Covert Distributed Computing Using Java Through Web Spoofing

Jeffrey Horton and Jennifer Seberry

Centre for Computer Security Research
School of Information Technology and Computer Science
University of Wollongong
Northfields Avenue, Wollongong
{jeffh, j.seberry}@cs.uow.edu.au

Abstract. We use the Web Spoofing attack reported by Cohen and also the Secure Internet Programming Group at Princeton University to give a new method of achieving covert distributed computing with Java. We show how Java applets that perform a distributed computation can be inserted into vulnerable Web pages. This has the added feature that users can rejoin a computation at some later date through bookmarks made while the pages previously viewed were spoofed. Few signs of anything unusual can be observed. Users need not *knowingly* revisit a particular Web page to be victims.
We also propose a simple countermeasure against such a spoofing attack, which would be useful to help users detect the presence of Web Spoofing. Finally, we introduce the idea of browser users, as clients of Web-based services provided by third parties, "paying" for these services by running a distributed computation applet for a short period of time.

1 Introduction

There are many problems in computer science which may be solved most easily through the application of brute force. An example of such a problem is determination of the key used to encrypt a block of data with an algorithm such as DES (Data Encryption Standard). The computer time required could be obtained with the full knowledge and cooperation of the individuals controlling the resources, or covertly without their knowledge by some means. A past suggestion for the covert accomplishment of tasks such as this involved the use of computer viruses to perform distributed computations [1].

Java is a general purpose object-oriented programming language introduced in 1995 by Sun Microsystems. It is similar in many ways to C and C++. Programs written in Java may be compiled to a platform-independent bytecode which can be executed on any platform to which the Java runtime system has been ported; the Java bytecodes are commonly simply interpreted, however speed of execution of Java programs can be improved by using a runtime system which translates the bytecodes into native machine instructions at execution time. Such systems, incorporating these Just-in-time (JIT) compilers, are becoming more

common. The Java system includes support for easy use of multiple threads of execution, and network communication at a low level using sockets, or a high level using URL objects [2].

One of the major uses seen so far for Java is the creation of applets to provide executable content for HTML pages on the World Wide Web. Common Web browsers such as Netscape Navigator and Microsoft Internet Explorer include support for downloading and executing Java applets. There are various security restrictions imposed upon applets that are intended to make it safer for users to execute applets from unknown sources on their computers. One such restriction is that applets are usually only allowed to open a network connection to the host from which the applet was downloaded. A number of problems with Java security have been discovered by various researchers [5] [7].

Java could also be applied to performing a distributed computation. Java's straightforward support for networking and multiple threads of execution make construction of an applet to perform the computing tasks simple. The possibility of using Java applets to covertly or otherwise perform a distributed computation is discussed by several researchers [4] [5] [7, pp. 112–114] [8].

There have been no suggestions, however, as to how this might be accomplished without requiring browser users to knowingly visit a particular page or Web server at the beginning or sometime during the course of each session with their Web browser, so that the applet responsible for performing the computation can be loaded. This paper describes how the Web spoofing idea described by the Secure Internet Programming Group at Princeton University can be used to pass a Java applet to perform a distributed computation to a client. The advantage is that clients do not have to *knowingly* (re)visit a particular site each time, but may rejoin the computation through bookmarks made during a previous session.

There will be some indications visible in the browser when rejoining a computation through a bookmark that the user has not reached the site they may have been expecting; however, these signs are small, and the authors believe, mostly correctable using the same techniques as employed in a vanilla Web Spoofing attack.

2 About Web Spoofing

Web Spoofing was first described briefly by Cohen [3]. The Web Spoofing attack was later discussed in greater detail and elaborated upon by the Secure Internet Programming Group at Princeton University [6]. Among other contributions, the Princeton group introduced the use of JavaScript for the purposes of concealing the operation of the Web Spoofing attack and preventing the browser user from escaping from the spoofed context. JavaScript is a scripting language that is supported by some common Web browsers. JavaScript programs may be embedded in an HTML page, and may be executed when the HTML page is loaded by the browser, or when certain events occur, such as the browser user holding the mouse pointer over a hyperlink on the page.