

## LTTng-tools - Bug #861

### UST subbuffers silently dropped on moderate trace traffic

11/18/2014 11:31 AM - Mathieu Desnoyers

<b>Status:</b>	Resolved	<b>Start date:</b>	11/18/2014
<b>Priority:</b>	Critical	<b>Due date:</b>	
<b>Assignee:</b>	Mathieu Desnoyers	<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	2.4		
<b>Description</b>			
<p>ltnng-tools commit 02b3d1769d5f8a33e4109b1e681141c9295dfda6 introduced an important regression for ltnng-ust tracing in the consumer daemon: after reading a sub-buffer, a check has been added to see whether there are more sub-buffers available to read, and if it is the case, it ensures the wakeup pipe will be awakened again.</p> <p>The issue lies in the use of <code>ustctl_put_next_subbuf()</code> in this check. This acts as if the sub-buffer has been read, when in reality it has not been read. It therefore trashes the data contained by this sub-buffer.</p> <p>This check should use <code>ustctl_put_subbuf()</code>, which does not move the consumer position.</p>			

#### Associated revisions

##### Revision 5420e5db - 11/18/2014 04:57 PM - Mathieu Desnoyers

Fix: UST subbuffers silently dropped on moderate trace traffic

Well, it looks like we really screwed up on this one.

ltnng-tools commit 02b3d1769d5f8a33e4109b1e681141c9295dfda6 introduced an important regression for ltnng-ust tracing in the consumer daemon: after reading a sub-buffer, a check has been added to see whether there are more sub-buffers available to read, and if it is the case, it ensures the wakeup pipe will be awakened again.

The issue lies in the use of `ustctl_put_next_subbuf()` in this check. This acts as if the sub-buffer has been read, when in reality it has not been read. It therefore trashes the data contained by this sub-buffer.

This check should use `ustctl_put_subbuf()`, which does not move the consumer position.

This is a severe bug, and the fix needs to be applied to stable-2.6, stable-2.5, and stable-2.4.

Fixes #861

Signed-off-by: Mathieu Desnoyers <[mathieu.desnoyers@efficios.com](mailto:mathieu.desnoyers@efficios.com)>

Signed-off-by: Jérémie Galarneau <[jeremie.galarneau@efficios.com](mailto:jeremie.galarneau@efficios.com)>

##### Revision 4dbc8d72 - 11/18/2014 04:58 PM - Mathieu Desnoyers

Fix: UST subbuffers silently dropped on moderate trace traffic

Well, it looks like we really screwed up on this one.

ltnng-tools commit 02b3d1769d5f8a33e4109b1e681141c9295dfda6 introduced an important regression for ltnng-ust tracing in the consumer daemon: after reading a sub-buffer, a check has been added to see whether there are more sub-buffers available to read, and if it is the case, it ensures the wakeup pipe will be awakened again.

The issue lies in the use of `ustctl_put_next_subbuf()` in this check. This acts as if the sub-buffer has been read, when in reality it has not been read. It therefore trashes the data contained by this sub-buffer.

This check should use `ustctl_put_subbuf()`, which does not move the consumer position.

This is a severe bug, and the fix needs to be applied to stable-2.6,

stable-2.5, and stable-2.4.

Fixes #861

Signed-off-by: Mathieu Desnoyers <[mathieu.desnoyers@efficios.com](mailto:mathieu.desnoyers@efficios.com)>

Signed-off-by: Jérémie Galarneau <[jeremie.galarneau@efficios.com](mailto:jeremie.galarneau@efficios.com)>

**Revision 00e77320 - 11/18/2014 04:58 PM - Mathieu Desnoyers**

Fix: UST subbuffers silently dropped on moderate trace traffic

Well, it looks like we really screwed up on this one.

lttng-tools commit 02b3d1769d5f8a33e4109b1e681141c9295dfda6 introduced an important regression for lttng-ust tracing in the consumer daemon: after reading a sub-buffer, a check has been added to see whether there are more sub-buffers available to read, and if it is the case, it ensures the wakeup pipe will be awakened again.

The issue lies in the use of `ustctl_put_next_subbuf()` in this check. This acts as if the sub-buffer has been read, when in reality it has not been read. It therefore trashes the data contained by this sub-buffer.

This check should use `ustctl_put_subbuf()`, which does not move the consumer position.

This is a severe bug, and the fix needs to be applied to stable-2.6, stable-2.5, and stable-2.4.

Fixes #861

Signed-off-by: Mathieu Desnoyers <[mathieu.desnoyers@efficios.com](mailto:mathieu.desnoyers@efficios.com)>

Signed-off-by: Jérémie Galarneau <[jeremie.galarneau@efficios.com](mailto:jeremie.galarneau@efficios.com)>

**Revision 520d93fd - 11/18/2014 04:59 PM - Mathieu Desnoyers**

Fix: UST subbuffers silently dropped on moderate trace traffic

Well, it looks like we really screwed up on this one.

lttng-tools commit 02b3d1769d5f8a33e4109b1e681141c9295dfda6 introduced an important regression for lttng-ust tracing in the consumer daemon: after reading a sub-buffer, a check has been added to see whether there are more sub-buffers available to read, and if it is the case, it ensures the wakeup pipe will be awakened again.

The issue lies in the use of `ustctl_put_next_subbuf()` in this check. This acts as if the sub-buffer has been read, when in reality it has not been read. It therefore trashes the data contained by this sub-buffer.

This check should use `ustctl_put_subbuf()`, which does not move the consumer position.

This is a severe bug, and the fix needs to be applied to stable-2.6, stable-2.5, and stable-2.4.

Fixes #861

Signed-off-by: Mathieu Desnoyers <[mathieu.desnoyers@efficios.com](mailto:mathieu.desnoyers@efficios.com)>

Signed-off-by: Jérémie Galarneau <[jeremie.galarneau@efficios.com](mailto:jeremie.galarneau@efficios.com)>

**History**

---

**#1 - 11/18/2014 12:15 PM - Mathieu Desnoyers**

- Status changed from *Confirmed* to *Resolved*

- % Done changed from 0 to 100

Applied in changeset tools|commit:5420e5dbd5fd06f2d1ba524d44eed601daef792c.