# strongSwan - Bug #411

## valgrind reports invalid read size

12.09.2013 19:29 - Piyush Patel

| | | | |
|---|---|---|---|
| **Status:** | Closed | **Start date:** | 12.09.2013 |
| **Priority:** | Normal | **Due date:** | |
| **Assignee:** | Tobias Brunner | **Estimated time:** | 0.00 hour |
| **Category:** | charon | | |
| **Target version:** | 5.1.1 | | |
| **Affected version:** | 5.1.0 | **Resolution:** | Fixed |

**Description**

Valgrind outputted the below error. Looks like it's due to ipsec SA rekeying. This test was done against strongswan 5.1.0:

```
==31002== Invalid read of size 8
==31002==    at 0x52CBF40: have_equal_ts (task_manager_v1.c:1769)
==31002==    by 0x52CE227: queue_child_rekey (task_manager_v1.c:1788)
==31002==    by 0x52ADB44: rekey_child_sa (ike_sa.c:1409)
==31002==    by 0x52A8DB5: execute (rekey_child_sa_job.c:68)
==31002==    by 0x4E5BE12: process_jobs (processor.c:235)
==31002==    by 0x4E5EDDF: thread_main (thread.c:309)
==31002==    by 0x5502E99: start_thread (pthread_create.c:308)
==31002==  Address 0x63dda18 is 8 bytes inside a block of size 32 free'd
==31002==    at 0x4C2A82E: free (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so)
==31002==    by 0x52CBF3C: have_equal_ts (task_manager_v1.c:1768)
==31002==    by 0x52CE227: queue_child_rekey (task_manager_v1.c:1788)
==31002==    by 0x52ADB44: rekey_child_sa (ike_sa.c:1409)
==31002==    by 0x52A8DB5: execute (rekey_child_sa_job.c:68)
==31002==    by 0x4E5BE12: process_jobs (processor.c:235)
==31002==    by 0x4E5EDDF: thread_main (thread.c:309)
==31002==    by 0x5502E99: start_thread (pthread_create.c:308)
==31002==
==31002== Invalid free() / delete / delete[] / realloc()
==31002==    at 0x4C2A82E: free (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so)
==31002==    by 0x52CBF42: have_equal_ts (task_manager_v1.c:1769)
==31002==    by 0x52CE227: queue_child_rekey (task_manager_v1.c:1788)
==31002==    by 0x52ADB44: rekey_child_sa (ike_sa.c:1409)
==31002==    by 0x52A8DB5: execute (rekey_child_sa_job.c:68)
==31002==    by 0x4E5BE12: process_jobs (processor.c:235)
==31002==    by 0x4E5EDDF: thread_main (thread.c:309)
==31002==    by 0x5502E99: start_thread (pthread_create.c:308)
==31002==  Address 0x63dda10 is 0 bytes inside a block of size 32 free'd
==31002==    at 0x4C2A82E: free (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so)
==31002==    by 0x52CBF3C: have_equal_ts (task_manager_v1.c:1768)
==31002==    by 0x52CE227: queue_child_rekey (task_manager_v1.c:1788)
==31002==    by 0x52ADB44: rekey_child_sa (ike_sa.c:1409)
==31002==    by 0x52A8DB5: execute (rekey_child_sa_job.c:68)
==31002==    by 0x4E5BE12: process_jobs (processor.c:235)
==31002==    by 0x4E5EDDF: thread_main (thread.c:309)
==31002==    by 0x5502E99: start_thread (pthread_create.c:308)
==31002==
```

Looks like code is freeing same object twice. The below patch might be what was intended. Applying it fixed the issue:

```
--- src/libcharon/sa/ikev1/task_manager_v1.c.orig    2013-09-12 08:45:38.161869385 -0700
+++ src/libcharon/sa/ikev1/task_manager_v1.c    2013-09-12 09:19:12.628216083 -0700
@@ -1766,7 +1766,7 @@
         equal = ts1->equals(ts1, ts2);
     }
     e1->destroy(e1);
-    e1->destroy(e1);
```

```
+     e2->destroy(e2);

      return equal;
 }
```

### Associated revisions

**Revision fafa7684 - 13.09.2013 10:14 - Tobias Brunner**

ikev1: Fix double free when searching for redundant CHILD_SAs

Fixes #411.

### History

**#1 - 13.09.2013 11:21 - Tobias Brunner**

*- Tracker changed from Issue to Bug*

*- Category set to charon*

*- Status changed from New to Resolved*

*- Assignee set to Tobias Brunner*

*- Target version set to 5.1.1*

*- Resolution set to Fixed*

Thanks for report. Fixed with the associated commit.

**#2 - 18.09.2013 14:55 - Tobias Brunner**

*- Status changed from Resolved to Closed*