Secure Coding.
Practical steps to defend your web apps.

Comparison of Java Applets and ActiveX Controls

Jim Hurst

Comparison of Java Applets and ActiveX Controls

Introduction

The Internet is a window to the world. News, games, entertainment, mail, and shared documents are only a web browser away. Content-rich websites are the norm. Animation and interactive pages are useful and entertaining. They are with us to stay.

This paper examines two of the first technologies used to deliver rich content to the desktop: Java applets and ActiveX controls. They use different approaches for downloadable code to provide a similar functionality.

Both technologies are based on code that is downloaded from the web server and then executed on the local computer. The world was a more innocent place, at least in terms of computer security, when the World Wide Web was new. Downloadable code has become much more suspect in the past 5 years. Dynamic web sites now often use a back-end database and languages such as PHP to create customized pages for each user. Managing the delivery of downloadable code securely turned out to be a difficult proposition, and web designers are moving away from it.

Java Programming Language

A discussion of Java applets begins with the Java language itself. Java is an object-oriented programming language developed by Sun Microsystems. The most interesting thing about Java is that it was designed from the ground up to be platform independent. Developers can write code in Java, and it can run on any operating system (at least in theory; in practice it's slightly more complex.).

Java's platform independence is made possible by compiling the source code to bytecode, which is binary code meant to be executed by a virtual machine, rather than actual hardware. The virtual machine is a piece of software that resides between the running Java program and the operating system. A program running on a virtual machine makes all its requests for services to the virtual machine. The virtual machine then translates those requests into requests to the operating system and actual hardware.

What this means for Java is that Java programs can run on any platform that supports a Java Virtual Machine (JVM), and most modern platforms support a JVM. The virtual machine manages all interactions between the Java program and the platform.

Java Security Model

The primary focus here is on security for Java applets. But what exactly does it means for a programming technology to be secure? This goes back to the first principles of information security: The programming language should support the confidentiality, integrity, and availability of the information on the computer system. This means that the language should be not allow malicious programs (trojans and viruses) to harm the user's computer. It should protect private information on the host computer and the network. Where required, it should support authentication and encryption.

Java tackles these requirements with the sandbox model. This model places strict limitations on the resources available to a running program. The sandbox model can control resource access at a number of levels: computer memory, filesystem, http access to web servers, and general network access. The Java security model defines how these resources are to be protected. Sandbox settings control what level of access is allowed, from the absolute minimum needed to run a program all the way to full access to all system resources.

The anatomy of a Java program strongly reflects the security model. Java objects are contained in class files. The class loader loads these files after validation by the bytecode verifier, which enforces language rules and memory protections. The Core Java API (applications programming interface) uses the loaded classes to interact with the operating system via the security manager. The security manager is the subsystem tasked with allowing or preventing access to system resources.

Java Applets

An applet is a software component that runs inside some larger application. Java applets run in the context of a Java-enabled web browser. The web browser is responsible for maintaining the sandbox that manages the applet's resource access. In practice, this usually means preventing the applet from accessing the local filesystem. The browser downloads the applet code from a web server and either embeds the applet into an html page or opens a new browser window to show the applet user interface. The default security manager denies applets all access to the filesystem and all network access except to the web host that supplied the applet.

Java applets are marvelous things. They provide interactive features to web pages far beyond what can be accomplished in html. Because they are platform independent, they can run on Windows, Mac OS, Linux, and other platforms. A perceived advantage of applets in the early days was to move processing from the server to the client, which made web applications more scalable for busy servers. This is a less important feature today because processor costs drop and multiprocessor systems become the norm. Java applets can run in real time, which makes them useful for monitoring and control applications.

Applets have several drawbacks Applets require a Java plug-in, which is not always available. Some organizations do not allow users to install software, so these users might not view applets by default. There are also performance issues. The applet cannot run until the Java Virtual Machine is initialized, and this delay can be significant. Applets usually execute at a speed that is comparable to, but slower than, compiled applications. Finally, Java applets are considered more difficult and expensive to develop than html based pages.

