Apache + SSL (for Windows) 환경 구축

1. Apache + SSL 배포판 구하기
 공식 Apache 홈페이지에는 Windows용의 경우 noSSL 버전만 배포하고 있습니다. 그러나 다음 위치에서 받을 수 있습니다.
  - http://mirror.apache-kr.org/httpd/binaries/win32/ 의 apache_x.x.x-win32-x86-openssl-0.x.xx.msi 형태의 설치

2. Apache + SSL 설치하기
  apache_2.0.63-win32-x86-openssl-0.9.7m.msi 더블클릭하여 gui 방식으로 설치. 설치위치는 c:₩Apache₩Apache2로 함.

 o c:₩Apache₩Apache2₩bin 디렉토리로 이동("c:₩Apache₩Apache2₩bin")하여
   설치디렉토리 밑의 bin 디렉토리에 있는 ssleay32.dll, libeay32.dll 파일을 시스템디렉토리,
   (Windows 2000의 경우 C:₩WINNT, Windows XP, 2003등은 C:₩Windows)아래 system32
   디렉토리로 복사한다.

   PROMPT> copy ssleay32.dll C₩WINDOWS₩system32₩
   PROMPT> copy libeay32.dll C₩WINDOWS₩system32₩

  * [내 컴퓨터] - [속성] - [고급 - 환경 변수]에서 시스템 변수중 Path에 C:₩WINDOWS₩system32;C:₩Apache₩Apache2₩bin 추가

3. 웹브라우져 접속(http://cjo.klid.or.kr)을 통한 정상기동 여부 확인
  - 10.60.183.14   cjo.klid.or.kr

4. C:₩Apache2₩conf₩openssl.cnf 수정

  [req_distinguished_name]
countryName_default   = KR ==> KR 입력

#stateOrProvinceName   = State or Province Name (full name) ==> 주석 처리
#stateOrProvinceName_default   = Some-State   ==> 주석 처리

#localityName   = Locality Name (eg, city) ==> 주석 처리

0.organizationName_default   = Government of Korea ==> Government of Korea 입력

organizationalUnitName_default = Group of Server ==> 주석 제거 및 Group of Server 입력

#emailAddress   = Email Address ==> 주석 처리
#emailAddress_max   = 64   ==> 주석 처리

5. key 및 csr 파일 생성 경로(C:₩Apache₩Apache2₩conf₩ 하위디렉토리) 생성

C:₩Apache₩Apache2₩conf₩ssl

6. key쌍 생성

[명령] openssl genrsa -out <key filename> 1024 ===> OS가 Windows일 경우
       openssl genrsa -des3 -out <key filename 1> 1024 ===> OS가 Unix일 경우

==>

c:₩Apache₩Apache2₩bin₩openssl genrsa -out c:₩Apache₩Apache2₩conf₩ssl₩cjo.klid.key 1024

c:₩Apache₩Apache2₩bin₩openssl genrsa -des3 -out c:₩Apache₩Apache2₩conf₩ssl₩cjo.klid1.key 1024

```
Loading 'screen' into random state - done
Generating RSA private key, 1024 bit long modulus
..++++++
.......++++++
e is 65537 (0x10001)
Enter pass phrase for c:₩Apache₩Apache2₩conf₩ssl₩cjo.klid1.key:
Verifying - Enter pass phrase for c:₩Apache₩Apache2₩conf₩ssl₩cjo.klid1.key:
C:₩>
```

 [참고]
 Error: Init: SSLPassPhraseDialog builtin is not supported on Win32 에러

  : key 생성시 옵션에서 -des3 를 주지 않으면 위 에러가 해결 될 수 있습니다.
  ==> c:₩Apache₩Apache2₩bin₩openssl genrsa -out c:₩Apache₩Apache2₩conf₩ssl₩cjo.klid.key 1024 < no 패스워드 >

      c:₩Apache₩Apache2₩bin₩openssl genrsa -des3 -out c:₩Apache₩Apache2₩conf₩ssl₩cjo.klid1.key 1024 < 패스워드를 물어봄 >

