Alpine Linux - Bug #10156

Bareos-director segfaults on startup

03/26/2019 07:24 AM - Simon F

Status:	Closed	**Start date:**	03/26/2019
Priority:	High	**Due date:**	
Assignee:		**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	3.9.3		
Affected versions:	3.9.0, 3.9.1, 3.9.2, 3.9.3, 3.10.0	**Security IDs:**	

Description

Bareos in alpine >= 3.9 segfaults on startup.
The issue is the "bsmtp" program which fails and tears down the whole director.

Reproducible with:

$> docker run -it alpine:3.9 /bin/ash
/ # apk add bareos
[...]
/ # bsmtp
Segmentation fault (core dumped)

Associated revisions

Revision 1b3fe8f6 - 04/08/2019 06:23 PM - Simon F

community/bareos: Fix segfaults on startup

This PR fixes two undefined behaviours with pthreads leading to
segfaults.

Fixes #10156
https://bugs.alpinelinux.org/issues/10156

(cherry picked from commit 37923bd0f6d4e33a3b7de6c23e708787c5d7d3bf)

History

#1 - 03/26/2019 07:24 AM - Simon F

- Subject changed from Bareos segfaults on startup to Bareos-director segfaults on startup

#2 - 03/26/2019 07:32 AM - Simon F

More information about the director crash:
It seems the director only crashes when bconsole connects to it or a job is about to run.

```
[07:27] root@obelix:~/dockerfiles/docker-bareos# bconsole
Connecting to Director localhost:9101
^C
[07:28] root@obelix:~/dockerfiles/docker-bareos# docker-compose logs bareos
Attaching to docker-bareos_bareos_1
bareos_1        | created directory: '/run/bareos'
bareos_1        | BAREOS interrupted by signal 11: Segmentation violation
bareos_1        | Kaboom! bareos-dir, obelix-dir got signal 11 - Segmentation violation. Attempting traceback.
bareos_1        | Kaboom! exepath=/usr/sbin/
bareos_1        | Calling: /usr/sbin/btraceback /usr/sbin/bareos-dir 11 /var/lib/bareos
bareos_1        | execv: /usr/sbin/btraceback failed: ERR=Permission denied
bareos_1        | The btraceback call returned 255
bareos_1        | Dumping: /var/lib/bareos/obelix-dir.11.bactrace
```

```
/ # cat /var/lib/bareos/obelix-dir.11.bactrace
Attempt to dump locks
threadid=0x0000007f3f39c11b max=0 current=-1
threadid=0x0000007f3f39c34b max=1 current=-1
```

```
threadid=0x0000007f3f39c7ab max=0 current=-1
threadid=0x0000007f3f39c57b max=0 current=-1
threadid=0x0000007f3f3a066b max=0 current=-1
threadid=0x0000007f3f3b5028 max=2 current=-1
Attempt to dump current JCRs. njcrs=2
threadid=0x0000007f3f3b5028 JobId=0 JobStatus=R jcr=0x556256f73308 name=*JobMonitor*.2019-03-26_07.27.56_01
threadid=0x1000007f3f3b5028 killable=0 JobId=0 JobStatus=R jcr=0x556256f73308 name=*JobMonitor*.2019-03-26_07.
27.56_01
        use_count=1
        JobType=I JobLevel=
        sched_time=26-Mar-2019 07:27 start_time=26-Mar-2019 07:27
        end_time=01-Jan-1970 00:00 wait_time=01-Jan-1970 00:00
        db=0 db_batch=0 batch_started=0
threadid=0x0000007f3f39c34b JobId=0 JobStatus=R jcr=0x556256f75008 name=*StatisticsCollector*.2019-03-26_07.27
.56_02
threadid=0x0000007f3f39c34b killable=0 JobId=0 JobStatus=R jcr=0x556256f75008 name=*StatisticsCollector*.2019-
03-26_07.27.56_02
        use_count=1
        JobType=I JobLevel=
        sched_time=26-Mar-2019 07:27 start_time=26-Mar-2019 07:27
        end_time=01-Jan-1970 00:00 wait_time=01-Jan-1970 00:00
        db=0x556256f77ce8 db_batch=0 batch_started=0
```

```
/var/lib/bareos # gdb -c core
GNU gdb (GDB) 8.2.1
[...]
[New LWP 49]
Core was generated by `/usr/sbin/bsmtp -h localhost -f root@localhost -s Bareos NONE traceback of bare'.
Program terminated with signal SIGSEGV, Segmentation fault.
#0  0x00007f67cbfeb2c3 in ?? ()
(gdb) bt
#0  0x00007f67cbfeb2c3 in ?? ()
#1  0x00007f67cbf67928 in ?? ()
#2  0x00000000000000aa in ?? ()
#3  0x00007f67cbf6cb76 in ?? ()
#4  0x000055770f1466ae in ?? ()
#5  0x0000000000000000 in ?? ()
(gdb)
```

