

Tvheadend - Bug #4595

EMM harvesting with Mux scheduler not working?

2017-09-15 01:10 - LubosD -

Status:	New	Start date:	2017-09-15
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	Descrambling	Estimated time:	0.00 hour
Target version:		Affected Versions:	
Found in version:	git master		

Description

I'm trying to make use of the Mux scheduler feature to make sure my Sky card (that I use for "Freesat from Sky") always has the entitlements.

Sadly, it doesn't work. I added a schedule for 11170.75H @ 28.2E and Tvheadend is configured to connect to OSCAM via Newcamd protocol to talk to the smart card. I was able to watch encrypted channels whilst I still had entitlements.

When the scheduler job is triggered, a subscription with weight 100 appears, but Tvheadend doesn't connect to OSCAM and hence doesn't pass any EMMs.

Apart from this, I've also noticed that when I tune in to an encrypted channel and my card's entitlements are already expired (so TVH is receiving failures), Tvheadend doesn't seem to pass any EMMs. So it seems I can't use Tvheadend to revive a "dead" card.

History

#1 - 2017-09-15 09:00 - Jaroslav Kysela

The newcamd connection should be persistent, so TVH should be connected to oscam all time.

Provide: '--trace descrambler,cwc' log: <https://tvheadend.org/projects/tvheadend/wiki/Traces>

#2 - 2017-09-15 22:36 - LubosD -

It for sure isn't persistent. It disconnects from OSCAM once the connection is not used by anything.

#3 - 2017-09-16 13:46 - Jaroslav Kysela

I really doubt. The TCP connection to the server is persistent. You probably don't see any ECMs in oscam (so it shows as inactive), but if you look to log, there shouldn't be disconnects.

Anyway, I need logs to see where's the issue. Also, you may give feedback, if you see any changes in the EMM counters (reader statistics) in oscam. Perhaps, the issue is in the EMM reassembly code.

#4 - 2017-09-16 14:15 - saen acro

[LubosD](#) -

In user config is keepalive and sleep active, also is **Keepalive** in newcam server active?

#5 - 2018-07-21 17:40 - Joe User

I also have never gotten this to work. I am using a DVBAPI connection to oscam and the connection is fine and works for descrambling. When a channel is being recorded or streamed, the emms are processed normally, but the mux scheduler does not send any information to oscam (oscam debug set to 192, 128=dvbapi, 64=emm). Manually playing an entire mux has the same result - nothing sent to oscam.

Here is a log. I ran it for just one minute to create the log, but I have run it for as long as twenty minutes with no results.