7. key쌍 통한 csr 생성

[명령] openssl req -new -config <openssl cnf file> -key <key filename> -out <csr filename>
[예] c:₩Apache₩Apache2₩bin₩openssl req -config c:₩Apache₩Apache2₩conf₩openssl.cnf -new -key
c:₩Apache₩Apache2₩conf₩ssl₩cjo.klid.key -out c:₩Apache₩Apache2₩conf₩ssl₩cjo.klid.csr

==>

c:₩Apache₩Apache2₩bin₩openssl req -config c:₩Apache₩Apache2₩conf₩openssl.cnf -new -key
c:₩Apache₩Apache2₩conf₩ssl₩cjo.klid.key -out c:₩Apache₩Apache2₩conf₩ssl₩cjo.klid.csr

You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

Country Name (2 letter code) [KR]:KR
Organization Name (eg, company) [Government of Korea]:Government of Korea
Organizational Unit Name (eg, section) [Group of Server]:Group of Server
Common Name (eg, YOUR name) []:cjo.klid.or.kr

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:cho1203
An optional company name []:cho1203

C:₩Apache₩Apache2₩bin>

8. csr 파일을 통한 인증관리센터 홈페이지에서 인증서(*.p7b 화일) 발급

9. 발급받은 pkcs#7(*.p7b 화일) ==> pem 변환

[명령] openssl pkcs7 -in <p7b filename> -out <pem filename> -print_certs -text

c:₩Apache₩Apache2₩bin₩openssl pkcs7 -in c:₩Apache₩Apache2₩conf₩ssl₩cjo.klid.or.kr.p7b -out
c:₩Apache₩Apache2₩conf₩ssl₩cjo.klid.pem -print_certs -text

C:₩>

10. *.pem 파일을 통한 *.pem 파일 생성(ca.pem, rootcaChain.pem) ==> CA 및 체인

 1) "Certificate:"와 "-----END CERTIFICATE-----" 사이에 "Subject: C=KR, O=Government of Korea, OU=GPKI, CN=CA131000001" 가 있는 부분을 복사하여
 편집기에 붙여넣어 ca.pem 파일 생성

 2) "Subject: C=KR, O=Government of Korea, OU=GPKI, CN=Root CA" 와 "Subject: C=KR, O=Government of Korea, OU=GPKI, CN=CA131000001" 가 있는
 "Certificate:" 에서 "-----END CERTIFICATE-----" 까지 복사하여
 편집기에 붙여넣어 rootcaChain.pem 파일 생성

11. httpd.conf 파일 수정
설치경로 : c:₩Apache₩Apache2₩conf₩httpd.conf

LoadModule ssl_module modules/mod_ssl.so ==> 주석 제거

#
Bring in additional module-specific configurations ==> ssl.conf 로딩 구문 확인
#

<IfModule mod_ssl.c>
    Include conf/ssl.conf

```
</IfModule>
```

12. conf/ssl.conf 수정 내용

```
SSLRandomSeed startup builtin
SSLRandomSeed connect builtin
#SSLRandomSeed startup file:/dev/random  512
#SSLRandomSeed startup file:/dev/urandom 512
#SSLRandomSeed connect file:/dev/random  512
#SSLRandomSeed connect file:/dev/urandom 512

<IfDefine SSL>

Listen 443

SSLPassPhraseDialog  builtin

SSLSessionCache        none
#SSLSessionCache        shmht:logs/ssl_scache(512000)
#SSLSessionCache        shmcb:logs/ssl_scache(512000)
#SSLSessionCache         dbm:logs/ssl_scache
SSLSessionCacheTimeout  300

##
## SSL Virtual Host Context
##

<VirtualHost *:443>

#   General setup for the virtual host
DocumentRoot "C:/Apache/Apache2/htdocs"
ServerName cjo.klid.or.kr:443
ServerAdmin cjo@klid.or.kr
ErrorLog logs/sslerror_log
TransferLog logs/sslaccess_log

#   SSL Engine Switch:
#   Enable/Disable SSL for this virtual host.
SSLEngine on

SSLCertificateFile conf/ssl/cjo.klid.pem  ===> 웹에서 발급받은 SSL인증서
SSLCertificateKeyFile conf/ssl/cjo.klid.key
SSLCertificateChainFile conf/ssl/rootca.pem
SSLCACertificatePath conf/ssl
SSLCACertificateFile conf/ssl/ca.pem

</VirtualHost>

</IfDefine>
```

