**SUMMARY**

A concurrency bug in Apache httpd-2 2.2.6.

**DETAILS**

Some details can also be found at: https://bz.apache.org/bugzilla/show\_bug.cgi?id=44402

This bug is due to a data race.

|  |  |
| --- | --- |
| Thread1 (fdqueue.c) | Thread2 (worker.c) |
| apr\_status\_t ap\_queue\_info\_set\_idle(fd\_queue\_info\_t \*queue\_info,  apr\_pool\_t \*pool\_to\_recycle)  {  ..  for (;;) {  new\_recycle->next = queue\_info->recycled\_pools;  if (apr\_atomic\_casptr((volatile void\*\*)&(queue\_info->recycled\_pools),  new\_recycle, new\_recycle->next) ==  new\_recycle->next) {  break;  } | static void \* APR\_THREAD\_FUNC worker\_thread(apr\_thread\_t \*thd, void \* dummy)  {  ..  apr\_pool\_clear(ptrans);  last\_ptrans = ptrans;  // new\_recycle->next will be corrupted and point to a deleted value. |
|  | APR\_DECLARE(void) apr\_pool\_clear(apr\_pool\_t \*pool)  {  apr\_memnode\_t \*active;  /\* Destroy the subpools. The subpools will detach themselves from  \* this pool thus this loop is safe and easy.  \*/  while (pool->child)  apr\_pool\_destroy(pool->child);  /\* Run cleanups \*/  run\_cleanups(&pool->cleanups);  pool->cleanups = NULL;  pool->free\_cleanups = NULL;  /\* Free subprocesses \*/  free\_proc\_chain(pool->subprocesses);  pool->subprocesses = NULL;  /\* Clear the user data. \*/  pool->user\_data = NULL;  /\* Find the node attached to the pool structure, reset it, make  \* it the active node and free the rest of the nodes.  \*/  active = pool->active = pool->self;  active->first\_avail = pool->self\_first\_avail;  if (active->next == active)  return;  \*active->ref = NULL;  allocator\_free(pool->allocator, active->next);  active->next = active;  active->ref = &active->next;  } |
|  | static APR\_INLINE  void allocator\_free(apr\_allocator\_t \*allocator, apr\_memnode\_t \*node)  {  ..  while (freelist != NULL) {  node = freelist;  freelist = node->next;  free(node);  }  } |
| 0-lock | 0-lock |

Crash 1 :

(dbx) where

current thread: t@76

=>[1] allocator\_free(allocator = 0x101f870, node = (nil)), line 331 in "apr\_pools.c"

[2] apr\_pool\_clear(pool = 0x102fb88), line 710 in "apr\_pools.c"

[3] ap\_core\_output\_filter(f = 0x1020550, b = 0x101f9e8), line 899 in

"core\_filters.c"

[4] ap\_pass\_brigade(next = 0x1020550, bb = 0x101f9e8), line 526 in "util\_filter.c"

[5] logio\_out\_filter(f = 0x10204e0, bb = 0x101f9e8), line 135 in "mod\_logio.c"

[6] ap\_pass\_brigade(next = 0x10204e0, bb = 0x101f9e8), line 526 in "util\_filter.c"

[7] ap\_flush\_conn(c = 0x101fd00), line 84 in "connection.c"

[8] ap\_lingering\_close(c = 0x101fd00), line 123 in "connection.c"

[9] process\_socket(p = 0x101f968, sock = 0x101f9e8, my\_child\_num = 1,

my\_thread\_num = 227, bucket\_alloc = 0x1029a88), line 545 in "worker.c"

[10] worker\_thread(thd = 0x5bed38, dummy = 0x6dbac0), line 894 in "worker.c"

[11] dummy\_worker(opaque = 0x5bed38), line 142 in "thread.c"

[12] \_thr\_setup(0x0, 0x0, 0x0, 0x0, 0x0, 0x0), at 0xfffffd7ffef5d8f7

[13] \_lwp\_start(0x0, 0x0, 0x0, 0x0, 0x0, 0x0), at 0xfffffd7ffef5dba0

Crash 2 :

(dbx) where

current thread: t@363

=>[1] apr\_palloc(pool = 0x21680007952225ff, size = 18446744073323675656U), line

601 in "apr\_pools.c"

[2] apr\_sockaddr\_ip\_get(addr = 0xcda3d0, sockaddr = 0x42d790), line 104 in

"sockaddr.c"

[3] core\_create\_conn(ptrans = 0xcda2d8, server = 0x4bf600, csd = 0xcda358, id

= 360, sbh = 0xcda378, alloc = 0xd147e8), line 3895 in "core.c"

[4] ap\_run\_create\_connection(0x0, 0x0, 0x0, 0x0, 0x0, 0x0), at 0x45fe03

[5] process\_socket(p = 0xcda2d8, sock = 0xcda358, my\_child\_num = 0,

my\_thread\_num = 360, bucket\_alloc = 0xd147e8), line 542 in "worker.c"

[6] worker\_thread(thd = 0x7192f8, dummy = 0x7e45a0), line 894 in "worker.c"

[7] dummy\_worker(opaque = 0x7192f8), line 142 in "thread.c"

[8] \_thr\_setup(0x0, 0x0, 0x0, 0x0, 0x0, 0x0), at 0xfffffd7ffef5d8f7

[9] \_lwp\_start(0x0, 0x0, 0x0, 0x0, 0x0, 0x0), at 0xfffffd7ffef5dba0