The security model for applets is considered to be sound. Several vulnerabilities have been found in the Java Runtime Environment, which is the virtual machine used by the browser.

ActiveX Controls

ActiveX controls are a Microsoft implementation of downloadable code. This discussion focuses on ActiveX controls as they are incorporated into the Internet Explorer web browser, but readers should be aware that the technology is not limited to browsers. ActiveX controls are based on the Component Object Model and Object Linking and Embedding, so a quick explanation of these topics follows.

OLE and COM

Object Linking and Embedding (OLE) is a protocol developed by Microsoft to support distributed objects. It provides for complex multimedia documents and the sharing of pieces of documents (which could be text, spreadsheets, images, or audio objects) between applications. OLE 1.0 was introduced in 1990 and evolved to become the basis of the software component architecture known as the Component Object Model (COM), and later, the Distributed Component Object Model (DCOM). COM and DCOM provide a language-neutral method of creating objects that can be shared across applications, and even machines.

OLE 2.0 was renamed ActiveX in 1996. ActiveX introduced ActiveX controls and ActiveX Scripting. ActiveX was popular in years past with web designers for adding multimedia content to web pages. Today, COM, DCOM, and ActiveX have been largely replaced by Microsoft's .NET initiative.

ActiveX Controls

ActiveX controls are software components based on the COM. Like Java applets, ActiveX controls can be used to add rich content to web pages. Unlike applets, ActiveX controls are limited to use in Microsoft's Internet Explorer web browser.

ActiveX controls are normally, but not always, visual in nature. A visual control can be used to display a movie. A nonvisual application of an ActiveX control might be to download a keystroke logger from an FTP site and install it. A malicious ActiveX control might do both, without the user realizing that malware was being installed as the movie was playing.

There is nothing inherently insecure about ActiveX controls, although they have been blamed for numerous malware incidents. ActiveX controls are a programming technology and are just as secure as other technologies. But because ActiveX controls are Microsoft Win32 components, they have full access to the operating system, with all the risk that this implies.

The risk is that Internet Explorer users will allow ActiveX controls from nontrusted sites on the Internet. In some cases, legitimate ActiveX controls have been repurposed with hostile intent to attack the local computer.

This potential for mischief in ActiveX controls has been widely exploited by generations of malware. After an Internet Explorer user turns on the browser's ability to download and activate ActiveX controls on web pages, a malicious website can run any code on that machine. Spam and phishing attacks have been used to draw users to hostile websites that contain malicious ActiveX components.

Authentication

Microsoft responded to these risks by developing a registration system for browsers to authenticate controls before downloading them. Companies authoring ActiveX controls register with Microsoft and use digital signing to validate both their identity and the ActiveX control itself. Technically, this code-signing approach is an adequate solution. In practice, digital signing is not well understood by many users. If the users want to run the program, they might ignore the fact that Microsoft refused to authenticate the code.

Contrasting the Two Approaches

Java applets and ActiveX controls provide a similar functionality to the browser: multimedia content that can be embedded in HTML pages, based on downloadable code. Beyond that, they are quite different.

ActiveX is an API designed to allow the sharing of objects between programs. Support for ActiveX must be "baked in" to applications during their development. The most important application to incorporate ActiveX controls is Microsoft's Internet Explorer web browser. If the browser is permitted to run the ActiveX control, the control has full access to the operating system. There is no middle ground: Security is based on accepting or rejecting an ActiveX control based on the perceived trustworthiness of the website.

Java applets can be run in any browser with the Java plug-in. Applets run in a virtual machine, which limits the access to resources according to the security policy in place. The virtual machine uses a sandbox model, which protects against memory corruption, buffer overflow, and unauthorized access to system resources. Java applets can be written with hostile intent, but normally, applets are prevented from reading and writing files to the client system, and from making network connections to any system except the originating host.

