Product Briefing

SEPTEMBER 2002

Secure Design with Flash Player

Macromedia's Flash Player provides a rich set of multimedia capabilities that permits the creation of compelling applications. Because security features are essential in the widespread deployment of applications, the Flash Player provides functionality that protects both application and host platform. This brief identifies design and development practices that make effective use of the security features within Flash Player.

@stake partnered with Macromedia to assess the design and operation of Flash Player 6 (hereafter referred to simply as Flash Player) with respect to industry best practices in security.

- An **Application Architecture Assessment** included a review of the application's design, its interactions with the operating system and external components, and its handling of potentially confidential data.
- An **Application Penetration Assessment** provided a practical demonstration of the security aspects of the Flash Player by attempting to circumvent those features intended to protect the confidentiality and integrity of the application, its data, and its host platform.

Like all software, the Flash Player functions according to the requirements and expectations of its design scope. This document describes the boundaries within which the Flash Player meets requirements to protect its data and host platform. To retain the intended security properties of the Flash Player platform, developers should design and develop their applications within these boundaries.

Flash Player Architecture Overview

Macromedia's Flash Player is comprised of several components providing functionality at different levels. The core player consists of platform-independent code that governs the interpretation, execution, and rendering of Flash content. Operating system-specific modules implement the interactions between the core player and the operating system, including file system operations, multimedia device access, and browser interaction. Each component provides security features that protect the application, data, and platform within its scope of operation.

Specific security features implemented in the Flash Player software include:

- The **security sandbox** to facilitate controlled access to data.
- **Data container files** to insulate the file system from direct access by applications.
- The **ActionScript interpreter**, which allows Flash applications to interact with the user and target platform within well-defined constraints.
- **User controls** that govern the use of multimedia input devices and the downloading of player components.
- **Network access constraints** that limit Flash Player applications' interactions to specific protocols and hosts.

Flash Player Deployment Options

The Flash Player can be deployed on a target platform in two forms. The intended deployment model is the standard installation of the Flash Player on the target platform, and is invoked after the user has obtained the desired content. Alternatively, a developer of Flash content can create a "projector" that bundles a Flash Player application with a standalone Flash Player executable; when invoked on the target machine, this executable plays the bundled application.

However, distribution of projectors becomes a less attractive option for content development and distribution as users become increasingly wary of the dangers posed to their computers by executable applications arriving from third parties who they might not trust. In addition, because a projector is unable to fully implement the complete Flash Player security model, this deployment option should be avoided in all but tightly controlled situations.

Flash Player Security Features

This section analyzes the key Flash Player security features reviewed by @stake.

Reliance on DNS Security

Domain Name Service (DNS) is the infrastructure component of the Internet that acts as a directory and allows mnemonic names, such as "www.macromedia.com," to identify computers. Several Flash Player security features depend on accurate DNS lookups when making access control decisions. In particular, data sharing between individual Flash Player applications relies on the accuracy of DNS to identify that the two applications originated from the same source or that they are authorized to access shared data.

DNS query results can be trusted within certain narrowly defined limits. These limits include situations where:

- The DNS server and any forwarders upon which it relies are trusted to provide accurate lookups and are authoritative for every domain being queried.
- DNS queries are not subject to eavesdropping or interception at any point on the network.

Generally, these constraints are difficult to meet except within carefully controlled corporate environments.

Flash Player Application Data Security

The Flash Player security sandbox permits peer Flash Player applications to share data in the following two circumstances:

- **Data sharing between two peer Flash Player applications originating from machines within the same DNS domain.** Before permitting one Flash Player application to access data belonging to another, the Flash Player examines the DNS domains from which each of the two applications originated. If the domains, except for the host name, match, then each application can access the other's local shared objects without restriction.
- **Data sharing between two peer Flash Player applications originating from machines in different DNS domains.** A Flash Player application can inform the Flash Player that it intends to provide applications from other domains with access to its shared objects. As in the case with peer Flash Player applications from the same domain, the Flash Player will identify these secondary applications based on their domain names.

The Flash Player can only enforce access restrictions to the extent that DNS provides accurate resolutions. Therefore, in situations where DNS resolution cannot be controlled entirely by the owner of an application, the developer must avoid placing confidential or sensitive information in shared objects.

Client System Security

The Flash Player is designed to protect the target platform from unauthorized access while providing certain aspects of the platform for use by Flash content. Of primary interest are the Flash Player's ability to limit access to the file system, media inputs, and code execution.

File System

The Flash Player does not permit Flash Player applications to access the file system directly; instead, it requires indirect access through methods it provides. This encapsulation permits the Flash Player to control the precise location of all the data a an application stores. The player stores data in files within a player-specified directory hierarchy, using specific naming conventions that preclude the creation of arbitrary files within the target platform's file system. The player stores the names and contents of local shared objects within the data files, which permit developers to use any arbitrary name and data without interfering with the normal, secure operation of the target platform.

