

# [NT] Technical Information on Un-patched MS Java Vulnerabilities

*Source:* <http://www.derkeiler.com/Mailing-Lists/Securiteam/2002-11/0026.html>

---

*From:* [support@securiteam.com](mailto:support@securiteam.com)

*Date:* 11/10/02

From: [support@securiteam.com](mailto:support@securiteam.com)

To: [list@securiteam.com](mailto:list@securiteam.com)

Date: 10 Nov 2002 16:18:07 +0200

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.securiteam.com>

-- promotion

Beyond Security would like to welcome Tiscali World Online to our service provider team.

For more info on their service offering IP-Secure, please visit [http://www.worldonline.co.za/services/work\\_ip.asp](http://www.worldonline.co.za/services/work_ip.asp)

-----

Technical Information on Un-patched MS Java Vulnerabilities

---

## SUMMARY

These are some technical details about the security vulnerabilities Jouko has found in Microsoft's Java implementation. They were reported to the vendor mostly during August 2002.

## DETAILS

There were more than 10 vulnerabilities found in the Microsoft's Java implementation. The vendor has published a bulletin and patch addressing four of them (without mentioning the source of the information though). The rest is listed here. Technical details of the four already patched vulnerabilities were published in his previous message to BugTraq.

This list contains a brief explanation and enough information for system administrators, security professionals, and IE users to confirm the existence of the flaws and determine if their software is vulnerable. This requires some knowledge about Java; no exploit code is disclosed here. The impact of some of these issues isn't known as they would require more investigation and co-operation with the vendor.

## Securiteam: [NT] Technical Information on Un-patched MS Java Vulnerabilities

These issues were also reported to Sun Microsystems; their Java implementation appears to be unaffected.

### 1) URL parsing error

Impact: Impersonating a web site, cookie theft

Java code parses URLs wrong if they contain a colon used to indicate the port number. E.g. directing the user to the URL

80@www.bank.com/bankapplet.html">http://www.evilsite.com:80@www.bank.com/bankapplet.html causes the browser

to load the web page from www.bank.com, but due to the error, the Java engine loads the applet code from a wrong site, www.evilsite.com. This can be exploited at least to steal cookies related to www.bank.com if the applet tag on www.bank.com contains the MAYSCRIPT keyword (via netscape.javascript.\*). The attack requires that a Java applet exists on a web page on www.bank.com.

### 2) Stack overflow in class loader

Impact: Most likely only DoS

An overflow happens when a class with a long name is attempted to load. This can happen with e.g. Class.forName() or ClassLoader.loadClass(). This results in the browser crashing. It looks unlikely that this could be exploited to run a shellcode.

### 3) File path discovery

Impact: Finding out the current directory and username

Due to insufficient security checks any Java applet may find out the current directory of the Internet Explorer process by doing new File(".").getAbsolutePath(). This usually gives the desktop path which includes the username on multi-user operating systems. All local file access is supposed to be denied from untrusted applets. The information retrieved in this way may be used in conjunction with other vulnerabilities.

### 4) INativeServices memory access

Impact: Reading memory space, may lead to delivery and execution of any code

Any applet may get an instance of com.ms.awt.peer.INativeServices by calling SystemX.getNativeServices(). Its methods may be invoked indirectly via the java.lang.reflect.\* methods. The methods of INativeServices take memory addresses etc. as parameters without checking them. It's easy to crash the browser by passing bogus parameters. It's also possible to read the process's memory space via the method pGetFontEnumeratedFamily() and retrieve sensitive information such as cookies and addresses of visited websites. In particular, this can be used to find out the exact path to IE's cache directories. This allows certain codebase related attacks, for instance starting another applet having a file: codebase (see vulnerability 6) which can then browse the hard disks and shares and read

any file. This could be used for instance to read cookies, passwords, and other sensitive information, or perhaps to launch other codebase attacks to run arbitrary code.

5) INativeServices clipboard access

Impact: Any applet can get or set the contents of clipboard

The methods `ClipboardGetText()` and `ClipboardSetText()` of the class `INativeServices` can be used to access and modify clipboard contents. The methods are accessible by any applet. The clipboard may obviously contain very sensitive information. The methods have to be called indirectly via the package `java.lang.reflect.*`.

6) `file://` codebase when using shares

Impact: Any applet may get global file read access

The codebase in the applet tag can be set to "`file://%00`" which causes the applet to gain read access to all local files and network shares. The applet may also list directory contents. This requires that the applet is loaded from a publicly readable network share. The consequences are the same as described for vulnerability 4).

7) `StandardSecurityManager` restriction bypassing

Impact: Bypassing package access restrictions

The class `com.ms.security.StandardSecurityManager` can be extended by any applet. The protected static fields containing package access restrictions (`deniedDefinitionPackages`, `deniedAccessPackages`) can be altered or emptied. Thus, any applet can bypass these restrictions. They originate from the registry and aren't used by default, so this flaw doesn't probably pose a big risk on default systems.

8) `com.ms.vm.loader.CabCracker`

Impact: An applet may load any local .cab archive

The method `load()` of the `CabCracker` class is used to load archives from hard disk. The method does security checks and asks confirmation from the user, and then calls `load0()` if the tests are successfully passed. However the `load0()` method is declared public, so any applet can call it directly and so skip the security checks. This would require some more investigation (i.e. what's possible with these cab archives; Microsoft hasn't commented this in any way). In any case, an untrusted applet isn't supposed to be able to access the local file system in this way.

9) Problems with HTML object passed to Java applets via JavaScript

Impact: Unknown

JavaScript code can pass references of HTML objects to an applet. The applet may invoke methods of some proprietary MS interfaces on them. Some of these crash the browser due to illegal memory accesses. This may be a similar case as `INativeServices/JdbcOdbc`.

10) HTML <applet> tag may be used to bypass Java class restrictions  
Impact: Unknown

An applet tag can be used to instantiate objects whose constructors are private. Intention of them shouldn't be possible. E.g. <applet code=java.lang.Class> instantiates a Class object. Some of its native methods crash the browser when called on this new instance, because they presume the object can't be instantiated this way. As usual, IE crashing means it might be possible to trick it into modifying memory in arbitrary addresses and compromise the system.

**Workaround:**

The only known workaround for these issues is to disable Java support in Internet Options -> Security -> Internet -> Custom level -> Microsoft VM / Java permissions / Disable Java or use an alternative web browser and mail client.

**ADDITIONAL INFORMATION**

The information has been provided by <mailto:jouko@solutions.fi> Jouko Pynnonen.

=====

This bulletin is sent to members of the SecuriTeam mailing list.

To unsubscribe from the list, send mail with an empty subject line and body to:

[list-unsubscribe@securiteam.com](mailto:list-unsubscribe@securiteam.com)

In order to subscribe to the mailing list, simply forward this email to: [list-subscribe@securiteam.com](mailto:list-subscribe@securiteam.com)

=====

=====

**DISCLAIMER:**

The information in this bulletin is provided "AS IS" without warranty of any kind.

In no event shall we be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages.

- 
- **Previous message:** [support@securiteam.com](mailto:support@securiteam.com): "[NT] LiteServe Directory Index Cross-Site Scripting"
  - **Messages sorted by:** [\[ date \]](#) [\[ thread \]](#) [\[ subject \]](#) [\[ author \]](#) [\[ attachment \]](#)