**SUMMARY**

A concurrency bug in Tomcat 6-6.0.29.

**DETAILS**

Some details can also be found at: <https://bz.apache.org/bugzilla/show_bug.cgi?id=49972>

This bug is due to a data race.

Case1:

Thread1 executes:

long now = System.currentTimeMillis();

if ((now - currentDateGenerated) > 1000) {

synchronized (format) {

if ((now - currentDateGenerated) > 1000) {

currentDateGenerated = now;

Thread2 executes:

long now = System.currentTimeMillis();

if ((now - currentDateGenerated) > 1000) {

Thread2 skips the "if" part.

If these were the first calls to this function the return value in Thread2 will be null.

-----

Case2:

Thread1 executes:

currentDate = format.format(new Date(now));

Thread2 goes through fast path and executes

return currentDate;

The object currentDate was not properly published and according to the Java memory model (if I understand it correctly) may be incomplete.

|  |  |
| --- | --- |
| Thread1 (FastHttpDateFormat.java) | Thread2 (FastHttpDateFormat.java) |
| public static final String getCurrentDate() {  long now = System.currentTimeMillis();  if ((now - currentDateGenerated) > 1000) {  synchronized (format) {  if ((now - currentDateGenerated) > 1000) {  currentDateGenerated = now;  currentDate = format.format(new Date(now));  }  }  }  return currentDate;  } | public static final String getCurrentDate() {  long now = System.currentTimeMillis();  if ((now - currentDateGenerated) > 1000) {  synchronized (format) {  if ((now - currentDateGenerated) > 1000) {  currentDateGenerated = now;  currentDate = format.format(new Date(now));  }  }  }  return currentDate;  } |
| 0-lock | 1-lock |