

Tvheadend - Bug #5353

Locked up with "too much queued input data" and "too much queued table input data"

2018-11-26 21:35 - Flole Systems

Status:	Fixed	Start date:	2018-12-10
Priority:	Normal	Due date:	
Assignee:		% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:		Affected Versions:	
Found in version:	4.3-1544~gfb329606b		

Description

I just experienced a complete lockup of tvheadend. In the logs it complained about too much queued input data. Running gdb and running "info threads" gave this:

```
* 1 Thread 0x7f8f91706bc0 (LWP 44826) "tvheadend" __lll_lock_wait () at ../sysdeps/unix/sysv/linux/x86_64/lowlevellock.S:135
  2 Thread 0x7f8f8c045700 (LWP 44828) "tvh:log" 0x00007f8f8faf79f3 in futex_wait_cancelable (private=<optimized out>, expected=0, futex_word=0x562a553e872c <tvhlog_cond+44>)
    at ../sysdeps/unix/sysv/linux/futex-internal.h:88
  3 Thread 0x7f8f8be44700 (LWP 44832) "tvh:notify" __lll_lock_wait () at ../sysdeps/unix/sysv/linux/x86_64/lowlevellock.S:135
  4 Thread 0x7f8f8bc43700 (LWP 44833) "tvheadend" 0x00007f8f8e785bb7 in epoll_wait (epfd=14, events=0x7f8f7c000b80, maxevents=2, timeout=500) at ../sysdeps/unix/sysv/linux/epoll_wait.c:30
  5 Thread 0x7f8f8b442700 (LWP 44834) "tvh:save" 0x00007f8f8faf79f3 in futex_wait_cancelable (private=<optimized out>, expected=0, futex_word=0x562a55287f88 <save_cond+40>)
    at ../sysdeps/unix/sysv/linux/futex-internal.h:88
  6 Thread 0x7f8f8b241700 (LWP 44835) "tvh:mtick" __clock_nanosleep (clock_id=1, flags=0, req=0x7f8f8b240920, rem=0x7f8f8b240920) at ../sysdeps/unix/sysv/linux/clock_nanosleep.c:48
  7 Thread 0x7f8f8b040700 (LWP 44836) "tvh:mtimer" __lll_lock_wait () at ../sysdeps/unix/sysv/linux/x86_64/lowlevellock.S:135
  8 Thread 0x7f8f8ae3f700 (LWP 44837) "tvh:tasklet" 0x00007f8f8faf79f3 in futex_wait_cancelable (private=<optimized out>, expected=0, futex_word=0x562a55287ccc <tasklet_cond+44>)
    at ../sysdeps/unix/sysv/linux/futex-internal.h:88
  9 Thread 0x7f8f8ac3e700 (LWP 44846) "tvh:fsmonitor" 0x00007f8f8fafb384 in __libc_read (fd=15, buf=0x7f8f8ac3d4e0, nbytes=160) at ../sysdeps/unix/sysv/linux/read.c:27
 10 Thread 0x7f8f8aa3d700 (LWP 44855) "tvh:imagecache" 0x00007f8f8faf79f3 in futex_wait_cancelable (private=<optimized out>, expected=0, futex_word=0x562a5528b9cc <imagecache_cond+44>)
    at ../sysdeps/unix/sysv/linux/futex-internal.h:88
 11 Thread 0x7f8f8a83c700 (LWP 44856) "tvh:http" 0x00007f8f8e785bb7 in epoll_wait (epfd=18, events=0x562a571b9940, maxevents=1, timeout=-1) at ../sysdeps/unix/sysv/linux/epoll_wait.c:30
 12 Thread 0x7f8f8a63b700 (LWP 44860) "tvh:service" 0x00007f8f8faf79f3 in futex_wait_cancelable (private=<optimized out>, expected=0, futex_word=0x562a55288a0c <pending_save_cond+44>)
    at ../sysdeps/unix/sysv/linux/futex-internal.h:88
 13 Thread 0x7f8f8a43a700 (LWP 44861) "tvh:capmt" 0x00007f8f8e785bb7 in epoll_wait (epfd=47, events=0x7f8f74005300, maxevents=1, timeout=500) at ../sysdeps/unix/sysv/linux/epoll_wait.c:30
 14 Thread 0x7f8f8a239700 (LWP 44865) "tvh:iptv" 0x00007f8f8e785bb7 in epoll_wait (epfd=23, events=0x562a571c5a70, maxevents=1, timeout=-1) at ../sysdeps/unix/sysv/linux/epoll_wait.c:30
 15 Thread 0x7f8f8a038700 (LWP 44866) "tvh:iptv" 0x00007f8f8e785bb7 in epoll_wait (epfd=24, events=0x562a571f7f20, maxevents=1, timeout=-1) at ../sysdeps/unix/sysv/linux/epoll_wait.c:30
 16 Thread 0x7f8f89e37700 (LWP 44867) "tvh:iptv" 0x00007f8f8e785bb7 in epoll_wait (epfd=25, events=0x562a571bd860, maxevents=1, timeout=-1) at ../sysdeps/unix/sysv/linux/epoll_wait.c:30
 17 Thread 0x7f8f89c36700 (LWP 44868) "tvh:iptv" 0x00007f8f8e785bb7 in epoll_wait (epfd=26, events=0x562a5727a0b0, maxevents=1, timeout=-1) at ../sysdeps/unix/sysv/linux/epoll_wait.c:30
 18 Thread 0x7f8f89a35700 (LWP 44869) "tvh:iptv" 0x00007f8f8e785bb7 in epoll_wait (epfd=27, events=0x562a571d1fb0, maxevents=1, timeout=-1) at ../sysdeps/unix/sysv/linux/epoll_wait.c:30
 19 Thread 0x7f8f89834700 (LWP 44870) "tvh:iptv" 0x00007f8f8e785bb7 in epoll_wait (epfd=28, events=0x562a571b8800, maxevents=1, timeout=-1) at ../sysdeps/unix/sysv/linux/epoll_wait.c:30
 20 Thread 0x7f8f89633700 (LWP 44871) "tvh:iptv" 0x00007f8f8e785bb7 in epoll_wait (epfd=29, events=0x562a571c9270, maxevents=1, timeout=-1) at ../sysdeps/unix/sysv/linux/epoll_wait.c:30
 21 Thread 0x7f8f89432700 (LWP 44872) "tvh:iptv" 0x00007f8f8e785bb7 in epoll_wait (epfd=30, events=0x562a571b9270, maxevents=1, timeout=-1) at ../sysdeps/unix/sysv/linux/epoll_wait.c:30
 22 Thread 0x7f8f89231700 (LWP 44873) "tvh:iptv" 0x00007f8f8e785bb7 in epoll_wait (epfd=31, eve
```

```
nts=0x562a571f7ea0, maxevents=1, timeout=-1) at ../sysdeps/unix/sysv/linux/epoll_wait.c:30
23 Thread 0x7f8f89030700 (LWP 44874) "tvh:iptv" 0x00007f8f8e785bb7 in epoll_wait (epfd=32, eve
nts=0x562a571e1930, maxevents=1, timeout=-1) at ../sysdeps/unix/sysv/linux/epoll_wait.c:30
24 Thread 0x7f8f88e2f700 (LWP 44879) "tvh:tshift-reap" 0x00007f8f8faf79f3 in futex_wait_cancel
able (private=<optimized out>, expected=0, futex_word=0x562a5528dlcc <timeshift_reaper_cond+44>)
at ../sysdeps/unix/sysv/linux/futex-internal.h:88
25 Thread 0x7f8f88c2e700 (LWP 44880) "tvh:tcp-loop" __l1l_lock_wait () at ../sysdeps/unix/sysv
/linux/x86_64/lowlevellock.S:135
26 Thread 0x7f8f88a2d700 (LWP 44881) "tvh:upnp" 0x00007f8f8e785bb7 in epoll_wait (epfd=36, eve
nts=0x7f8f78000b80, maxevents=2, timeout=1000) at ../sysdeps/unix/sysv/linux/epoll_wait.c:30
27 Thread 0x7f8f8882c700 (LWP 44882) "tvh:svcmmap" 0x00007f8f8faf79f3 in futex_wait_cancelable
(private=<optimized out>, expected=0, futex_word=0x562a5528ba68 <service_mapper_cond+40>)
at ../sysdeps/unix/sysv/linux/futex-internal.h:88
28 Thread 0x7f8f73fff700 (LWP 44917) "tvh:epggrabso" 0x00007f8f8fafb6d7 in __libc_accept (fd=3
9, addr=..., len=0x0) at ../sysdeps/unix/sysv/linux/accept.c:26
29 Thread 0x7f8f8862b700 (LWP 44919) "tvh:epggrabi" 0x00007f8f8fafb384 in __libc_read (fd=41,
buf=0x7f8edc919aa0, nbytes=8192) at ../sysdeps/unix/sysv/linux/read.c:27
30 Thread 0x7f8f8842a700 (LWP 44922) "tvh:epgdata" 0x00007f8f8faf79f3 in futex_wait_cancelable
(private=<optimized out>, expected=0, futex_word=0x562a552883a8 <epggrab_data_cond+40>)
at ../sysdeps/unix/sysv/linux/futex-internal.h:88
31 Thread 0x7f8f88229700 (LWP 44927) "tvh:dvr-inotify" 0x00007f8f8fafb384 in __libc_read (fd=4
0, buf=0x7f8f88228830, nbytes=335) at ../sysdeps/unix/sysv/linux/read.c:27
32 Thread 0x7f8f72f7e700 (LWP 44936) "tvh:satip-rtcp" __clock_nanosleep (clock_id=1, flags=0,
req=0x7f8f72f7d330, rem=0x7f8f72f7d330) at ../sysdeps/unix/sysv/linux/clock_nanosleep.c:48
33 Thread 0x7f8f72d7d700 (LWP 44937) "tvh:avahi" 0x00007f8f8e778bf9 in __GI___poll (fds=0x7f8f
68004550, nfds=2, timeout=-1) at ../sysdeps/unix/sysv/linux/poll.c:29
34 Thread 0x7f8f72338700 (LWP 44940) "tvh:mi-table" 0x00007f8f8faf79f3 in futex_wait_cancelabl
e (private=<optimized out>, expected=0, futex_word=0x562a5723dfa0)
at ../sysdeps/unix/sysv/linux/futex-internal.h:88
35 Thread 0x7f8f72137700 (LWP 44941) "tvh:mi-main" 0x00007f8f8faf79f3 in futex_wait_cancelable
(private=<optimized out>, expected=0, futex_word=0x562a5723de98)
at ../sysdeps/unix/sysv/linux/futex-internal.h:88
36 Thread 0x7f8f71f36700 (LWP 44942) "tvh:mi-table" 0x00007f8f8faf79f3 in futex_wait_cancelabl
e (private=<optimized out>, expected=0, futex_word=0x562a571d8e10)
at ../sysdeps/unix/sysv/linux/futex-internal.h:88
37 Thread 0x7f8f71d35700 (LWP 44943) "tvh:mi-main" 0x00007f8f8faf79f3 in futex_wait_cancelable
(private=<optimized out>, expected=0, futex_word=0x562a571d8d08)
at ../sysdeps/unix/sysv/linux/futex-internal.h:88
38 Thread 0x7f8f71b34700 (LWP 44944) "tvh:mi-table" 0x00007f8f8faf79f3 in futex_wait_cancelabl
e (private=<optimized out>, expected=0, futex_word=0x562a57266a50)
at ../sysdeps/unix/sysv/linux/futex-internal.h:88
39 Thread 0x7f8f71933700 (LWP 44945) "tvh:mi-main" 0x00007f8f8faf79f3 in futex_wait_cancelable
(private=<optimized out>, expected=0, futex_word=0x562a57266948)
---Type <return> to continue, or q <return> to quit---
at ../sysdeps/unix/sysv/linux/futex-internal.h:88
40 Thread 0x7f8f71732700 (LWP 44946) "tvh:mi-table" 0x00007f8f8faf79f3 in futex_wait_cancelabl
e (private=<optimized out>, expected=0, futex_word=0x562a57264e50)
at ../sysdeps/unix/sysv/linux/futex-internal.h:88
41 Thread 0x7f8f71531700 (LWP 44947) "tvh:mi-main" 0x00007f8f8faf79f3 in futex_wait_cancelable
(private=<optimized out>, expected=0, futex_word=0x562a57264d48)
at ../sysdeps/unix/sysv/linux/futex-internal.h:88
42 Thread 0x7f8f71330700 (LWP 44948) "tvh:mi-table" 0x00007f8f8faf79f3 in futex_wait_cancelabl
