Technical Note

Java Applet Signing Guide

Brendon J. Wilson
April 7, 2000
Version 1.1

Copyright Notice

Table of Contents

Overview

This document introduces the methods and tools required to digitally sign Java applet code, enabling the end user to verify the integrity of the Java code. By providing additional credentials with the applet code, the user's web browser can endow the applet with additional access to resources and operations normally prohibited by the restrictions of the Java security sandbox.

The techniques discussed in this document only address the signing techniques and tools required to produce signed code for the Netscape Navigator and MS Internet Explorer web browsers; no instructions for producing signed code under the Java 1.2 platform are provided at this time. In addition, only instructions for code signing under Microsoft Windows are provided; currently the Microsoft signing tools are only available for the Windows platforms, tying the process of creating a signed applet to the Windows platform.

Origin of this Document

This document is based on a document originally created during the HushMail project at Hush Communications. As the HushMail applet required access to resources normally restricted by the Java sandbox, the applet needed to be signed; the research conducted into code/applet signing has been captured in this document to reduce the learning curve incurred by Java developers who are new to the requirements of code signing.

Intended Audience

This document is intended for Java developers and architects who need a step-by-step procedure to guide them through the process of creating signed Java applets for Microsoft Internet Explorer and Netscape Communicator.

Relation to Other Systems

The code signing techniques described here rely on X.509 certificates, usually issued by a Certificate Authority such as Verisign, and the trust hierarchies explicitly specified by X.509 certificates.

Detailed Description

Introduction

Applet signing allows users to verify the code being downloaded by their web browser has not been replaced during the transmission from server to client by a malicious attacker. The two dominant distribution platforms for signed applets are the two major web browsers, Microsoft Internet Explorer (IE) and Netscape Communicator (NS). Unfortunately, the format of signed applets for these two browsers and the Java 1.2 Security Model released by JavaSoft are wholly incompatible. This document details the process and tools required to produce signed applets for both of these platforms.

In addition to the tools provided by Microsoft and Netscape, developers of signed code will require a certificate with which to sign their code. Though both IE and NS use industry standard X.509 certificates and PKCS #7 keys, each browser uses these elements in a proprietary manner; the IE scheme is known as Authenticode, and the NS scheme is known as Object Signing. Due to these incompatibilities, developers who wish to produce signed code for both platforms will require a certificate of each type, issued by a Certificate Authority. Luckily, both IE and NS tools allow the generation of test certificates for development and testing purposes.

Applet Signing With Microsoft SDK For Java

In order to produce signed applet code for the Microsoft Internet Explorer web browser, code must be packaged in the Microsoft CAB archive format, before being signed using the Microsoft `signcode` utility. This section outlines the process required to obtain and install the SDK, generate a test certificate, package files in a CAB, and sign the CAB with a certificate using the Microsoft SDK for Java.

Obtaining Microsoft SDK For Java

The latest available SDK release at this time is the Microsoft SDK for Java 4.0; the SDK can be downloaded for free from Microsoft:

http://www.microsoft.com/java/sdk/40/

In the event that this SDK is no longer available, an updated version should be available from Microsoft's Java web site:

http://www.microsoft.com/java/

Installing Microsoft SDK For Java 4.0

Once you have obtained a copy of the Microsoft SDK For Java 4.0 installation program, described above, run the `sdkjava40.exe` file and follow the installation instructions:

1. Click **Next** to bypass the installation information.
2. Click **Yes** to agree to the license agreement.
3. Click **Next**, leaving the installation directory as the default.

4. Click **Next**, leaving the default installation options.
5. The setup will run the installation.
6. Click **Finish**.

The Microsoft SDK For Java 4.0 is now installed on your machine. Documentation is available in the `Docs` directory under the default installation directory `C:\Program Files\Microsoft SDK For Java 4.0`. In order to be able to run any of the tools in the SDK from the command line, the system's PATH environment variable will need to be set by appending:

```
SET PATH=%PATH%;C:\PROGRA~1\MICROS~1.0\bin;
```

to the AUTOEXEC.bat file and restarting, or by manually invoking this command from the command line before attempting to use the SDK tools. Note that you may need to omit the first term (`%PATH%;`) if your PATH is already too long; if you invoke this command from the command line, this additional PATH will only affect your immediate DOS session.

Generating a Test Certificate

If an Authenticode certificate isn't available, you can use the SDK to generate a test certificate using the following procedure:

1. Go to the DOS prompt (**Start | Programs | MS-DOS Prompt**).
2. Change into a working directory.
3. Type `makecert -sk` **`MyKeyName`** `-n "CN=My Company Name"` **`MyTestCert.cer`**, where **`MyKeyName`** is the name of the key, and **`MyTestCert.cer`** is the file to contain the certificate. This command will generate a test certificate, and place it in the working directory.
4. Type `cert2spc` **`MyTestCert.cer MyTestCert.spc`** to convert the **`MyTestCert.cer`** certificate to a Software Publisher Certificate stored in the file **`MyTestCert.spc`**.

You now have a test certificate (**`MyTestCert.spc`**) suitable for signing CAB archive files for distribution via Microsoft Internet Explorer.