13. Apache Start

```
C:₩Apache₩Apache2₩bin>apache -D SSL -k start
```

14. Apache Stop

```
C:₩Apache₩Apache2₩bin>apache -D SSL -k stop
```

```
=================> conf/ssl.conf Sample <=================
#
# This is the Apache server configuration file providing SSL support.
# It contains the configuration directives to instruct the server how to
# serve pages over an https connection. For detailing information about these
# directives see <URL:http://httpd.apache.org/docs/2.0/mod/mod_ssl.html>
#
# Do NOT simply read the instructions in here without understanding
# what they do.  They're here only as hints or reminders.  If you are unsure
# consult the online docs. You have been warned.
#

#
# Pseudo Random Number Generator (PRNG):
# Configure one or more sources to seed the PRNG of the SSL library.
# The seed data should be of good random quality.
```

```
# WARNING! On some platforms /dev/random blocks if not enough entropy
# is available. This means you then cannot use the /dev/random device
# because it would lead to very long connection times (as long as
# it requires to make more entropy available). But usually those
# platforms additionally provide a /dev/urandom device which doesn't
# block. So, if available, use this one instead. Read the mod_ssl User
# Manual for more details.
#
# Note: This must come before the <IfDefine SSL> container to support
#       starting without SSL on platforms with no /dev/random equivalent
#       but a statically compiled-in mod_ssl.
#
SSLRandomSeed startup builtin
SSLRandomSeed connect builtin
#SSLRandomSeed startup file:/dev/random  512
#SSLRandomSeed startup file:/dev/urandom 512
#SSLRandomSeed connect file:/dev/random  512
#SSLRandomSeed connect file:/dev/urandom 512

<IfDefine SSL>

#
# When we also provide SSL we have to listen to the
# standard HTTP port (see above) and to the HTTPS port
#
# Note: Configurations that use IPv6 but not IPv4-mapped addresses need two
#       Listen directives: "Listen [::]:443" and "Listen 0.0.0.0:443"
#
Listen 443

##
##  SSL Global Context
##
##  All SSL configuration in this context applies both to
##  the main server and all SSL-enabled virtual hosts.
##

#
#   Some MIME-types for downloading Certificates and CRLs
#
AddType application/x-x509-ca-cert .crt
AddType application/x-pkcs7-crl    .crl

#   Pass Phrase Dialog:
#   Configure the pass phrase gathering process.
#   The filtering dialog program (`builtin' is a internal
#   terminal dialog) has to provide the pass phrase on stdout.
SSLPassPhraseDialog  builtin

#   Inter-Process Session Cache:
#   Configure the SSL Session Cache: First the mechanism
#   to use and second the expiring timeout (in seconds).
SSLSessionCache        none
#SSLSessionCache        shmht:logs/ssl_scache(512000)
#SSLSessionCache        shmcb:logs/ssl_scache(512000)
#SSLSessionCache         dbm:logs/ssl_scache
SSLSessionCacheTimeout  300

#   Semaphore:
#   Configure the path to the mutual exclusion semaphore the
#   SSL engine uses internally for inter-process synchronization.
SSLMutex default
#SSLMutex  none

##
## SSL Virtual Host Context
##

<VirtualHost *:443>

#   General setup for the virtual host
DocumentRoot "C:/Apache/Apache2/htdocs"
ServerName cjo.klid.or.kr:443
ServerAdmin cjo@klid.or.kr
ErrorLog logs/sslerror_log
TransferLog logs/sslaccess_log
```

```
#   SSL Engine Switch:
#   Enable/Disable SSL for this virtual host.
SSLEngine on

#   SSL Cipher Suite:
#   List the ciphers that the client is permitted to negotiate.
#   See the mod_ssl documentation for a complete list.
SSLCipherSuite ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:+eNULL

