

Kerbal Space Program - Bug #7515

1.1 prerelease won't start without pulseaudio

03/30/2016 02:26 AM - kshade

Status:	Closed	Start date:	03/30/2016
Severity:	Normal	% Done:	100%
Assignee:	NathanKell		
Category:	Application		
Target version:			
Version:	Build 01172	Language:	English (US)
Platform:	Linux	Mod Related:	No
Expansion:			

Description

Steps to reproduce:

1. Install KSP 1.1 Prerelease on a Linux installation without Pulseaudio
2. Try to launch the game from Steam or a console
3. Black window briefly comes up, then disappears

Diagnosis by stupid enduser:

This is the relevant section from the Player.log:

```
FMOD failed to get number of drivers ... An error occured that wasn't supposed to. Contact support.
```

```
(Filename: Line: 533)
```

Looking for that error on Google (misspelled "occured" included) reveals that KSP isn't the only Unity 5 game suffering from this issue. I've found this thread on the Unity forums: <http://forum.unity3d.com/threads/failed-to-get-number-of-drivers.377026/>

Expected results:

The game should look for a running Pulseaudio instance, use it when there is one, use ALSA instead if there isn't (like in the Unity 4 version).

Workaround:

Install Pulseaudio. Scrub hands until you feel clean again.

History

#1 - 03/30/2016 02:28 AM - kshade

Filed this as low priority because most distributions come with Pulseaudio by default. Hope that's appropriate.

#2 - 03/30/2016 09:30 AM - sal_vager

- Category changed from Audio to Unity3D

I have bad news kshade, KSP relies on the Unity engine for sound and Unity5 needs pulseaudio on Linux, there's no getting around it on Squads end, Unity Technologies would have to fix their version of FMOD :/

Sorry.

#3 - 03/30/2016 09:43 AM - sal_vager

- Status changed from New to Unity Bug

#4 - 03/30/2016 11:06 AM - kshade

Judging from that thread they are on it though, so that's good.

#5 - 03/31/2016 04:36 AM - steve_v

I see. This is a deal breaker for me as I rely on advanced ALSA functionality and will have no sound at all (in any application) if pulseaudio is running. I cannot uninstall it completely either, as large parts of the system are linked against it's libraries.

I don't think this is low priority BTW, to quote the priority table:

Critical: A large portion of the game is unplayable.

"large portion unplayable" covers "CTD on startup", right?

I'd love to help with finding a workaround I can actually use, but I don't own KSP on steam. Options?

#6 - 03/31/2016 06:07 AM - NathanKell

- Assignee set to NathanKell

- Severity changed from Low to Critical

I agree it is critical, but it is also a deep Unity bug and therefore highly unlikely there's anything we can do on our side. However, I will look into it on the off chance there is.

Also, that thread linked talked about *two* workarounds. One is to install PulseAudio, but the other is to remove it completely rather than having it disabled. Does that work for you kshade?

#7 - 03/31/2016 09:12 AM - kshade

It does, but I had to remove the library it looks for from Steam's Linux runtime - not brilliant.

#9 - 04/02/2016 04:58 AM - silverwingedseraph

Is there a way to inform KSP that it should not use Steam's runtime? I don't have PulseAudio installed (though I do have libpulse, so if that's what breaks FMOD then this won't help), but the Steam Runtime does have it, and I have all the other libraries Kerbal Space Program needs.