Creating a CAB Archive File

Before signing any code, you must package the code into a single CAB archive file. For this step, you'll require a working directory containing all of the compiled code to be packaged. To package the code:

1. Go to the DOS prompt (**Start | Programs | MS-DOS Prompt)**
2. Change to the working directory containing the code to package in a CAB
3. Type `cabarc /r /p n` **`MyApplet.cab`** `*.class *.properties` to create a new (**n**) CAB called **`MyApplet.cab`**, containing all of the .class and .properties files in the current directory and subdirectories (**`/r`**), while preserving the directory paths (**`/P`**).

You now have a single CAB archive file (**`MyApplet.cab`**) suitable for signing.

Signing a CAB Archive File

Using the CAB file prepared in the previous section and a certificate, it's a simple matter to sign the CAB. For this step, you'll require a working directory containing both the CAB to be signed, and the certificate to use for the signature. To sign the CAB with the certificate:

1. Go to the DOS prompt (**Start | Programs | MS-DOS Prompt**).
2. Change to the working directory containing both the CAB and the certificate to use for the signature.
3. Type `signcode -j javasign.dll -jp` **`medium`** `-spc` **`MyTestCert.spc -k MyKeyName MyApplet.cab`** to sign the **`MyApplet.cab`** CAB using the key **`MyKeyName`** found in the **`MyTestCert.spc`** certificate. This signature will request the code in the CAB be endowed with **`medium`** risk permissions; for more information on IE permission levels, see the documentation accompanying the SDK.

You now have a signed CAB (**`MyApplet.cab`**) that can be distributed using Internet Explorer. Note that if a test certificate has been used to sign the code, this code will not be recognized as secure by IE unless the test certificate has been installed in the web browser and the test root authority has been enabled. The next section details the process of installing the test certificate.

Installing a Test Certificate in Internet Explorer

To enable trust on the test root, you'll need to use the `setreg` utility provided with the SDK using the following command:

```
setreg 1 TRUE
```

in order to set option 1 to TRUE; option 1 corresponds to the 'Trust the Test Root' option in the registry. Once the 'Test Root' has been granted trust authority, you can check that a CAB signed with the test certificate is valid by executing the following command on the signed CAB:

```
chkjava MyApplet.cab
```

where `MyApplet.cab` is the CAB to verify. This utility will bring up a dialog detailing the signature on the CAB.

Applet Signing With Netscape Signtool

In order to produce signed applet code for the Netscape Communicator web browser, code must be packaged in the JavaSoft JAR archive format before being signed using the `signtool` utility. This section outlines the process required to obtain and install the Signtool utility, generate a test certificate, package files in a JAR, and sign the JAR with the test certificate using the Netscape Signtool utility.

Obtaining Netscape Signtool

The latest available Signtool utility release at this time is the Netscape Signtool 1.3; the utility can be downloaded for free from Netscape:

http://developer.netscape.com/software/signedobj/jarpack.html#signtool1.3

In the event that this release is no longer available, an updated version should be available from Netscape's Object Signing site:

http://developer.netscape.com/docs/manuals/signedobj/overview.html

Installing Netscape Signtool 1.3

Once you have obtained a copy of the Netscape Signtool 1.3 installation program, described above, unzip the `signtool13WIN95.zip` file to the `C:\Program Files` directory. The Netscape Signtool is now installed on your machine; to be able to run any of the tools from the command line, the system's PATH environment variable will need to be set by appending:

```
SET PATH = %PATH%;C:\PROGRA~1\SIGNTO~1;
```

to the AUTOEXEC.bat file and restarting, or by manually invoking this command from the command line before attempting to use the Netscape tools. Note that you may need to omit the first term (`%PATH%;`) if your PATH is already too long; if you invoke this command from the command line, this additional PATH will only affect your immediate DOS session.

Generating a Test Certificate

If an Authenticode certificate isn't available, you can use Signtool to generate a test certificate. Before generating a test certificate, you must have Netscape Communicator installed on your machine, and you must have already set a password to protect your certificates. To set the password on the certificates in your browser:

1. Start Netscape Communicator.
2. Go to **Communicator | Tools | Security Info**.
3. Select **Passwords**.
4. Click **Set Password**.
5. Enter a password and a verification password, and click **OK**.
6. Click **OK** in the confirmation dialog.
7. Click **OK** to dismiss the Security Info dialog.

To generate a test certificate, ensure that Netscape is closed, and use the following procedure:

1. Go to the DOS prompt (**Start | Programs | MS-DOS Prompt**).
2. Change into a working directory.
3. Type **`signtool -G MyTestCert –d ..\Netscape\Users\Name`** to create a key named **`MyTestCert`**. The relative directory is required to allow the tool to

locate the keys3.db and certs7.db files for the user profile **Name**, and register the test certificate with the local copy of Netscape Communicator.

