EUROPEAN COMMISSION
DIRECTORATE-GENERAL
PERSONNEL AND ADMINISTRATION
Directorate DS - Security
Coordination and Informatics Security

Brussels, 21/06/2011
HR.DS5/GV/ac ARES (2011) 663475
SEC20.10.05/04 - Standards

European Commission

Information System Security Policy

C(2006) 3602

STANDARD ON MOBILE CODE

ADOPTED BY MRS. IRENE SOUKA,

DIRECTOR-GENERAL OF DG HUMAN RESOURCES AND SECURITY, ON 21/06/2011

TABLE OF CONTENTS

1. ADOPTION PROCEDURE

This Security Standard is adopted in accordance with Article 10(3) of Commission Decision C(2006) 3602 concerning the security of information systems used by the European Commission, adopted on 16 August 2006.

It is drawn up under the responsibility of the Security Directorate pursuant to Article 9(1)(b) and takes into account the items listed in Article 10(2) of Commission Decision C(2006)3602, in particular internationally recognised norms and standards applicable in the field of information systems security.

Under Article 10(3) of Commission Decision C(2006) 3602, the implementing rules may be supplemented by measures of a technical, physical, procedural or organisational nature proposed by the Director of the Security Directorate and adopted by the Director-General of the Directorate-General for Human Resources and Security in consultation with departments that have a legitimate interest. These supplementary measures are called 'security standards' where their application is mandatory, or 'security guidelines' where their application is optional or where they provide guidance on security standards implementation

2. INTRODUCTION

Mobile code is any code that may be downloaded and executed directly on an end user device, often without the knowledge or intervention of the user. Mobile code is sometimes capable of running on multiple platforms. There is a wide range of uses for mobile code, including legitimate uses such as provide rich user interfaces in web browsers, and malicious use such as spreading computer viruses. Consequently, measures must be in place to permit authorised use and block malicious code.

3. OBJECTIVES

The purpose of this standard is to provide measures that must be applied for the safe execution of legitimate imported code on the local systems, and to prevent them from exploiting systems weaknesses and spreading across the European Commission network.

4. SCOPE

This standard establishes policy on the implementation and use of mobile code in all European Commission Information systems. It covers all kinds of mobile code technologies with their potentially dangerous associated content.

5. THREATS COVERED

Security controls defined in this security standard will help to reduce the impact of the following threats (their description is in the *Standard on Information Security Risk Management*).

T23 – Disclosure

T24 – Data from untrustworthy sources

T26 – Tampering with software

T30 – Saturation of the information system

T31 – Software malfunction

T36 – Corruption of data

T39 – Abuse of rights

6. TERMINOLOGY

ActiveX: a set of interfaces from Microsoft that provide tools for linking desktop applications to the Web.

Applet: an application that has limited features, requires limited memory resources, and is usually portable between operating systems.

Certificate: an electronic text that is issued by a certification authority (CA) and establishes the credentials of the owner. It contains information such as the common name, a serial number, expiration dates, a copy of the certificate holder's public key (used for encrypting messages and electronic signatures), and the electronic signature of the certificate-issuing authority so that a recipient can verify that the certificate is real.

Code Signing: the process of digitally signing executables and scripts to confirm the software author and guarantee that the code has not been altered or corrupted since it was signed, by use of a cryptographic hash.

Flash: a multimedia platform used to add animation, video, and interactivity to Web pages. Flash is frequently used for advertisements and games. More recently, it has been positioned as a tool for the so-called "Rich Internet Application".

Java: a programming language designed to develop applications, especially ones for the Internet, that can operate on different platforms.

JavaScript: a general purpose, cross-platform scripting language available in most Web browsers. It can be embedded within standard Web pages to create interactive documents.

LotusScript: a dialect of the BASIC programming language, very similar to Visual Basic, that is used by Lotus Notes and other IBM Lotus Software products.

Mediated access: access to system resources subject to the control and approval of a runtime-enforced security policy, either during execution or at the beginning of execution.

Mobile agent: a combination of computer software and data that is able to migrate from one computer to another autonomously and continue its execution on the destination computer

Mobile code: software or any code that may be downloaded and executed directly on an end user device, often without the knowledge or intervention of the user, and is sometimes capable of running on multiple platforms. In other words, it is code sourced from remote, possibly untrusted systems, but executed on the local system. Mobile code has many synonyms, including mobile agents, downloadable code, executable content, or remote code. These mechanisms are often used to provide rich and dynamic content to users visiting web sites, or to automate common tasks.