If it's useful, here's the result of the crash when running KSP in GDB:

```
Program received signal SIGPWR, Power fail/restart.
[Switching to Thread 0x7ffff1835700 (LWP 7307)]
0x00007ffff79c60c9 in futex_abstimed_wait (cancel=true,
    private=<optimized out>, abstime=0x0, expected=0, futex=0x7ffff247e070)
    at sem_waitcommon.c:42
```

then

```
(gdb) step
```

```
Program received signal SIGXCPU, CPU time limit exceeded.
0x00007ffff60fa657 in do_sigsuspend (set=0x7ffff2480900)
    at ../sysdeps/unix/sysv/linux/sigsuspend.c:29
```

Every thread dies to these signals, until finally:

```
Program received signal SIGSEGV, Segmentation fault.
[Switching to Thread 0x7ffff7fc6780 (LWP 7363)]
0x0000000000b4acfa in ?? ()
```

```
Program received signal SIGABRT, Aborted.
0x00007ffff60fa267 in __GI_raise (sig=sig@entry=6)
    at ../sysdeps/unix/sysv/linux/raise.c:55
55  ../sysdeps/unix/sysv/linux/raise.c: No such file or directory.
```

```
(gdb)
```

Continuing.

```
[Thread 0x7fff937fe700 (LWP 7392) exited]
[Thread 0x7fffe802c700 (LWP 7399) exited]
[Thread 0x7fff92ffd700 (LWP 7393) exited]
[Thread 0x7fff93fff700 (LWP 7391) exited]
[Thread 0x7fff0ff9700 (LWP 7390) exited]
[Thread 0x7fffb17fa700 (LWP 7389) exited]
[Thread 0x7fffb1ffb700 (LWP 7388) exited]
[Thread 0x7fffb27fc700 (LWP 7387) exited]
[Thread 0x7fffb2ffd700 (LWP 7386) exited]
[Thread 0x7fffb37fe700 (LWP 7385) exited]
[Thread 0x7fffb3fff700 (LWP 7384) exited]
[Thread 0x7fffd4fff9700 (LWP 7383) exited]
```

```
[Thread 0x7ffffd57fa700 (LWP 7382) exited]
[Thread 0x7ffffd5ffb700 (LWP 7381) exited]
[Thread 0x7ffffd67fc700 (LWP 7380) exited]
[Thread 0x7ffffd6ffd700 (LWP 7379) exited]
[Thread 0x7ffffd77fe700 (LWP 7378) exited]
[Thread 0x7ffffd7fff700 (LWP 7377) exited]
[Thread 0x7ffffe8dac700 (LWP 7376) exited]
[Thread 0x7ffffe95ad700 (LWP 7375) exited]
[Thread 0x7ffffe9dae700 (LWP 7374) exited]
[Thread 0x7ffffea5af700 (LWP 7373) exited]
[Thread 0x7ffffeb5b1700 (LWP 7371) exited]
[Thread 0x7ffffebfff700 (LWP 7370) exited]
[Thread 0x7fffff1634700 (LWP 7369) exited]
[Thread 0x7fffff1835700 (LWP 7368) exited]
[Thread 0x7fffff18c6700 (LWP 7367) exited]
[Thread 0x7fffff7fc6780 (LWP 7363) exited]
```

Program terminated with signal SIGABRT, Aborted.
The program no longer exists.

#10 - 04/02/2016 05:29 AM - HebaruSan

All of my log files from #7514 belong over here. I tried the suggestions on that thread in vain, but turning pulseaudio back on fixed the problem I was having. Can those attachments be transferred by a moderator?

#11 - 04/02/2016 07:31 AM - steve_v

silverwingedseraph wrote:

Is there a way to inform KSP that it should not use Steam's runtime?

IIRC, steam just launches the binary with LD_LIBRARY_PATH set... so launching KSP outside of steam should do what you ask.

On that note, removing libpulse from the steam runtime should work for *steam users*, but copies from the store (when it drops) are going to be using system libs... and you can't remove the system copy of libpulse without a heap of collateral damage.

How practical would shipping a "steamified" (bundle the steam runtime, it's FOSS right?) version from the store be? at least then there's a workaround for those (like me) who cannot/will not install the pulse binaries.

Bundling the steam runtime would also solve any other distro-specific weirdness, as KSP would be using Ubuntu libraries.