I have trace set for descrambler and descrambler-emm (as well as others...)

```
2018-07-21 15:54:00.006 [ DEBUG] mpegts: 11804.2V in NET - add raw service
2018-07-21 15:54:00.006 [ TRACE] subscription: 0036: creating subscription for service 11804.2V in NET weight
 0 using profile <none>
2018-07-21 15:54:00.008 [ TRACE] subscription: 0036: find instance for 11804.2V in NET weight 100
2018-07-21 15:54:00.008 [ DEBUG] service: 7: 11804.2V in NET si 0x7faf9c0089d0 Silicon Labs Si2183 #1 : DVB-S
 #1 weight 0 prio 10 error 0
2018-07-21 15:54:00.008 [ TRACE] service: will start new instance 7
```

```

2018-07-21 15:54:00.008 [ TRACE] service: starting 11804.2V in NET
2018-07-21 15:54:00.008 [ INFO] mpegts: 11804.2V in NET - tuning on Silicon Labs Si2183 #1 : DVB-S #1
2018-07-21 15:54:00.008 [ TRACE] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - frontend clear
2018-07-21 15:54:00.008 [ TRACE] linuxdvb: tuner Silicon Labs Si2183 #1 : DVB-S #1 tuning to DVB-S2 9.0E freq
 11804200 V sym 27500000 fec AUTO mod QPSK roff AUTO is_id -1 pls_mode ROOT pls_code 1 (freq 1204200)
2018-07-21 15:54:00.008 [ TRACE] linuxdvb: S2CMD 17 => 6
2018-07-21 15:54:00.008 [ TRACE] linuxdvb: S2CMD 03 => 1204200
2018-07-21 15:54:00.008 [ TRACE] linuxdvb: S2CMD 06 => 2
2018-07-21 15:54:00.008 [ TRACE] linuxdvb: S2CMD 08 => 27500000
2018-07-21 15:54:00.008 [ TRACE] linuxdvb: S2CMD 09 => 9
2018-07-21 15:54:00.008 [ TRACE] linuxdvb: S2CMD 04 => 0
2018-07-21 15:54:00.008 [ TRACE] linuxdvb: S2CMD 12 => 2
2018-07-21 15:54:00.008 [ TRACE] linuxdvb: S2CMD 13 => 3
2018-07-21 15:54:00.008 [ TRACE] linuxdvb: S2CMD 42 => 4294967295
2018-07-21 15:54:00.008 [ TRACE] linuxdvb: S2CMD 01 => 0
2018-07-21 15:54:00.035 [ DEBUG] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - starting 11804.2V in NET
2018-07-21 15:54:00.035 [ TRACE] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - tuning
2018-07-21 15:54:00.035 [ TRACE] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - frontend clear
2018-07-21 15:54:00.035 [ TRACE] linuxdvb: tuner Silicon Labs Si2183 #1 : DVB-S #1 tuning to DVB-S2 9.0E freq
 11804200 V sym 27500000 fec AUTO mod QPSK roff AUTO is_id -1 pls_mode ROOT pls_code 1 (freq 1204200)
2018-07-21 15:54:00.035 [ TRACE] linuxdvb: S2CMD 17 => 6
2018-07-21 15:54:00.035 [ TRACE] linuxdvb: S2CMD 03 => 1204200
2018-07-21 15:54:00.035 [ TRACE] linuxdvb: S2CMD 06 => 2
2018-07-21 15:54:00.035 [ TRACE] linuxdvb: S2CMD 08 => 27500000
2018-07-21 15:54:00.035 [ TRACE] linuxdvb: S2CMD 09 => 9
2018-07-21 15:54:00.035 [ TRACE] linuxdvb: S2CMD 04 => 0
2018-07-21 15:54:00.035 [ TRACE] linuxdvb: S2CMD 12 => 2
2018-07-21 15:54:00.035 [ TRACE] linuxdvb: S2CMD 13 => 3
2018-07-21 15:54:00.035 [ TRACE] linuxdvb: S2CMD 42 => 4294967295
2018-07-21 15:54:00.035 [ TRACE] linuxdvb: S2CMD 01 => 0
2018-07-21 15:54:00.035 [ DEBUG] mpegts: 11804.2V in NET - started
2018-07-21 15:54:00.035 [ DEBUG] mpegts: 11804.2V in NET - open PID 0000 (0) [2/0x56350073b970]
2018-07-21 15:54:00.035 [ TRACE] subscription: 0036: linking sub 0x5635003d70a0 to svc 0x56350073b970 type 1
2018-07-21 15:54:00.035 [ INFO] subscription: 0036: "" subscribing to mux "11804.2V", weight: 100, adapter:
"Silicon Labs Si2183 #1 : DVB-S #1", network: "NET", service: "Raw PID Subscription"
2018-07-21 15:54:00.035 [ TRACE] subscription: 0036: chain 01: input
2018-07-21 15:54:00.035 [ TRACE] subscription: 0036: chain 02: mux sched input
2018-07-21 15:54:00.058 [ TRACE] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - checking FE status (ready)
2018-07-21 15:54:00.116 [ TRACE] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - status 4 (0000)
2018-07-21 15:54:00.158 [ TRACE] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - checking FE status (ready)
2018-07-21 15:54:00.162 [ TRACE] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - status 4 (0000)
2018-07-21 15:54:00.208 [ TRACE] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - checking FE status (ready)
2018-07-21 15:54:00.212 [ DEBUG] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - status GOOD (SIGNAL | CARRI
ER | VITERBI | SYNC | LOCK)
2018-07-21 15:54:00.212 [ DEBUG] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - locked
2018-07-21 15:54:00.212 [ TRACE] thread: created thread 140392099911424 [tvh:linuxdvb-front / 0x5634fef1fdca0(0x
5635000e4040)]
2018-07-21 15:54:00.212 [ TRACE] mpegts: table: mux 0x56350010bc70 add pat 00/00 (0) pid 0000 (0)
2018-07-21 15:54:00.212 [ DEBUG] mpegts: 11804.2V in NET - open PID 0000 (0) [20/0x7faf9c004b60]
2018-07-21 15:54:00.212 [ TRACE] mpegts: table: mux 0x56350010bc70 add cat 01/FF (1) pid 0001 (1)
2018-07-21 15:54:00.212 [ DEBUG] mpegts: 11804.2V in NET - open PID 0001 (1) [16/0x7faf9c006030]
2018-07-21 15:54:00.212 [ TRACE] mpegts: table: mux 0x56350010bc70 add nit 00/00 (0) pid 0010 (16)
2018-07-21 15:54:00.212 [ DEBUG] mpegts: 11804.2V in NET - open PID 0010 (16) [20/0x7faf9c008ec0]
2018-07-21 15:54:00.212 [ TRACE] mpegts: table: mux 0x56350010bc70 add sdt 40/F8 (64) pid 0011 (17)
2018-07-21 15:54:00.212 [ DEBUG] mpegts: 11804.2V in NET - open PID 0011 (17) [20/0x7faf9c00a390]
2018-07-21 15:54:00.212 [ TRACE] mpegts: table: mux 0x56350010bc70 add bat 48/F8 (72) pid 0011 (17)
2018-07-21 15:54:00.212 [ DEBUG] mpegts: 11804.2V in NET - open PID 0011 (17) [16/0x7faf9c00b860]
2018-07-21 15:54:00.212 [ TRACE] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - weighted PIDs 0(1000)
2018-07-21 15:54:00.212 [ TRACE] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - open PID 0000 (0) fd 40
2018-07-21 15:54:00.213 [ TRACE] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - weighted PIDs 0(1000),1(999),1
6(999),17(999)
2018-07-21 15:54:00.213 [ TRACE] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - open PID 0001 (1) fd 41
2018-07-21 15:54:00.213 [ TRACE] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - open PID 0010 (16) fd 43
2018-07-21 15:54:00.213 [ TRACE] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - open PID 0011 (17) fd 44
2018-07-21 15:54:01.208 [ TRACE] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - checking FE status (ready)
2018-07-21 15:54:01.212 [ TRACE] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - status 1 (001F)
2018-07-21 15:54:01.361 [ DEBUG] service: 11804.2V in NET: Status changed to [Demuxed packets]
2018-07-21 15:54:01.361 [ DEBUG] service: 11804.2V in NET: Status changed to [Demuxed packets] [Reassembled p
ackets]
2018-07-21 15:54:01.361 [ TRACE] mpegts: input Silicon Labs Si2183 #1 : DVB-S #1 got 8460 bytes
2018-07-21 15:54:01.361 [ TRACE] descrambler: CAT data (len 31)
2018-07-21 15:54:01.361 [ TRACE] descrambler: 09 04 56 04 E0 30 09 04 0E 00 00 5A 09 04 0E 00 ..V..0.....Z...
.
2018-07-21 15:54:01.361 [ TRACE] descrambler: E0 5A 5F 04 53 41 50 53 FD 05 53 41 50 53 00 ..Z..SAPS..SAPS.
2018-07-21 15:54:01.361 [ TRACE] mpegts: table: mux 0x56350010bc70 add eit 00/00 (0) pid 0012 (18)