#3 - 03/27/2019 07:38 AM - Simon F

Ok, after some struggling around with the stripped debug infos i got the following bt from gdb:

```
0x00007ffff7fbe2c3 in tss_get () from /lib/ld-musl-x86_64.so.1
(gdb) bt
#0  0x00007ffff7fbe2c3 in tss_get () from /lib/ld-musl-x86_64.so.1
#1  0x00007ffff7f2d94f in get_jobid_from_tsd () at jcr.c:739
#2  0x00007ffff7f34279 in p_msg (file=0x555555558006 "bsmtp.c", line=364,
    level=0, fmt=0x555555558568 "Fatal error: no recipient given.\n")
    at message.c:1343
#3  0x00005555555564d5 in main (argc=0, argv=<optimized out>) at bsmtp.c:364
```

#4 - 04/04/2019 06:49 AM - Michael Cassaniti

Looking through the code further it seems that p_msg is expecting to be called from a thread. The function get_jobid_from_tsd is called when logging the message to provide some form of job ID, but there is no thread running. There should be a thread key set called jcr_key, but its initial value is undefined. This is passed into tss_get which is a C11 wrapper for pthread_getspecific. Below is a snippet from the man page at https://linux.die.net/man/3/pthread_setspecific.

```
The effect of calling pthread_getspecific() or pthread_setspecific() with a key value not obtained from pthrea
d_key_create() or after key has been deleted with pthread_key_delete() is undefined.
```

This indicates an upstream bug. I wish I didn't suddenly learn all this stuff, but oh well I've done it now. It also seems that this bug is still there in version 18.2 upstream. Can an upstream bug be raised?

I also found this bug which lists the same issue: https://github.com/openwrt/telephony/issues/99

#5 - 04/05/2019 04:12 AM - Michael Cassaniti

The below patch fixes the issue with bsmtp, but it doesn't fix bareos-dir from crashing. The code will make sure that the job control record key is initialized before using it.

```
diff --git a/src/lib/jcr.c b/src/lib/jcr.c
index 00bfe8c87..338f90e59 100644
--- a/src/lib/jcr.c
+++ b/src/lib/jcr.c
@@ -77,6 +77,7 @@ static pthread_mutex_t jcr_lock = PTHREAD_MUTEX_INITIALIZER;
 static pthread_mutex_t job_start_mutex = PTHREAD_MUTEX_INITIALIZER;
 static pthread_mutex_t last_jobs_mutex = PTHREAD_MUTEX_INITIALIZER;

+static bool jcr_initialized = false;
 #ifdef HAVE_WIN32
 static bool tsd_initialized = false;
 static pthread_key_t jcr_key;          /* Pointer to jcr for each thread */
@@ -351,6 +352,8 @@ static void create_jcr_key()
        berrno be;
        Jmsg1(NULL, M_ABORT, 0, _("pthread key create failed: ERR=%s\n"),
              be.bstrerror(status));
+   } else {
+     jcr_initialized = true;
    }
 }

@@ -719,7 +722,10 @@ void set_jcr_in_tsd(JCR *jcr)
  */
 JCR *get_jcr_from_tsd()
 {
-   JCR *jcr = (JCR *)pthread_getspecific(jcr_key);
+   JCR *jcr = (JCR *)INVALID_JCR;
+   if (jcr_initialized){
+     jcr = (JCR *)pthread_getspecific(jcr_key);
+   }

    /*
     * Set any INVALID_JCR to NULL which the rest of BAREOS understands
@@ -736,7 +742,7 @@ JCR *get_jcr_from_tsd()
  */
 uint32_t get_jobid_from_tsd()
 {
-   JCR *jcr = (JCR *)pthread_getspecific(jcr_key);
+   JCR *jcr = get_jcr_from_tsd();
    uint32_t JobId = 0;

    if (jcr && jcr != INVALID_JCR) {
```

#6 - 04/05/2019 05:45 AM - Simon F

Ok, I found the other issue with bareos-dir.
See https://github.com/alpinelinux/aports/pull/6951 for details

#7 - 04/05/2019 10:30 AM - Simon F

Opened an upstream issue: https://bugs.bareos.org/view.php?id=1073

#8 - 04/08/2019 06:24 PM - Natanael Copa

Thanks for tracking this down and reporting it upstream! Good job!

#9 - 04/08/2019 06:26 PM - Simon F

- Status changed from New to Resolved

- % Done changed from 0 to 100

Applied in changeset 1b3fe8f6f7bcaaa6789d90017f419f0887029260.

#10 - 04/08/2019 08:20 PM - Natanael Copa

- Status changed from Resolved to Closed