TelForm Applet Security

Problems and Solutions as applied to TelForm Applets

Java was the first programming language written with networking, and therefore Security, in mind. The aim is to protect the Java user from applets written by those with hostile intent and by those who have not tested their code against unwanted side-effects. Following is a description of some of the sandbox limitations of Java Applets:-
No Reading-of or Writing-to Local storage or memory areas not directly relevant to the applet.
Making a network connection to any computer except the applet's host is forbidden.
Launching a new process or programme is not possible.
Libraries with the above or similar abilities may not be loaded.

For a long time in Java's history there was very little cooperation between the few PC Operating System and Web Browser corporations and therefore no agreed mechanism to grant permission to over-ride these restrictions. In 1997 Elliotte Rusty Harold recommended writing applications instead (See references). On the other hand since the Netscape 2 web browser was released in 1995 and Internet Explorer shortly afterwards, scripts have been able to make server requests and responses unknown to the Client or the Server, and can call on any other network script. Consequently script authors have had access to user data and have been at the root of major known security scares. It is therefore easy to see why Web Developers tended to take the easy option and use lax script languages, rather than sandboxed Java. Operating System and Web Browser patches and updates still struggle to lessen the effects of malware or poor scripting, with varying degrees of success and unknown measures of failure.

Java Security Manager

The Java language and the Java Virtual Machine which runs the Java code has a number of security safety features. The JVM has inbuilt mechanisms which enforce the Sandbox. Among these are are
The Class Loader architecture
The Class file verifier
The Java Security Manager

The Java Security Manager is the programmable mechanism for administering Java Security which has developed with Java releases.

The first version of Java had no practicable means of qualifying exceptions to the above limitations on applets. In Java 1.1 the Java Security Manager was provided with the ability of verifying that the Class files in a Jar file were signed with a Digital Signature. This should prove that the code comes from a trusted verifiable source and has not been altered. However in Java 1.1 all files in the java archive were granted all permissions, it was not until Java 1.2, released in 1999, that more sophisticated access control for individual classes was introduced.

Digital Signatures

Digital Signatures are files of numerical data issued by Certificate Authorities to individuals and organisations to be used in Encryption of network data and Authentication of data senders. Explanations of how Public Key Infrastructure PKI works may be found elsewhere and a reasonable understanding will be assumed here. Decryption is the reverse process of Encryption.

Even the original proponents admit that because the same certificate is used for both Encryption and Authentication the entire process is made more complicated and Encryption is weakened, because the public key is made highly visible during the Authentication process and cannot be

easily changed. Digital certificates are used to convince the user of Java applets that the file has been produced by a trusted author, has not been altered, and may therefore be permitted exceptions to the normal restrictions.

Web Browsers contain a store of Root Certificates - certificates issued by CAs within the top tiers of the hierarchy of Certificate Authorities. Certificates issued by intermediate CAs lower down the CA tree are trusted only if a verification chain to a root CA is proven as unbroken by the Web Browser's authentication process. If the Certificate proves valid, the applet runs unhindered, if not the user is presented with a pop-up dialogue, warning the user to allow the applet to continue at their peril. Not surprisingly most developers, and their customers even more so, will pay a lot to avoid this. For this reason very few PC users are even aware of Digital Certificates while relying on them for 'secure' trading over the Internet.

We can assume CAs have a commercial relationship with Web Browser companies. Netscape 4.0.2 included Thawte in their root certificate cache, but Microsoft did not do so until Verisign bought the company. A major part of the Certificate Authority's role is to verify the identity of the Digital Certificate purchaser, and presumably are a lot more stringent in this since Verisign issued an individual fraudulently purporting to be the Microsoft Corporation in 2001, from which both companies and their Customers took years to recover. In 2009 a Digital Certificate costs between $300-$2,500 per year. The actual practice of checking authentication documents such as Business Registration documents, Driving Licences and similar may be done by Registration Authorities.

A few years ago in the UK, I helped two friends with problems with their PCs (they were not known to each other). I noticed in both Internet Explorer Certificate Stores they had the same Certificates for the same pornographic sites, I was convinced of their innocence not only on a character basis but also on technical and circumstantial grounds. More investigation led me to believe that somebody had distributed a bogus Internet Explorer update CD to the clone PC shops and 'PC doctors' in the city. After an anti-virus service or upgrade, the PC was returned to the unsuspecting Customers with security even weaker than before. In the end, the security of a Computer system is more related to the security features of the Computer file system than network restrictions.

Self Signed Certificates

An individual or organisation may generate a Self Signed Certificate using freely available tools or those that come with some application suites. Again though support for these may not be universal and it is alleged that some applications will delete these from Certificate stores.

Once an application for a Digital Certificate has been authenticated by a Registration Authority, the public key of the Certificate with the appropriate information is transmitted with the encrypted information and may be stored. But individuals and Organisations change. Important individuals such as the Technical Contact may change jobs, or there may be an address change. Mergers & Acquisitions can affect corporate identities, and even internal re-organisation may mean that the identity on a Digital Certificate issued to a sub-organisation is rendered inaccurate. This may mean that some browsers will then challenge the user to accept or reject the Certificate. The current culture of protecting the PC user from responsibility for Security is therefore unsatisfactory.

Revoked Certificates

At this time the main method of checking the validity of a Digital Certificate is by checking the latest Certificate Revocation List CRL to see if it has been withdrawn. This requires additional effort on the part of the Certificate Authority and the Certificate end user. Some applications can automate this process but the CRL then also requires security precautions.

A big improvement is imminent with the increasing adoption of the Online Certificate Status Protocol (OCSP). Here a clients such as a browser can send a request to an OCSP responder to verify the status of a Certificate.

Richer Clients, Stronger Servers

The more processing performed by a Client programme, the less effort is needed by the Server which reduces processing time. It also means it is easier for the Server to detect bogus requests and it is logical to assume this gives the Server greater resilience from Denial Of Service attacks.

Bibliography

Reference: Java Network Programming by Elliotte Rusty Harold, O'Reilly, 1997. ISBN 1-56592-227-1

Reference: Inside the Java 2 Virtual Machine by Bill Venners, McGraw Hill, 1999. ISBN 0-07-135093-4

Reference: Advanced Programming for Java 2 platform by Austin and Pawlan, Addison-Wesley, 2000. ISBN 0-201-71501-5

Reference: Java Developers Almanac by Patrick Chan, Sun, 2000. ISBN 0-201-43299-4

Reference: Java J2SE 1.4 Core Platform Update by James Hart, Wrox, 2002. ISBN 1-861007-27-2