e (private=<optimized out>, expected=0, futex_word=0x562a57237a80)
at ../sysdeps/unix/sysv/linux/futex-internal.h:88
43 Thread 0x7f8f7112f700 (LWP 44949) "tvh:mi-main" 0x00007f8f8faf79f3 in futex_wait_cancelable
(private=<optimized out>, expected=0, futex_word=0x562a57237978)
at ../sysdeps/unix/sysv/linux/futex-internal.h:88
44 Thread 0x7f8f70f2e700 (LWP 44950) "tvh:mi-table" 0x00007f8f8faf79f3 in futex_wait_cancelabl
e (private=<optimized out>, expected=0, futex_word=0x562a5721e130)
at ../sysdeps/unix/sysv/linux/futex-internal.h:88
45 Thread 0x7f8f70d2d700 (LWP 44951) "tvh:mi-main" 0x00007f8f8faf79f3 in futex_wait_cancelable
(private=<optimized out>, expected=0, futex_word=0x562a5721e028)
at ../sysdeps/unix/sysv/linux/futex-internal.h:88
46 Thread 0x7f8f70b2c700 (LWP 44952) "tvh:mi-table" 0x00007f8f8faf79f3 in futex_wait_cancelabl
e (private=<optimized out>, expected=0, futex_word=0x562a5723c760)
at ../sysdeps/unix/sysv/linux/futex-internal.h:88
47 Thread 0x7f8f7092b700 (LWP 44953) "tvh:mi-main" 0x00007f8f8faf79f3 in futex_wait_cancelable
```

```
(private=<optimized out>, expected=0, futex_word=0x562a5723c658)
  at ../sysdeps/unix/sysv/linux/futex-internal.h:88
48 Thread 0x7f8f7072a700 (LWP 44954) "tvh:mi-table" 0x00007f8f8faf79f3 in futex_wait_cancelabl
e (private=<optimized out>, expected=0, futex_word=0x562a572086b0)
  at ../sysdeps/unix/sysv/linux/futex-internal.h:88
49 Thread 0x7f8f70529700 (LWP 44955) "tvh:mi-main" 0x00007f8f8faf79f3 in futex_wait_cancelable
(private=<optimized out>, expected=0, futex_word=0x562a572085a8)
  at ../sysdeps/unix/sysv/linux/futex-internal.h:88
50 Thread 0x7f8f70328700 (LWP 44956) "tvh:mi-table" 0x00007f8f8faf79f3 in futex_wait_cancelabl
e (private=<optimized out>, expected=0, futex_word=0x562a57561aa0)
  at ../sysdeps/unix/sysv/linux/futex-internal.h:88
51 Thread 0x7f8f53fff700 (LWP 44957) "tvh:mi-main" 0x00007f8f8faf79f3 in futex_wait_cancelable
(private=<optimized out>, expected=0, futex_word=0x562a5756199c)
  at ../sysdeps/unix/sysv/linux/futex-internal.h:88
52 Thread 0x7f8f53dfe700 (LWP 44958) "tvh:mi-table" 0x00007f8f8faf79f3 in futex_wait_cancelabl
e (private=<optimized out>, expected=0, futex_word=0x562a572435f4)
  at ../sysdeps/unix/sysv/linux/futex-internal.h:88
53 Thread 0x7f8f53bfd700 (LWP 44959) "tvh:mi-main" 0x00007f8f8faf79f3 in futex_wait_cancelable
(private=<optimized out>, expected=0, futex_word=0x562a572434e8)
  at ../sysdeps/unix/sysv/linux/futex-internal.h:88
54 Thread 0x7f8f72b7c700 (LWP 44960) "tvh:satip-front" 0x00007f8f8e785bb7 in epoll_wait (epfd=
57, events=0x7f8f44000b80, maxevents=1, timeout=-1) at ../sysdeps/unix/sysv/linux/epoll_wait.c:30
55 Thread 0x7f8f539fc700 (LWP 44961) "tvh:satip-front" 0x00007f8f8e785bb7 in epoll_wait (epfd=
60, events=0x7f8f48000b80, maxevents=1, timeout=-1) at ../sysdeps/unix/sysv/linux/epoll_wait.c:30
56 Thread 0x7f8f537fb700 (LWP 44962) "tvh:satip-front" 0x00007f8f8e785bb7 in epoll_wait (epfd=
63, events=0x7f8f3c000b80, maxevents=1, timeout=-1) at ../sysdeps/unix/sysv/linux/epoll_wait.c:30
57 Thread 0x7f8f535fa700 (LWP 44963) "tvh:satip-front" 0x00007f8f8e785bb7 in epoll_wait (epfd=
54, events=0x7f8f40000b80, maxevents=1, timeout=-1) at ../sysdeps/unix/sysv/linux/epoll_wait.c:30
58 Thread 0x7f8f533f9700 (LWP 44964) "tvh:mi-table" 0x00007f8f8faf79f3 in futex_wait_cancelabl
e (private=<optimized out>, expected=0, futex_word=0x7f8f4c007e04)
  at ../sysdeps/unix/sysv/linux/futex-internal.h:88
59 Thread 0x7f8f531f8700 (LWP 44965) "tvh:mi-main" 0x00007f8f8faf79f3 in futex_wait_cancelable
(private=<optimized out>, expected=0, futex_word=0x7f8f4c007cf8)
  at ../sysdeps/unix/sysv/linux/futex-internal.h:88
60 Thread 0x7f8f52ff7700 (LWP 44966) "tvh:mi-table" 0x00007f8f8faf79f3 in futex_wait_cancelabl
e (private=<optimized out>, expected=0, futex_word=0x7f8f4c006240)
  at ../sysdeps/unix/sysv/linux/futex-internal.h:88
61 Thread 0x7f8f52df6700 (LWP 44967) "tvh:mi-main" 0x00007f8f8faf79f3 in futex_wait_cancelable
(private=<optimized out>, expected=0, futex_word=0x7f8f4c006138)
  at ../sysdeps/unix/sysv/linux/futex-internal.h:88
62 Thread 0x7f8f52bf5700 (LWP 44968) "tvh:mi-table" 0x00007f8f8faf79f3 in futex_wait_cancelabl
e (private=<optimized out>, expected=0, futex_word=0x7f8f4c005ad4)
  at ../sysdeps/unix/sysv/linux/futex-internal.h:88
63 Thread 0x7f8f529f4700 (LWP 44969) "tvh:mi-main" 0x00007f8f8faf79f3 in futex_wait_cancelable
(private=<optimized out>, expected=0, futex_word=0x7f8f4c0059cc)
  at ../sysdeps/unix/sysv/linux/futex-internal.h:88
64 Thread 0x7f8f527f3700 (LWP 44970) "tvh:mi-table" __lll_lock_wait () at ../sysdeps/unix/sysv
/linux/x86_64/lowlevellock.S:135
65 Thread 0x7f8f525f2700 (LWP 44971) "tvh:mi-main" 0x00007f8f8faf79f3 in futex_wait_cancelable
(private=<optimized out>, expected=0, futex_word=0x7f8f4c00531c)
  at ../sysdeps/unix/sysv/linux/futex-internal.h:88
66 Thread 0x7f8f523f1700 (LWP 44972) "tvh:satip-front" 0x00007f8f8e785bb7 in epoll_wait (epfd=
72, events=0x7f8f38000b80, maxevents=1, timeout=-1) at ../sysdeps/unix/sysv/linux/epoll_wait.c:30
67 Thread 0x7f8f521f0700 (LWP 44973) "tvh:satip-front" 0x00007f8f8e785bb7 in epoll_wait (epfd=
69, events=0x7f8f34000b80, maxevents=1, timeout=-1) at ../sysdeps/unix/sysv/linux/epoll_wait.c:30
68 Thread 0x7f8f51fef700 (LWP 44974) "tvh:satip-front" 0x00007f8f8e785bb7 in epoll_wait (epfd=
75, events=0x7f8f30000b80, maxevents=1, timeout=-1) at ../sysdeps/unix/sysv/linux/epoll_wait.c:30
69 Thread 0x7f8f51dee700 (LWP 44975) "tvh:satip-front" 0x00007f8f8e785bb7 in epoll_wait (epfd=
53, events=0x7f8f2c000b80, maxevents=1, timeout=-1) at ../sysdeps/unix/sysv/linux/epoll_wait.c:30
70 Thread 0x7f8f51bed700 (LWP 44976) "tvh:mi-table" 0x00007f8f8faf79f3 in futex_wait_cancelabl
e (private=<optimized out>, expected=0, futex_word=0x7f8f4c00a384)
---Type <return> to continue, or q <return> to quit---
  at ../sysdeps/unix/sysv/linux/futex-internal.h:88
71 Thread 0x7f8f519ec700 (LWP 44977) "tvh:mi-main" 0x00007f8f8faf79f3 in futex_wait_cancelable
(private=<optimized out>, expected=0, futex_word=0x7f8f4c00a27c)
  at ../sysdeps/unix/sysv/linux/futex-internal.h:88
72 Thread 0x7f8f517eb700 (LWP 44978) "tvh:mi-table" __lll_lock_wait () at ../sysdeps/unix/sysv
```

```

/linux/x86_64/lowlevellock.S:135
 73 Thread 0x7f8f515ea700 (LWP 44979) "tvh:mi-main" 0x00007f8f8faf79f3 in futex_wait_cancelable
(private=<optimized out>, expected=0, futex_word=0x7f8f4c0091fc)
  at ../sysdeps/unix/sysv/linux/futex-internal.h:88
 74 Thread 0x7f8f513e9700 (LWP 44980) "tvh:mi-table" 0x00007f8f8faf79f3 in futex_wait_cancelabl
e (private=<optimized out>, expected=0, futex_word=0x7f8f4c008b74)
  at ../sysdeps/unix/sysv/linux/futex-internal.h:88
 75 Thread 0x7f8f511e8700 (LWP 44981) "tvh:mi-main" __pthread_mutex_lock_full (mutex=0x7f8e9400
2378) at ../nptl/pthread_mutex_lock.c:313
 76 Thread 0x7f8f50fe7700 (LWP 44982) "tvh:mi-table" 0x00007f8f8faf79f3 in futex_wait_cancelabl
e (private=<optimized out>, expected=0, futex_word=0x7f8f4c0083b0)
  at ../sysdeps/unix/sysv/linux/futex-internal.h:88
 77 Thread 0x7f8f50de6700 (LWP 44983) "tvh:mi-main" 0x00007f8f8faf79f3 in futex_wait_cancelable
(private=<optimized out>, expected=0, futex_word=0x7f8f4c0082a8)
  at ../sysdeps/unix/sysv/linux/futex-internal.h:88
 78 Thread 0x7f8ea71f8700 (LWP 1727) "tvh:tcp-start" __l1l_lock_wait () at ../sysdeps/unix/sysv
/linux/x86_64/lowlevellock.S:135
 79 Thread 0x7f8f7297b700 (LWP 4353) "tvh:tcp-start" __l1l_lock_wait () at ../sysdeps/unix/sysv
/linux/x86_64/lowlevellock.S:135
 80 Thread 0x7f8f50be5700 (LWP 31697) "tvh:tcp-start" __l1l_lock_wait () at ../sysdeps/unix/sys
v/linux/x86_64/lowlevellock.S:135
 81 Thread 0x7f8ea7dfe700 (LWP 48132) "tvh:dvr" 0x00007f8f8faf79f3 in futex_wait_cancelable (pr
ivate=<optimized out>, expected=0, futex_word=0x562a589d2290) at ../sysdeps/unix/sysv/linux/futex-
internal.h:88
 82 Thread 0x7f8ea59ec700 (LWP 32182) "tvh:dvr" 0x00007f8f8faf79f3 in futex_wait_cancelable (pr
ivate=<optimized out>, expected=0, futex_word=0x562a5a1da350) at ../sysdeps/unix/sysv/linux/futex-
internal.h:88
 83 Thread 0x7f8ea5dee700 (LWP 32183) "tvh:dvr" 0x00007f8f8faf79f3 in futex_wait_cancelable (pr
ivate=<optimized out>, expected=0, futex_word=0x562a5a034e14) at ../sysdeps/unix/sysv/linux/futex-
internal.h:88
 84 Thread 0x7f8ea4fe7700 (LWP 9893) "tvh:dvr" 0x00007f8f8faf79f3 in futex_wait_cancelable (pri
vate=<optimized out>, expected=0, futex_word=0x562a592abca4) at ../sysdeps/unix/sysv/linux/futex-i
nternal.h:88