```



```

2018-07-21 15:54:07.469 [ TRACE] mpegts: input Silicon Labs Si2183 #1 : DVB-S #1 got 18800 bytes
2018-07-21 15:54:07.525 [ TRACE] mpegts: input Silicon Labs Si2183 #1 : DVB-S #1 got 3196 bytes
2018-07-21 15:54:07.755 [ TRACE] mpegts: input Silicon Labs Si2183 #1 : DVB-S #1 got 18800 bytes
2018-07-21 15:54:08.059 [ TRACE] mpegts: input Silicon Labs Si2183 #1 : DVB-S #1 got 18800 bytes
2018-07-21 15:54:08.209 [ TRACE] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - checking FE status (ready)
2018-07-21 15:54:08.213 [ TRACE] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - status 1 (001F)
2018-07-21 15:54:08.345 [ TRACE] mpegts: input Silicon Labs Si2183 #1 : DVB-S #1 got 19552 bytes
2018-07-21 15:54:08.621 [ TRACE] mpegts: input Silicon Labs Si2183 #1 : DVB-S #1 got 18236 bytes
2018-07-21 15:54:08.847 [ TRACE] mpegts: input Silicon Labs Si2183 #1 : DVB-S #1 got 18800 bytes
2018-07-21 15:54:09.144 [ TRACE] mpegts: input Silicon Labs Si2183 #1 : DVB-S #1 got 18988 bytes
2018-07-21 15:54:09.209 [ TRACE] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - checking FE status (ready)
2018-07-21 15:54:09.213 [ TRACE] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - status 1 (001F)
2018-07-21 15:54:09.437 [ TRACE] mpegts: input Silicon Labs Si2183 #1 : DVB-S #1 got 18800 bytes
2018-07-21 15:54:09.621 [ TRACE] mpegts: input Silicon Labs Si2183 #1 : DVB-S #1 got 12408 bytes
2018-07-21 15:54:09.855 [ TRACE] mpegts: input Silicon Labs Si2183 #1 : DVB-S #1 got 18988 bytes
2018-07-21 15:54:10.148 [ TRACE] mpegts: input Silicon Labs Si2183 #1 : DVB-S #1 got 18800 bytes
2018-07-21 15:54:10.209 [ TRACE] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - checking FE status (ready)
2018-07-21 15:54:10.213 [ TRACE] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - status 1 (001F)
2018-07-21 15:54:10.424 [ TRACE] mpegts: input Silicon Labs Si2183 #1 : DVB-S #1 got 18800 bytes
2018-07-21 15:54:10.692 [ TRACE] mpegts: input Silicon Labs Si2183 #1 : DVB-S #1 got 18800 bytes
2018-07-21 15:54:10.720 [ TRACE] mpegts: input Silicon Labs Si2183 #1 : DVB-S #1 got 3572 bytes
2018-07-21 15:54:11.209 [ TRACE] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - checking FE status (ready)
2018-07-21 15:54:11.213 [ TRACE] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - status 1 (001F)
2018-07-21 15:54:11.491 [ TRACE] mpegts: input Silicon Labs Si2183 #1 : DVB-S #1 got 18800 bytes
2018-07-21 15:54:11.724 [ TRACE] mpegts: input Silicon Labs Si2183 #1 : DVB-S #1 got 17296 bytes
2018-07-21 15:54:11.993 [ TRACE] mpegts: input Silicon Labs Si2183 #1 : DVB-S #1 got 18988 bytes
2018-07-21 15:54:12.209 [ TRACE] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - checking FE status (ready)
2018-07-21 15:54:12.213 [ TRACE] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - status 1 (001F)
2018-07-21 15:54:12.286 [ TRACE] mpegts: input Silicon Labs Si2183 #1 : DVB-S #1 got 18800 bytes
2018-07-21 15:54:12.562 [ TRACE] mpegts: input Silicon Labs Si2183 #1 : DVB-S #1 got 18988 bytes
2018-07-21 15:54:12.785 [ TRACE] mpegts: input Silicon Labs Si2183 #1 : DVB-S #1 got 18800 bytes
2018-07-21 15:54:12.823 [ TRACE] mpegts: input Silicon Labs Si2183 #1 : DVB-S #1 got 2444 bytes
2018-07-21 15:54:13.117 [ TRACE] mpegts: input Silicon Labs Si2183 #1 : DVB-S #1 got 18988 bytes
2018-07-21 15:54:13.209 [ TRACE] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - checking FE status (ready)
2018-07-21 15:54:13.213 [ TRACE] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - status 1 (001F)