4. Type **y** to confirm Netscape is not currently open.
5. Provide the following information:

```
certificate common name: MyKeyName
organization: My Company Name
organization unit: My Department
state or province: BC
country (must be exactly 2 characters): CA
username: myname
email address: me@mycompany.com
```

You now have a test certificate (two files: **x509.cacert**, and **x509.raw**) suitable for signing JAR archive files for distribution via Netscape Communicator. Note that the actual certificates used for the signing are the versions installed in Netscape Communicator; the files created by the tool are only required to install the test certificate in additional copies of Netscape Communicator.

Signing a JAR Archive File

Using the test certificate installed in Netscape Communicator, it's a simple matter to sign the JAR. For this step, you'll require a working directory containing the files to be signed and packages into a JAR. To create the JAR and sign it with the test certificate:

1. Go to the DOS prompt (**Start | Programs | MS-DOS Prompt**).
2. Change to the working directory containing the JAR to be signed.
3. Type `signtool -k` **`MyTestCert`** `-Z` **`MyApplet.jar`** `-d` **`..\Netscape\Users\Name`** `.` to sign the code in the current directory and subdirectories using the certificate **MyTestCert**, and place the resulting signed code in a JAR called **MyApplet.jar**. The relative directory is required to allow the tool to locate the keys3.db and certs7.db files for the user profile **Name**, in order to find the certificate in the local copy of Netscape Communicator.

You now have a signed JAR (**MyApplet.jar**) that can be distributed using Netscape Communicator. Note that if a test certificate has been used to sign the code, Netscape Communicator will not recognize this code as secure unless the test certificate has been installed in the web browser. The next section details the process of installing the test certificate.

Installing a Test Certificate in Netscape Communicator

If a test certificate used to sign the applet has not been installed in the web browser, you'll need to install the certificate in the web browser before JAR archives signed with the test certificate will be recognized as valid. To install the test certificate in Netscape Communicator, you only need to create a local HTML file containing the line:

```
<a href="x509.cacert">Import My Test Certificate</a>
```

in the same directory as the “x509.cacert” test certificate. Start Netscape, load the HTML file, and click the link; the certificate will be installed into your local copy of Netscape Communicator.

Once the certificate is installed in Netscape, you can also verify the signature of the signed JAR from the command line:

1. Go to the DOS prompt (**Start | Programs | MS-DOS Prompt**).
2. Change to the working directory containing the JAR to verify.
3. Type `signtool -v` **`MyApplet.jar`** `-d` **`..\Netscape\Users\Name`**, which will verify the **`MyApplet.jar`** using the local copy of Netscape. The relative directory is required to allow the tool to locate the keys3.db and certs7.db files for the user profile **`Name`**, in order to find the test certificate in the local copy of Netscape Communicator.

HTML Invocation of Signed Applets

In order to include signed archives in web pages, the signed archive must be included in the APPLET tag that calls code contained within archive. However, due to the proprietary nature of the code signing mechanism, it should come as no surprise that the tags required to included signed archives is also browser-dependent. Fortunately, a mutually compatible method is available to include both signed CABs and JARs in a single APPLET tag, as shown below:

```
<APPLET code="MyApplet" WIDTH=10 HEIGHT=10 ARCHIVE="MyApplet.jar">
  <PARAM NAME=cabbase VALUE=MyApplet.cab>
  . . .
</APPLET>
```

As with a regular APPLET tag, the signed JAR is included in the ARCHIVE attribute of the APPLET tag; however, to allow for inclusion of a signed CAB, the additional PARAM “cabbase” is added. In the case of Netscape Communicator, the “cabbase” parameter is passed to the applet as normal, and should be ignored by the applet. In the case of Internet Explorer, when the “cabbase” parameter is found, the ARCHIVE attribute on the APPLET tag is ignored, and the “cabbase” is used to provide the location of the signed archive file.

GNU Free Documentation License

In order to limit the size of this document, a copy of the GNU Free Documentation Version 1.1 has not been included within the document itself. A copy of the license is freely available for download from the Free Software Foundation:

http://www.gnu.org/copyleft/fdl.html

Alternatively, a copy of the license terms can be obtained by writing the Free Software Foundation at:

Free Software Foundation, Inc.
59 Temple Place, Suite 330,
Boston, MA
02111-1307
USA

References

Microsoft Corporation. Microsoft SDK for Java, version 4.0 [online]: Microsoft Corporation, 2000 [cited April 2000]. Available from the World Wide Web: (http://www.microsoft.com/java/sdk/40/)

Microsoft Corporation. Microsoft Technologies for Java [online]: Microsoft Corporation, 2000 [cited April 2000]. Available from the World Wide Web: (http://www.microsoft.com/java/)

Netscape Communications Corporation. Object-Signing Resources [online]: Netscape Communications Corporation, 2000 [cited April 2000]. Available from the World Wide Web: (http://developer.netscape.com/docs/manuals/signedobj/overview.html)

Netscape Communications Corporation. Object-Signing Tools [online]: Netscape Communications Corporation, 2000 [cited April 2000]. Available from the World Wide Web: (http://developer.netscape.com/software/signedobj/signtool1.3)

Revision History

Version	Author	Date	Description
1.0	Brendon J. Wilson	April 7, 2000	Document Created
1.1	Brendon J. Wilson	June 9, 2001	Added GNU license info.