```

Unfortunately I did not run a "thread apply all bt full" but only a "bt full", so attached is what I have available, even though this probably not be much help (I really hope that the "info threads" will help to figure out what happened though) sad.png

Subtasks:

Bug # 5412: Interface crashes

Rejected

Associated revisions

Revision 80ea669a - 2018-12-03 07:40 - Jaroslav Kysela

tvh_thread: do not use debug code when not activated, issue #5353, issue #5389

Revision 11f6531a - 2018-12-09 16:12 - Jaroslav Kysela

fix some problems detected by cppcheck, issue #5353

Revision f0524db4 - 2018-12-09 16:34 - Jaroslav Kysela

iptv http input: play with the locking, issue #5353

Revision bceba085 - 2018-12-10 14:36 - Jaroslav Kysela

timers - change locking schema, fixes #5413, issue #5353

Revision 7ce391fc - 2018-12-11 08:24 - Jaroslav Kysela

iptv: add missing lock to the iptv_http_kick_cb, fixes #5415, issue #5353

Revision fb4410ad - 2018-12-11 08:51 - Jaroslav Kysela

iptv: correction for the previous patch, fixes #5415, issue #5353

Revision bd662457 - 2019-01-13 20:17 - Jaroslav Kysela

parse_ac3: avoid the endless loop for the AC3/EAC3 auto-detection, issue #5353

History

#1 - 2018-11-26 22:05 - Flole Systems

Good/Bad new is I got another crash and this time I got the backtrace right and also got a core file.

I have removed all sensitive Information (hopefully) and uploaded the backtrace here: <https://cloud.flole.de/index.php/s/X2rtHB9GHPY5HR8>

If anyone of the developers need the core file, feel free to email me at the address in my profile.

#2 - 2018-11-27 08:23 - Luis Alves

Not very helpful I think.

I had a similar issue - please check: <https://tvheadend.org/issues/5295>

I wrote step by step what I have done to find out where the lock is (basically when you have the lock, print the "global_lock", find out in which thread it is stuck and why).

#3 - 2018-11-27 08:25 - Jaroslav Kysela

Try latest, add '--thrdebug 1' to the tvh command line arguments. Show 'CFGDIR/mutex-deadlock.txt' contents after the crash.

#4 - 2018-11-27 08:25 - Luis Alves

Step-by-step on post [#10](#) (<https://tvheadend.org/issues/5295#note-10>)

#5 - 2018-11-27 08:53 - Jaroslav Kysela

[Luis Parada](#), basically the new thread debug code does something similar but without gdb.

#6 - 2018-11-27 10:40 - Luis Alves

Jaroslav Kysela wrote:

[Luis Parada](#), basically the new thread debug code does something similar but without gdb.

Yeah! I submitted my post before seeing yours.

That's a cool feature to easily debug these nasty lockups.

#7 - 2018-11-30 17:37 - Flole Systems

Unfortunately I am unable to reproduce the crash reliably, no clue how I managed to have it crash 2 times in a few minutes. I tried to use the print global_locks on the Stacktrace with the original core dump and binary nothing was found.

#8 - 2018-12-01 11:21 - Ricardo Rocha

Jaroslav Kysela wrote:

Try latest, add '--thrdebug 1' to the tvh command line arguments. Show 'CFGDIR/mutex-deadlock.txt' contents after the crash.

i did enable --thrdebug 1 on options and with 1602 i am getting constant stucked system with messages :

```
Dec 01 11:12:27 host tvheadend[7424]: mpegts: too much queued table input data (over 2MB) for SAT>IP DVB-C Tuner, discarding new
```

however i can't see any mutex-deadlock.txt on my cfg dir

#9 - 2018-12-01 11:46 - Ricardo Rocha

```
Dec 01 11:40:43 lp7v tvheadend[7969]: capmt: osc: No free adapter slot available for service "Name"  
Dec 01 11:41:08 lp7v tvheadend[7969]: mpegts: too much queued table input data (over 2MB) for SAT>IP DVB-C Tuner, discarding new
```

it's the 11 time it happens only on the morning... i was checking and last 4 always appear after the capmt: osc: No free adapter slot available for service "Name"

#10 - 2018-12-01 16:49 - Jaroslav Kysela

I added another 'profile' code which can be activated using this: <https://tvheadend.org/projects/tvheadend/wiki/Debugging#Mutex-profiling>

#11 - 2018-12-02 22:10 - Flole Systems

Will that code also help if a mutex never get's a lock because they are in a situation where 2 locks are waiting for the opposite one to become available?

#12 - 2018-12-02 22:56 - Pablo R.

Why if running tvh with '--thrdebug 10020' it never crash, and when I dont put it hangs instantly?

#13 - 2018-12-03 07:40 - Jaroslav Kysela

Pablo R. wrote:

Why if running tvh with '--thrdebug 10020' it never crash, and when I dont put it hangs instantly?

Really, really good point. v4.3-1617-g80ea669a5

#14 - 2018-12-05 22:28 - Flole Systems

Got another lockup right now, and the thrdebug did not output anything yet (there was some output during the last few days so it's definitely working). Right now it's still in that hung state, I will now do the gdb stuff and then restart, so if there's anything I should do now for testing I would need to know that in the next 15 or so minutes.

I assume that what I said earlier just happened: The mutex never get's the lock so it will never output that it took too long.

#15 - 2018-12-05 22:34 - Pablo R.

Flole Systems wrote:

Got another lockup right now, and the thrdebug did not output anything yet (there was some output during the last few days so it's definitely working). Right now it's still in that hung state, I will now do the gdb stuff and then restart, so if there's anything I should do now for testing I would need to know that in the next 15 or so minutes.

I assume that what I said earlier just happened: The mutex never get's the lock so it will never output that it took too long.

Maybe Valgrind can help?

#16 - 2018-12-05 22:35 - Flole Systems

Ahhh this looks very promising I think (although I am unable to interpret what exactly this all means):

```
(gdb) print global_lock
$1 = {mutex = pthread_mutex_t = {Type = Normal, Status = Acquired, possibly with waiters, Owner ID = 28562, Robust = No, Shared = No, Protocol = None}, thread = 140383409391360, filename = 0x5616a7cfff13a "src/main.c", lineno = 623, tstamp = 4264962751427, link = {tqe_next = 0x7fad5c02f3c0, tqe_prev = 0x5616a82f9ca0 <thrwatch_mutexes>}}
(gdb) thread find 28562
Thread 7 has target id 'Thread 0x7fad8f3fc700 (LWP 28562)'
(gdb) info threads 7
  Id   Target Id         Frame
  7    Thread 0x7fad8f3fc700 (LWP 28562) "tvh:mtimer" __l1ll_lock_wait () at ../sysdeps/unix/sysv/linux/x86_64/lowlevellock.S:135
(gdb) thread 7
[Switching to thread 7 (Thread 0x7fad8f3fc700 (LWP 28562))]
#0  __l1ll_lock_wait () at ../sysdeps/unix/sysv/linux/x86_64/lowlevellock.S:135
#135  in ../sysdeps/unix/sysv/linux/x86_64/lowlevellock.S
(gdb) bt
#0  __l1ll_lock_wait () at ../sysdeps/unix/sysv/linux/x86_64/lowlevellock.S:135
#1  0x00007fad97f04023 in __GI___pthread_mutex_lock (mutex=0x7fad5c02f3c0) at ../nptl/pthread_mutex_lock.c:78
#2  0x00005616a6d0f3a6 in tvh__mutex_lock (mutex=0x7fad5c02f3c0, filename=0x5616a7dcfe00 "src/input/mpegts/mpegts_service.c", lineno=781) at src/tvh_thread.c:208
#3  0x00005616a6e2d575 in mpegts_service_pid_list_ (t=0x5616aa902e00, owner=0x5616aa902e00) at src/input/mpegts/mpegts_service.c:781
#4  0x00005616a6e2d6f5 in mpegts_service_pid_list (t=0x5616aa902e00) at src/input/mpegts/mpegts_service.c:800
#5  0x00005616a6d44ad3 in subscription_create_msg (s=0x7fad4c030380, lang=0x0) at src/subscriptions.c:1067
#6  0x00005616a6d44e3e in subscription_status_callback (p=0x0) at src/subscriptions.c:1125
#7  0x00005616a6cf7142 in mtimer_thread (aux=0x0) at src/main.c:646
#8  0x00005616a6d0eebb in timer_wrapper (p=0x5616aa2c1e90) at src/tvh_thread.c:93
#9  0x00007fad97f016db in start_thread (arg=0x7fad8f3fc700) at pthread_create.c:463
#10 0x00007fad96b9588f in clone () at ../sysdeps/unix/sysv/linux/x86_64/clone.S:95
```

Does this help?

