**SUMMARY**

A concurrency bug in NSS-Libraries-3.4

**DETAILS**

Some details can also be found at: https://bugzilla.mozilla.org/show\_bug.cgi?id=124923

This bug is due to a data race.

The dynamic table (oid\_d\_hash) gets modified (new entries added) when the user loads new crypto modules. It needs to be protected by a lock.

|  |  |
| --- | --- |
| Thread1 () | Thread2 () |
| static SECOidData \*  secoid\_FindDynamic(SECItem \*key) {  SECOidData \*ret = NULL;  if (secoidDynamicTable == NULL) {  /\* PORT\_SetError! \*/  return NULL;  if (secoidLastHashEntry != secoidLastDynamicEntry) {  SECStatus rv = secoid\_DynamicRehash();  if ( rv != SECSuccess ) {  return NULL;  }  ret = (SECOidData \*)PL\_HashTableLookup (oid\_d\_hash, key);  return ret;    } | SECStatus  SECOID\_Shutdown(void)  {  int i;  if (oidhash) {  PL\_HashTableDestroy(oidhash);  oidhash = NULL;  }  if (oidmechhash) {  PL\_HashTableDestroy(oidmechhash);  oidmechhash = NULL;  }  if (oid\_d\_hash) {  PL\_HashTableDestroy(oid\_d\_hash);  oid\_d\_hash = NULL;  }  if (secoidDynamicTable) {  for (i=0; i < secoidLastDynamicEntry; i++) {  PORT\_Free(secoidDynamicTable[i]);  }  PORT\_Free(secoidDynamicTable);  secoidDynamicTable = NULL;  secoidDynamicTableSize = 0;  secoidLastDynamicEntry = 0;  secoidLastHashEntry = 0;  }  return SECSuccess;  } |
| static SECOidData \*  secoid\_FindDynamicByTag(SECOidTag tagnum)  {  int tagNumDiff;    if (secoidDynamicTable == NULL) {  return NULL;  }  if (tagnum < SEC\_OID\_TOTAL) {  return NULL;  }  tagNumDiff = tagnum - SEC\_OID\_TOTAL;  if (tagNumDiff >= secoidLastDynamicEntry) {  return NULL;  }    return(secoidDynamicTable[tagNumDiff]);  } |  |
| 0-lock | 0-lock |