**SUMMARY**

A concurrency bug in Tomcat 5-5.5.9.

**DETAILS**

Some details can also be found at: https://bz.apache.org/bugzilla/show\_bug.cgi?id=37458

This bug is due to a data race.

This is one possible scence to cause an exception: suppose two threads t1 and t2, both are trying to load a same new class. t1 goes first and hit the line 1630, definePackage(packageName, entry.manifest, entry.codeBase), and stops before the method call. Then t2 goes ahead and it can hit the same method call too since the entry.manifest is still not null at that time. And then t2 or t1 continues and loads the class and sets entry.manifest at the end. After this, when the other thread tries to finish the call to definePackage, an exception will be thrown since entry.manifest and entry.codebase have become null now.

|  |  |
| --- | --- |
| Thread1 (WebappClassLoader.java) | Thread2 (WebappClassLoader.java) |
| protected Class findClassInternal(String name)  throws ClassNotFoundException {  ..  if (pkg == null) {  if (entry.manifest == null) {  definePackage(packageName, null, null, null, null, null,  null, null);  } else {  definePackage(packageName, entry.manifest, entry.codeBase);  }  }  .. | protected Class findClassInternal(String name)  throws ClassNotFoundException {  ..  synchronized (this) {  if (entry.loadedClass == null) {  ..  entry.manifest = null;  entry.certificates = null;  } else {  clazz = entry.loadedClass;  }  }    .. |
| 0-lock | 1-lock |