Unfortunately valgrind can not be attached to a running program though.

#17 - 2018-12-05 23:13 - Flole Systems

I just searched where mi_output_lock is in use, and it's also being used by sat ip and the descrambler..... I was watching 2 different encrypted channels using tvheadend as sat ip client at that moment, maybe the issue lies somewhere in there... I have just checked through all the occurrences

of mi_output_lock and didn't find any obvious issues, but I do not know the code well enough to see them just by looking at them.

Maybe a possible solution would be to use pthread_mutex_trylock() in a while loop to do the thrdebug stuff and write a message when we're stuck waiting for a lock (for example longer than 5 seconds), I do not know how this will affect performance though to have such a loop. Maybe even only do that in case the debugging is enabled.

#18 - 2018-12-06 15:59 - Jaroslav Kysela

You're very close, but we need to know, which thread acquired the mi_output_lock, too. Those commands might help (example for output in comment#16):

```
gdb) frame 2 # look for frame with tvh__mutex_lock function
gdb) print mutex
... continue with backtrace of thread (Owner ID) shown
```

It seems that there is somewhere lock disorder like (global_lock->mi_output_lock and mi_output_lock->global_lock).

#19 - 2018-12-08 00:41 - Flole Systems

As I have written a core file at that lockup, I tried to use that one now to get the information we need. What I did was (I was unsure if you wanted the "frame 2" from thread 7 or the main thread):

```
#gdb tvheadend core
(gdb) frame 2
#2 0x00007fad97f07d9d in __pthread_cond_wait_common (abstime=0x7ffe003238e0, mutex=0x5616a84cfe00 <global_loc
k>, cond=0x5616a836f080 <gtimer_cond>) at pthread_cond_wait.c:645
645 pthread_cond_wait.c: No such file or directory.
(gdb) print mutex
$1 = (pthread_mutex_t *) 0x5616a84cfe00 <global_lock>
(gdb) thread 7
[Switching to thread 7 (Thread 0x7fad8f3fc700 (LWP 28562))]
#0 __lll_lock_wait () at ../sysdeps/unix/sysv/linux/x86_64/lowlevellock.S:135
135 in ../sysdeps/unix/sysv/linux/x86_64/lowlevellock.S
(gdb) frame 2
#2 0x00005616a6d0f3a6 in tvh__mutex_lock (mutex=0x7fad5c02f3c0, filename=0x5616a7dcfe00 "src/input/mpegts/mpe
gts_service.c", lineno=781) at src/tvh_thread.c:208
208 int r = pthread_mutex_lock(&mutex->mutex);
(gdb) print mutex
$2 = (tvh_mutex_t *) 0x7fad5c02f3c0
```

I should be able to get the information we need from that core dump, right? Unfortunately I can not see the thread that should be shown there, or does this only work during a "real lockup" and not when using the core dump? As far as I know core dumps can be used in gdb just like this would be at the time of the lockup.

Also I tried something myself (and I think I found the "partner" of the lockup, at least we now have 2 threads that are somehow "connected" both waiting for a mutex):

```
(gdb) thread 7
(gdb) bt
#0 __lll_lock_wait () at ../sysdeps/unix/sysv/linux/x86_64/lowlevellock.S:135
#1 0x00007fad97f04023 in __GI___pthread_mutex_lock (mutex=0x7fad5c02f3c0) at ../nptl/pthread_mutex_lock.c:78
#2 0x00005616a6d0f3a6 in tvh__mutex_lock (mutex=0x7fad5c02f3c0, filename=0x5616a7dcfe00 "src/input/mpegts/mpe
gts_service.c", lineno=781) at src/tvh_thread.c:208
#3 0x00005616a6e2d575 in mpegts_service_pid_list_ (t=0x5616aa902e00, owner=0x5616aa902e00) at src/input/mpeg
ts/mpegts_service.c:781
#4 0x00005616a6e2d6f5 in mpegts_service_pid_list (t=0x5616aa902e00) at src/input/mpegts/mpegts_service.c:800
#5 0x00005616a6d44ad3 in subscription_create_msg (s=0x7fad4c030380, lang=0x0) at src/subscriptions.c:1067
#6 0x00005616a6d44e3e in subscription_status_callback (p=0x0) at src/subscriptions.c:1125
#7 0x00005616a6cf7142 in mtimer_thread (aux=0x0) at src/main.c:646
#8 0x00005616a6d0eebb in thread_wrapper (p=0x5616aa2c1e90) at src/tvh_thread.c:93
#9 0x00007fad97f016db in start_thread (arg=0x7fad8f3fc700) at pthread_create.c:463
#10 0x00007fad96b9588f in clone () at ../sysdeps/unix/sysv/linux/x86_64/clone.S:95
(gdb) p *((tvh_mutex_t *) 0x7fad5c02f3c0)
$7 = {mutex = pthread_mutex_t = {Type = Normal, Status = Acquired, possibly with waiters, Owner ID = 28809, Ro
bust = No, Shared = No, Protocol = None}, thread = 140382425728768, filename = 0x5616a7dcd670 "src/input/mpeg
ts/mpegts_input.c", lineno = 1697,
  tstamp = 4264962719426, link = {tqe_next = 0x0, tqe_prev = 0x5616a84cfe48 <global_lock+72>}}
(gdb) thread find 28809
Thread 77 has target id 'Thread 0x7fad549e4700 (LWP 28809)'quit
(gdb) info threads 77
  Id   Target Id         Frame
  77   Thread 0x7fad549e4700 (LWP 28809) 0x00007fad97f0bf27 in __pause_nocancel () at ../sysdeps/unix/sysv/lin
ux/pause.c:41
```

```
(gdb) thread 77
[Switching to thread 77 (Thread 0x7fad549e4700 (LWP 28809))]
#0 0x00007fad97f0bf27 in __pause_nocancel () at ../sysdeps/unix/sysv/linux/pause.c:41
41      ../sysdeps/unix/sysv/linux/pause.c: No such file or directory.
(gdb) bt
#0 0x00007fad97f0bf27 in __pause_nocancel () at ../sysdeps/unix/sysv/linux/pause.c:41
#1 0x00007fad97f03a45 in __pthread_mutex_lock_full (mutex=0x7facec0311f8) at ../nptl/pthread_mutex_lock.c:429
#2 0x00005616a6d0f3a6 in tvh__mutex_lock (mutex=0x7facec0311f8, filename=0x5616a7dcdca0 "src/input/mpegts/tsdemux.c", lineno=265) at src/tvh_thread.c:208
#3 0x00005616a6e1ed68 in ts_recv_raw (t=0x7facec031010, tspos=5603360492, tsb=0x7fad2c0038a1 "G@\\021\\025", len=188) at src/input/mpegts/tsdemux.c:265
#4 0x00005616a6e1b4f2 in mpegts_input_process (mi=0x7fad5c02f140, mpkt=0x7fad2c003880) at src/input/mpegts/mpegts_input.c:1467
#5 0x00005616a6e1c30f in mpegts_input_thread (p=0x7fad5c02f140) at src/input/mpegts/mpegts_input.c:1707
#6 0x00005616a6d0e6bb in thread_wrapper (p=0x7fad5800e6d0) at src/tvh_thread.c:93
#7 0x00007fad97f016db in start_thread (arg=0x7fad549e4700) at pthread_create.c:463
#8 0x00007fad96b9588f in clone () at ../sysdeps/unix/sysv/linux/x86_64/clone.S:95
(gdb)
```

#20 - 2018-12-08 09:07 - Luis Alves

It's a mess! smile.png

- 1) Thread 7 acquired "s_stream_mutex" (src/subscriptions.c line 1064)
- 2) Thread 77 acquired "mi_output_lock" (src/input/mpegts/tsdemux.c line 265)
- 3) Thread 7 is waiting to get "mi_output_lock" locked by thread 77
- 4) Thread 77 is waiting to get "s_stream_mutex" locked by thread 7

#21 - 2018-12-08 09:10 - Luis Alves

Oopps, this is wrong: "1) Thread 7 acquired "s_stream_mutex" (src/subscriptions.c line 1064)"
Someone else has that lock...

#22 - 2018-12-08 09:18 - Flole Systems

So they are locked up waiting for each other, right?

Line 1064 in src/subscriptions.c is an unlock by the way, so it did not acquire it there, right?

Any idea how I could further track this down? We now know what thread has what lock and where they're stuck, is this already enough so Jaroslav know's where the issue is (because he knows all the code and how it's supposed to work)?

#23 - 2018-12-08 09:24 - Flole Systems

I started to go backwards during

```
#2 0x00005616a6d0f3a6 in tvh__mutex_lock (mutex=0x7fad5c02f3c0, filename=0x5616a7dcfe00 "src/input/mpegts/mpegts_service.c", lineno=781) at src/tvh_thread.c:208
#3 0x00005616a6e2d575 in mpegts_service_pid_list_ (t=0x5616aa902e00, owner=0x5616aa902e00) at src/input/mpegts/mpegts_service.c:781
#4 0x00005616a6e2d6f5 in mpegts_service_pid_list (t=0x5616aa902e00) at src/input/mpegts/mpegts_service.c:800
#5 0x00005616a6d44ad3 in subscription_create_msg (s=0x7fad4c030380, lang=0x0) at src/subscriptions.c:1067
```

and noticed the following: in subscriptions.c:1064 the s_stream_mutex is unlocked, in subscriptions.c:1067 mpegts_service.c:781 the mi_output_lock is being waited for. Now what could be done is move the tv_mutex_lock(s_stream_mutex) in subscriptions.c:1064 to a line after subscriptions.c:1067 (it would be 1082 most likely).

That could be a completely wrong analysis and there might be a reason this is done the way it is done, and possibly tracing the other lock backwards would show something similar....

#24 - 2018-12-08 09:32 - Flole Systems

Just to get a little more experienced I've done the same "backwards-tracking" on the other lock aswell, there the output lock is acquired in src/input/mpegts/mpegts_input.c:1697, then in src/input/mpegts/mpegts_input.c:1707 the mpegts_input_process is called which needs the s_stream_mutex, and finally in src/input/mpegts/mpegts_input.c:1710 the mi_output_lock is unlocked again. Moving that unlock backwards a few lines is probably dangerous though.