#12 - 04/02/2016 11:00 AM - kspflo

You could always hide your system Pulse libraries using a mount namespace. Let me see, if I can whip something up.

#13 - 04/02/2016 11:45 AM - kshade

KSP isn't the only game affected by this (Stardrive 2 also has this issue, for example), I'd hope that Unity fixes this problem soon:

<http://forum.unity3d.com/threads/several-games-segfault-at-fmod-failed-to-get-number-of-drivers-when-pulseaudio-isnt-running.369943/>
<http://forum.unity3d.com/threads/failed-to-get-number-of-drivers.377026/#post-2544187>

#14 - 04/02/2016 01:30 PM - kspflo

- File *hidepulse.c* added
- File *hidepulse.x86* added
- File *hidepulse.x86_64* added

I believe to have developed a functional workaround using the approach I alluded to earlier. It's an ugly hack, but it should work. (I use Pulseaudio, so my testing has been limited.)

Attached you'll find the C source code and prebuilt binaries for your convenience.

Note 1: It's a setuid executable since mount(2) and unshare(2) require root privileges.

Note 2: Never execute binaries requiring root privileges you downloaded from a stranger on the internet, unless you have some confirmation it's not malicious.

How do you use it?

1. Download the prebuilt binary for 32 or 64bit as appropriate or see below on how to build from source.
2. Install it into your system using *one* of the following commands depending on which binary you downloaded:

```
sudo install -o root -g root -m 4755 -s hidepulse.x86 /usr/bin/  
sudo install -o root -g root -m 4755 -s hidepulse.x86_64 /usr/bin/
```

Should you not wish to install it systemwide, you still *MUST* change ownership to root and *MUST* set the setuid permission bit, or start it using sudo.

3. Now you can launch KSP using *one* of the following commands, again depending on the binary name:

```
hidepulse.x86 /path/to/KSP/KSP.x86  
hidepulse.x86_64 /path/to/KSP/KSP.x86_64
```

(Despite requiring root privileges for the hack, when following the above instructions, KSP will not be launched as root.)

4. On the terminal you should be seeing output like this:

```
hidepulse: Omitting directory 'pulseaudio'  
hidepulse: Omitting file 'libpulse-mainloop-glib.so'  
hidepulse: Omitting file 'libpulse-mainloop-glib.so.0'  
hidepulse: Omitting file 'libpulse-mainloop-glib.so.0.0.5'  
hidepulse: Omitting file 'libpulse-simple.so'  
hidepulse: Omitting file 'libpulse-simple.so.0'  
hidepulse: Omitting file 'libpulse-simple.so.0.1.0'  
hidepulse: Omitting file 'libpulse.so'  
hidepulse: Omitting file 'libpulse.so.0'  
hidepulse: Omitting file 'libpulse.so.0.19.0'
```

5. Enjoy playing KSP!?!

What does it do?

It figures out where your Pulseaudio libraries reside and then creates a new directory containing all libraries other than those related to Pulseaudio. More specifically, it mounts a tmpfs, bind mounts all your libraries into that directory, and then "moves" it over the original directory, without actually physically touching any of your files. By creating a new mount namespace, only child processes will be affected.

How to build (optional):

```
gcc -std=gnu99 -O2 -o hidepulse hidepulse.c -ldl
```

#15 - 04/02/2016 01:48 PM - **steve_v**

kspflo wrote:

I believe to have developed a functional workaround using the approach I alluded to earlier. It's an ugly hack, but it should work.

Thanks boss, I'm not sure I *like* this approach, but I guess it's no more ugly than the ld_preload hack I was contemplating.

Since I haven't actually gotten around to implementing my idea, I'll gladly take yours :)

I don't actually have access to the prerelease, so testing it will have to wait. :(

#16 - 04/02/2016 02:13 PM - **kshade**

it's no more ugly than the ld_preload hack

I disagree, LD_PRELOAD wouldn't require root access.

