**SUMMARY**

A concurrency bug in Core- JavaScript Engine.

**DETAILS**

Some details can also be found at: https://bugzilla.mozilla.org/show\_bug.cgi?id=622691

This bug is due to a data race.

The constructor acquires cx->runtime->atomState.lock and then writes to cx->runtime->defaultCompartmentIsLocked (fine).

The destructor drops the lock, then writes to cx->runtime->defaultCompartmentIsLocked :-(, so it is not consistently protected by the lock.

|  |  |
| --- | --- |
| Thread1 () | Thread2 () |
| AutoLockDefaultCompartment(JSContext \*cx) : cx(cx) {  JS\_LOCK(cx, &cx->runtime->atomState.lock);  #ifdef JS\_THREADSAFE  cx->runtime->defaultCompartmentIsLocked = true;  #endif  } | ~AutoLockDefaultCompartment() {  JS\_UNLOCK(cx, &cx->runtime->atomState.lock);  #ifdef JS\_THREADSAFE  cx->runtime->defaultCompartmentIsLocked = false;  #endif  } |
| 1-lock | 0-lock |