Chapter 7.14
Mobile Code and Security Issues

E. S. Samundeeswari
Vellalar College for Women, India

F. Mary Magdalene Jane
P. S. G. R. Krishnammal, India

ABSTRACT

Over the years, computer systems have evolved from centralized monolithic computing devices supporting static applications, into client-server environments that allow complex forms of distributed computing. Throughout this evolution, limited forms of code mobility have existed. The explosion in the use of the World Wide Web, coupled with the rapid evolution of the platform-independent programming languages, has promoted the use of mobile code and, at the same time, raised some important security issues. This chapter introduces mobile code technology and discusses the related security issues. The first part of the chapter deals with the need for mobile codes and the various methods of categorising them. One method of categorising the mobile code is based on code mobility. Different forms of code mobility, like code on demand, remote evaluation, and mobile agents, are explained in detail. The other method is based on the type of code distributed. Various types of codes, like source code, intermediate code, platform-dependent binary code, and just-in-time compilation, are explained. Mobile agents, as autonomously migrating software entities, present great challenges to the design and implementation of security mechanisms. The second part of this chapter deals with the security issues. These issues are broadly divided into code-related issues and host-related issues. Techniques, like sandboxing, code signing, and proof-carrying code, are widely applied to protect the hosts. Execution tracing, mobile cryptography, obfuscated code, and cooperating agents are used to protect the code from harmful agents. The security mechanisms, like language support for safety, OS level security, and safety policies, are discussed in the last section. In order to make the mobile code approach practical, it is essential to understand mobile code technology. Advanced and innovative solutions are to be developed to restrict the operations that mobile code can perform, but without unduly restricting its functionality. It is also necessary to develop formal, extremely easy-to-use safety measures.

INTRODUCTION

Mobile code computation is a new paradigm for structuring distributed systems. Mobile programs migrate from remote sites to a host, and interact with the resources and facilities local to that host. This new mode of distributed computation promises great opportunities for electronic commerce, mobile computing, and information harvesting. There has been a general consensus that security is the key to the success of mobile code computation.

Distributed applications involve the coordination of two or more computers geographically apart and connected by a physical network. Most distributed applications deploy the client/server paradigm. There are certain problems with the client/server paradigm, such as the requirement of a high-network bandwidth and continuous user-computer interactivity. Hence, the mobile code paradigm has been developed as an alternative approach for distributed application design.

In the client/server paradigm, programs cannot move across different machines and must run on the machines they reside on. The mobile-code paradigm, on the other hand, allows programs to be transferred among, and executed on, different computers. By allowing code to move between hosts, programs can interact on the same computer instead of over the network. Therefore, communication cost can be reduced. Besides, one form of mobile code is a program that can be designed to work on behalf of users autonomously. This autonomy allows users to delegate their tasks to the mobile code, and not to stay continuously in front of the computer terminal.

With the growth of distributed computer and telecommunications systems, there have been increasing demands to support the concept of ”mobile code,” sourced from remote, possibly untrustworthy systems, but executed locally.

MOBILE CODE

Mobile code consists of small pieces of software obtained from remote systems outside the enclave boundary, transferred across a network, and then downloaded and executed on a local system without explicit installation or execution by the recipient.

The mobile-code paradigm encompasses programs that can be executed on one or several hosts other than the one that they originate from. Mobility of such programs implies some built-in capability for each piece of code to travel smoothly from one host to another. A mobile code is associated with at least two parties: its producer and its consumer, the consumer being the host that runs the code.

Examples of mobile code include a Java script embedded within an HTML page, a visual basic script contained in a WORD document, an HTML help file, an ActiveX Control, a Java applet, a transparent browser plug-in or DLL, a new document viewer installed on demand, an explicitly downloaded executable binary, and so forth. Since mobile code runs in the execution context of the user that downloads the code, it can issue any system calls that the user is allowed to make, including deleting files, modifying configurations or registry entries, ending e-mails, or installing back-door programs in the home directory. The most common type of malicious mobile code is an e-mail attachment.

Mobile-code systems range from simple applets to intelligent software agents. These systems offer several advantages over the more traditional distributed computing approaches, like flexibility in software design beyond the well-established object-oriented paradigm and bandwidth optimization. As usual, increased flexibility comes with a cost, which is increased vulnerability in the face of malicious intrusion scenarios akin to Internet. Possible vulnerabilities with mobile code fall in one of two categories: attacks performed