#17 - 04/02/2016 02:30 PM - **kspflo**

Well, I had most of the mount namespace code lying around, so it was less work for me than writing an LD_PRELOAD "solution" from scratch. You're of course free to come up with your own hack or wait for a Unity fix.

#18 - 04/02/2016 02:37 PM - **steve_v**

kshade wrote:

I disagree, LD_PRELOAD wouldn't require root access.

That is a point, though there's nothing particularly dangerous in there.

I might have a poke at my idea but I kinda need the game for that, as otherwise I don't know which syscalls to hijack :). Later perhaps.

#19 - 04/02/2016 03:14 PM - **kspflo**

steve_v wrote:

I might have a poke at my idea but I kinda need the game for that, as otherwise I don't know which syscalls to hijack ;). Later perhaps.

```
$ strace -fe trace=file ./KSP.x86_64 2>&1 | grep libpulse.so
[pid 6925] open("/usr/lib64/pulseaudio/tls/x86_64/libpulse.so.0", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
[pid 6925] open("/usr/lib64/pulseaudio/tls/libpulse.so.0", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
[pid 6925] open("/usr/lib64/pulseaudio/x86_64/libpulse.so.0", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
[pid 6925] open("/usr/lib64/pulseaudio/libpulse.so.0", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
[pid 6925] open("/usr/lib64/libpulse.so.0", O_RDONLY|O_CLOEXEC) = 21
```

Only one choice, it looks like. No prior stat() or access(). Good luck!

Personally, I dislike the idea of intercepting open() even more and consider the namespace approach a bit more surgical.

#20 - 04/02/2016 03:33 PM - steve_v

kspflo wrote:

No prior stat() or access(). Good luck!

Arse.

Personally, I dislike the idea of intercepting open() even more and consider the namespace approach a bit more surgical.

Yeah, so do I TBH. I might still code this up at some point, (my free-weekend is rapidly vanishing) but I was hoping to leave open() alone. For the present I think I'll just nick yours & file it until release. :)

#21 - 04/02/2016 03:51 PM - kspflo

- File *hidepulse.c* added

- File *hidepulse.x86* added

- File *hidepulse.x86_64* added

I could not resist to make one last improvement. There's really no good reason to bind mount symlinks and not doing so cuts the number of bind mounts by a sizable 60% on my system. With almost 5000 files and directories in /usr/lib64, that's quite significant. Also, I forgot to dclose().

Updated files attached. Instructions from v1 remain valid.

#22 - 04/02/2016 03:52 PM - kshade

Third approach: I loaded up the binary in ghex, looked for libpulse-simple and replaced the l with an x. Strace confirms that it's now trying to load a library that doesn't exist:

```
open("/lib64/tls/x86_64/libpuxse-simple.so.0", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
stat("/lib64/tls/x86_64", 0x7ffc7523e530) = -1 ENOENT (No such file or directory)
open("/lib64/tls/libpuxse-simple.so.0", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
...
```

[kspflo](#): Wasn't trying to be a dick to you by the way, I think your approach is actually pretty cool - it's just not something I can see as an "official" workaround.

#23 - 04/02/2016 04:21 PM - kspflo

kshade wrote:

[kspflo](#): Wasn't trying to be a dick to you by the way, I think your approach is actually pretty cool - it's just not something I can see as an "official" workaround.

No offence taken. Fully understand your reservations.

kshade wrote:

Third approach: I loaded up the binary in ghex, looked for libpulse-simple and replaced the l with an x. Strace confirms that it's now trying to load a library that doesn't exist:

So, for an enduser-friendly workaround, one could load the KSP exe into memory (memfd_create(), or a pipe + vmsplice() on Linux <3.17), patch it, and run it (fexecve()).

I'd add the memory roundtrip, so the file isn't restored by the patcher, if that's a concern. I have a Steam copy, I only ever read about the patcher in devnotes.

I'm genuinely curious how that would work and I might just try for the fun of it.

