Signed Applet

In Explorer

● 작성 : 오 광 신

2000년 4월 1일

raytrust@nser.co.kr

raytrust@dreamwiz.com

●이 문서는 웹에 있는 정보들을 제 나름대로 정리한 것입니다.

1. Microsoft SDK for JAVA(http://www.microsoft.com/java/)를 다운 로드 한 후 설치한다.

2. 설치한 Microsoft SDK for JAVA의 bin 디렉토리를 path에 설정해 준다.

예, path = %path%;E:\Program Files\Microsoft SDK for Java 4.0\bin

3. makecert 명령어를 사용하여 인증서를 생성한다.

makecert -sk [인증서 별명] -n "CN=[인증 이름]" [인증서 화일명]

-. 공개키와 비밀키 쌍이 생성되어지고, 비밀키는 레지스트리에 저장된다.

-. 인증서가 주어진 파일명으로 생성된다.

예, makecert -sk MyCertification -n "CN=KwangShin" KSCertification.cer

➔ KSCertificaiton.cer 파일이 생성된다.

makecert 명령어의 option 설명

- -sk "KeyName"
  보관되어 있는 키 이름.
  없을 경우 이 이름으로 레지스트리에 키를 생성한다.
- -ss "Store"
  인증서의 보관소 명
- -sr "Location"
  레지스트리 내의 인증서의 보관소 위치를 나타냄.
  (CurrentUser | LocalMachine) 중 하나
- -# "Number"
  1~130사이의 수.
- -$ "Authority"
  인증서를 발급한 기관 형태 (individual | commercial) 중 하나
- -n "X.509 name"
  X.500의 구분되는 이름 (예 : CN=Chungnam)
- -?
  기본 옵션의 목록 설명
- -!
  확장 옵션의 목록 설명

4. java 코드에 sign을 할 때, .spc 형식의 서명 파일이 필요하므로 cert2spc 명령을 사용하여 .cer 서명 파일에서 .spc 형식의 서명 파일을 만든다.

cert2spc [인증서 화일명] [.spc 화일]

예 > cert2spc KSCertification.cer KSCertification.spc

➔ KSCertification.spc 파일이 생성된다.

5. 서명이 필요한 파일들을 cabarc 명령을 사용하여 cab 형식의 파일로 만든다.

cabarc - r -p n [.cab 파일] [서명될 파일들]

예 > cabarc -r -p n KSSignedApplet.cab SignedApplet.class

➔ KSSignedApplet.cab 파일이 생성된다.

cabarc 명령어의 option 설명

- L
  cab 파일의 목록 보기 (예 : cabarc L test.cab)
- N
  새로운 cab 파일 생성 (예 : cabarc N test.cab *.c app.mak *.h)
- X
  cab 파일로부터 파일 꺼내기 (예 : cabarc X test.cab foo*.c)
- -c
  작업할 파일 확인 하기
- -o
  파일을 꺼낼 때 사용자에게 묻지 않고 덮어쓰기
- -m
  압축 형태 지정 (LZX:<15..21>|MSZIP|NONE). 기본값은 MSZIP
- -p
  파일명에 파일 경로 포함하기 (단, 상대 경로만 허용)
- -P
  파일 포함시 지정된 접두어 생략하기 (주로 파일 경로의 일부가 지정됨)
- -r
  하위 디렉토리의 파일까지 포함하기 (주로 -p 옵션과 함께 사용)
- -s
  cab파일내에 사인 정보를 둘 여유공간 확보 (예: -s 6144 => 6KB). SDK2.0 부터는 필요 없음
- -l
  cab파일 생성시 'cabinet set id'지정 (기본ID는 0)

6. signcode 명령어를 사용하여 앞에서 만든 인증서를 cab 파일에 포함 시키고, 레지스트리의 비밀키로 서명을 한다.

signcode -j javasign.dll -jp [레벨] -spc [.spc 파일] [.cab 파일]

