Suricata - Bug #1786
spm crash on rule reload
05/19/2016 04:54 PM - Victor Julien

Description
[30695] 19/5/2016 -- 23:42:48 -- (detect.c:3959) <Info> (SigAddressCleanupStage1) -- cleaning up signature grouping structure... complete
ASAN:SIGSEGV

==30695==ERROR: AddressSanitizer: SEGV on unknown address 0x000000000000 (pc 0x0000007a276a bp 0x7ffe7f116170 sp 0x7ffe7f116170 T0)
#0 0x7a2769 in SpmDestroyGlobalThreadCtx (/home/victor/dev/suricata/src/suricata+0x7a2769)
#1 0x51bfec in DetectEngineCtxFree (/home/victor/dev/suricata/src/suricata+0x51bfec)
#2 0x5257d7 in DetectEnginePruneFreeList (/home/victor/dev/suricata/src/suricata+0x5257d7)
#3 0x525c89 in DetectEngineReload (/home/victor/dev/suricata/src/suricata+0x525c89)
#4 0x41f48f in main (/home/victor/dev/suricata/src/suricata+0x41f48f)
#5 0x7fe55d79c82f in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x2082f)
#6 0x421798 in _start (/home/victor/dev/suricata/src/suricata+0x421798)

AddressSanitizer can not provide additional info.
SUMMARY: AddressSanitizer: SEGV ??:0 SpmDestroyGlobalThreadCtx
==30695==ABORTING

Happens with both --set spm-algo=hs and --set spm-algo=bm

History
#1 - 05/19/2016 04:55 PM - Victor Julien
Btw this works to prevent the crash, but I'm not sure it's the proper fix.

diff --git a/src/util-spm.c b/src/util-spm.c
index 04e7676..5c9bc05 100644
--- a/src/util-spm.c
+++ b/src/util-spm.c
@@ -113,8 +113,11 @@ SpmGlobalThreadCtx *SpmInitGlobalThreadCtx(uint16_t matcher)
 void SpmDestroyGlobalThreadCtx(SpmGlobalThreadCtx *global_thread_ctx)
 {
+ if (global_thread_ctx != NULL) {
   uint16_t matcher = global_thread_ctx->matcher;
+ if(spm_table[matcher].DestroyGlobalThreadCtx!= NULL)
   spm_table[matcher].DestroyGlobalThreadCtx(global_thread_ctx);
+ }
  }

SpmThreadCtx *SpmMakeThreadCtx(const SpmGlobalThreadCtx *global_thread_ctx)

#2 - 05/19/2016 04:57 PM - Victor Julien
Ah it's not actually a full rule reload. It's start up with delayed detect enabled. This starts with an empty (minimal) detection engine, then forces a reload. So perhaps the issue is that the 'minimal' detect engine is freed too aggressively.

#3 - 05/19/2016 07:59 PM - Justin Viiret
So, it looks like init of the SPM table needs to happen earlier (or more often). I'll take a look at this today.
Actually, ignore my last comment, spm_table is fine.

Victor: as you thought, this is the minimal detection engine, which gets a NULL pointer for its SpmGlobalThreadCtx. I think that allowing the Destroy functions in the SPM API to accept NULL pointers is a reasonable fix.

I've written and tested a patch -- will file a PR in a moment.

#5 - 05/20/2016 05:02 PM - Victor Julien

- Status changed from Assigned to Closed
- Target version changed from 70 to 3.1rc1

https://github.com/inliniac/suricata/pull/2088