Runtime environment: any execution environment, be it an application or part of the Operating System, which is installed on the end user device and can run mobile code. Runtime environments include but are not limited to: Microsoft .NET, the Java Runtime Environment (JRE), the Perl interpreter, and in the sense of this standard, also web browsers, Office Automation software, document viewers/readers, the Windows batch interpreter, and various Unix shells.

Script: a program or sequence of instructions that is interpreted or carried out by another program, instead of being compiled and executed by the computer processor.

Self signed certificate: a digital certificate that is signed by the creator of the certificate. In a Public Key Infrastructure, only the root Certificate Authorities possess self-signed certificates. Self-signed certificates confuse the issue of identity verification, as without further proof, anyone can claim to represent a particular entity.

Shockwave: a multimedia player program that allows Adobe Director applications to be published on the Internet and viewed in a web browser on any computer which has the Shockwave plug-in installed.

Silverlight: Microsoft Silverlight is an application framework for writing and running business and media applications (rich Internet applications), with features and purposes similar to those of Adobe Flash. The run-time environment for Silverlight is available as a plug-in for most web browsers.

Unmediated access: direct use of system resources, not subject to any approval or control by a runtime-enforced security policy beyond that imposed on conventional user applications.

VBScript: a programming language developed by Microsoft. It can be embedded in Web pages for viewing with Internet Explorer or run via the Windows Script Host. Browsers other than Internet Explorer do not support VBScript.

Visual Basic for Applications (VBA): an implementation of Microsoft's event-driven programming language Visual Basic, and associated integrated development environment (IDE), which is built into most Microsoft Office applications. VBA is often used to write macros that are contained in Microsoft Office documents and can contain malicious code.

7. BACKGROUND INFORMATION

7.1. Definition of mobile code

With the growth of distributed computer and telecommunications systems, there have been increasing demands to support the concept of mobile code, which is code sourced from remote (and possibly untrusted) systems but executed on the end user device (PC, laptop, PDA, smartphone etc).

Mobile code is any code that is transferred and subsequently executed directly on an end user device, with or without the knowledge or intervention of the user. Examples of mobile code include, but are not limited to:

- Applets on Internet web pages (Java Applets, JavaScript, VBScript, ActiveX controls, Shockwave or Flash animation…)
- HTML e-mails with active content
- Office documents containing macros
- PDF files containing active content
- Downloaded executable files

As an exception to the above, the following do not constitute mobile code in the sense of this standard:

- Code that only runs on an EC server but never on an EC end user device
- Office or PDF documents without macros or active contents[1]
- EC workstation and laptop logon scripts and centrally distributed applications, if deployed over a secured internal network channel

Mobile code is often used in the context of web applications or emails. Occasionally, other programmes, such as Office Automation Suites and Java Web Start, download mobile code as well.

Mobile code requires a "runtime environment" to be executed on the end user equipment. The definition of "runtime environment" in this standard is very broad; for details consult the terminology section of this standard.

[1] Malformed documents exploiting vulnerabilities such as buffer overflows are covered by the *Standard on Controls against Malicious Code*

7.2. Risks of mobile code

Under the guise of providing flexible services for legitimate programmers, some mobile code technologies allow the code to have full access to system resources and services on workstations, laptops, and mobile devices such as PDAs and smartphones. However, supporting mobile code introduces a number of serious security issues that must be addressed. Some of the dangers are:

- modification of user data
- modification of system files, the Windows Registry, or security settings
- installation of hidden malicious code that runs at a later time (e.g. Trojan Horses)
- e-mailing the malicious code to other users
- sending sensitive information

Mobile code technologies can serve as vehicle for security compromise. Some technologies allow the code to have full access to system resources and services on mobile devices, workstations and servers. They may perform activities such as deletion, modification or extraction of a user's data, or degrade the performance of a system in a way that could compromise its integrity or availability.

According to their potential for causing harm, mobile code technologies can be classed in two risk categories:

1) **High risk** mobile code has full functionality through unmediated access[2] to local and remote systems services and resources. This category of mobile code presents serious dangers since there are few or no countermeasures once access is obtained. This category of mobile code either executes with full access to all system services or does not execute at all.

   Examples of high risk mobile code technologies include binary executables, Active X, Java programmes, Windows Scripting Host scripts, and any form of shell or batch scripts.