예 > signcode -j javasign.dll -jp low -spc KSCertification.spc -k MyCertification KSSignedApplet.cab

➔ Sign된 KSSignedApplet.cab 파일이 된다.

cabarc 명령어의 option 설명

■ -spc "file"
SPC를 포함하는 파일명

■ -v "pvkFile"
비밀키를 포함하고 있는 파일명

■ -k "KeyName"
레지스트리 내의 키 이름

■ -n "name"
사인할 내용에 대한 텍스트 이름

■ -l "info"
사인할 내용에 대한 부가 설명이 있는 장소 (예: URL)

■ -p "provider"
시스템 내의 암호화 시스템 제공자 이름

■ -y "type"
시스템 내의 암호화 시스템 제공자 형태

■ -ky "keyType"
키의 종류 (signature | exchange | 정수) 중 하나

■ -$ "authority"
인증서를 인증한 기관의 종류 (individual | commercial) 중 하나

■ -a "algorithm"
사인에 이용된 해쉬 알고리즘. (md5 | sha1) 중 하나. 기본값 : md5

■ -t "URL"
타임스탬프를 찍어줄 서버의 HTTP주소

■ -tr "number"
타임스탬프 서버 접속 실패시 재시도 횟수. 기본은 1회

■ -tw "number"
타임스탬프간 간격 (초단위). 기본은 0초

■ -j "dllName"

사인에 필요한 부가 특성들을 포함하는 DLL 파일명 (예 : 보안 레벨)

■ -jp "param"

DLL파일에 넘길 파라미터

■ -c "file"

인코딩된 SPC를 포함한 X.509파일명

■ -s "Store"

인증서를 가지고 있는 인증서 보관소명 . 기본은 mystore

■ -r "location"

레지스트리 내의 인증서 보관소의 위치 (localMachine | currentUser)중 하나. 기본은 currentUser

■ -sp "policy"

인증서 검증에 필요한 모든 인증서를 포함할 것인가 아니면 SPC보관소에 들어있는 인증서가 나올 때까지 포함할 것인가에 대한 정책. (chain | spcstore)중 하나. 기본은 spcstore

■ -cn "name"

인증서 일반 이름 (별명)

■ -x

사인하지 말고 타임스탬프만 받을 것을 명시

7. html 페이지에 다음과 같이 넣는다.

```
<APPLET CODE="SignedApplet.class" WIDTH="700" HEIGHT="500" >
<PARAM name="cabbase" value="KSSignedApplet.cab">
</APPLET>
```

8. 위와 같이 하면, 클라이언트가 연결 되었을 때 윈도우가 뜨면서 인증할 것인지를 물어봅니다. 여기서, YES를 해주면 모든 권한을 가지게 된다.

일부의 권한만을 주기 위해서는 프로그램 내부에서 다음과 같이 해 주면 된다.

```
if (Class.forName("com.ms.security.PolicyEngine")!=null) {
   PolicyEngine.assertPermission(PermissionID.FILEIO);
}
```

위의 예제 중에서 다음 부분에 여러 가지 권한들을 부여할 수 있다.

```
PolicyEngine.assertPermission(PermissionID.???????);
```

"???????" 부분은 다음과 같은 ID가 올 수 있습니다.

1. SYSTEM
2. FILEIO
3. NETIO
4. THREAD
5. PROPERTY
6. EXEC
7. REFLECTION
8. PRINTING
9. SECURITY
10. REGISTRY
11. CLIENTSTORE
12. UI
13. SYSSTREAMS
14. USERFILEIO
15. MULTIMEDIA

* 참고 : 대개의 문서들이 위와 같이 각각의 권한을 주지 않으면 모든 권한을 준다고 말하고 있는데, 실제로는 그렇게 하면 안 돌아 가더군요. 그래서, 자기가 쓰고자 하는 권한을 꼭! 주어야 합니다.