#24 - 04/02/2016 04:25 PM - kshade

Isn't the non-Steam patcher just Launcher.x86_64? That's in the Steam distribution, just not used.

Another reason for writing a small executable that patches the executable: It could detect if Pulse is actually running, and skip/revert the patch if it is.

#25 - 04/02/2016 04:35 PM - kshade

~~Little correction, libpulse-simple isn't the thing that needs "mangling", there are other references to PulseAudio that do. I'm not sure which one is the actual culprit.~~

#26 - 04/02/2016 04:41 PM - steve_v

kshade wrote:

Isn't the non-Steam patcher just Launcher.x86_64? That's in the Steam distribution, just not used.

AFAICT, the mysterious patcher is supposed to be retrieved by Launcher.x86[64] - but it's been MIA for a long time, and gets an html page instead.

Another reason for writing a small executable that patches the executable: It could detect if Pulse is actually running, and skip/revert the patch if it is.

I like it.

That said, the accepted workaround for the last Unity SNAFU was just some arguments for xxd posted in the Linux thread... Not exactly "official", but it's been in use for some time.

#27 - 04/02/2016 04:57 PM - kshade

Good idea, actually:

```
echo "013ab3d0: 006c 6962 7075 7873 652d 7369 6d70 6c65" | xxd -r - ~/.local/share/Steam/steamapps/common/Kerbal\ Space\ Program/KSP.x86_64
```

Turns out my previous comment wasn't right, it's just that one instance that needs replacing.

#28 - 04/03/2016 10:56 AM - kspflo

- File pulsenomore.c added

- File pulsenomore.x86 added

- File pulsenomore.x86_64 added

New day, new hack.

pulsenomore checks if Pulseaudio is running, if it is, executes the supplied program directly, otherwise loads the executable into memory, replaces "libpulse-simple.so.0" with "/dev/null", and then executes that.

It requires the memfd_create() syscall introduced in 3.17. I actually tried to provide a fallback (open() with O_TMPFILE), but the file descriptor must not be opened with write permission, or ETXTBSY is returned, when trying to execute it. If needed, one could of course write the patched file to disk, but I'll wait for someone without memfd_create() to come along first.

I disabled Pulseaudio autospawn and KSP crashed as expected. Using pulsenomore, it launches successfully.

No more setuid needed, just make it executable, and run it.

Edit:

Compile with: gcc -std=gnu99 -O2 -o pulsenomore pulsenomore.c -ldl

#29 - 04/03/2016 11:21 AM - steve_v

kspflo wrote:

New day, new hack.

kspflo, you are legendary. Thanks again for the effort on this one.

This thing is pretty elegant, I already have some unrelated ideas I might steal it for. ;)

I'll wait for someone without memfd_create() to come along first.

Looks like even Debian stable has it (backported), so with any luck this will never happen. :)

#30 - 04/03/2016 11:43 AM - kshade

[kspflo](#): Works perfectly fine from what I can see, thanks! I put it into the game's launch options like this:

```
/home/kshade/bin/pulsenomore.x86_64 %command%_64
```

#31 - 04/03/2016 01:07 PM - kspflo

Glad to hear it's working.

steve_v wrote:

kspflo, you are legendary. Thanks again for the effort on this one.

Thanks for the praise. If nothing else, it was educational. Tried a few things before I settled on this.

steve_v wrote:

This thing is pretty elegant, I already have some unrelated ideas I might steal it for. :)

Steal away. ;)

steve_v wrote:

Looks like even Debian stable has it (backported), so with any luck this will never happen. :)

Yes, I was thinking about Debian. Good to know they backport these things.

kshade wrote:

```
%command%_64
```

It's so obvious. How did I miss this? Thank you!

Remember though, according to the pre-release announcement, Squad intends to make the 64bit exe the default, once 1.1 is released.

