### Tails - Bug #11729

**Emergency shutdown triggered after resuming from suspend**

08/26/2016 02:50 AM - segfault

<table>
<thead>
<tr>
<th>Status:</th>
<th>Resolved</th>
<th>Start date:</th>
<th>08/26/2016</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
<td>Due date:</td>
<td></td>
</tr>
<tr>
<td>Assignee:</td>
<td></td>
<td>% Done:</td>
<td>100%</td>
</tr>
<tr>
<td>Category:</td>
<td></td>
<td>Estimated time:</td>
<td>0.00 hour</td>
</tr>
<tr>
<td>Target version:</td>
<td>Tails_3.13.2</td>
<td>Spent time:</td>
<td>0.00 hour</td>
</tr>
<tr>
<td>Feature Branch:</td>
<td>feature/14556-show-suspend-button</td>
<td>Starter:</td>
<td></td>
</tr>
<tr>
<td>Type of work:</td>
<td>Code</td>
<td>Affected tool:</td>
<td></td>
</tr>
<tr>
<td>Blueprint:</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Description**

Sometimes Tails will immediately shut down after resuming from suspend.

**Related issues:**

- Related to Tails - Feature #14556: Show a suspend to RAM button in the status... Resolved 08/30/2017
- Related to Tails - Bug #16787: Trigger emergency shutdown on resume when the ... Confirmed

**Associated revisions**

- **Revision 0f9a9a04 - 03/27/2019 07:46 PM - segfault**
  Disable emergency shutdown during suspend (refs: #11729)

- **Revision d6b6302e - 03/28/2019 10:06 PM - segfault**
  Disable emergency shutdown during suspend (refs: #11729)

- **Revision e4e7f812 - 04/04/2019 09:37 AM - segfault**
  Disable emergency shutdown during suspend (refs: #11729)

- **Revision 268474a4 - 04/04/2019 09:51 AM - segfault**
  Document that emergency shutdown is disabled during suspend (refs: #11729)

- **Revision a7c35d0f - 04/04/2019 11:19 AM - segfault**
  Document that emergency shutdown is disabled during suspend (refs: #11729)

- **Revision 8f4bf911 - 04/04/2019 01:05 PM - intrigeri**
  Merge remote-tracking branch 'origin/feature/14556-show-suspend-button' into stable (Fix-committed: #11729, #14556)

- **Revision 6ea13262 - 04/04/2019 01:06 PM - intrigeri**
  Design doc: add link to implementation (refs: #11729)

**History**

#### #2 - 08/26/2016 02:52 AM - segfault

This happens to me once in a while. I think this might be a blocker for #11395, because it could lead to loss of unsaved work.

#### #3 - 08/26/2016 04:16 AM - intrigeri

- Target version deleted (2017)

(Generally, our roadmap should reflect collective decisions. Now, sometimes some of us add to it stuff that they intend to do themselves, which is fine. But adding to the roadmap without any assignee feels wrong to me.)
Confirmed by several reports and by myself: after suspend, Tails shuts down when running from USB stick but not when running from SD card, on a Thinkpad X201

#5 - 08/28/2016 06:30 AM - intrigeri

At this point, I feel the need to point out that we started polishing the support of suspend-to-RAM on the grounds that it was already mostly working and could therefore be advertised a bit better. The previous status quo was that developers (at least myself) thought that suspend-to-RAM did not work and was not supported, but power users started using it and relying on it. And now we're discovering serious issues with this feature, one after the other, some of them being potentially hard to fix. So I'd like us to take a step back and double-check that it still seems reasonable to add this feature on the list of things we need to make work nicely and keep working in the foreseeable future. At least I need to know if "we" is "the Foundations Team" in this case; I'm not convinced it should be.

#6 - 08/28/2016 09:42 AM - anonym

I agree with intrigeri's concern, but I'm just gonna drop a suggestion for how we can fix this issue: we simply hook into the suspend action to kill the udev watchdog, and restart it when we resume. This assumes that the cause of this issue is that some hardware detaches the USB drive on suspend, and reattaches it on resume, and that that doesn't cause other issues.

#9 - 08/29/2016 07:20 AM - sajolida

- Assignee set to intrigeri
- QA Check set to Info Needed

Yeap, we shouldn't allow users to suspend from the desktop if resuming shuts down their session.

intrigeri: could you be more explicit regarding "discovering serious issues [...] one after the other". Which issues other than this one are you referring to? Maybe we should add Redmine relationships or discuss this outside of this ticket if these issues go beyond the scope of this ticket only or if the discussion gets complicated.

#10 - 08/31/2016 05:59 AM - intrigeri

- Assignee changed from intrigeri to sajolida

intrigeri: could you be more explicit regarding "discovering serious issues [...] one after the other". Which issues other than this one are you referring to?

