Have Java's Security Issues Gotten out of Hand?

Lee Garber

In the past year, security experts have found many vulnerabilities, some critical, in Java. This represents a serious trend because Java is so widely used.

Java is among the world's most popular software platforms. It offers the ability to develop and run applications that can be used across operating systems.

However, some experts say Java is now also a major security risk. Researchers have found many flaws that enable hackers to bypass security measures, take over computing systems, steal data, and cause other problems.

"They've found dozens in 2012 and dozens in 2011," said Atif Mushtaq, senior staff scientist with security vendor FireEye.

One recent vulnerability affects every version of Java issued since 2004, including those used in most of today's smartphones. To make matters worse, online exploit collections for hackers and penetration-testing applications for security professionals now include the code for easy-to-implement Java attacks, as the sidebar "Exploit Toolkits" explains.

Many experts say that if you don't absolutely need Java, uninstall it; if you do need it for some applications, take precautions.

In response to Java's problems, Apple decided to stop bundling it with OS X and has taken other protective measures. The Mozilla Foundation has blocked older, unpatched Java plug-ins from running on its Firefox browser.

As the "Java Backgrounder" sidebar discusses, Oracle, gained control of Java when it merged with Sun Microsystems in 2010. The company did not respond to multiple requests for comment.

SPATE OF PROBLEMS

Over time, Java has faced many security problems, which have gained a higher profile this year.

Researchers are discovering unique vulnerabilities throughout the code base that exist only in the Java language and the Java virtual machine (JVM), said Dan Guido, CEO of security vendor Trail of Bits. Because these vulnerabilities are specific to Java, traditional exploit mitigations don't help, he noted.

Security Explorations, a Polish consultancy, has been particularly active in investigating Java problems. It analyzed binary and source code for Java implementations from Oracle, IBM and Apple, and found multiple critical security vulnerabilities, said company CEO Adam Gowdiak.

Just since 2 April 2012, Security Explorations has uncovered many flaws, including critical sandbox bypasses that let hackers attack and gain control of systems.

"We found 50 security issues in total: 31 were reported to Oracle, including 17 complete JVM sandbox-bypass exploits;" Gowdiak said, "two were reported to Apple, including one sandbox-bypass exploit; and 17 were reported to IBM, including 10 sandbox bypass exploits."

Oracle has been eliminating the recent Java problems this year and says it will fix the last two in February 2013, according to Gowdiak. He noted that hackers have exploited some of the Oracle and Apple flaws but none of the IBM vulnerabilities.

IBM, which didn't respond to requests for comment, plans to release patches this month, Gowdiak said. Apple, which also didn't respond to requests for information, has been working on patches, too.

Published by the IEEE Computer Society

0018-9162/12/$31.00 © 2012 IEEE

"Most of the bugs have their origin in insecure implementation of Java code," Gowdiak explained. "That naturally makes them platform independent. This means that the vulnerabilities could be easily exploited on all operating system platforms supporting vulnerable Java SE versions." They could also work on both 32- and 64-bit systems.

Two flaws are particularly troubling and exemplify the recent Java security issues.

Java 5, 6, and 7 flaws

Security Explorations found and showed how to exploit a sandbox-bypass flaw in Java Standard Edition 5, 6, and 7, which are all of the versions issued since 2004 and which are found in most smartphones. This problem puts hundreds of millions of Java users at risk. The company tested the flaw on the Chrome, Firefox, Internet Explorer, Opera, and Safari browsers running on a fully patched 32-bit computer running Windows 7.

"This is bigger than any previous issue we found as part of our Java security research project," Gowdiak said. The flaw lets attackers violate the JVM's type-safety security, which is supposed to keep code from running on parts of a system it isn't authorized to access.

Such a problem, Gowdiak explained, "can lead to unrestricted access to Java classes, their fields and methods, [regardless] of Java security access controls. In the most obvious attack scenario, such a condition could lead to full privilege elevation."

A hacker could set the global value of Java's SecurityManager to null, disabling all security checks in the target JVM and enabling the hacker to cause problems such as installing malware or backdoors, as well as stealing, changing, or deleting data.

Gowdiak said that he provided Oracle with detailed information about the vulnerability and that it hasn't been exploited in the wild yet except for one zero-day attack in August.

Java 7 vulnerability

Several security investigators found another critical problem affecting Java SE 7, which hundreds of hackers have taken advantage of. Security researchers traced one of the early attacks to servers in China.