2) **Standard risk** mobile code either has full functionality through mediated or controlled access, or limited functionality, with no unmediated access to local system services and resources. This category of mobile code can be harmful (due to malicious exploits).

   Examples of standard risk mobile code technologies include Java applets, Visual Basic for Applications, JavaScript, Shockwave, Flash, and Microsoft Silverlight.

[2] See section 6 for definitions of mediated and unmediated access.

7.3. Usage scenarios of mobile code

As well as the inherent risk in the technology employed, there are different usage scenarios for mobile code, and these scenarios require different security measures. The basic scenarios may be described as follows:

- EC mobile code – information systems owned and controlled by the European Commission may make use of mobile code for client interactions. While it may be assumed that these are not malicious[3], they must be protected from interference or replacement by malicious replicas.

- External non-malicious mobile code – users within the EC frequently execute external mobile code for legitimate reasons, such as accessing web-based systems. This is normal and acceptable use, and the security measures must be designed to allow it while ensuring that covert malicious activity does not occur.

- External malicious mobile code – this type of mobile code must be identified as soon as possible and prevented from executing. To guard against accidental execution of mobile code, security measures must be implemented to restrict the damage that this type of mobile code can cause.

The rules in this standard are intended to permit legitimate use and block malicious mobile code. To cover the scenarios described above, they are split into rules for information systems that provide mobile code ("server side"), and rules for end user devices that execute it ("client side").

8. SECURITY CONTROLS

5.4..2 In addition to ensuring that mobile code does not contain malicious code, mobile code usage shall be controlled to avoid unauthorised use or disruption of system, network, or application resources and other breaches of information security.

8.1. Introduction

The rules in this standard are split into two parts, dealing respectively with the server side and the client side of the mobile code architecture. The server side concerns information systems that are owned and/or controlled by the European Commission, and therefore the EC is the provider of mobile code (which may be executed by end users within or outside the EC). On this side, the rules focus on the integrity of the code provided to end users.

The client side concerns end user devices that are executing mobile code which may originate from EC systems or other sources (such as Internet

[3] This assumption is based on the premise that the rules in the standards relating to secure systems development have been followed in the selection or development of the system (see the *Standard on Secure Systems Development*).

sites, external e-mails etc.). The rules for the client side focus on preventing malicious mobile code from causing harm to EC systems or data.

8.2. Server side

This section provides rules for EC information systems that use mobile code as part of their architecture, such as web-based applications with a front-end applet that runs in a browser. The rules in this section are intended to ensure that this type of mobile code is protected from being compromised by changing the code that is sent to the user device. These rules must be applied at the server side of Commission information systems that use mobile code.

8.2.1. Choice of mobile code technologies

Whenever an information system that uses mobile code is acquired by or developed for use by the Commission, the related risks must be assessed and appropriate measures implemented to address these risks. The *Standard on Secure Systems Development* provides general rules for the development of information systems, but, in relation to mobile code, the following issues must be specifically addressed:

- Availability of code signing technologies. Preference should be given to technologies that support code signing using a Public Key Infrastructure.

- Identification of the technology used for the mobile code and its associated risk category (high or standard as described in section 6 above). If the technology has not yet been determined (e.g. for a system to be developed), preference should be given to lower risk technologies in order to reduce the potential vulnerability of the system.

- Specific threats to the confidentiality, integrity or availability of the system that could be exploited through the abuse or manipulation of the mobile code used by the system.

- Whether the mobile code will be executed on end user devices that are not controlled by the Commission (i.e. devices that may have a different security configuration)

- What type of information is handled by the mobile code, and how it is transmitted across the networks between the front and back ends of the system

According to the risk category of the technology used, the relevant security measures described below must be implemented.

8.2.2. Security Measures

The principal measure for ensuring the integrity of both high and standard risk EC mobile code is code signing. This enables the client to verify the authenticity of the code provided (see section 8.3 below). Code signing is not possible for all types of mobile code, but it should be used where possible.

If code signing is used, the mobile code must be signed before being installed on a server. Code should be signed using a certificate issued by an EC CA, although this can be replaced by commercial code signing if it is not technically possible. Mobile code should not rely on self-signed certificates, since an adversary could easily create a self-signed certificate that looks trustworthy.

Input validation must be performed on the server side for any data received from clients. Please refer to the *Standard on Secure Systems Development*.

