

[NEWS] Security Vulnerability in SUN's Java Virtual Machine Implementation ('/ Replaces '.')

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2003-10/0105.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 10/23/03

To: list@securiteam.com

Date: 23 Oct 2003 18:17:14 +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

The SecuriTeam alerts list – Free, Accurate, Independent.

Get your security news from a reliable source.

<http://www.securiteam.com/maillinglist.html>

Security Vulnerability in SUN's Java Virtual Machine Implementation ('/ Replaces '.')

SUMMARY

LSD has found a security vulnerability in the SUN's implementation of the Java Virtual Machine. The vulnerability allows the creation of a malicious applet that could *completely* bypass applet sandbox restrictions.

DETAILS

Vulnerable systems:

- * SDK and JRE 1.4.1_03 and earlier
- * SDK and JRE 1.3.1_08 and earlier
- * SDK and JRE 1.2.2_015 and earlier.

The vulnerability stems from the fact that Class Loader's checks are not done properly with regard to package access. Specifically, due to the logic flaw in the implementation of the loadClass method of the sun.applet.AppletClassLoader class it is possible to load any class into JVM without issuing a call to checkPackageAccess method of the Security Manager. The implementation of the vulnerable loadClass method is presented below:

```
public synchronized Class loadClass(String s, boolean flag)
    throws ClassNotFoundException
{
    int i = s.lastIndexOf('.');
    if(i != -1)
    {
        SecurityManager securitymanager = System.getSecurityManager();
        if(securitymanager != null)
            securitymanager.checkPackageAccess(s.substring(0, i));
    }
    return super.loadClass(s, flag);
}
```

Whenever a user or JVM itself issues a call to loadClass method of the corresponding Class Loader object, a check is done to see whether the requested class is defined in any package. For that purpose, a simple check detecting whether the loaded class name contains the '.' character is done. If this is the case, a proper call to Security Manager's checkPackageAccess method is issued. This is done in order to check whether the requested class belongs to the package that can be actually accessed (loaded into JVM in this specific case). However, such an implementation of this check is not sufficient. This is mainly due to the fact that JVM uses internally slightly different class naming convention in which all fully qualified class names have the '/' character as a package name separator instead of the '.' one.

The aforementioned check can be simply bypassed by using the '/' character instead of the '.' one, while defining package name. For example, due to the default security policy in JRE (package.access=sun.), any attempt to access a class from the "sun." package tree should cause a security exception to be thrown. This works very well when the loadClass method is issued with for example "sun.some_package.some_class" argument. However, this does not work when the call is made with the class name argument set to "sun/some_package/some_class". In this latter case, checkPackageAccess method of Security Manager class isn't invoked at all and the class can be successfully loaded into JVM .

Impact:

The described vulnerability allows for the creation of a malicious applet that could *completely* bypass applet sandbox restrictions. We developed proof of concept code that successfully exploited this vulnerability in Netscape 6 and 7 as well as Mozilla web browsers environment using vulnerable versions of JRE Plugin.

It should be noted, that all users incorporating the vulnerable version of the Java Runtime Environment Plugin in their web browsers are at risk. This could potentially include users of other web browsers (Opera, Internet Explorer), but we have not investigated that further.

In about 4 weeks time, we will release an updated version of our Java/JVM security paper [2] in which we will publish all exploitation details with

Securiteam: [NEWS] Security Vulnerability in SUN's Java Virtual Machine Implementation ('/ Replaces '.')

regard to the presented flaw along with some other flaw that affects all old and not supported any more Netscape 4.x browsers.

Vendor response:

SUN was informed about this issue on June 2 2003 and has already addressed it in their latest SDK/JRE versions. Please, see Sun Alert Notification numbered

http://sunsolve.sun.com/pub-cgi/retrieve.pl?doc=fsalert%2F57221&zone_32=category%3Asecurity 57221 for more information about the patched SDK/JRE releases.

ADDITIONAL INFORMATION

The information has been provided by <mailto:contact@lsd-pl.net> Last Stage of Delirium.

=====

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

In order to subscribe to the mailing list, simply forward this email to: 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.