Researchers said the Java 7 vulnerability is highly exploitable, is not easily detectable by security software, and doesn't have effects that users would readily notice.

"It's not a bug, it's a design flaw," said FireEye's Mushtaq.

The Java Runtime Environment (JRE) vulnerability in Java 7 let hackers execute arbitrary code via an applet that exploits the flaw. "The applet must be specially crafted and use the vulnerable function in a certain way," noted Mushtaq.

Attackers use the Java 7 flaw to arbitrarily change the software's security settings, allowing malware to read, write, and execute code on an infected system.

"This method lets hackers obtain privileged references to private fields of arbitrary classes," Gowdiak explained. "This means attackers could obtain references to any field from the Java SE class environment and also set its value to an arbitrary value."

In essence, Mushtaq noted, this lets hackers change security settings and bypass Java's SecurityManager restrictions, enabling them to run code with full privileges. "It gives

EXPLOIT TOOLKITS

A factor that has made Java flaws more dangerous in recent years has been the inclusion by malicious groups and security developers of ready-to-launch attacks in exploit kits for hackers and penetration-testing kits for security professionals.

Major Java attacks have been found in kits such as Metasploit, Blackhole, VulnDisco SA Canvas, Eleonore, and Crimepack.

Security developers and hacker groups generally sell the kits online, which make the attacks available in an easy-to-implement form to a large number of people, including unskilled hackers who otherwise couldn't exploit the Java flaws, noted Gary McGraw, chief technology officer of software-security consultancy Cigital.

The incorporation of Java problems into exploit kits started in the third quarter of 2010, said Dan Guido, CEO of security vendor Trail of Bits. A recent company study of the 15 most popular toolkits showed that 11 had at least one Java exploit and two-thirds had at least two Java exploits. Now, Guido said, "almost all exploits kits available include at least one recent Java exploit."

Atif Mushtaq, senior staff scientist with security vendor FireEye, said, "We're seeing hundreds of groups exploiting this vulnerability. I guess people are not really updating their software."

JAVA BACKGROUNDER

Sun Microsystems began work on Java as an internal project in 1990 and released it publicly in 1995. Oracle acquired Java when it merged with Sun in 2010.

Java enables the development and deployment of applications on multiple computing platforms. The key is the use of a Java virtual machine (JVM)—unique to each platform—that executes the same set of Java bytecode on all supported systems. The technology is used in many environments, such as embedded and mobile devices, PCs, enterprise servers, websites, and even supercomputers.

Java includes various forms of security, including cryptography, authentication, authorization, and a public-key infrastructure. A key part of the security model is use of a sandbox to keep code from escaping the Java environment and affecting the rest of the host system.

On its website, Oracle says that Java runs on 850 million PCs and billions of other devices worldwide.

Numerous companies—including Apple, Hewlett-Packard, and IBM—have developed their own proprietary JVMs. There are also many open source JVMs.

the hacker access to a system. The first group of hackers used the flaw to deliver the Poison Ivy remote-access toolkit to infected systems. Now, hackers could use it to install and execute malware on the system." They could also steal information.

Investigators have identified phishing campaigns with e-mails—purportedly from major companies—that contain links that direct users to websites that launch the Java 7 exploits.

The SANS Institute, a security research and education organization, found an exploit that used a fake Microsoft e-mail—built with the template of a real message from the company—about an actual change to Microsoft's terms of service. The e-mail included a link that sent victims to a compromised website, which took advantage of the Java 7 vulnerability to deliver the Zeus Trojan. Zeus steals data from a victim's computer.

Another attack, which security vendor Websense uncovered, uses what is purportedly an e-mail from Amazon. The message asks the user to click on a link to verify an order. Clicking on the link takes the victim to a webpage containing a Java exploit. So far, researchers have found exploits on about 100 websites.

Security Explorations privately notified Oracle of the problem in April 2012, and FireEye did the same in August. However, Oracle reportedly didn't issue a patch—which also addressed two other Java issues—until 30 August, outside its normal quarterly cycle of Java-related fixes.

Gowdiak said his team investigated the patch and found that it created yet another security issue that enabled exploitation of some bugs that Oracle hadn't addressed yet.

BEHIND THE PROBLEMS

A study by security vendor Qualys based on its BrowserCheck application—which scans browsers and plug-ins for problems—showed that about 80 percent of the computers analyzed have Java enabled and 40 percent run versions with critical vulnerabilities. According to Qualys, this makes it the most vulnerable browser plug-in.