In retrospect, I think I was erroneously mixing together things related to locking screen and things related to suspend. Sorry for the confusion.

#11 - 11/07/2016 02:27 PM - sajolida
Is this still the case with 3.0?

Related to Feature #14556: Show a suspend to RAM button in the status menu added

I'll try to reproduce with Tails 3.4.

can't reproduce with Tails 3.4, closing it.

Do you still experience this with 3.4?

This didn't happen every time, but only now and then. And I don't use Tails regularly enough currently to tell if this is fixed or not. Maybe we can close it if there are no bug reports since Tails 3.0.
Do you still experience this with 3.4?

This didn’t happen every time, but only now and then. And I don’t use Tails regularly enough currently to tell if this is fixed or not. Maybe we can close it if there are no bug reports since Tails 3.0.

Marking it as resolved accordingly.

---

**#19** - 03/24/2019 11:23 AM - segfault

- Status changed from Resolved to Confirmed
- Assignee changed from mercedes508 to segfault
- QA Check deleted (Info Needed)

I can still reproduce this issue. On my test hardware, it happens in about 1 out of 3 times I resume from suspend. It prints the following log messages:

```
brcmsmac bcma0:1: brcms_ops_bss_info_changed: qos enabled: false (implement)
brcmsmac bcma0:1: brcms_ops_config: change power-save mode: false (implement)
brcmsmac bcma0:1: brcms_ops_bss_info_changed: qos enabled: false (implement)
brcmsmac bcma0:1: brcms_ops_config: change power-save mode: false (implement)
systemd-udevd: Process 'lmt-udev force' terminated by signal TERM.
systemd-udevd: Failed to wait for spawned command 'lmt-udev force': Input/output error
```

Followed by a lot of squashfs error messages:

```
SQUASHFS error: Unable to read data cache entry
SQUASHFS error: Unable to read page, block X, size Y
```

---

**#20** - 03/24/2019 01:28 PM - segfault

This seems to be indeed caused by our udev-watchdog-wrapper, because I don’t see this issue after stopping the watchdog, I suspended and resumed 10 times in a row now.

Next step: Try to stop the watchdog before suspending (and restarting it afterwards), as already suggested by anonym in #note-6.
So we could use systemd's inhibitor locks for this:

https://www.freedesktop.org/wiki/Software/systemd/inhibit/

This would certainly make the suspend button in the system menu work without causing this issue. But the documentation includes this note:

Note that this will only be sent out for suspend/resume cycles done via logind, i.e. generally only for high-level user-induced suspend cycles, and not automatic, low-level kernel induced ones which might exist on certain devices with more aggressive power management.

Now I'm wondering whether we could hook into somewhere more low level to also prevent this issue for these "devices with more aggressive power management". I will do some more research.

Now I'm wondering whether we could hook into somewhere more low level to also prevent this issue for these "devices with more aggressive power management".

AFAIK computers that would auto-suspend via low-level (driver, hardware) means are basically smartphones and possibly some tablets. While it's great that the systemd developers and doc are so forward-thinking, at this point I don't think Tails runs on any such device in a remotely usable way. (FWIW, I'm testing Tails on a tablet regularly and despite continuous improvements, it's not remotely usable yet; I've never seen that device auto-suspend anyway.) So I would suggest you don't need to do more research :)

The kernel supports registering suspend notifier functions which are then called before the suspend, see register_pm_notifier in


Unfortunately, I can't find a kernel module which uses these notifiers and exposes them to the user space.

logind seems not to make use of this kernel functionality, it only sends the PrepareForSleep signal when the suspend is triggered via logind itself:

https://github.com/systemd/systemd/blob/e62a7f7ea757259eb330da5b6d3ab4ede46400a2/src/login/logind-dbus.c#L1659
Now I'm wondering whether we could hook into somewhere more low level to also prevent this issue for these "devices with more aggressive power management".

AFAIK computers that would auto-suspend via low-level (driver, hardware) means are basically smartphones and possibly some tablets. While it's great that the systemd developers and doc are so forward-thinking, at this point I don't think Tails runs on any such device in a remotely usable way. (FWIW, I'm testing Tails on a tablet regularly and despite continuous improvements, it's not remotely usable yet; I've never seen that device auto-suspend anyway.) So I would suggest you don't need to do more research :)

Ah OK, I thought there might also be laptops which do this. Well then I would just go on and implement it the systemd way.

I think the right place to do this is in the tails-shutdown-on-media-removal.service / udev-watchdog-wrapper itself, which should just stop watching for the udev signals during suspend. But I would very much like to do this in Python instead of bash, so I'm tempted to rewrite the udev-watchdog-wrapper in Python (this would also allow us to use udisks via the D-Bus API instead of parsing udevadm output).

Alternatively, I could write a udev-watchdog-wrapper-wrapper which simply kills / restarts the udev-watchdog-wrapper before / after suspend.

