valgrind: UninitCondition in ceph::crypto::onwire::AES128GCM_OnWireRxHandler::authenticated_decrypt_update_final()

Description

<error>
<unique>0x0</unique>
</error>
<tid>12</tid>
</error>
<threadname>msgr-worker-2</threadname>
</error>
<kind>UninitCondition</kind>
<what>Conditional jump or move depends on uninitialised value(s)</what>
</error>
<stack>
<frame>
<ip>0x10430C7C</ip>
</frame>
<frame>
<ip>0x1042CBD6</ip>
</frame>
<frame>
<ip>0x5368F14</ip>
</frame>
<frame>
<ip>0x5358151</ip>
</frame>
</stack>
/a/kchai-2019-03-20_05:45:48-rados-wip-kefu-testing-2019-03-20-1120-distro-basic-smithi/3751697/remote/smithi039/log/valgrind/m
on.a.log.gz

<table>
<thead>
<tr>
<th>Related issues:</th>
<th>Can't reproduce</th>
<th>Resolved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Related to RADOS - Bug #44362: osd: uninitialized memory in sendmsg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Copied to RADOS - Backport #41534: nautilus: valgrind: UninitCondition in cep...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**History**

**#1 - 03/20/2019 11:24 AM - Kefu Chai**

The test branch contains [https://github.com/ceph/ceph/pull/27012](https://github.com/ceph/ceph/pull/27012)

**#2 - 03/20/2019 11:40 AM - Radoslaw Zarzynski**

- Status changed from New to In Progress
- Assignee set to Radoslaw Zarzynski

**#3 - 05/09/2019 04:22 PM - J. Eric Ivancich**

- Priority changed from Normal to High

Is this being actively worked on?

How close are we to a fix on this?

I would like to make this a high priority bug, perhaps even urgent. This results in a lot of valgrind test failures on teuthology, thereby creating a lot of noise and adding to the workload of developers doing QA.

As an example, most of the failures in this teuthology run appear to be as a result of this bug:


**#4 - 05/16/2019 04:21 PM - Ali Maredia**

The RGW verify suite has commented out the lines running valgrind on the mon.

[https://github.com/ceph/ceph/pull/28155](https://github.com/ceph/ceph/pull/28155)

Before this bug is resolved, that PR needs to be undone to ensure valgrind is being run on the mon in the rgw verify suite.

**#5 - 05/28/2019 05:13 PM - Radoslaw Zarzynski**

This bug looks like being duplicated by of [http://tracker.ceph.com/issues/39449](http://tracker.ceph.com/issues/39449) which has been addressed with a pair of interconnected PRs:

- [https://github.com/ceph/ceph/pull/27265](https://github.com/ceph/ceph/pull/27265) for ceph merged on Apr 2.
- [https://github.com/ceph/teuthology/pull/1274](https://github.com/ceph/teuthology/pull/1274) for teuthology on Apr 30.


```xml
<!-- the msgr stuff -->
<frame>
  <ip>0x5358BA4</ip>
  <obj>/usr/lib64/ceph/libceph-common.so.0</obj>
  <fn>EventCenter::process_events(unsigned int, std::chrono::duration&lt;unsigned long, std::ratio&lt;1l, 1000000000l&gt; &gt;*)</fn>
</frame>
<frame>
  <ip>0x535E8D6</ip>
</frame>
```

06/05/2021
while in the whitelist we expect:

```c
+++ the msgr stuff
fun:_ZN11EventCenter14process_eventsEjPNSt6chrono8durationImSt5ratioILl1ELl10000000000000000EE
fun:operator()
fun:_ZNSt17_Function_handlerIFvvEZN12NetworkStack10add_threadEjEUlvE_E9_M_invokeERKSt9_Any_data
fun:execute_native_thread_routine
fun:start_thread
fun:clone
```

It's a bit surprising as the unresolved symbols come from libceph-common. Anyway, I'm tuning the whitelist now.
Scheduled a resurrected run for validation: 


Status changed from In Progress to Fix Under Review

https://github.com/ceph/ceph/pull/28305

Status changed from Fix Under Review to Resolved

Status changed from Resolved to Pending Backport
Backport set to nautilus

seeing this in the rgw suite for nautilus runs, so tagging for backport of https://github.com/ceph/ceph/pull/28305

Copied to Backport #41534: nautilus: valgrind: UninitCondition in ceph::crypto::onwire::AES128GCM_OnWireRxHandler::authenticated_decrypt_update_final() added

Pull request ID set to 28305

Status changed from Pending Backport to Resolved

While running with --resolve-parent, the script "backport-create-issue" noticed that all backports of this issue are in status "Resolved".

Related to Bug #44362: osd: uninitialized memory in sendmsg added