#   Server Certificate:
#   Point SSLCertificateFile at a PEM encoded certificate.  If
#   the certificate is encrypted, then you will be prompted for a
#   pass phrase.  Note that a kill -HUP will prompt again.  Keep
#   in mind that if you have both an RSA and a DSA certificate you
#   can configure both in parallel (to also allow the use of DSA
#   ciphers, etc.)
SSLCertificateFile conf/ssl/cjo.klid.pem
#SSLCertificateFile conf/ssl.crt/server-dsa.crt

#   Server Private Key:
#   If the key is not combined with the certificate, use this
#   directive to point at the key file.  Keep in mind that if
#   you've both a RSA and a DSA private key you can configure
#   both in parallel (to also allow the use of DSA ciphers, etc.)
SSLCertificateKeyFile conf/ssl/cjo.klid.key
#SSLCertificateKeyFile conf/ssl.key/server-dsa.key

#   Server Certificate Chain:
#   Point SSLCertificateChainFile at a file containing the
#   concatenation of PEM encoded CA certificates which form the
#   certificate chain for the server certificate. Alternatively
#   the referenced file can be the same as SSLCertificateFile
#   when the CA certificates are directly appended to the server
#   certificate for convinience.
SSLCertificateChainFile conf/ssl/rootca.pem
#SSLCertificateChainFile conf/ssl.crt/ca.crt

#   Certificate Authority (CA):
#   Set the CA certificate verification path where to find CA
#   certificates for client authentication or alternatively one
#   huge file containing all of them (file must be PEM encoded)
#   Note: Inside SSLCACertificatePath you need hash symlinks
#         to point to the certificate files. Use the provided
#         Makefile to update the hash symlinks after changes.
SSLCACertificatePath conf/ssl
SSLCACertificateFile conf/ssl/ca.pem

#   Certificate Revocation Lists (CRL):
#   Set the CA revocation path where to find CA CRLs for client
#   authentication or alternatively one huge file containing all
#   of them (file must be PEM encoded)
#   Note: Inside SSLCARevocationPath you need hash symlinks
#         to point to the certificate files. Use the provided
#         Makefile to update the hash symlinks after changes.
#SSLCARevocationPath conf/ssl.crl
#SSLCARevocationFile conf/ssl.crl/ca-bundle.crl

#   Client Authentication (Type):
#   Client certificate verification type and depth.  Types are
#   none, optional, require and optional_no_ca.  Depth is a
#   number which specifies how deeply to verify the certificate
#   issuer chain before deciding the certificate is not valid.
#SSLVerifyClient require
#SSLVerifyDepth  10

#   Access Control:
#   With SSLRequire you can do per-directory access control based
#   on arbitrary complex boolean expressions containing server
#   variable checks and other lookup directives.  The syntax is a
#   mixture between C and Perl.  See the mod_ssl documentation
#   for more details.
#<Location />
#SSLRequire (    %{SSL_CIPHER} !~ m/^(EXP|NULL)/ ₩
#            and %{SSL_CLIENT_S_DN_O} eq "Snake Oil, Ltd." ₩
#            and %{SSL_CLIENT_S_DN_OU} in {"Staff", "CA", "Dev"} ₩
```

```
#            and %{TIME_WDAY} >= 1 and %{TIME_WDAY} <= 5 ₩
#            and %{TIME_HOUR} >= 8 and %{TIME_HOUR} <= 20       ) ₩
#          or %{REMOTE_ADDR} =~ m/^192₩.76₩.162₩.[0-9]+$/
#</Location>