2018-07-21 15:54:13.414 [ TRACE] mpegts: input Silicon Labs Si2183 #1 : DVB-S #1 got 18988 bytes
2018-07-21 15:54:13.690 [ TRACE] mpegts: input Silicon Labs Si2183 #1 : DVB-S #1 got 18800 bytes
2018-07-21 15:54:13.824 [ TRACE] mpegts: input Silicon Labs Si2183 #1 : DVB-S #1 got 12220 bytes
2018-07-21 15:54:14.117 [ TRACE] mpegts: input Silicon Labs Si2183 #1 : DVB-S #1 got 18800 bytes
2018-07-21 15:54:14.209 [ TRACE] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - checking FE status (ready)
2018-07-21 15:54:14.213 [ TRACE] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - status 1 (001F)
2018-07-21 15:54:14.389 [ TRACE] mpegts: input Silicon Labs Si2183 #1 : DVB-S #1 got 18800 bytes
2018-07-21 15:54:14.676 [ TRACE] mpegts: input Silicon Labs Si2183 #1 : DVB-S #1 got 18988 bytes
2018-07-21 15:54:14.962 [ TRACE] mpegts: input Silicon Labs Si2183 #1 : DVB-S #1 got 16168 bytes
2018-07-21 15:54:15.209 [ TRACE] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - checking FE status (ready)
2018-07-21 15:54:15.218 [ TRACE] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - status 1 (001F)
2018-07-21 15:54:15.651 [ TRACE] mpegts: input Silicon Labs Si2183 #1 : DVB-S #1 got 18800 bytes
2018-07-21 15:54:15.874 [ TRACE] mpegts: input Silicon Labs Si2183 #1 : DVB-S #1 got 18800 bytes
2018-07-21 15:54:15.927 [ TRACE] mpegts: input Silicon Labs Si2183 #1 : DVB-S #1 got 3008 bytes
...
...
(just this sequence repeats continually no matter how long the mux scheduler runs for.)
...
...
2018-07-21 15:54:59.214 [ TRACE] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - checking FE status (ready)
2018-07-21 15:54:59.218 [ TRACE] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - status 1 (001F)
2018-07-21 15:54:59.653 [ TRACE] mpegts: input Silicon Labs Si2183 #1 : DVB-S #1 got 18800 bytes
2018-07-21 15:54:59.861 [ TRACE] mpegts: input Silicon Labs Si2183 #1 : DVB-S #1 got 18800 bytes
2018-07-21 15:54:59.978 [ TRACE] mpegts: input Silicon Labs Si2183 #1 : DVB-S #1 got 7708 bytes
2018-07-21 15:55:00.006 [ INFO] subscription: 0036: "" unsubscribing
2018-07-21 15:55:00.006 [ DEBUG] mpegts: 11804.2V in NET - close PID 0000 (0) [2/0x56350073b970]
2018-07-21 15:55:00.006 [ DEBUG] mpegts: 11804.2V in NET - stopping mux
2018-07-21 15:55:00.006 [ DEBUG] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - stopping 11804.2V in NET
2018-07-21 15:55:00.006 [ TRACE] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - waiting for dvr thread
2018-07-21 15:55:00.007 [ DEBUG] linuxdvb: Silicon Labs Si2183 #1 : DVB-S #1 - stopped dvr thread
2018-07-21 15:55:00.007 [ TRACE] mpegts: Silicon Labs Si2183 #1 : DVB-S #1 - flush subscribers
2018-07-21 15:55:00.007 [ TRACE] mpegts: 11804.2V in NET - flush tables
2018-07-21 15:55:00.007 [ TRACE] descrambler: mux 0x56350010bc70 - flush tables
2018-07-21 15:55:00.007 [ TRACE] mpegts: table: mux 0x56350010bc70 destroy pmt 02/FF (2) pid 105D (4189)
2018-07-21 15:55:00.007 [ TRACE] mpegts: table: mux 0x56350010bc70 free pmt 02/FF (2) pid 105D (4189)
2018-07-21 15:55:00.007 [ TRACE] mpegts: table: mux 0x56350010bc70 destroy pmt 02/FF (2) pid 151A (5402)
2018-07-21 15:55:00.007 [ TRACE] mpegts: table: mux 0x56350010bc70 free pmt 02/FF (2) pid 151A (5402)
2018-07-21 15:55:00.007 [ TRACE] mpegts: table: mux 0x56350010bc70 destroy pmt 02/FF (2) pid 1519 (5401)
2018-07-21 15:55:00.007 [ TRACE] mpegts: table: mux 0x56350010bc70 free pmt 02/FF (2) pid 1519 (5401)
2018-07-21 15:55:00.007 [ TRACE] mpegts: table: mux 0x56350010bc70 destroy pmt 02/FF (2) pid 141C (5148)