#25 - 2018-12-08 09:40 - Luis Alves

Flole Systems wrote:

So they are locked up waiting for each other, right?

No my mistake.

Line 1064 in src/subscriptions.c is an unlock by the way, so it did not acquire it there, right?

Yep, the mistake was right there.

If you still have the core file, try to find the owner of the "s_stream_mutex" mutex. Something like:

```
thread 77
p *((tvh_mutex_t *) 0x7facec0311f8)
```

Then find the thread and do a bt.

How big is the core file? If not too big, upload it somewhere.

#26 - 2018-12-08 10:18 - Flole Systems

Now it gets weird:

```
gdb) thread 77
[Switching to thread 77 (Thread 0x7fad549e4700 (LWP 28809))]
#0  0x00007fad97f0bf27 in __pause_nocancel () at ../sysdeps/unix/sysv/linux/pause.c:41
41      ../sysdeps/unix/sysv/linux/pause.c: Datei oder Verzeichnis nicht gefunden.
(gdb) bt
#0  0x00007fad97f0bf27 in __pause_nocancel () at ../sysdeps/unix/sysv/linux/pause.c:41
#1  0x00007fad97f03a45 in __pthread_mutex_lock_full (mutex=0x7facec0311f8) at ../nptl/pthread_mutex_lock.c:429
#2  0x00005616a6d0f3a6 in tvh__mutex_lock (mutex=0x7facec0311f8, filename=0x5616a7dcdca0 "src/input/mpegts/tsdemux.c", lineno=265) at src/tvh_thread.c:208
#3  0x00005616a6e1ed68 in ts_recv_raw (t=0x7facec031010, tspos=5603360492, tsb=0x7fad2c0038a1 "G@\\021\\025", len=188) at src/input/mpegts/tsdemux.c:265
#4  0x00005616a6e1b4f2 in mpegts_input_process (mi=0x7fad5c02f140, mpkt=0x7fad2c003880) at src/input/mpegts/mpegts_input.c:1467
#5  0x00005616a6e1c30f in mpegts_input_thread (p=0x7fad5c02f140) at src/input/mpegts/mpegts_input.c:1707
#6  0x00005616a6d0eebb in thread_wrapper (p=0x7fad5800e6d0) at src/tvh_thread.c:93
#7  0x00007fad97f016db in start_thread (arg=0x7fad549e4700) at pthread_create.c:463
#8  0x00007fad96b9588f in clone () at ../sysdeps/unix/sysv/linux/x86_64/clone.S:95
(gdb) p *((tvh_mutex_t *) 0x7facec0311f8)
$1 = {mutex = pthread_mutex_t = {Type = Normal, Status = Acquired, possibly with waiters, Owner ID = 394242232, Robust = No, Shared = No, Protocol = Priority inherit}, thread = 140380670726864, filename = 0x7facec031110 "s\\312\\342\\246\\026v", lineno = 541938527, tstamp = 3699224701937669705, link = {tqe_next = 0x585649445f565020, tqe_prev = 0x565020345649445f}}
(gdb) thread find 394242232
No threads match '394242232'
(gdb)
```

My thread IDs don't go past 29000 so that is probably not the correct thread.....

I think the first thing I wrote might actually be it: The mutex is unlocked before the other one is locked, so they might lock out each other?

Or do we have a memory corruption here? That would explain the gdb output I just posted.

As the core file contains passwords and other stuff I'd rather not post it (and it's 6.5GB big [biggrin.png](#))

#27 - 2018-12-08 10:53 - Flole Systems

I just tried to use valgrind with ttheadend and it goes absolutely crazy....

```
More than 10000000 total errors detected. I'm not reporting any more.
Final error counts will be inaccurate. Go fix your program!
```

Most of them are "Use of uninitialised value of size 8" or "Conditional jump or move depends on uninitialised value(s)" and in the end I have

```
definitely lost: 17,738 bytes in 29 blocks
indirectly lost: 264 bytes in 1 blocks
possibly lost: 0 bytes in 0 blocks
still reachable: 5,403 bytes in 40 blocks
suppressed: 72 bytes in 2 blocks
```

That's quite a lot if that is accurate! Also I tried actually using ttheadend while using valgrind but streaming was almost impossible as it was so slow....

#28 - 2018-12-08 14:59 - Luis Alves

Instead of casting the pointer to "tvh_mutex_t" try just printing mutex on the current frame:

```
thread 77  
print mutex
```

#29 - 2018-12-08 17:58 - Jaroslav Kysela

I continue to improve the mutex code in the latest (fixed some bugs, added magic check for the memory overwrite), so I would suggest to upgrade and retest with this:

<https://tvheadend.org/projects/tvheadend/wiki/Debugging#Mutex-profiling>

The debugging code should print all locked and waiting mutexes with the source code filename/line numbers. I also added magic number checks (to detect the memory corruptions) for mutex and internal timers.

#30 - 2018-12-08 21:08 - Luis Alves

I just got the "too much queued..." msg!

Updated to latest git but something is wrong... It crashes on start:

mutex-deadlock.txt:

```
REASON: magic  
mutex 0x7ff8217ca778 locked in: (null):0 (thread 0)
```

log:

```
Dec 8 21:00:32 server tvheadend[793]: CRASH: Signal: 6 in PRG: /home/tvheadend/tvheadend/build.linux/tvheadend (4.3-1637~g1fad380e1) [d8aad91a922ddf898c12fd3a882170df6ebcf9fb] CWD: /home/tvheadend  
Dec 8 21:00:32 server tvheadend[793]: CRASH: Fault address 0x3e900000319 (N/A)  
Dec 8 21:00:32 server tvheadend[793]: CRASH: Loaded libraries: linux-vdso.so.1 /usr/lib/x86_64-linux-gnu/libd  
vbcsa.so.1 /usr/lib/x86_64-linux-gnu/libssl.so.1.1 /usr/lib/x86_64-linux-gnu/libcrypto.so.1.1 /lib/x86_64-linu  
x-gnu/libz.so.1 /usr/lib/x86_64-linux-gnu/libpcre2-8.so.0 /usr/lib/x86_64-linux-gnu/liburiparser.so.1 /usr/lib  
/x86_64-linux-gnu/libavahi-common.so.3 /usr/lib/x86_64-linux-gnu/libavahi-client.so.3 /usr/local/lib/libva.so.  
2 /lib/x86_64-linux-gnu/libdbus-1.so.3 /lib/x86_64-linux-gnu/libdl.so.2 /lib/x86_64-linux-gnu/libpthread.so.0  
/lib/x86_64-linux-gnu/libm.so.6 /lib/x86_64-linux-gnu/librt.so.1 /lib/x86_64-linux-gnu/libmvec.so.1 /usr/lib/x  
86_64-linux-gnu/libstdc++.so.6 /usr/local/lib/libva-drm.so.2 /usr/local/lib/libva-x11.so.2 /usr/lib/x86_64-lin  
ux-gnu/libX11.so.6 /lib/x86_64-linux-gnu/libc.so.6 /lib/x86_64-linux-gnu/libsystemd.so.0 /lib64/ld-linux-x86-6  
4.so.2 /lib/x86_64-linux-gnu/libgcc_s.so.1 /usr/lib/x86_64-linux-gnu/libdrm.so.2 /usr/lib/x86_64-linux-gnu/lib  
Xext.so.6 /usr/lib/x86_64-linux-gnu/libXfixes.so.3 /usr/lib/x86_64-linux  
Dec 8 21:00:32 server tvheadend[793]: CRASH: Register dump [23]: 0000000000000000000000000000000000000000  
00080000000000000024600000000000004200000000000000200007ff8215c998000007ff8217ca7680000000000000200007ff821  
5c965000007ff8215c98a80000000000000000000000000000000000000000000000000000000000000007ff85fd5be9700007ff8215c965000007ff8  
5fd5be97000000000000246002b0000000003300000000000000000000000000000000000000000000000000000fffff7fbbba1300000000000000000  
Dec 8 21:00:32 server tvheadend[793]: CRASH: STACKTRACE  
Dec 8 21:00:32 server tvheadend[793]: CRASH: /SCRATCH/repos/tvheadend/src/trap.c:176 0x55c2e2f489ed 0x55c2e2d29000  
Dec 8 21:00:33 server tvheadend[793]: CRASH: ??:0 0x7ff8613ba890 0x7ff8613a8000  
Dec 8 21:00:33 server tvheadend[793]: CRASH: gsignal+0xc7 (/lib/x86_64-linux-gnu/libc.so.6)  
Dec 8 21:00:33 server tvheadend[793]: CRASH: abort+0x141 (/lib/x86_64-linux-gnu/libc.so.6)  
Dec 8 21:00:33 server tvheadend[793]: CRASH: /SCRATCH/repos/tvheadend/src/tvh_thread.c:487 0x55c2e2f0961e 0x55c2e2d29000  
Dec 8 21:00:33 server tvheadend[793]: CRASH: /SCRATCH/repos/tvheadend/src/tvh_thread.c:148 0x55c2e2f09669 0x55c2e2d29000  
Dec 8 21:00:33 server tvheadend[793]: CRASH: /SCRATCH/repos/tvheadend/src/tvh_thread.c:359 0x55c2e2f0a015 0x55c2e2d29000  
Dec 8 21:00:33 server tvheadend[793]: CRASH: /SCRATCH/repos/tvheadend/src/htsp_server.c:3342 0x55c2e2f3981d 0x55c2e2d29000  
Dec 8 21:00:33 server tvheadend[793]: CRASH: /SCRATCH/repos/tvheadend/src/tvh_thread.c:91 0x55c2e2f09438 0x55c2e2d29000
```

(had to revert the last 2 commits)

Will analyze the core dump from previous run now...

#31 - 2018-12-08 21:22 - Luis Alves

I don't get it...