I think the right place to do this is in the tails-shutdown-on-media-removal.service / udev-watchdog-wrapper itself, which should just stop watching for the udev signals during suspend.

Another implementation idea would be to use sleep.target, somehow, to toggle on/off its reaction to udev events. For example:
a /lib/systemd/system-sleep/udev-watchdog script (systemd-sleep(8)):
- if passed "pre", runs touch /run/something-clever/sleeping
- if passed "post", runs rm /run/something-clever/sleeping
- udev-watchdog does not react to udev events if /run/something-clever/sleeping exists

This should guarantee that udev events are not acted upon by udev-watchdog while suspending or resuming. It's less nice than some other options, such as reacting to D-Bus stuff sent by logind but it probably requires much less work to get it right, and avoids the need to rewrite code into a different language.

But I would very much like to do this in Python instead of bash,

Indeed, I'm not looking forward to reviewing code that takes an inhibitor lock via the logind D-Bus API in shell.

so I'm tempted to rewrite the udev-watchdog-wrapper in Python (this would also allow us to use udisks via the D-Bus API instead of parsing udevadm output).

Why not, but keep in mind that this is a persistent daemon and Python will eat more RAM. (I guess this would be a perfect use case for Go or Rust, in passing.)

---

#27 - 03/27/2019 07:48 PM - segfault
- Status changed from Confirmed to In Progress

Applied in changeset tails/0f9a9a043b0a383541ef3c8d0d674a56b7fe9331.

#28 - 03/28/2019 10:37 PM - segfault
- Assignee deleted (segfault)
- QA Check set to Ready for QA

a /lib/systemd/system-sleep/udev-watchdog script (systemd-sleep(8)):

that's a great idea, thanks for the hint to systemd-sleep that makes things so much simpler :) I put a trivial script in /lib/systemd/system-sleep which simply stops the tails-shutdown-on-media-removal.service before suspend and starts it again. I tested it and it seems to work.

so I'm tempted to rewrite the udev-watchdog-wrapper in Python (this would also allow us to use udisks via the D-Bus API instead of parsing
udevadm output).

Why not, but keep in mind that this is a persistent daemon and Python will eat more RAM. (I guess this would be a perfect use case for Go or Rust, in passing.)

Indeed, Go or Rust would be better than Python for a daemon process. I could rewrite it in Go at some point, but I think the system-sleep script already solves the problem at hand.

#29 - 03/28/2019 10:39 PM - segfault
- Feature Branch set to feature/14556-show-suspend-button

#30 - 04/04/2019 07:27 AM - intrigeri
- Assignee set to segfault
- Target version set to Tails_3.14
- QA Check changed from Ready for QA to Dev Needed

Regarding /lib/systemd/system-sleep/stop-tails-shutdown-on-media-removal.sh:

- It's worth mentioning it in wiki/src/contribute/design/memory_erasure.mdwn
- I suggest s/stop/toggle/ in the script's name, to better express what it does.

Other than this, @segfault: code review passes and it works fine on a X200.

#31 - 04/04/2019 11:13 AM - segfault
- Assignee changed from segfault to intrigeri
- QA Check changed from Dev Needed to Ready for QA

intrigeri wrote:

Regarding /lib/systemd/system-sleep/stop-tails-shutdown-on-media-removal.sh:

- It's worth mentioning it in wiki/src/contribute/design/memory_erasure.mdwn
- I suggest s/stop/toggle/ in the script's name, to better express what it does.

Done
#32 - 04/04/2019 12:33 PM - intrigeri
Added a link in the design doc to the implementation, code review passes, will now test.

#33 - 04/04/2019 01:07 PM - intrigeri
- Status changed from In Progress to 11
- % Done changed from 0 to 100

Applied in changeset tails/84bf911c03b195ff615d2c47b45e9a2f3779d62.

#34 - 04/04/2019 01:07 PM - intrigeri
- Status changed from 11 to In Progress

Applied in changeset tails/6ea1326297ae072e5464fb7bb1d5b76f6a031df.

#35 - 04/04/2019 01:08 PM - intrigeri
- Status changed from In Progress to 11
- Assignee deleted (intrigeri)
- QA Check changed from Ready for QA to Pass

#36 - 05/05/2019 08:23 AM - intrigeri
- Target version changed from Tails_3.14 to Tails_3.13.2

#37 - 05/06/2019 03:00 PM - anonym
- Status changed from 11 to Resolved

#38 - 05/06/2019 03:03 PM - anonym
- Target version changed from Tails_3.13.2 to Tails_3.14

#39 - 05/06/2019 06:15 PM - intrigeri
- Target version changed from Tails_3.14 to Tails_3.13.2

#40 - 06/07/2019 10:44 AM - intrigeri
- Related to Bug #16787: Trigger emergency shutdown on resume when the boot device is missing added