```



```

2018-07-21 15:55:00.008 [ TRACE] mpegts: table: mux 0x56350010bc70 destroy nit 00/00 (0) pid 0010 (16)
2018-07-21 15:55:00.008 [ TRACE] mpegts: table: mux 0x56350010bc70 free nit 00/00 (0) pid 0010 (16)
2018-07-21 15:55:00.008 [ TRACE] mpegts: table: mux 0x56350010bc70 destroy cat 01/FF (1) pid 0001 (1)
2018-07-21 15:55:00.008 [ TRACE] mpegts: table: mux 0x56350010bc70 free cat 01/FF (1) pid 0001 (1)
2018-07-21 15:55:00.008 [ TRACE] mpegts: table: mux 0x56350010bc70 destroy pat 00/00 (0) pid 0000 (0)
2018-07-21 15:55:00.008 [ TRACE] mpegts: table: mux 0x56350010bc70 free pat 00/00 (0) pid 0000 (0)
2018-07-21 15:55:00.008 [ TRACE] mpegts: 11804.2V in NET - mi=0x5635000e4040
2018-07-21 15:55:00.008 [ DEBUG] mpegts: 11804.2V in NET - close PID 0000 (0) [20/0x7faf9c004b60]
2018-07-21 15:55:00.008 [ DEBUG] mpegts: 11804.2V in NET - close PID 0001 (1) [16/0x7faf9c006030]
2018-07-21 15:55:00.008 [ DEBUG] mpegts: 11804.2V in NET - close PID 0010 (16) [20/0x7faf9c008ec0]
2018-07-21 15:55:00.008 [ DEBUG] mpegts: 11804.2V in NET - close PID 0011 (17) [16/0x7faf9c00b860]
2018-07-21 15:55:00.008 [ DEBUG] mpegts: 11804.2V in NET - close PID 0011 (17) [20/0x7faf9c00a390]
2018-07-21 15:55:00.008 [ DEBUG] mpegts: 11804.2V in NET - close PID 0012 (18) [20/0x7faf60033c80]
2018-07-21 15:55:01.977 [ TRACE] linuxdVB: Silicon Labs Si2183 #1 : DVB-S #1 - checking FE status

