strongSwan - Bug #1269

Data race in ike_sa_manager.c

13.01.2016 09:17 - Avinoam Meir

Status:	Closed	**Start date:**	13.01.2016
Priority:	Normal	**Due date:**	
Assignee:	Tobias Brunner	**Estimated time:**	0.00 hour
Category:	libcharon		
Target version:	5.4.0		
Affected version:	5.3.5	**Resolution:**	Fixed

Description

I run this (https://code.google.com/p/data-race-test/wiki/ThreadSanitizer) tool, and found possible data race:

Trread 1 call stack:

```
0 checkout_by_message src/libcharon/sa/ike_sa_manager.c:1341
1 execute src/libcharon/processing/jobs/process_message_job.c:66
2 process_job src/libstrongswan/processing/processor.c:235
3 process_jobs src/libstrongswan/processing/processor.c:321
4 thread_main src/src/libstrongswan/threading/thread.c:303
```

Thread 2 call stack:

```
0 heckout_by_message src/libcharon/sa/ike_sa_manager.c:1296
1 execute src/libcharon/processing/jobs/process_message_job.c:66
2 process_job src/libstrongswan/processing/processor.c:235
3 process_jobs src/libstrongswan/processing/processor.c:321
4 thread_main src/libstrongswan/threading/thread.c:303
```

Suggested fix:

Change those lines (ike_sa_manager.c:1296):

```
entry->checked_out = TRUE;
unlock_single_segment(this, segment);
entry->processing = get_message_id_or_hash(message);
entry->init_hash = hash;
```

to that:

```
entry->checked_out = TRUE;
entry->processing = get_message_id_or_hash(message);
entry->init_hash = hash;
unlock_single_segment(this, segment);
```

Associated revisions

Revision e663d8e2 - 01.02.2016 10:39 - Tobias Brunner

ike-sa-manager: Don't update entries for init messages after unlocking segment

If the retransmit of an initial message is processed concurrently with the original message it might not have been handled as intended as the thread processing the retransmit might not have seen the correct value of entry->processing set by the thread handling the original request.

For IKEv1, i.e. without proper message IDs, there might still be races e.g. when receiving a retransmit of the initial IKE message while processing the initiator's second request.

Fixes #1269.

History

#1 - 18.01.2016 17:57 - Tobias Brunner

- Status changed from New to Feedback

- Assignee set to Tobias Brunner

- Target version set to 5.4.0

Agreed, retransmits of initial IKE messages received practically concurrently with the original message might not have been handled as intended (i.e. by simply logging ignoring request with ID ..., already processing). Fix can be found in the *1269-checkout-by-message* branch. As mentioned in the commit message there might still be races for IKEv1 that I don't think are currently handled correctly by the IKEv1 task manager.

#2 - 24.01.2016 14:15 - Avinoam Meir

Test the 1269-checkout-by-message branch and it works well.

#3 - 01.02.2016 10:40 - Tobias Brunner

- Status changed from Feedback to Closed

- Resolution set to Fixed

Thanks for testing! Applied the fix to master.