#   SSL Engine Options:
#   Set various options for the SSL engine.
#   o FakeBasicAuth:
#     Translate the client X.509 into a Basic Authorisation.  This means that
#     the standard Auth/DBMAuth methods can be used for access control.  The
#     user name is the `one line' version of the client's X.509 certificate.
#     Note that no password is obtained from the user. Every entry in the user
#     file needs this password: `xxj31ZMTZzkVA'.
#   o ExportCertData:
#     This exports two additional environment variables: SSL_CLIENT_CERT and
#     SSL_SERVER_CERT. These contain the PEM-encoded certificates of the
#     server (always existing) and the client (only existing when client
#     authentication is used). This can be used to import the certificates
#     into CGI scripts.
#   o StdEnvVars:
#     This exports the standard SSL/TLS related `SSL_*' environment variables.
#     Per default this exportation is switched off for performance reasons,
#     because the extraction step is an expensive operation and is usually
#     useless for serving static content. So one usually enables the
#     exportation for CGI and SSI requests only.
#   o CompatEnvVars:
#     This exports obsolete environment variables for backward compatibility
#     to Apache-SSL 1.x, mod_ssl 2.0.x, Sioux 1.0 and Stronghold 2.x. Use this
#     to provide compatibility to existing CGI scripts.
#   o StrictRequire:
#     This denies access when "SSLRequireSSL" or "SSLRequire" applied even
#     under a "Satisfy any" situation, i.e. when it applies access is denied
#     and no other module can change it.
#   o OptRenegotiate:
#     This enables optimized SSL connection renegotiation handling when SSL
#     directives are used in per-directory context.
#SSLOptions +FakeBasicAuth +ExportCertData +CompatEnvVars +StrictRequire
<FilesMatch "₩.(cgi|shtml|phtml|php3?)$">
    SSLOptions +StdEnvVars
</FilesMatch>
<Directory "C:/Apache/Apache2/cgi">
    SSLOptions +StdEnvVars
</Directory>

#   SSL Protocol Adjustments:
#   The safe and default but still SSL/TLS standard compliant shutdown
#   approach is that mod_ssl sends the close notify alert but doesn't wait for
#   the close notify alert from client. When you need a different shutdown
#   approach you can use one of the following variables:
#   o ssl-unclean-shutdown:
#     This forces an unclean shutdown when the connection is closed, i.e. no
#     SSL close notify alert is send or allowed to received.  This violates
#     the SSL/TLS standard but is needed for some brain-dead browsers. Use
#     this when you receive I/O errors because of the standard approach where
#     mod_ssl sends the close notify alert.
#   o ssl-accurate-shutdown:
#     This forces an accurate shutdown when the connection is closed, i.e. a
#     SSL close notify alert is send and mod_ssl waits for the close notify
#     alert of the client. This is 100% SSL/TLS standard compliant, but in
#     practice often causes hanging connections with brain-dead browsers. Use
#     this only for browsers where you know that their SSL implementation
#     works correctly.
#   Notice: Most problems of broken clients are also related to the HTTP
#   keep-alive facility, so you usually additionally want to disable
#   keep-alive for those clients, too. Use variable "nokeepalive" for this.
#   Similarly, one has to force some clients to use HTTP/1.0 to workaround
#   their broken HTTP/1.1 implementation. Use variables "downgrade-1.0" and
#   "force-response-1.0" for this.

SetEnvIf User-Agent ".*MSIE.*" ₩
         nokeepalive ssl-unclean-shutdown ₩
         downgrade-1.0 force-response-1.0

#   Per-Server Logging:
#   The home of a custom SSL log file. Use this when you want a
#   compact non-error SSL logfile on a virtual host basis.
```

```
CustomLog logs/ssl_request_log ₩
          "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x ₩"%r₩" %b"

</VirtualHost>

</IfDefine>
```