6.915 Computer and Network Security **Fall Term, 1995**

Lecture 23 : November 30, 1995

Lecturer: Ron Rivest *Scribe: Greg Pal*

1 Agenda

There were 4 Handouts:

- 2 problem set solutions (PS 6 and PS 7)
- 2 readings (Network Firewalls, Security Flaws in the HotJava Web Browser)

Today we will talk about

- The Web and Web Browsers
- Java and HotJava
- Security in HotJava

2 Web Development

During the last two years, the World Wide Web had grown into an immense network of documents scattered across the Internet. Written in the Hypertext Markup Language (HTML), documents on the Web integrate text and images to present information to Web surfers. Over time, though, site designers have demanded more features in their pages. Instead of modifying HTML to accommodate each of these demands, Sun Microsystems developed the Java programming language to allow developers to dynamically expand the capabilities of Web browsers.

2.1 Web Browsers

There are currently two different browsers which support "applets" written in the Java programming language, HotJava and Netscape Navigator v2. The HotJava browser

was developed by Sun Microsystems, while the Netscape Navigator was developed by Netscape Communications Corp.

3 Java and HotJava

Sun Microsystems originally developed the Java programming language for set-top boxes. With the advent of the World Wide Web, they modified the language to support the execution of code on Web clients. In many ways, Java is similar to the C++ programming language. Both are object-oriented and support classes, objects, inheritance, and modularity. Java simplifies many "troubling" aspects of C++, though. For example, Java does not support pointers and is much more type-safe than C++. Java is also multi-threaded and uses different name spaces to resolve names. For a complete description of the Java programming language, please refer to The Java Language Environment and The Java Language Specification.

Java's popularity has grown so much that the December 4, 1995 issue of Business Week magazine has a feature article on the software revolution that Java may cause. The following is an excerpt from their article:

> The computer (Java interpreter) is a "virtual machine," a piece of software that lets any computer simulate, perfectly, an ideal and standardized Java computer. As far as any Java program knows, this simulation is the genuine article - a complete computer with disk drive, display, memory, and everything else. This virtual machine executes Java programs by "interpreting" their commands one by one and commanding the underlying computer to perform all tasks needed to write text to a screen or pull data from a disk say. There's a small penalty to be paid in performance because of this interpretation step. But the two-step procedure also helps: No Java program is allowed to really penetrate your computer, so you can be reasonably certain it won't unleash viruses to infect your system.

When a person writes Java code, they compile their textual source code into an intermediate Java "byte code". This byte code is then transmitted over the Web to a client computer where it is interpreted. The fact that Java is interpreted helps accomplish the following purposes:

- verification of untrusted code downloaded over the Internet

- portability of Java byte code
- protection of source against decompilation
- efficiency (compared with transmitting source code and performing the entire compile on the client)

HotJava is a web browser which is implemented using the Java programming language.

3.1 Security in HotJava

What are we worried about:

- Confidentiality: We do not want an applet to be able to steal information from the client and transmit it to another party.
- Integrity: We do not want an applet to be able to corrupt the local file system.
- Availability: An applet should not be able to "hog" system resources and starve out other client processes.
- Denial of service: An multi-threaded applet should not be able to deadlock and freeze the client machine.

Attacks: Drew Dean and Dan S. Wallach of Princeton University wrote a paper entitled Security Flaws in the HotJava Web Browser in which they investigate different attacks that can be mounted against the HotJava browser. It should be noted that Dean and Wallach had access to the source code for HotJava when they designed these attacks.

Covert Channels: Using the accept() system call, an applet can permit other hosts to create covert channels to a client computer. Given this system call, the only problem that remains is how to notify the foreign host that the client is accessible. The authors devised three methods to create this connection. To quote the authors: "If the Web server is running an SMTP mail daemon, the applet can connect to it and transmit an e-mail message to any machine on the net. Additionally, the Domain Name System (DNS) can be used as a two-way communications channel [and] ... If the Web server is cooperating, the ability for an applet to retrieve a URL is another covert channel. The format of a URL is protocol://hostname/server-defined, so any arbitrary information can be encoded in a URL."

Information Available to Applets: Given the ability to create a covert channel between a host and the client, what interesting information could be revealed about the client? (To be perfectly honest, I don't think any of this information is particularly interesting) Basically, the only client information that an applet can capture is the user's environment information using the System.getenv() system call.

Denial of Service In figure 2 of their paper, the authors demonstrate a piece of Java code which can lock the status line of the HotJava browser and prevent further pages from being retrieved.

4 Concluding Notes

In the remainder of their paper, the authors explore the conflict between delivering powerful applications over the World Wide Web and the security concerns that these applications introduce.