Referring to the sandbox-related and other Java problems that have cropped up this year, said Gary McGraw, chief technology officer of software-security consultancy Cigital."They look exactly like the Java vulnerabilities from 10 years ago. It's déjà vu all over again."

Java lets untrusted code run and assumes its sandbox will contain any problems, McGraw said. "This assumes [Java's] sandbox is working properly and doesn't have implementation errors. But it's had a history of implementation errors."

Some experts say that Java now poses a major security risk for users.

"Java is beginning to show its age," he stated. "There are many newer platforms that might be better from a security perspective, such as Ruby on Rails, HTML5, and .NET."

Java is a good target for hackers because it is on so many computers, McGraw said. The software has come bundled with some operating systems, and many people have downloaded it, he explained. In addition, it works across platforms, so it is on many different types of machines.

Hackers look to attack a large, widely deployed code base like Java, noted FireEye's Mushtaq. Java has been around a long time and past flaws may still be in systems, at least in legacy code and older versions.

Some organizations and individuals use applications requiring older Java versions, which, if unpatched, could contain flaws. Even after updates, vulnerable legacy versions of the JRE usually remain on a computer.

Meanwhile, some users don't patch their software regularly, even when they could implement automatic updates. Others aren't even aware that their browsers have Java plug-ins enabled by default.

And now that vendors have made OSs and browsers safer, hackers are increasingly attacking browser plug-ins.

WHAT LIES AHEAD

To cope with possible problems, Mushtaq said, "If you don't need Java, you should just disable it or simply uninstall it."

Many programs for which Java is supposedly required will run as well or almost as well without the technology. However, disabling or deleting Java isn't convenient for all users. For example, Adobe's popular Creative Suite of graphic design, video editing, and Web development applications have required users to run a JRE.

Organizations that need the technology could utilize one browser that has Java disabled or deleted for general browsing and a second with Java enabled for websites that require it.

"Java wasn't really designed for modern [security] guidelines," Mushtaq said. "If Oracle wants its software to be popular, it needs to improve its code and design process. They need to put more effort into security. Now, they're putting more effort into features."

McGraw said, "It's not clear to me that Oracle's going to invest in pushing Java ahead. I hope they do. This latest round of problems has certainly shone the spotlight on Java in a negative way. Sometimes that spurs companies to do a better job." For Java to be more secure, he stated, they have to build security in from the beginning of the development process, which hasn't always been done in the past.

"Oracle needs to learn from Microsoft how to handle vulnerabilities," Mushtaq said. "A quarterly patch cycle is inadequate. I hope that they've learned their lesson."

With few exceptions, Oracle rolls out Java patches several times a year. For example, next year, the patches are slated for 19 February, 18 June, and 15 October.

In a dynamically changing security world, Oracle should issue patches more often and fix serious problems right away, said Security Exploration's Gowdiak. The failure to do so in the past, he contended, "speaks for itself."

Trail of Bits' Guido agreed that Oracle's response to Java problems has been inadequate. "There is not much they can do to remove all of these vulnerabilities from their codebase," he said. "They need to shift to patching quickly to drive down the value an attacker can derive from exploiting these flaws in the wild. Their quarterly patch schedule creates an enormous window of opportunity for an attacker."

Also, Gowdiak noted, "They could invest resources in developing a real sandbox, one that operates at a lower level of privilege so that when the JVM is exploited, attackers are still not in the position to do anything malicious to the exploited computer."

According to Gowdiak, "The education of software engineers should help decrease the number of new bugs introduced to the code. Engineers need to be aware of common Java security pitfalls. Code-review efforts should help catch security bugs prior to final product release."

Guido said Java might not present such a security problem in the future because browser vendors are increasingly moving to protect their users and preventing Java and other plug-ins from running without users enabling it. "This will decrease the rates of success of attackers using exploits for Java and force them to move to other vectors to achieve the same effect," he stated.

FireEye's Mushtaq, on the other hand, said "Overall, I'm not really optimistic about the future of Java in the browser because of its vulnerabilities." And users don't need it because websites don't use it much any more.

The key to Java becoming safer, Cigital's McGraw said, is whether people care enough about security to push for it. He said he was pessimistic, adding, "I think we can make a prediction that [Java security] will [still] be a major issue in 2028."

Lee Garber is the IEEE Computer Society's senior news editor. Contact him at lgarber@computer.org.