Deadlock when detection thread dies and main thread attempts to restart

When a thread dies and is restarted there is a deadlock. TmThreadCheckThreadState() takes a lock on tv_root_lock as it iterates through the list of ThreadVars and performs a check as to whether the thread is running or not. If the thread has failed, it will potentially attempt to restart the thread calling TmThreadRestartThread(). This subsequently calls TmThreadSpawn() which eventually calls TmThreadAppend() which also attempts to claim a lock on tv_root_mutex. Since the lock was already taken by TmThreadCheckThreadState() this eventually causes a deadlock.

Though not the best solution, the easiest solution would be to remove SCMutexLock/Unlock calls from TmThreadAppend() as TmThreadAppend() is only called from TmThreadSpawn(). It seems like TmThreadAppend() could also be moved into an internal static/inline function as well. It looks like there may be other functions with potentially similar issues. For example TmThreadRemove() is not called by any function, has SCMutexLock/Unlock calls and if there was a higher order function that calls it, the same issue would occur. This may be a candidate for internal static/inline as well.

I think the better solution would be to add in recursive mutexes using PTHREAD_RECURSIVE_MUTEX_INITIALIZER and then define tv_root_lock as a recursive mutex. I'm not suggesting everything switch to recursive mutexes, just tv_root_lock initially.

History

#1 - 09/08/2016 02:38 PM - Andreas Herz
- Assignee set to OISF Dev
- Target version set to TBD

#2 - 11/29/2016 04:17 AM - Victor Julien
- Status changed from New to Closed
- Assignee changed from OISF Dev to Victor Julien
- Target version deleted (TBD)

I have removed the thread restart logic as it was broken in various ways.