```

I do see that descrambler sees the CAT, but it doesn't seem to do anything with it...

```

2018-07-21 15:54:01.361 [ TRACE] descrambler: CAT data (len 31)
2018-07-21 15:54:01.361 [ TRACE] descrambler: 09 04 56 04 E0 30 09 04 0E 00 00 5A 09 04 0E 00 ..V..0.....Z...
.
2018-07-21 15:54:01.361 [ TRACE] descrambler: E0 5A 5F 04 53 41 50 53 FD 05 53 41 50 53 00 ..Z..SAPS..SAPS.

```

As a workaround, I have a dummy timer set to record for twenty minutes each day, but sometimes this is not enough to catch all the emms needed.

#6 - 2018-11-06 16:13 - Joe User

- File `mpegts_mux_sched.patch` added

Looking at the code, it appears that `dvbcam` is the only CA which has a `"cac_cat_update"` module, and thus the only CA which will process EMM messages during a "mux schedule".

I made a patch to turn the "mux" scheduler into a "service" scheduler instead. After this patch, you select a service encrypted by the system you want to process EMMs for. Downside is it will also descramble the service, wasting CPU. But it is much better than having to create a dummy timer recording. It is also much nicer since you can take advantage of the cron method of scheduling to have it run multiple times a day instead of having to create multiple timer recordings.

```

diff --git a/src/input/mpegts/mpegts_mux_sched.c b/src/input/mpegts/mpegts_mux_sched.c
index 46ca4d18e..9e484cf7e 100644
--- a/src/input/mpegts/mpegts_mux_sched.c
+++ b/src/input/mpegts/mpegts_mux_sched.c
@@ -90,13 +90,13 @@ mpegts_mux_sched_class_mux_list ( void *o, const char *lang )
    htmsmsg_t *m, *p;

    p = htmsmsg_create_map();
-   htmsmsg_add_str (p, "class", "mpegts_mux");
+   htmsmsg_add_str (p, "class", "mpegts_service");
    htmsmsg_add_bool(p, "enum", 1);

    m = htmsmsg_create_map();
    htmsmsg_add_str (m, "type", "api");
    htmsmsg_add_str (m, "uri", "idnode/load");
-   htmsmsg_add_str (m, "event", "mpegts_mux");
+   htmsmsg_add_str (m, "event", "mpegts_service");
    htmsmsg_add_msg (m, "params", p);

    return m;
@@ -142,9 +142,9 @@ const idclass_t mpegts_mux_sched_class =
    },
    {
        .type     = PT_STR,
-       .id       = "mux",
-       .name     = N_("Mux"),
-       .desc     = N_("The mux to play when the entry is triggered."),
+       .id       = "service",
+       .name     = N_("Service"),
+       .desc     = N_("The service to play when the entry is triggered."),
        .off      = offsetof(mpegts_mux_sched_t, mms_mux),
        .list     = mpegts_mux_sched_class_mux_list,
    },
@@ -161,9 +161,10 @@ const idclass_t mpegts_mux_sched_class =
    .type     = PT_INT,
    .id       = "timeout",
    .name     = N_("Timeout (secs)"),
-   .desc     = N_("The length of time (in seconds) to play the mux "
+   .desc     = N_("The length of time (in seconds) to play the service "