On my core dump global_lock got stuck at:

```
(gdb) print global_lock  
$1 = {mutex = pthread_mutex_t = {Type = Normal, Status = Acquired, possibly with waiters, Owner ID = 19443, Ro
```

```
bust = No, Shared = No,
  Protocol = None}, tid = 0, filename = 0x0, lineno = 0, tstamp = 0, waiters = {lh_first = 0x0}, link = {tqe
_next = 0x0,
  tqe_prev = 0x0}}
```

```
(gdb) thread find 19443
Thread 7 has target id 'Thread 0x7f84c1569700 (LWP 19443)'
```

```
(gdb) thread 7
[Switching to thread 7 (Thread 0x7f84c1569700 (LWP 19443))]
#0 0x00007f84c5b639f3 in futex_wait_cancelable (private=<optimized out>, expected=0, futex_word=0x5562046a7e8
8 <http_cond+40>)
  at ../sysdeps/unix/sysv/linux/futex-internal.h:88
#8  ../sysdeps/unix/sysv/linux/futex-internal.h: No such file or directory.
(gdb) bt
```

```
#0 0x00007f84c5b639f3 in futex_wait_cancelable (private=<optimized out>, expected=0, futex_word=0x5562046a7e8
8 <http_cond+40>)
  at ../sysdeps/unix/sysv/linux/futex-internal.h:88
#1 __pthread_cond_wait_common (abstime=0x0, mutex=0x5562046a7ea0 <http_lock>, cond=0x5562046a7e60 <http_cond>
) at pthread_cond_wait.c:502
#2 __pthread_cond_wait (cond=cond@entry=0x5562046a7e60 <http_cond>, mutex=mutex@entry=0x5562046a7ea0 <http_lo
ck>)
  at pthread_cond_wait.c:655
#3 0x0000556202f36e64 in tvh_cond_wait (cond=cond@entry=0x5562046a7e60 <http_cond>, mutex=mutex@entry=0x55620
46a7ea0 <http_lock>)
  at src/tvh_thread.c:334
#4 0x0000556202f8b2cf in http_client_close (hc=0x7f848814d9e0) at src/httpc.c:1637
#5 0x00005562030487b7 in iptv_http_stop (mi=<optimized out>, im=0x556205ae23a0) at src/input/mpegts/iptv/iptv
_http.c:562
#6 0x0000556203046793 in iptv_input_stop_mux (mi=0x5562058eae0, mmi=<optimized out>) at src/input/mpegts/ipt
v/iptv.c:445
#7 0x0000556203001129 in mpegts_mux_stop (mm=0x556205ae23a0, force=<optimized out>, reason=0) at src/input/mp
egts/mpegts_mux.c:869
#8 0x00005562030054cd in mpegts_service_stop (t=0x556205ae2a50) at src/input/mpegts/mpegts_service.c:432
#9 0x0000556202f5e0d5 in service_stop (t=t@entry=0x556205ae2a50) at src/service.c:259
#10 0x0000556202f5b87c in subscription_unlink_service0 (s=s@entry=0x7f82ec31bfe0, reason=reason@entry=101, res
ched=resched@entry=1)
  at src/subscriptions.c:164
#11 0x0000556202f5c226 in subscription_reschedule () at src/subscriptions.c:397
#12 0x0000556202f28ecc in mtimer_thread (aux=<optimized out>) at src/main.c:646
#13 0x0000556202f36348 in thread_wrapper (p=0x556205136130) at src/tvh_thread.c:87
#14 0x00007f84c5b5d6db in start_thread (arg=0x7f84c1569700) at pthread_create.c:463
#15 0x00007f84c45ec88f in clone () at ../sysdeps/unix/sysv/linux/x86_64/clone.S:95
```

#32 - 2018-12-09 09:11 - Luis Alves

Another lock, same bt as above.
Something is wrong:

```
(gdb) print http_lock
$3 = {mutex = pthread_mutex_t = {Type = Normal, Status = Not acquired, Robust = No, Shared = No, Protocol = No
ne}, tid = 0,
  filename = 0x0, lineno = 0, tstamp = 0, waiters = {lh_first = 0x0}, link = {tqe_next = 0x0, tqe_prev = 0x0}}
```

"http_lock" should be locked at this point: <https://github.com/tvheadend/tvheadend/blob/master/src/httpc.c#L1637>
But it isn't...

httpc thread is stuck here: <https://github.com/tvheadend/tvheadend/blob/master/src/httpc.c#L1453>

#33 - 2018-12-09 09:24 - Luis Alves

By the way, the last mutex magic code is still crashing tvh on start. (had to revert the last 2 commits)

```
REASON: magic
mutex 0x7f29a3bfc778 locked in: (null):0 (thread 0)
```

#34 - 2018-12-09 11:04 - Luis Alves

Found the crash on start. The htsp_out_mutex needs to init the magic's:

```
diff --git a/src/htsp_server.c b/src/htsp_server.c
index 04b04c3c2..49af61b17 100644
```

```

--- a/src/htsp_server.c
+++ b/src/htsp_server.c
@@ -3420,6 +3420,8 @@ htsp_serve(int fd, void **opaque, struct sockaddr_storage *source,
     htsp.htsp_peer = source;
     htsp.htsp_writer_run = 1;

+ tvh_mutex_init(&htsp.htsp_out_mutex, NULL);
+
     LIST_INSERT_HEAD(&htsp.connections, &htsp, htsp_link);
     tvh_mutex_unlock(&global_lock);

```

#35 - 2018-12-09 14:16 - Flole Systems

I'm still trying to figure out the memory corruption so I can cppcheck today and it found quite a lot, most of the errors are "uninitialized variables", so that shouldn't matter. The other things it found are

```

[config.c:2388]: (error) Shifting signed 32-bit value by 31 bits is undefined behaviour
[config.c:2399]: (error) Shifting signed 32-bit value by 31 bits is undefined behaviour
[descrambler/cccam.c:337]: (warning) Possible null pointer dereference: saa
[epggrab/module/opentv.c:238]: (error) Memory leak: nentry
[htsmmsg.c:1458]: (error) Common realloc mistake: 'ret' nulled but not freed upon failure
[htsmmsg.c:1461]: (error) Common realloc mistake: 'ret' nulled but not freed upon failure
[htsmmsg.c:1464]: (error) Common realloc mistake: 'ret' nulled but not freed upon failure
[htsp_server.c:3408]: (error) Address of local auto-variable assigned to a function parameter.
[idnode.c:854]: (error) Shifting signed 32-bit value by 31 bits is undefined behaviour
[input/mpegts/linuxdvb/linuxdvb_ca.c:501]: (warning) Possible null pointer dereference: data
[input/mpegts/mpegts_input.c:1845]: (error) syntax error
[parsers/parser_hevc.c:171]: (error) Shifting signed 32-bit value by 31 bits is undefined behaviour
[prop.c:145]: (error) Shifting signed 32-bit value by 31 bits is undefined behaviour
[prop.c:160]: (error) Shifting signed 32-bit value by 31 bits is undefined behaviour
[prop.c:315]: (error) Shifting signed 32-bit value by 31 bits is undefined behaviour
[prop.c:327]: (error) Shifting signed 32-bit value by 31 bits is undefined behaviour
[prop.c:544]: (error) Shifting signed 32-bit value by 31 bits is undefined behaviour
[satip/rtsp.c:192]: (error) syntax error
[webui/xmltv.c:141]: (error) syntax error

```

I have only briefly looked it through, but the memory leak in opentv.c is definitely there. The syntax errors are from when it disabled those defines, they can be ignored safely. With "Common realloc mistake: 'ret' nulled but not freed upon failure" it probably means that if the realloc call fails you should call free on it? The null pointer references all basically don't apply, it just didn't catch all the checks before as far as I can see. And I have absolutely no clue what it means with "Shifting signed 32-bit value by 31 bits is undefined behaviour".

I will try with some other tools aswell and see if there is anything else that is being found.

#36 - 2018-12-09 15:37 - Luis Alves

Jaroslav,
In my case, this is what's happening:

thread a)
"global_lock" held by "mpegts_service_stop" (that calls "http_client_close" down the stack)
and "http_client_close" is waiting for a signal on "http_cond"

thread b)
The httpc thread (which should send the signal) is waiting for the global lock to continue (and eventually send the signal)
and is stuck at: "http_client_run->http_client_data_received->http_client_data_copy->iptv_http_data->iptv_http_safe_global_lock"

This lock is easily reproducible with this setup:
2 tvheadends, each with 1 auto iptv networks - with 10 max # inputs streams (4 should be enough to get a lock)
tvh1 points to an external playlist (could be another tvh I guess)
tvh2 playlist pointing at tvh1

Doing a "Force scan" on tvh2 iptv network, tvh1 will lock after a few seconds.

#37 - 2018-12-09 16:34 - Jaroslav Kysela

[Luis Parada](#): retest with v4.3-1638-gf0524db40

#38 - 2018-12-09 19:53 - David jrm

Jaroslav Kysela wrote:

[Luis Parada](#): retest with v4.3-1638-gf0524db40

It keep crashing..... sad.png


```
394 0x55f69a5af000
2018-12-10 16:28:45.019 [ ALERT] CRASH: /SCRATCH/repos/tvheadend/src/input/mpegts/iptv/iptv_http.c:342 0x55f6
9a8a307b 0x55f69a5af000
2018-12-10 16:28:45.030 [ ALERT] CRASH: /SCRATCH/repos/tvheadend/src/httpc.c:784 0x55f69a7e144d 0x55f69a5af00
0
2018-12-10 16:28:45.042 [ ALERT] CRASH: /SCRATCH/repos/tvheadend/src/httpc.c:919 0x55f69a7e15be 0x55f69a5af00
0
2018-12-10 16:28:45.054 [ ALERT] CRASH: /SCRATCH/repos/tvheadend/src/httpc.c:1133 0x55f69a7e3d2d 0x55f69a5af0
00
2018-12-10 16:28:45.066 [ ALERT] CRASH: /SCRATCH/repos/tvheadend/src/httpc.c:1195 0x55f69a7e433a 0x55f69a5af0
00
2018-12-10 16:28:45.078 [ ALERT] CRASH: /SCRATCH/repos/tvheadend/src/httpc.c:1453 0x55f69a7e4492 0x55f69a5af0
00
2018-12-10 16:28:45.088 [ ALERT] CRASH: /SCRATCH/repos/tvheadend/src/tvh_thread.c:91 0x55f69a78f8d8 0x55f69a5
af000
2018-12-10 16:28:45.107 [ ALERT] CRASH: ??:0 0x7f0310d036db 0x7f0310cfc000
Aborted (core dumped)
```

This started with commit f0524db

#42 - 2018-12-10 16:59 - Luis Alves

From a quick test I can see that it the assert happens when I start a subscription for an iptv stream.

Seems to be something left out of a lock that is getting "corrupted".

By the way, do you prefer a new issue to be opened on new bugs like this or it's ok to keep discussing on the previous one?

Flole Systems,

Have you tried the latest code to see if your lock is also solved?

#43 - 2018-12-10 22:53 - Flole Systems

I haven't been brave enough yet wink.png Reading here about the issues it's still having I decided to wait and an hour ago my monday recording crazyness has started and I'm recording 4 channels right now and that will last 2 more hours wink.png I can probably give it a try tomorrow

And I just got a crash.... I suspect a memory corruption again?

```
CRASH: Register dump [23]: 00000000000000000000000000100007f26e43a05d0000000000000000600007f26e43a0b40000
0000000000000007f26e000d93000007f27190028d0032219d403e519d300000000000001100007f26e43a06e000000000000000
000000000000011032219d403e519d3000f0000000000000007f26e43a06d00000558a1946231a000000000010202002b00000000003
3000000000000000000000000000dfffffff7ffbbba130000000000000000
CRASH: STACKTRACE
CRASH: tvheadend/src/trap.c:176 0x558a193a785e 0x558a19172000
CRASH: ??:0 0x7f271dabf890 0x7f271daad000
CRASH: tvheadend/src/input/mpegts.h:113 (discriminator 1) 0x558a1946231a 0x558a19172000
CRASH: tvheadend/src/input/mpegts/tsdemux.c:272 0x558a19462dd5 0x558a19172000
CRASH: tvheadend/src/input/mpegts/mpegts_input.c:1467 (discriminator 3) 0x558a1945f4f2 0x558a19172000
CRASH: tvheadend/src/input/mpegts/mpegts_input.c:1707 0x558a1946030f 0x558a19172000
CRASH: tvheadend/src/tvh_thread.c:93 0x558a19352ebb 0x558a19172000
CRASH: ??:0 0x7f271dab46db 0x7f271daad000
CRASH: clone+0x3f (/lib/x86_64-linux-gnu/libc.so.6)
```

#44 - 2018-12-11 08:26 - Jaroslav Kysela

[luis Parada](#): The IPTV HTTP issue should be fixed through [#5353](#) . Please, open new issue next time.

