

Do Java Users Live Dangerously?

Java is giving researchers a dose of excitement by allowing them to visualize or analyze data with software commandeered from distant machines (see main text). It is also fueling visions of a bustling Internet economy. This operating system, which makes it possible to download foreign programs over the Internet and run them in a local computer, will allow users to invite electronic shopkeepers into their home machines to conduct transactions. But these very attributes are also giving security experts the jitters.

The ability to download and run foreign programs—applets, in Java-speak—could in theory expose a host machine to computer viruses and other digital mischief. Java's designers were well aware of this vulnerability, so they included an array of software guardians that screen each applet for admission and then keep it in what Java engineers call a sandbox, where it can't run amok. But several computer experts have learned to use bugs in Java to bypass its safeguards. Each new security breach has made the vision of a bustling Internet economy seem more distant and even raised concerns about other uses of Java. They have also rattled staff at JavaSoft.



Finding fault. Oxford's David Hopwood.

"I've named gray hairs on my head after [the attacks]," says Marianne Mueller, a staff engineer at the company.

The latest Java bug was found by David Hopwood, a student at Oxford University. On 1 June, he announced on the Web that he had created an applet capable of undermining Java's security safeguards. "He found a bug—a subtle bug," agrees Mueller. Hopwood's subtle bug exposes the "security manager" at Java's front door to attack. The security manager acts as a gatekeeper. Whenever an applet tries to do something potentially mischievous, like renaming files or creating new directories on the hard drive, the security manager slams the door. But Hopwood wrote a program that, when downloaded by a Java host, kills the security manager and replaces it with an impostor—a phony who dozes at the gate, leaving it open for later invaders.

To make matters worse, three Princeton University researchers had already found a bug that enables a rogue applet to escape from the sandbox. "If you combine the two attacks," says Hopwood, "you can run any code"—even code that tells your computer to send bank records to Taiwan or to erase your hard drive. "It's a vicious attack," agrees Drew Dean, a member of the Princeton team.

"I'm feeling kind of bloodied," admits Mueller. JavaSoft and Netscape, which developed the World Wide Web browser on which Java runs, are both working to patch the holes. But others keep cropping up, which is in part a reflection of Java's power. "There's a tension between being secure and doing interesting things," says Mueller. "Often, we're between a rock and a hard place." Security experts also blame a hurried production schedule. "Overall, companies are racing too rapidly to add new features" to software including Java, says Edward Felten, head of the Java research effort at Princeton. "And new code means new bugs." But he notes that JavaSoft is working hard to identify any new vulnerabilities—even going as far as funding the Princeton group's efforts.

So far, the attacks seem to be limited to laboratory exercises. Although rumors of Java viruses are rife, Hopwood calls them "hype. Sensationalist hype." Even so, Hopwood, Dean, and Felten all disable Java and Javascript on their browsers when wandering through the Internet.

Although only the computer experts are nervous now, consumers might have reason to worry in the future. If Java becomes the backbone of a new virtual marketplace, applets will interact with shoppers on their home computers. This means that Java applets will have to be able to accept money—and hostile applets could easily eavesdrop on transactions and skim off some of the proceeds. In that case, "[a Java-based attack] would be a good way to steal money a little at a time from a lot of people," says Dean. "It's all fun and games until there's real money involved."

—Charles Seife

Charles Seife is a free-lance science writer in Scarsdale, New York.

guage, or HTML, and as languages go it's pretty static. "HTML basically works with forms, like documents. There are whole classes of things that you can't do with it," Jamison says. "You can highlight a chunk of sequence, for instance, but to do something with it, you have to fill in its parameters on a form. Then the server does the processing and ships it back to you." To focus on a smaller chunk of the sequence, you have to fill in more parameters and resubmit. "HTML is back and forth, back and forth, every time you want to add a term. Or maybe you want to rotate a protein view or loosen a folding bond. HTML can't do that for you."

Some genes with your Java? But the Java language can, to a certain extent. "What Java does is move the code over to the client," says Gregg Helt, a biologist in the University of California, Berkeley's, *Drosophila* Genome Project and author of the Java-based *Drosophila* Genome Browser prototype. "And that

makes things more dynamic." Java applets are embedded in HTML pages, the standard form for displaying information on the Web, but they run on the user's machine and pull data off the server they came from to reshape it at the user's command. The user doesn't need to go back to the original server for each operation. So, says Helt, you get the advantages of the Web's universal access and Java's facile visualization and interaction.

Actually, Helt says, "I was opposed to Java at first" because its graphics seemed limited and he worried about bugs. "I set out to prove it didn't work. What I found was that, hey, it works pretty well." It was good enough, in fact, for him to put the genome browser on the Web earlier this year. It shows a 3-megabase region of the *Drosophila* genome as a physical map, with chromosome bands. Users can zoom in on a band to see the subbands and still finer landmarks known as contigs and P1s. "You can't do that with HTML,"

notes Helt. "It would have to go to the server and get another picture file."

Then, says Helt, you can unpack and analyze the information hidden in each of those features. "Click on the P1, and you get a window with an annotated map of that sequence: BLAST homologies, gene predictions, and known GenBank entries. You can get down to the DNA level, and for a P1 that's 80,000 base pairs. You get a DNA viewer in a pop-up window, and you can select features in the annotated map and the viewer will move to them." And once users have selected a gene, another window will let them see what cells in the early fruit fly embryo express it.

Helt has also included a primer-prediction tool for scientists who want to amplify a particular segment with the polymerase chain reaction. A window lets a user specify primer length, temperature, and other options; the tool then recommends primer sequences, and arrows appear on the map to show where they