

mgr - Bug #24282

mgr segfaults on create-self-signed-cert

05/24/2018 03:17 PM - Abhishek Lekshmanan

Status:	Resolved	% Done:	0%
Priority:	High	Spent time:	0.00 hour
Assignee:	Abhishek Lekshmanan		
Category:	Build		
Target version:			
Source:		Affected Versions:	
Tags:		ceph-qa-suite:	
Backport:		Pull request ID:	
Regression:	No	Crash signature (v1):	
Severity:	3 - minor	Crash signature (v2):	
Reviewed:			

Description

vstart clusters seem to segfault when dashboard or restful tries to create a self signed cert:

```
#0 0x00007f071876dadb in raise () from /lib64/libpthread.so.0
#1 0x0000560c18eee291 in reraise_fatal (signum=11)
    at /ssd/builds/cpp/ceph_mimic/src/global/signal_handler.cc:74
#2 handle_fatal_signal (signum=11) at /ssd/builds/cpp/ceph_mimic/src/global/signal_handler.cc:138
#3 <signal handler called>
#4 0x00007f0708dae023 in ?? ()
    from /usr/lib64/python2.7/site-packages/cryptography/hazmat/bindings/_openssl.so
#5 0x00007f071aa4a64e in PyEval_EvalFrameEx () from /usr/lib64/libpython2.7.so.1.0
#6 0x00007f071aa4a434 in PyEval_EvalFrameEx () from /usr/lib64/libpython2.7.so.1.0
#7 0x00007f071aa4ffa4 in PyEval_EvalCodeEx () from /usr/lib64/libpython2.7.so.1.0
#8 0x00007f071aa4ac1c in PyEval_EvalFrameEx () from /usr/lib64/libpython2.7.so.1.0
#9 0x00007f071aa4ffa4 in PyEval_EvalCodeEx () from /usr/lib64/libpython2.7.so.1.0
#10 0x00007f071a9f376f in ?? () from /usr/lib64/libpython2.7.so.1.0
#11 0x00007f071a9dcbd6 in PyObject_Call () from /usr/lib64/libpython2.7.so.1.0
#12 0x00007f071a9e382e in ?? () from /usr/lib64/libpython2.7.so.1.0
#13 0x00007f071a9dcbd6 in PyObject_Call () from /usr/lib64/libpython2.7.so.1.0
#14 0x00007f071a9dceab in PyObject_CallMethod () from /usr/lib64/libpython2.7.so.1.0
#15 0x0000560c18dfa761 in ActivePyModule::handle_command (this=0x560c1b03abe0,
    cmdmap=std::map with 1 element = {...}, ds=ds@entry=0x7f0703a391b0, ss=ss@entry=0x7f0703a39040
)
    at /ssd/builds/cpp/ceph_mimic/src/mgr/ActivePyModule.cc:126
#16 0x0000560c18dc81e1 in ActivePyModules::handle_command (this=<optimized out>, module_name="dash
board",
    cmdmap=<error reading variable: Cannot access memory at address 0x2d3797ec9974cc28>,
    ds=ds@entry=0x7f0703a391b0, ss=ss@entry=0x7f0703a39040)
    at /ssd/builds/cpp/ceph_mimic/src/mgr/ActivePyModules.cc:769
#17 0x0000560c18de71c1 in PyModuleRegistry::handle_command (this=<optimized out>,
    module_name="dashboard", cmdmap=..., ds=ds@entry=0x7f0703a391b0, ss=ss@entry=0x7f0703a39040)
    at /ssd/builds/cpp/ceph_mimic/src/mgr/PyModuleRegistry.cc:294
#18 0x0000560c18d9cf44 in DaemonServer::<lambda(int)>::operator()(int) const (__closure=0x560c1c3b
8570,
    r_=<optimized out>) at /ssd/builds/cpp/ceph_mimic/src/mgr/DaemonServer.cc:1730
#19 0x0000560c18db33bc in boost::function1<void, int>::operator()(a0=<optimized out>,
    this=<optimized out>)
    at /ssd/builds/cpp/ceph_mimic/build/boost/include/boost/function/function_template.hpp:759
#20 FunctionContext::finish (this=<optimized out>, r=<optimized out>)
    at /ssd/builds/cpp/ceph_mimic/src/include/Context.h:522
#21 0x0000560c18dae949 in Context::complete (this=0x560c1c3b8780, r=<optimized out>)
    at /ssd/builds/cpp/ceph_mimic/src/include/Context.h:77
#22 0x00007f071aff3e7e in Finisher::finisher_thread_entry (this=0x560c1c2c4148)
```

```
at /ssd/builds/cpp/ceph_mimic/src/common/Finisher.cc:68
#23 0x00007f0718765724 in start_thread () from /lib64/libpthread.so.0
#24 0x00007f07177dce8d in clone () from /lib64/libc.so.6
```

