**SUMMARY**

A concurrency bug in JavaScript Engine

**DETAILS**

Some details can also be found at: https://bugzilla.mozilla.org/show\_bug.cgi?id=73761

This bug is due to a data race.

The manipulation of the 'empty' flag in js\_FlushPropertyCacheByProp does not look safe to me. You could race with another thread and end up setting empty when the cache was not really empty. This \*could\* have been the cause of the sort of crash I saw because js\_FlushPropertyCache is a nop when the flag is set.

|  |  |
| --- | --- |
| Thread1 (jsgc.c) | Thread2 (jsgc.c) |
| js\_FlushPropertyCacheByProp(JSContext \*cx, JSProperty \*prop)  {  JSPropertyCache \*cache;  JSBool empty;  JSPropertyCacheEntry \*end, \*pce, entry;  JSProperty \*pce\_prop;    if (cache->empty)  return;    empty = JS\_TRUE;  end = &cache->table[PROPERTY\_CACHE\_SIZE];  for (pce = &cache->table[0]; pce < end; pce++) {  PCE\_LOAD(cache, pce, entry);  pce\_prop = PCE\_PROPERTY(entry);  if (pce\_prop) {  if (pce\_prop == prop) {  PCE\_OBJECT(entry) = NULL;  PCE\_PROPERTY(entry) = NULL;  PCE\_STORE(cache, pce, entry);  } else {  empty = JS\_FALSE;  }  }  }  cache->empty = empty; | js\_FlushPropertyCacheByProp(JSContext \*cx, JSProperty \*prop)  {  JSPropertyCache \*cache;  JSBool empty;  JSPropertyCacheEntry \*end, \*pce, entry;  JSProperty \*pce\_prop;    if (cache->empty)  return;    empty = JS\_TRUE;  end = &cache->table[PROPERTY\_CACHE\_SIZE];  for (pce = &cache->table[0]; pce < end; pce++) {  PCE\_LOAD(cache, pce, entry);  pce\_prop = PCE\_PROPERTY(entry);  if (pce\_prop) {  if (pce\_prop == prop) {  PCE\_OBJECT(entry) = NULL;  PCE\_PROPERTY(entry) = NULL;  PCE\_STORE(cache, pce, entry);  } else {  empty = JS\_FALSE;  }  }  }  cache->empty = empty; |
| 0-lock | 0-lock |