connect:SSLv3 read server key exchange A
SSL_connect:SSLv3 read server done A
SSL_connect:SSLv3 write client key exchange A
SSL_connect:SSLv3 write change cipher spec A
SSL_connect:SSLv3 write finished A
SSL_connect:SSLv3 flush data
SSL_connect:SSLv3 read finished A
---
Certificate chain
 0 s:/OU=Domain Control Validated/OU=PositiveSSL Multi-Domain/CN=www.koinos.co.kr
   i:/C=GB/ST=Greater Manchester/L=Salford/O=COMODO CA Limited/CN=COMODO RSA Domain Validation Secure Server CA
 1 s:/C=GB/ST=Greater Manchester/L=Salford/O=COMODO CA Limited/CN=COMODO RSA Domain Validation Secure Server CA
   i:/C=GB/ST=Greater Manchester/L=Salford/O=COMODO CA Limited/CN=COMODO RSA Certification Authority
 2 s:/C=GB/ST=Greater Manchester/L=Salford/O=COMODO CA Limited/CN=COMODO RSA Certification Authority
   i:/C=SE/O=AddTrust AB/OU=AddTrust External TTP Network/CN=AddTrust External CA Root
SSL handshake has read 5194 bytes and written 287 bytes
---
New, TLSv1/SSLv3, Cipher is DHE-RSA-AES256-SHA
Server public key is 2048 bit
Secure Renegotiation IS supported
Compression: NONE
Expansion: NONE
SSL-Session:
    Protocol  : TLSv1
    Cipher    : DHE-RSA-AES256-SHA
    Session-ID: 5FA564E9C62AB0943DFB6662E7AB76569BF0DCD60495E690734A33C10BC6006F
    Session-ID-ctx:
    Master-Key: 3129AEBF59123CDB8C488AA39C2CB85D2E3660DF9BB9F6ABD485630F11D6DDF9673F3FEAA44F0C7A71E78BDE369A30F1
    Key-Arg   : None
    Krb5 Principal: None
    Start Time: 1525338229
    Timeout  : 7200 (sec)
    Verify return code: 0 (ok)
```

⑥ 인터넷 브라우저 확인

- 웹브라우저 주소창에 https 로 시작하는 주소와 위에서 설정한 SSL 포트를 입력합니다
- 예) <https://www.koinos.co.kr:443>



- 위 https 로 연결시 브라우저에서 웹페이지가 나타나는 경우 SSL 설정이 정상적으로 적용된 것입니다.

5. HTTPS 클라이언트 연결 설정

웹서버에 SSL 을 설치하셨지만, 일반 클라이언트는 https 와 같은 프로토콜을 입력하지 않고 웹 페이지에 접속합니다.

그래서 서버 리디렉트(redirect) 또는 웹페이지의 소스를 수정하여 https 암호화 구간으로 고객을 이동시킬 필요가 있습니다.

(1) Apache Redirect 설정

- 다음은 서버에서 지시자를 사용하여 암호화되지 않은 경로(<http://domain.co.kr>)로 들어온 사용자를 강제로 리디렉션 시켜서 암호화 통신하는 예입니다.
- http 서버의 httpd.conf 또는 virtualhost.conf 파일을 아래와 같이 설정하시기 바랍니다.

```
<VirtualHost koinos.co.kr:80>
    ServerAdmin admin@koinos.co.kr
    ServerName koinow.co.kr
    DocumentRoot /usr/local/apache/docs
    CustomLog logs/innocert.co.kr-access_log common
    Redirect https://www.koinos.co.kr
</VirtualHost>
```

(2) 개인정보구간 부분 암호화 설정

- ID/PW , 주민등록번호등 개인정보가 전송되는 구간을 부분 암호화.
- Htps 부분 암호화 설정은 "전체암호화 설정"에 비해 서버부하를 최소화 합니다.

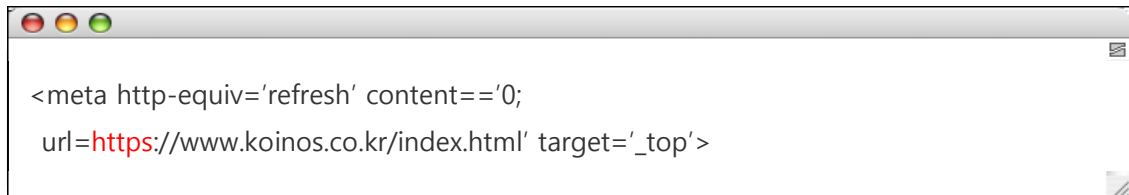
```
<form name='login_form' action='https://www.koinos.co.kr/login/ login.jsp'>
<td><input name="id" type="text" ></td>
<td><input name="pw" type="password" ></td>
</form>
```

(3) 웹 페이지 전체 암호화 설정

- 전체 페이지를 암호화 하는 방법은 아주 간단한 소스 수정을 통하여 적용할 수 있으나 암호화 적용이 필요 없는 부분까지 암호화하기 때문에 “부분 암호화” 보다 서버에 많은 부하를 줄 수 있습니다

i. HTML Tag 이용

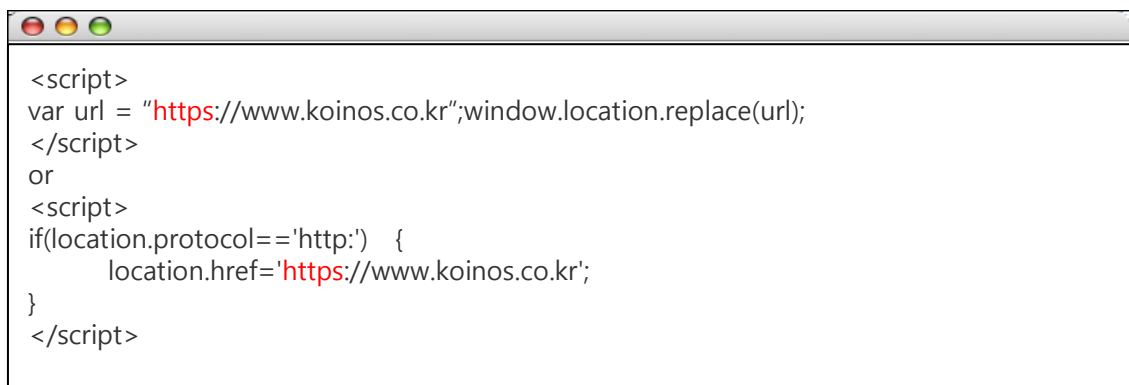
- 웹 페이지(index.html)에 아래와 같이 소스코드를 추가합니다.



```
<meta http-equiv='refresh' content=='0;
url=https://www.koinos.co.kr/index.html' target='_top'>
```

ii. Java Script 이용

- 웹 페이지(index.html)에 아래와 같이 소스코드를 추가합니다.



```
<script>
var url = "https://www.koinos.co.kr";window.location.replace(url);
</script>
or
<script>
if(location.protocol=='http:') {
    location.href='https://www.koinos.co.kr';
}
</script>
```