History

#1 - 06/06/2018 10:31 AM - Tim Serong

I've tried to reproduce this on the current master, plus also the current mimic branch, and everything seems to be working more or less OK, i.e. no segfaults (although the mimic build is complaining "Module 'dashboard' has failed: cannot import name UiApiController" in the mgr log).

This is on openSUSE Tumbleweed, with openssl 1.1.0h. I built with `./do_cmake.sh -DWITH_EMBEDDED=OFF -DWITH_MANPAGE=OFF -DWITH_PYTHON3=ON -DWITH_SYSTEMD=ON -DWITH_LTTNG=OFF -DHAVE_BABELTRACE=OFF` (although suspect that it's still using python 2.7), then ran `vstart.sh --debug --new -x --localhost --bluestore`

#2 - 06/06/2018 10:58 AM - Tim Serong

Ignore the "Module 'dashboard' has failed: cannot import name UiApiController" error - I had some stale .pyc files lying around after switching from master to mimic branch.

#3 - 06/08/2018 07:25 AM - Abhishek Lekshmanan

I did a git bisect and traced this back to 4860bb70e1f47377ff69e1dc44e9b11bc69a7c2a which replaces nss with openssl, maybe we have the issue of multiple initializations of openssl again, I'll try to dig a bit more into this

```
# bad: [5325701f88a398fe6cacf1a1c31a9d4a61a18674] Merge pull request #22194 from yehudasa/wip-fix-build
git bisect bad 5325701f88a398fe6cacf1a1c31a9d4a61a18674
# good: [1d684921e414981e04b83f8e040eb6578694c34f] Merge PR #21973 into wip-sage-testing-20180521.120735
git bisect good 1d684921e414981e04b83f8e040eb6578694c34f
# bad: [e823c15e0421579b4a16332078cab6f152743f9] Merge pull request #22083 from liewegas/wip-21480
git bisect bad e823c15e0421579b4a16332078cab6f152743f9
# bad: [e823c15e0421579b4a16332078cab6f152743f9] Merge pull request #22083 from liewegas/wip-21480
git bisect bad e823c15e0421579b4a16332078cab6f152743f9
# bad: [1642bc44919274ab7e21af3a6534b80e11d2b4ad] Merge pull request #22074 from dzafman/wip-parens
git bisect bad 1642bc44919274ab7e21af3a6534b80e11d2b4ad
# bad: [9860f53416875b15c641d1797a9c6e92a839cd3d] Merge PR #22091 into master
git bisect bad 9860f53416875b15c641d1797a9c6e92a839cd3d
# bad: [2535d11713aa837015e5028923ac97a271f41081] tests/crypto: add tests for the no-bl encrypt/decrypt, part
2.
git bisect bad 2535d11713aa837015e5028923ac97a271f41081
# bad: [4860bb70e1f47377ff69e1dc44e9b11bc69a7c2a] auth: CryptoAESKeyHandler switches from NSS to OpenSSL.
git bisect bad 4860bb70e1f47377ff69e1dc44e9b11bc69a7c2a
# good: [7635485d34c99671030a10e1da67945c19b4fc88] auth: the outbuf of AES should be multiple of block size
git bisect good 7635485d34c99671030a10e1da67945c19b4fc88
# first bad commit: [4860bb70e1f47377ff69e1dc44e9b11bc69a7c2a] auth: CryptoAESKeyHandler switches from NSS to
OpenSSL.
```

#4 - 06/11/2018 09:54 AM - Abhishek Lekshmanan

So this can be triggered by differing ssl libraries loaded by system python libraries vs what we link ceph against. From the commit listed above, pr <https://github.com/ceph/ceph/pull/21540> introduced linking ceph libraries against openssl instead of nss used previously. This will probably choose latest openssl if multiple ssl versions are offered.