```

```

                "(1 hour = 3600)."),
        .off      = offsetof(mpegts_mux_sched_t, mms_timeout),
+       .def.i    = 600,
    },
    {
        .type     = PT_BOOL,
@@ -216,7 +217,7 @@ static streaming_ops_t mpegts_mux_sched_input_ops = {
    static void
    mpegts_mux_sched_timer ( void *p )
    {
-   mpegts_mux_t *mm;
+   service_t *s;
        mpegts_mux_sched_t *mms = p;
        time_t nxt;

@@ -228,9 +229,9 @@ mpegts_mux_sched_timer ( void *p )
        if (!mms->mms_mux)
            return;

-   /* Find mux */
-   if (!(mm = mpegts_mux_find(mms->mms_mux))) {
-       tvhdebug(LS_MUXSCHED, "mux has been removed, delete sched entry");
+   /* Find service */
+   if (!(s = service_find_by_uuid(mms->mms_mux))) {
+       tvhdebug(LS_MUXSCHED, "service has been removed, delete sched entry");
        mpegts_mux_sched_delete(mms, 1);
        return;
    }
@@ -241,13 +242,13 @@ mpegts_mux_sched_timer ( void *p )

        if (!mms->mms_prch)
            mms->mms_prch = calloc(1, sizeof(*mms->mms_prch));
-       mms->mms_prch->prch_id = mm;
+       mms->mms_prch->prch_id = s;
        mms->mms_prch->prch_st = &mms->mms_input;

        mms->mms_sub
-       = subscription_create_from_mux(mms->mms_prch, NULL, mms->mms_weight,
+       = subscription_create_from_service(mms->mms_prch, NULL, mms->mms_weight,
                                         mms->mms_creator ?: "",
                                         SUBSCRIPTION_MINIMAL,
+                                         SUBSCRIPTION_CONTACCESS |SUBSCRIPTION_EMM,
                                         NULL, NULL, NULL);

        /* Failed (try-again soon) */
@@ -324,7 +325,7 @@ mpegts_mux_sched_create ( const char *uuid, htmsg_t *conf )
        idnode_load(&mms->mms_id, conf);

        /* Validate */
-       if (!mpegts_mux_find(mms->mms_mux ?: "")) ||
+       if (!service_find_by_uuid(mms->mms_mux ?: "")) ||
            !mms->mms_cronstr) {
            mpegts_mux_sched_delete(mms, 1);
            return NULL;

```

Note: after applying this patch, any "mux" schedules you had previously set up will be automatically removed. If you were using the mux scheduler for other purposes, you can recreate a schedule by selecting a service on the mux you previously had scheduled.

#7 - 2018-11-06 21:23 - Jaroslav Kysela

Some things:

- 1) We moved from newcamd to capmt (dvbapi) in the middle of the bug report. The EMM updates should be sent through newcamd (cac_caid_update callback).
- 2) For capmt (dvbapi) the oscam don't know which CAID should be updated until some service is subscribed. Oscam asks for PIDs in this case.