#32 - 04/09/2016 02:52 PM - sagir3

kspflo wrote:

New day, new hack.

pulsenomore checks if Pulseaudio is running, if it is, executes the supplied program directly, otherwise loads the executable into memory, replaces "libpulse-simple.so.0" with "/dev/null", and then executes that.

It requires the memfd_create() syscall introduced in 3.17. I actually tried to provide a fallback (open() with O_TMPFILE), but the file descriptor must not be opened with write permission, or ETXTBSY is returned, when trying to execute it. If needed, one could of course write the patched file to disk, but I'll wait for someone without memfd_create() to come along first.

I disabled Pulseaudio autospawn and KSP crashed as expected. Using pulsenomore, it launches successfully.

No more setuid needed, just make it executable, and run it.

Edit:

Compile with: gcc -std=gnu99 -O2 -o pulsenomore pulsenomore.c -ldl

Thank you so much! Working perfectly here on Arch Linux x64. I run KSP with the following command: ./pulsenomore

```
/<steamlibrary>/common/Kerbal\ Space\ Program/KSP.x86_64
```

I made an alias in .bashrc to make it easier.

#33 - 04/10/2016 04:01 PM - Lysius

I was wondering why it didn't work from Steam, didn't work when running KSP.x86_64 directly but did work when running KSP.x86: have pulse libraries installed for x86_64 but not for x86.

But I found a quite simple workaround:

```
run
```

```
ln -s /dev/null libpulse-simple.so.0
```

in the KSP directory
and change the launch options to

```
LD_LIBRARY_PATH=.:$LD_LIBRARY_PATH %command%
```

#34 - 04/15/2016 08:50 PM - NathanKell

Is there **anyone** for whom the workarounds do not work?

#35 - 04/17/2016 02:24 PM - Lysius

I have constantly had good success with my simple workaround of the symlink to /dev/null in a directory at the beginning of LD_LIBRARY_PATH. Setting that up (when no pulse is detected running) in a launcher shell script that is called instead of the binary could be the way to go. At least it does not require root and I prefer a simple shell script to a binary that does some black magic.

#36 - 04/21/2016 05:11 PM - psycho_zs

I use a script with bunch of stuff anyway, so adding this workaround was easy. It works.

Fixing this bug would require something like logic of 'pulsenomore' incorporated into KSP binary.

#37 - 09/14/2016 10:42 PM - psycho_zs

- % Done changed from 0 to 100

Seems fixed in 1.2

#38 - 03/15/2017 07:59 PM - Squelch

- Project changed from KSP Pre-Release to PrereleaseArchive

- Category changed from Unity3D to Unity3D

#39 - 03/24/2017 05:46 PM - sal_vager

- Project changed from PrereleaseArchive to Kerbal Space Program

- Category changed from Unity3D to 368

- Platform Linux added

Moving to main tracker for accessibility.

#40 - 03/20/2018 11:40 AM - Squelch

- Severity changed from Critical to Normal

#41 - 06/13/2019 11:18 PM - nestor

- Status changed from Unity Bug to Resolved

#42 - 06/17/2019 07:38 PM - chris.fulton

- Status changed from Resolved to Closed

Files

Player.log	15.8 KB	03/30/2016	kshade
sysinf.txt	1.93 KB	03/30/2016	kshade
hidepulse.c	3.97 KB	04/02/2016	kspflo
hidepulse.x86	13.9 KB	04/02/2016	kspflo
hidepulse.x86_64	16.1 KB	04/02/2016	kspflo
hidepulse.c	4.49 KB	04/02/2016	kspflo
hidepulse.x86	14.1 KB	04/02/2016	kspflo
hidepulse.x86_64	16.3 KB	04/02/2016	kspflo
pulsenomore.c	3.33 KB	04/03/2016	kspflo
pulsenomore.x86	13.6 KB	04/03/2016	kspflo
pulsenomore.x86_64	15.6 KB	04/03/2016	kspflo