However it can happen that the system python libraries are linked to the openssl library used for python which might be different,

```
ldd ./bin/ceph-mgr
linux-vdso.so.1 (0x00007ffe991b9000)
libceph-common.so.0 => /ssd/builds/cpp/ceph_mimic/build/lib/libceph-common.so.0 (0x00007f747c0d2000)
libpython3.4m.so.1.0 => /usr/lib64/libpython3.4m.so.1.0 (0x00007f747bc71000)
libblkid.so.1 => /usr/lib64/libblkid.so.1 (0x00007f747ba2c000)
libdl.so.2 => /lib64/libdl.so.2 (0x00007f747b828000)
libtcmalloc.so.4 => /usr/lib64/libtcmalloc.so.4 (0x00007f747b5b5000)
libresolv.so.2 => /lib64/libresolv.so.2 (0x00007f747b39e000)
libssl3.so => /usr/lib64/libssl3.so (0x00007f747b14c000)
libsmime3.so => /usr/lib64/libsmime3.so (0x00007f747af24000)
libnss3.so => /usr/lib64/libnss3.so (0x00007f747abfe000)
libnssutil3.so => /usr/lib64/libnssutil3.so (0x00007f747a9ce000)
libplds4.so => /usr/lib64/libplds4.so (0x00007f747a7ca000)
libplc4.so => /usr/lib64/libplc4.so (0x00007f747a5c5000)
libnspr4.so => /usr/lib64/libnspr4.so (0x00007f747a387000)
libssl.so.1.1 => /usr/lib64/libssl.so.1.1 (0x00007f747a11d000)
libcrypto.so.1.1 => /usr/lib64/libcrypto.so.1.1 (0x00007f7479c93000)
libpthread.so.0 => /lib64/libpthread.so.0 (0x00007f7479a76000)
libibverbs.so.1 => /usr/lib64/libibverbs.so.1 (0x00007f7479861000)
libz.so.1 => /lib64/libz.so.1 (0x00007f747964b000)
libstdc++.so.6 => /usr/lib64/libstdc++.so.6 (0x00007f74792c2000)
libm.so.6 => /lib64/libm.so.6 (0x00007f7478fc5000)
libgcc_s.so.1 => /lib64/libgcc_s.so.1 (0x00007f7478dae000)
libc.so.6 => /lib64/libc.so.6 (0x00007f7478a09000)
/lib64/ld-linux-x86-64.so.2 (0x00007f7485230000)
libutil.so.1 => /lib64/libutil.so.1 (0x00007f7478806000)
libuuid.so.1 => /usr/lib64/libuuid.so.1 (0x00007f7478601000)
libunwind.so.8 => /lib64/libunwind.so.8 (0x00007f74783e8000)
librt.so.1 => /lib64/librt.so.1 (0x00007f74781e0000)
libnl-route-3.so.200 => /usr/lib64/libnl-route-3.so.200 (0x00007f7477f8b000)
libnl-3.so.200 => /usr/lib64/libnl-3.so.200 (0x00007f7477d6d000)
```

ldd on python's hashlib which is dynamically loaded by mgr (from python-cryptography??)

```
ldd /usr/lib64/python3.4/lib-dynload/_hashlib.cpython-34m.so
linux-vdso.so.1 (0x00007ffc130ad000)
libssl.so.1.0.0 => /lib64/libssl.so.1.0.0 (0x00007f75a1fed000)
libcrypto.so.1.0.0 => /lib64/libcrypto.so.1.0.0 (0x00007f75a1b94000)
libpython3.4m.so.1.0 => /usr/lib64/libpython3.4m.so.1.0 (0x00007f75a1733000)
libpthread.so.0 => /lib64/libpthread.so.0 (0x00007f75a1516000)
libc.so.6 => /lib64/libc.so.6 (0x00007f75a1171000)
libdl.so.2 => /lib64/libdl.so.2 (0x00007f75a0f6d000)
libz.so.1 => /lib64/libz.so.1 (0x00007f75a0d57000)
libutil.so.1 => /lib64/libutil.so.1 (0x00007f75a0b54000)
libm.so.6 => /lib64/libm.so.6 (0x00007f75a0857000)
/lib64/ld-linux-x86-64.so.2 (0x00007f75a245e000)
```

#5 - 06/11/2018 03:38 PM - Abhishek Lekshmanan

I worked around the issue by installing a new python with pyenv that links to openssl 1.1 and explicitly setting this python while building, not sure whether having multiple openssl libraries in a system is something we should consider and fix for

#6 - 09/27/2018 02:56 AM - Kefu Chai

- *Category set to Build*
- *Status changed from New to Resolved*
- *Assignee set to Abhishek Lekshmanan*

<https://github.com/ceph/ceph/pull/22659>