#45 - 2018-12-11 08:49 - Luis Alves

Jaroslav Kysela wrote:

[luis Parada](#): The IPTV HTTP issue should be fixed through [#5353](#) . Please, open new issue next time.

Ok! By the way, the issue was not solved...

But then I tried to add the lock on the "iptv_input_mux_started" too and it seems to be fixed.

Something like:

```
iff --git a/src/input/mpegts/iptv/iptv_http.c b/src/input/mpegts/iptv/iptv_http.c
index f31ccd7c9..7ebc0a25e 100644
--- a/src/input/mpegts/iptv/iptv_http.c
+++ b/src/input/mpegts/iptv/iptv_http.c
@@ -195,13 +195,13 @@ iptv_http_kick_cb( void *aux )
     if (im == NULL) return;
     if (hp->flush) {
         hp->flush = 0;
```

```

+   tvh_mutex_lock(&iptv_lock);
   if (!hp->started) {
       iptv_input_mux_started(hp->mi, im);
   } else {
-   tvh_mutex_lock(&iptv_lock);
-   iptv_input_recv_flush(im);
-   tvh_mutex_unlock(&iptv_lock);
   }
+   tvh_mutex_unlock(&iptv_lock);
   hp->started = 1;
}

```

#46 - 2018-12-11 08:56 - Jaroslav Kysela

You're right. I've overlooked the sbuf touch in the iptv_mux_input_started().

#47 - 2018-12-11 16:43 - Luis Alves

Thanks you!

This last update completely fixed my hanging issue.

Not sure about the original issue reported by Flole.

#48 - 2018-12-12 02:10 - Flole Systems

I just installed it and will see now how it performs. It's never reproducible, so the cause is/was probably indeed a memory corruption but we will hopefully see that soon.

It might be a good idea to run coverity again, as far as I can see it hasn't been run since a about a year, and back then it detected quite some stuff. The chance that it might pick up the source of the memory corruption is definitely there. I also appreciate other ideas to track this down, unfortunately valgrind is not an option as tvheadend is unbelievable slow then, definitely unusable for live streaming multiple channels.

#49 - 2019-01-13 17:08 - Flole Systems

Unfortunately I just had another one, I was working on the coax cable at that point so maybe some data got lost on the way to the receiver at that time:

```

(gdb) print global_lock
$1 = {mutex = pthread_mutex_t = {Type = Normal, Status = Acquired, possibly with waiters, Owner ID = 10395, Robust = No, Shared = No, Protocol = None}, magic1 = 3614061450, tid = 10395, filename = 0x55b4451bdc32 "src/main.c", lineno = 703,
      tstamp = 7613493782568, waiters = {lh_first = 0x55b446d4f580}, link = {tqe_next = 0x55b445fbfb00, tqe_prev = 0x55b445836e78 <rtsp_lock+88>}, magic2 = 4181353505}
(gdb) thread find 10395
Thread 7 has target id 'Thread 0x7f77b9733700 (LWP 10395)'
(gdb) info threads 7
   Id Target Id         Frame
   7   Thread 0x7f77b9733700 (LWP 10395) __lll_lock_wait () at ../sysdeps/unix/sysv/linux/x86_64/lowlevellock.S:135
(gdb) thread 7
[Switching to thread 7 (Thread 0x7f77b9733700 (LWP 10395))]
#0 __lll_lock_wait () at ../sysdeps/unix/sysv/linux/x86_64/lowlevellock.S:135
135   in ../sysdeps/unix/sysv/linux/x86_64/lowlevellock.S
(gdb) bt
#0 __lll_lock_wait () at ../sysdeps/unix/sysv/linux/x86_64/lowlevellock.S:135
#1 0x00007f77be3f7023 in __GI___pthread_mutex_lock (mutex=0x55b445fbfb00) at ../nptl/pthread_mutex_lock.c:78
#2 0x000055b4441c0181 in tvh__mutex_lock (mutex=0x55b445fbfb00, filename=0x55b44529ab68 "src/input/mpegts/satip/satip_frontend.c", lineno=165) at src/tvh_thread.c:254
#3 0x000055b44432c093 in satip_frontend_signal_cb (aux=0x55b449138a90) at src/input/mpegts/satip/satip_frontend.c:165
#4 0x000055b4441a7a7b in mtimer_thread (aux=0x0) at src/main.c:706
#5 0x000055b4441bfac8 in thread_wrapper (p=0x55b445fbfb030) at src/tvh_thread.c:91
#6 0x00007f77be3f46db in start_thread (arg=0x7f77b9733700) at pthread_create.c:463
#7 0x00007f77bce7888f in clone () at ../sysdeps/unix/sysv/linux/x86_64/clone.S:95
(gdb) frame 2
#2 0x000055b4441c0181 in tvh__mutex_lock (mutex=0x55b445fbfb00, filename=0x55b44529ab68 "src/input/mpegts/satip/satip_frontend.c", lineno=165) at src/tvh_thread.c:254
254     int r = pthread_mutex_lock(&mutex->mutex);
(gdb) print mutex
$2 = (tvh_mutex_t *) 0x55b445fbfb00
(gdb) p *((tvh_mutex_t *) 0x55b445fbfb00)
$3 = {mutex = pthread_mutex_t = {Type = Normal, Status = Acquired, possibly with waiters, Owner ID = 10579, Robust = No, Shared = No, Protocol = None}, magic1 = 3614061450, tid = 10579, filename = 0x55b44528cb80 "src/input/mpegts/tsdemux.c", lineno = 164,

```

```

tstamp = 7613493746567, waiters = {lh_first = 0x55b4490f1fa0}, link = {tqe_next = 0x55b449138d38, tqe_prev =
0x55b445993218 <global_lock+88>}, magic2 = 4181353505}
(gdb) thread find 10579
Thread 70 has target id 'Thread 0x7f77af72b700 (LWP 10579)'
(gdb) thread 70
[Switching to thread 70 (Thread 0x7f77af72b700 (LWP 10579))]
#0 0x000055b4442629ff in parse_ac3 (t=0x55b447c72020, st=0x55b448dfa3c0, ilen=3854, next_startcode=445, sc_of
fset=0) at src/parsers/parsers.c:932
932     if (!(ver = ac3_valid_frame(p = buf + i))) continue;
(gdb) bt
#0 0x000055b4442629ff in parse_ac3 (t=0x55b447c72020, st=0x55b448dfa3c0, ilen=3854, next_startcode=445, sc_of
fset=0) at src/parsers/parsers.c:932
#1 0x000055b44426189b in parse_pes (t=0x55b447c72020, st=0x55b448dfa3c0, data=0x55b449b1e9ac "", len=184, sta
rt=1, vp=0x55b4442628e9 <parse_ac3>) at src/parsers/parsers.c:428
#2 0x000055b444262fba in parse_pes_ac3 (t=0x55b447c72020, st=0x55b448dfa3c0, data=0x55b449b1e9ac "", len=184,
start=1) at src/parsers/parsers.c:1026
#3 0x000055b44426552b in parse_mpeg_ts (t=0x55b447c72020, st=0x55b448dfa3c0, data=0x55b449b1e9ac "", len=184,
start=1, err=0) at src/parsers/parsers.c:1950
#4 0x000055b44425fce5 in parser_input_mpegts (prs=0x55b447c72020, pb=0x55b4487822d0) at src/parsers/message.c
:217
#5 0x000055b444260112 in parser_input (opaque=0x55b447c72020, sm=0x55b44632cfb0) at src/parsers/message.c:315
#6 0x000055b4441f2133 in streaming_target_deliver (st=0x55b447c72020, sm=0x55b44632cfb0) at src/streaming.h:4
61
#7 0x000055b4441f4245 in subscription_input_direct (opaque=0x55b4482ccdf0, sm=0x55b44632cfb0) at src/subscrip
tions.c:550
#8 0x000055b4441f44bd in subscription_input (opaque=0x55b4482ccdf0, sm=0x55b44632cfb0) at src/subscriptions.c
:635
#9 0x000055b4441lea7fd in streaming_target_deliver (st=0x55b4482ccf08, sm=0x55b44632cfb0) at src/streaming.h:4
61
#10 0x000055b4441eb51a in streaming_pad_deliver (sp=0x55b445fbfc00, sm=0x55b44632cfb0) at src/streaming.c:422
#11 0x000055b4441eb57c in streaming_service_deliver (t=0x55b445fbf910, sm=0x55b44632cfb0) at src/streaming.c:4
35
#12 0x000055b4442d1e23 in ts_flush (t=0x55b445fbf910, sb=0x55b445fbfd28) at src/input/mpegts/tsdemux.c:305
#13 0x000055b4442d1f2e in ts_remux (t=0x55b445fbf910,
src=0x55b4506663c5 "G\023\355\030\313\3b37\0m\276\353\352\324 @\212f\366\250\246 \001K\362\347\260T9\266i\3
41\az\203\205 Gx\232;\220\247\236r\253\255\323\371\245\366&\200\315)\335QS\3630\374\317\8\246W\337C\251S\24
1\260\371\261>\002q{h\220\377\313\340\336\315-'Q\375N\033\204T\222xf\370\251\335\006\352\320\027w\377\270\227C
gc\360\065\210\020\271(\337\340\327)\351\273)\264\224\262y\302\024\027#\352\326%\274\373\361<\306MF\221\212\2
11\004\316V\2368T\035G\361\273\b\221q\254\373A\341\005\275s\346\b\362\345\356\021\301\237\065\272\277\022\001"
, len=188, errors=0) at src/input/mpegts/tsdemux.c:332
#14 0x000055b4442d1436 in ts_recv_packet0 (t=0x55b445fbf910, st=0x55b445fbfd80,
tsb=0x55b4506663c5 "G\023\355\030\313\3b37\0m\276\353\352\324 @\212f\366\250\246 \001K\362\347\260T9\266i\3
41\az\203\205 Gx\232;\220\247\236r\253\255\323\371\245\366&\200\315)\335QS\3630\374\317\8\246W\337C\251S\24
1\260\371\261>\002q{h\220\377\313\340\336\315-'Q\375N\033\204T\222xf\370\251\335\006\352\320\027w\377\270\227C
gc\360\065\210\020\271(\337\340\327)\351\273)\264\224\262y\302\024\027#\352\326%\274\373\361<\306MF\221\212\2
11\004\316V\2368T\035G\361\273\b\221q\254\373A\341\005\275s\346\b\362\345\356\021\301\237\065\272\277\022\001"
, len=188) at src/input/mpegts/tsdemux.c:78
#15 0x000055b4442d1a91 in ts_recv_packet1 (t=0x55b445fbf910, tspos=551247208, pid=5101,
tsb=0x55b4506663c5 "G\023\355\030\313\3b37\0m\276\353\352\324 @\212f\366\250\246 \001K\362\347\260T9\266i\3
41\az\203\205 Gx\232;\220\247\236r\253\255\323\371\245\366&\200\315)\335QS\3630\374\317\8\246W\337C\251S\24
1\260\371\261>\002q{h\220\377\313\340\336\315-'Q\375N\033\204T\222xf\370\251\335\006\352\320\027w\377\270\227C
gc\360\065\210\020\271(\337\340\327)\351\273)\264\224\262y\302\024\027#\352\326%\274\373\361<\306MF\221\212\2
11\004\316V\2368T\035G\361\273\b\221q\254\373A\341\005\275s\346\b\362\345\356\021\301\237\065\272\277\022\001"
, len=188, table=1) at src/input/mpegts/tsdemux.c:216
#16 0x000055b4442ce5a8 in mpegts_input_process (mi=0x55b449138a90, mpkt=0x55b450661220) at src/input/mpegts/mp
egts_input.c:1529
#17 0x000055b4442cf1de in mpegts_input_thread (p=0x55b449138a90) at src/input/mpegts/mpegts_input.c:1746
#18 0x000055b4441bfac8 in thread_wrapper (p=0x55b4491167f0) at src/tvh_thread.c:91
#19 0x00007f77be3f46db in start_thread (arg=0x7f77af72b700) at pthread_create.c:463
#20 0x00007f77bce7888f in clone () at ../sysdeps/unix/sysv/linux/x86_64/clone.S:95

