**SUMMARY**

A concurrency bug in Apache Tomcat7.

**DETAILS**

Some details can also be found at: https://bz.apache.org/bugzilla/show\_bug.cgi?id=52356

This bug is due to a data race.

|  |  |
| --- | --- |
| Thread1 (FastQueue.java) | Thread2 (FastQueue.java) |
| public int getSize() {  return size;  } | public LinkObject remove() {  LinkObject element;  boolean gotLock;  if (!enabled) {  if (log.isInfoEnabled())  log.info("FastQueue.remove: queue disabled, remove aborted");  return null;  }  gotLock = lock.lockRemove();  try {  if (!gotLock) {  if (enabled) {  if (log.isInfoEnabled())  log.info("FastQueue.remove: Remove aborted although queue enabled");  } else {  if (log.isInfoEnabled())  log.info("FastQueue.remove: queue disabled, remove aborted");  }  return null;  }  if (log.isTraceEnabled()) {  log.trace("FastQueue.remove: remove starting with size " + size);  }  if (checkLock) {  if (inRemove)  log.warn("FastQueue.remove: Detected other remove");  inRemove = true;  if (inMutex)  log.warn("FastQueue.remove: Detected other mutex in remove");  inMutex = true;  }  element = first;  first = last = null;  size = 0;  if (checkLock) {  if (!inMutex)  log.warn("FastQueue.remove: Cancelled by other mutex in remove");  inMutex = false;  if (!inRemove)  log.warn("FastQueue.remove: Cancelled by other remove");  inRemove = false;  }  if (log.isTraceEnabled()) {  log.trace("FastQueue.remove: remove ending with size " + size);  }  } finally {  lock.unlockRemove();  }  return element; |
| 0-lock | 1-lock |