If code signing is not used, standard risk mobile code may be used but the reasons must be documented and approved in the system's security plan.

High risk mobile code (as defined in section 7.2 above) may be used in the European Commission Information systems provided that the code is delivered to the client securely. A secure manner is considered to be one of the following, or an equivalent:

- An encrypted connection (e.g. SSL, TLS or IPSec) from a trusted server using a European Commission or trusted commercial server certificate; or

- Mobile code is signed with an EC code-signing certificate prior to being installed on the servers, enabling the client to check the integrity of the mobile code. In case EC Code Signing cannot be used on technical grounds, it can be replaced by commercial code signing.

8.3. Client side rules

This section provides rules for mobile code enabled software (including, but not limited to, web browsers, e-mail clients, office applications, and other runtime environments) residing on workstations, servers and mobile devices. The rules in this section are intended to ensure that the end user devices are protected from being compromised by malicious mobile code. These rules must be applied at the client side of Commission information systems that use mobile code.

8.3.1. Controls applicable for all mobile code categories

The rules in this section must be applied to all end user computing devices that may execute mobile code, such as workstations and mobile devices.

- The number of runtime environments to be installed on each EC end user devices should be minimised.

- All runtime environments on end user devices must be securely configured, including where possible:

    - Blocking the execution of high risk mobile code originating from outside the EC's networks

    - Minimising the installation of add-ons to runtime environments (such as browser toolbars)

    - End users must be prevented from changing the security configuration of their runtime environments, or from installing other runtime environments that can execute mobile code (such as alternative web browsers)

    - Patch management procedures for all installed runtime environments must be in place to ensure that these are updated as relevant (see the *Standard on Management of Technical Vulnerabilities*)

    - End user devices must not run beta or other pre-release versions of runtime environments. These may only be run on developers' machines when required.

- All end user devices must have up-to-date anti-malware software (see the *Standard on Controls against Malicious Code*).

- All end user devices should have controls to restrict the installation or execution of unauthorised software (often termed "Application lockdown"). See the *Standard on Controls against Malicious Code*.

- The user should execute at least one action before the mobile code runs on his device. This implies that automatic execution of mobile code in e-mail bodies and attachments should be disabled.

- Information systems that are owned and controlled by the European Commission shall be trusted sources.

- When technically possible[4], the runtime environments that download and execute mobile code shall be capable of accepting and trusting EC and/or other trusted commercial code-signing certificate authorities.

8.3.2. Additional rules for high risk mobile code

The use of unsigned high risk mobile code (as defined in section 7.2 above) is prohibited when it comes from outside the European Commission and is to

[4] E.g. for browsers, e-mail clients, and the Java Runtime Environment.

be executed on EC end user devices. Consequently, where possible, all high risk unsigned mobile code must be blocked at the network boundary. In addition, where possible, all European Commission end user devices and runtime environments capable of executing mobile code must be configured to disable the execution of high risk mobile code obtained from outside the EC's networks.

High risk mobile code obtained from a trusted source shall be accepted for use within the European Commission Information systems provided that the code is delivered to the client securely using one of the methods described in section 8.2.2 above.

Web browsers and other mobile code enabled products must be configured to prompt the user for agreement prior to the execution of high risk mobile code[5].

9. REFERENCES

Note that standards marked (*) are in draft at the time of writing of this standard.

- Commission Decision (2001/844/EC, ECSC, Euratom) of 29/11/2001

- Commission Decision C(2006) 3602 of 16/8/2006

- Implementing rules for Commission Decision C(2006) 3602 of 16.8.2006

- Standard on Information Security Risk Management (*)

- Standard on Controls against Malicious Code (*)

- Standard on Secure Systems Development (*)

- Standard on Management of Technical Vulnerabilities (*)

10. RELATED DOCUMENTS

- International standard ISO/IEC 27001 – Second edition 2005-06-15

- International standard ISO/IEC 17799 – Second edition 2005-06-15

- NIST Special Publication 800-28 Version 2 – Guidelines on Active Content and Mobile Code

- Model-Carrying Code (MCC): A new paradigm for Mobile-Code Security; Department of Computer Science SUNY at Stony Brook, NY11794

[5] This prompt is different from the prompt that some technologies show (e.g. Java applets) before executing signed code.

- SANS white paper: Risk of Java Applets and Microsoft ActiveX Controls by Jennifer M. Marek, March 2002