Media Inputs

The Flash Player introduces the ability to use camera and microphone data within a larger application, presenting these streams to a remote server for handling within that application. Upon the first attempt by an application from a given DNS domain to access multimedia inputs, the player presents the user with a dialog box requesting permission to provide camera and microphone access to that domain. The user's decision persists through subsequent invocations of applications and player restarts, until the user invokes the Flash Player's Preference dialog box to change the setting.

Code Execution

A significant concern regarding downloadable applications is their interactions with the target operating system or machine and the security consequences to the platform. The Flash Player provides a proprietary, scripted execution environment that is the only way a Flash Player application can supply instructions for execution. This environment, known as the ActionScript Interpreter, limits the interactions applications can have with the target platform and other external entities by providing a set of objects and methods that constrain the interactions within safe boundaries.

ActionScript programs are not native to any target platform of the Flash Player, so a program that runs within the interpreter cannot run in the host operating system. Furthermore, objects that an ActionScript program manipulates are neither executed by the interpreter or the host platform, nor stored or transmitted in a form that could eventually interact with the platform. These features work together to prevent ActionScript programs from escaping to the host environment or otherwise interacting with the host system in an uncontrolled manner.

Network Security

The Flash Player provides network connectivity to applications interacting with other entities across a network. It presents network connectivity to applications as a set of objects and methods that limit the interactions an application has. The following sections discuss these areas in further detail.

AMF Protocol

The Action Message Format, or AMF, protocol permits Flash Player applications to invoke remote procedures, using HTTP or HTTPS as a transport and supplying ActionScript objects as parameters. The Flash Player restricts outbound AMF connections to those target hosts that share domain names with the application's source, except for the host name. This connection limitation is effective for applications originating from domains under the control of trusted entities, subject to the DNS constraints described above.

Within intranets that deploy the Flash Player, all AMF listeners with sensitive or non-public information should follow security best practices by requiring authentication of the AMF client to protect that information. Developers should protect sensitive and confidential information transmitted across an AMF connection by specifying an HTTPS:// (SSL) URL as the connection gateway.

XML/HTTP

The Flash Player provides an XML-over-HTTP transport for use by applications. The Flash Player packages this capability as an ActionScript object and imposes usage restrictions on it: a port limitation excludes connections to ports under 1023 to prevent accesses of well-known services, and the target host must share the domain name with the Flash Player application's source, except for the host name. This latter limitation is effective for applications originating from domains under the control of trusted entities, subject to the DNS constraints described above.

Product Downloads

The Flash Player enables the applications it plays to download new player components from Macromedia. When invoked, this download process first requests the user's permission to proceed, connects to a Macromedia server to obtain the component file, then moves the downloaded component to a designated location within the Flash Player installation. As this process relies on accurate DNS resolution, it is subject to the limitations of the DNS infrastructure as described above. Use of the Flash Player's component download process should be conducted in accordance with security best practices, which suggest exercising due caution when obtaining executable code from remote sources.

RTMP

RTMP, a streaming protocol designed to transmit data to the Macromedia Flash Communication Server MX, encapsulates media streams and data for transmission to the remote server. Macromedia recommends transmitting only information that is public and non-sensitive over RTMP. Please refer to the "Secure Design with Macromedia Flash Communication Server MX" document for further information about the Flash Player and the RTMP protocol.

Conclusion

The Flash Player provides a versatile platform for multimedia applications, with a feature set that includes security-conscious functionality. Developers with a basic understanding of the concepts represented in this document will be able to create applications that retain the intended security properties of the Flash Player platform. The information presented here will facilitate the introduction of the Flash Player into controlled environments, such as corporate intranets, by providing an understanding of the relationship and interactions among the Flash Player application, the Flash Player itself, the target platform, and the network.

About @stake, Inc.

@stake provides corporations with digital security services that secure critical infrastructure and electronic relationships. @stake applies industry expertise and pioneering research to design and build secure business solutions. As the first company to develop an empirical model measuring the Return On Security Investment (ROSI), @stake works where security and business intersect. Headquartered in Cambridge, MA, @stake has offices in Denver, Hamburg, London, New York, Raleigh, San Francisco, and Seattle. For more information, go to www.atstake.com.

Reproduction guidelines: you may make copies of this document unless otherwise noted. If you quote or reference this document, you must appropriately attribute the contents and authorship to @stake. Opinions presented in this document reflect judgment at the time of publication and are subject to change. While every precaution has been taken in the preparation of this document, @stake assumes no responsibility for errors, omissions, or damages resulting from the use of the information herein. Products or corporate names may be trademarks or registered trademarks of other companies and are used only for the explanation and to the owner's benefit, without intent to infringe.