Summary

Java applets are pieces of downloadable code used to add functionality to web pages. They are written in the Java programming language, which compiles to bytecode that runs in a virtual machine. The virtual machine for applets is the Java plug-in used in a web browser. This virtual machine denies all applets most access to system resources.

ActiveX controls are bits of downloadable code used by Microsoft's Internet Explorer browser to enhance web pages. ActiveX is a protocol for sharing objects between programs based on Object Linking and Embedding and the Common Object Model. Security for an ActiveX control is based upon the user verifying the author of the control, because when accepted, an ActiveX control has full access to the operating system. Microsoft has implemented a registry of digital signatures to verify ActiveX controls.

The age of downloadable code appears to be drawing to an end. Microsoft has deprecated ActiveX controls. Java applets are less popular than they were 5 years ago. These technologies will continue to be used, but they represent the past, not the future.

References

Books

Oaks, Scott. *Java Security*. O'Reilly Press, Sebastopol, CA. 1998.

Websites

Code Signing Digital IDs. Comodo Group.

http://www.instantssl.com/code-signing/

Designing Secure ActiveX Controls. Microsoft Corporation.

http://msdn.microsoft.com/workshop/components/activex/security.asp

Java Applet Security. Sun Microsystems.

http://java.sun.com/sfaq/

Last Updated: June 22nd, 2018

Upcoming SANS App Sec Training

SANSFIRE 2018	Washington, DC	Jul 14, 2018 - Jul 21, 2018	Live Event
SANS Northern Virginia- Alexandria 2018	Alexandria, VA	Aug 13, 2018 - Aug 18, 2018	Live Event
SANS Virginia Beach 2018	Virginia Beach, VA	Aug 20, 2018 - Aug 31, 2018	Live Event
Community SANS Nashville DEV540	Nashville, TN	Aug 27, 2018 - Aug 31, 2018	Community SANS
SANS Amsterdam September 2018	Amsterdam, Netherlands	Sep 03, 2018 - Sep 08, 2018	Live Event
Community SANS New York DEV522	New York, NY	Sep 17, 2018 - Sep 22, 2018	Community SANS
Community SANS Toronto DEV540 @ Security Compass	Toronto, ON	Sep 17, 2018 - Sep 21, 2018	Community SANS
SANS London September 2018	London, United Kingdom	Sep 17, 2018 - Sep 22, 2018	Live Event
SANS Network Security 2018	Las Vegas, NV	Sep 23, 2018 - Sep 30, 2018	Live Event
Community SANS Nashville DEV540	Nashville, TN	Oct 01, 2018 - Oct 05, 2018	Community SANS
SANS Brussels October 2018	Brussels, Belgium	Oct 08, 2018 - Oct 13, 2018	Live Event
SANS October Singapore 2018	Singapore, Singapore	Oct 15, 2018 - Oct 27, 2018	Live Event
Secure DevOps Summit & Training 2018	Denver, CO	Oct 22, 2018 - Oct 29, 2018	Live Event
SANS Gulf Region 2018	Dubai, United Arab Emirates	Nov 03, 2018 - Nov 15, 2018	Live Event
SANS London November 2018	London, United Kingdom	Nov 05, 2018 - Nov 10, 2018	Live Event
SANS Santa Monica 2018	Santa Monica, CA	Dec 03, 2018 - Dec 08, 2018	Live Event
Community SANS New York DEV540	New York, NY	Dec 10, 2018 - Dec 14, 2018	Community SANS
SANS Cyber Defense Initiative 2018	Washington, DC	Dec 11, 2018 - Dec 18, 2018	Live Event
SANS Security East 2019	New Orleans, LA	Feb 02, 2019 - Feb 09, 2019	Live Event
SANS OnDemand	Online	Anytime	Self Paced
SANS SelfStudy	Books & MP3s Only	Anytime	Self Paced