```

#50 - 2019-01-13 20:18 - Jaroslav Kysela

The parse_ac3() issue might be fixed in v4.3-1719-gbd662457d . Thanks for the gdb analysis.

#51 - 2019-01-21 22:28 - Rich 11

I have this same issue. It usually happens <1 hour of recording. Sometimes happens after 10 minutes, sometimes >2 hours. I assume it could be related to weak tv signal as someone else mentioned.

```

Jan 21 22:03:13 raspberrypi tvheadend[25546]: linuxdVB: Panasonic MN88472 #0 : DVB-T #0 - read() EOVERFLOW
Jan 21 22:03:13 raspberrypi tvheadend[25546]: TS: DVB-T Network/682MHz/BBC ONE HD: H264 @ #6601 Continuity cou
nter error (total 5)
Jan 21 22:03:13 raspberrypi tvheadend[25546]: TS: DVB-T Network/682MHz/BBC ONE HD: AAC @ #6602 Continuity coun
ter error (total 4)

```



```

Jan 21 22:03:14 raspberrypi tvheadend[25546]: TS: DVB-T Network/682MHz/BBC ONE HD: DVBSUB @ #6605 Continuity counter error (total 4)
Jan 21 22:03:17 raspberrypi tvheadend[25546]: mpegts: too much queued table input data (over 2MB) for Panasonic MN88472 #0 : DVB-T #0, disca
Jan 21 22:03:27 raspberrypi tvheadend[25546]: mpegts: too much queued table input data (over 2MB) for Panasonic MN88472 #0 : DVB-T #0, disca
Jan 21 22:03:37 raspberrypi tvheadend[25546]: mpegts: too much queued table input data (over 2MB) for Panasonic MN88472 #0 : DVB-T #0, disca
Jan 21 22:03:47 raspberrypi tvheadend[25546]: mpegts: too much queued table input data (over 2MB) for Panasonic MN88472 #0 : DVB-T #0, disca
Jan 21 22:03:57 raspberrypi tvheadend[25546]: mpegts: too much queued table input data (over 2MB) for Panasonic MN88472 #0 : DVB-T #0, disca
Jan 21 22:04:06 raspberrypi tvheadend[25546]: linuxdvb: Panasonic MN88472 #0 : DVB-T #0 - read() EOVERFLOW
Jan 21 22:04:06 raspberrypi tvheadend[25546]: TS: DVB-T Network/682MHz/BBC ONE HD: AAC @ #6602 Continuity counter error (total 5)
Jan 21 22:04:07 raspberrypi tvheadend[25546]: mpegts: too much queued table input data (over 2MB) for Panasonic MN88472 #0 : DVB-T #0, disca
Jan 21 22:04:17 raspberrypi tvheadend[25546]: mpegts: too much queued table input data (over 2MB) for Panasonic MN88472 #0 : DVB-T #0, disca
Jan 21 22:04:27 raspberrypi tvheadend[25546]: mpegts: too much queued table input data (over 2MB) for Panasonic MN88472 #0 : DVB-T #0, disca
Jan 21 22:04:37 raspberrypi tvheadend[25546]: mpegts: too much queued table input data (over 2MB) for Panasonic MN88472 #0 : DVB-T #0, disca
Jan 21 22:04:47 raspberrypi tvheadend[25546]: mpegts: too much queued table input data (over 2MB) for Panasonic MN88472 #0 : DVB-T #0, disca
Jan 21 22:04:58 raspberrypi systemd[1]: tvheadend.service: Child 25546 belongs to tvheadend.service
Jan 21 22:04:58 raspberrypi systemd[1]: tvheadend.service: cgroup is empty
Jan 21 22:04:58 raspberrypi systemd[1]: tvheadend.service: Changed running -> exited

```

It gets killed by OOM killer:

```

[Jan21 22:04] tvh:dvr invoked oom-killer: gfp_mask=0x6000c2(GFP_KERNEL|__GFP_HIGHMEM), nodemask=(null), order=0, oom_score_adj=0

```

If I try to use --thrdebug, it won't even start tvheadend:

```

hts@raspberrypi:~$ tvheadend --thrdebug 10020
2019-01-21 22:15:05.703 [ INFO] main: Log started
2019-01-21 22:15:05.829 [ INFO] http: Starting HTTP server 0.0.0.0:9981
2019-01-21 22:15:05.848 [ INFO] htsp: Starting HTSP server 0.0.0.0:9982
2019-01-21 22:15:06.303 [ INFO] config: loaded
2019-01-21 22:15:06.314 [ INFO] config: scanfile (re)initialization with path <none>
2019-01-21 22:15:06.734 [ INFO] descrambler: adding CAID 2600/FFFF as ConstCW interval 10000ms pc 20 ep default (BISS)
2019-01-21 22:15:06.742 [ INFO] descrambler: adding CAID 0E00/FFFF as MultiPID interval 1000ms pc 2 ep default (PowerVu)
2019-01-21 22:15:07.046 [ INFO] iptv: Using 2 input thread(s)
2019-01-21 22:15:10.435 [ INFO] dvr: Creating new configuration ''
2019-01-21 22:15:10.582 [ INFO] epgrab: module uk_freesat created
2019-01-21 22:15:10.583 [ INFO] epgrab: module uk_freesat_eit created
2019-01-21 22:15:10.583 [ INFO] epgrab: module uk_freeview created
2019-01-21 22:15:10.583 [ INFO] epgrab: module nz_freeview2 created
2019-01-21 22:15:10.583 [ INFO] epgrab: module nz_freeview1 created
2019-01-21 22:15:10.583 [ INFO] epgrab: module viasat_10.583 created
2019-01-21 22:15:10.584 [ INFO] epgrab: module Bulsatcom_39E created
2019-01-21 22:15:10.584 [ INFO] epgrab: module uk_cable_virgin created
2019-01-21 22:15:10.584 [ INFO] epgrab: module eit created
2019-01-21 22:15:10.584 [ INFO] epgrab: module psip created
2019-01-21 22:15:10.847 [ INFO] epgrab: module opentv-skyuk created
2019-01-21 22:15:10.913 [ INFO] epgrab: module opentv-ausat created
2019-01-21 22:15:10.920 [ INFO] epgrab: module opentv-skyit created
2019-01-21 22:15:10.954 [ INFO] epgrab: module opentv-skynz created
2019-01-21 22:15:10.964 [ INFO] epgrab: module xmltv created
2019-01-21 22:15:11.011 [ INFO] spawn: Executing "/usr/bin/tv_find_grabbers"
REASON: deadlock (src/tvh_thread.c:512)
mutex 0x823060 locked in: src/input/mpegts/linuxdvb/linuxdvb_adapter.c:606 (thread 25841)
mutex 0x823060 waiting in: src/service_mapper.c:369 (thread 25861)
mutex 0x823060 waiting in: src/notify.c:103 (thread 25844)
Aborted

REASON: deadlock (src/tvh_thread.c:512)
mutex 0x7bf060 locked in: src/input/mpegts/linuxdvb/linuxdvb_adapter.c:606 (thread 25435)
mutex 0x7bf060 waiting in: src/input/mpegts/scanfile.c:949 (thread 25443)
mutex 0x7bf060 waiting in: src/service_mapper.c:369 (thread 25455)
mutex 0x7bf060 waiting in: src/notify.c:103 (thread 25438)

```

Aborted

tvheadend: version 4.3-1732~g10ed59ce3

on rpi zero-w

#52 - 2019-01-21 22:40 - Flole Systems

Without looking at it using GDB it's just guessing what it might be. See what I did above, it's really easy, you just have to follow the path (see what lock is waiting, see what thread currently has that lock, see what that thread is currently doing or more specific where it's stuck)

The issue I had with "weak" TV Signal was me messing around with the cable of my SAT-IP Tuner and causing some packet loss there. I can try doing that again to see if it's fixed now, but you should look at your issue with gdb (take a coredump so you can continue using tvheadend and don't have to keep it in it's stuck state). Don't forget to install the debug version first, otherwise you won't see much.

#53 - 2020-04-04 02:49 - Flole Systems

- Status changed from New to Fixed

#54 - 2020-05-13 03:44 - Victor S

Nothing fixed there. Stop closing every ticket of this, when you didn't even confirm it was fixed. Annoying.

#55 - 2020-05-13 04:11 - Flole Systems

This was my ticket and I found the source and fixed it (something you should do aswell). Fix is mentioned above.

Stop spamming in every ticket or you will get banned. If you are having issues with a lockup in the ac3 parsing this issue is for you, if not this issue is not for you. I will say this one last time.

Also my motivation to fix your issue is 0, and I am sure every other person who is or was contributing is feeling the same so good luck getting it fixed. Maybe you should start with an apology for your spamming and inappropriate language, otherwise I don't see any chance that someone will ever look into any issue you created.

Files

gdb.txt	11 KB	2018-11-26	Flole Systems
---------	-------	------------	---------------