Regarding the proposed fix (previous comment) for capmt:

- 1) keep raw mux scheduler
- 2) create "base" class for mux scheduler
- 3) add service scheduler reusing the "base" class with the mux scheduler
- 4) create PR on github for review/merge, I cannot comment code nicely in bug reports

#8 - 2018-11-09 21:09 - Joe User

Sorry I did not make it more clear. This patch is only intended as a temporary workaround until a real solution is made. That is why I did not bother with a PR.

I looked into various ways to implement a fix, and did not see any easy solution. I did think of adding an entry to the "mux scheduler" to select a CAID to use, but I did not see an easy way to insert it into the descrambler.

Scheduling a service would only work if there could be a way to set some flag so that the service was not descrambled and yet EMMs would be processed. I tried the SUBSCRIPTION_NODESCR flag, but that stops the descrambler from even being started. Can you see some way to tell the descrambler to process EMMs but not descramble?

If a way is found, I was thinking instead of duplicating a lot of code for a "service scheduler", maybe there can be a "mux/service scheduler" where there are two columns, one to select the mux, and the second to optionally select a specific service from that mux. If no service is selected, then it would behave just as before.

There would need to be new code for retrieving the list of services. I copied the "mpegts_mux_sched_class_mux_list" to create this:

```
static htmsg_t *
mpegts_mux_sched_class_service_list ( void *o, const char *lang )
{
    htmsg_t *m, *p;

    p = htmsg_create_map();
    htmsg_add_str (p, "class", "mpegts_service");
    htmsg_add_bool (p, "enum", 1);

    m = htmsg_create_map();
    htmsg_add_str (m, "type", "api");
    htmsg_add_str (m, "uri", "idnode/load");
    htmsg_add_str (m, "event", "mpegts_service");
    htmsg_add_msg (m, "params", p);

    return m;
}
```

But it will return all services. How can I make it return only the services for the entered mux? (The UUID of the mux will be known) Can I add a filter for the "multiplex_uuid" somehow? I did not see any cases of using a filter with the idnode api.

If you give me a hint for that, I think I can create the rest of the code for a "mux/service scheduler". But again, that would be only the first step and ideally there would be a flag to tell the descrambler to only process EMMs and not actually descramble the service.

Also, it would also be nice to be able to filter only enabled services...

#9 - 2018-11-09 21:21 - Jaroslav Kysela

It's not possible to make a list which depends on other field (because values are handled only in the browser until they're saved). Keep separated mux / service entries.

But again, that would be only the first step and ideally there would be a flag to tell the descrambler to only process EMMs and not actually descramble the service.

No way for the oscam dvbapi (capmt) network protocol. The oscam cannot subscribe anything and it reacts only on service data once the client requests to descramble something. On other side, tvh does not know which CAID oscam supports through this protocol until the service descrambling is requested (PMT is sent).

#10 - 2018-11-11 21:38 - Joe User

Jaroslav Kysela wrote:

It's not possible to make a list which depends on other field (because values are handled only in the browser until they're saved). Keep separated mux / service entries.

I had already almost finished implementing a combined scheduler, so I finished it. I created a pull request: [[<https://github.com/tvheadend/tvheadend/pull/1210>]]

I also included a default timeout of 600 (10min) since without it, the default is 0, which seems to mean never timeout which is not good if someone forgets to set a timeout.

Also, in testing I noticed the settings were only validated when created, and not on a change, so I added a validation function which is called during creation and change.

No way for the oscam dvbapi (capmt) network protocol. The oscam cannot subscribe anything and it reacts only on service data once the client requests to descramble something. On other side, tvh does not know which CAID oscam supports through this protocol until the service descrambling is requested (PMT is sent).

I was thinking more of a flag to do everything except the actual descrambling in tvhcsa. Maybe pretend no valid keys? I will look into when I have time...

#11 - 2018-11-11 22:50 - Joe User

BTW, if the provider you want to process EMMs for has an encrypted radio channel, choosing it will significantly reduce cpu usage. smile.png

#12 - 2019-06-13 18:39 - matias segurola

Joe User wrote:

BTW, if the provider you want to process EMMs for has an encrypted radio channel, choosing it will significantly reduce cpu usage. smile.png

It is still working on this idea or left halfway

Files

mpegts_mux_sched.patch	3.33 KB	2018-11-06	Joe User
------------------------	---------	------------	----------