Ruby master - Bug #9659

crash in FIPS mode after unchecked algo->init_func failure

03/20/2014 07:50 PM - jared.jennings.ctr (Jared Jennings)

Status:	Feedback		
Priority:	Normal		
Assignee:			
Target version:			
ruby -v:	ruby 1.8.7 (2011-06-30 patchlevel 352) [x86_64-linux]	**Backport:**	2.0.0: DONTNEED, 2.1: DONTNEED

Description

This is just like #4944, but in the digest extension instead of the openssl extension.

On my host, which is configured for FIPS 140-2 compliance (this is a U.S. Government security standard), OpenSSL refuses to perform an MD5 checksum. It indicates this refusal when the digest algorithm initialization function is called: this function returns a 0 indicating failure instead of a 1 indicating success. But it's just a bunch of arithmetic; how can it fail? So the return code is ignored. But if the initialization fails, and we go on trying to use the algorithm, the Ruby interpreter crashes:

```
$ OPENSSL_FORCE_FIPS_MODE= ruby -rdigest -e "puts Digest::MD5.hexdigest('hi')"
md5_dgst.c(78): OpenSSL internal error, assertion failed: Digest MD5 forbidden in FIPS mode!
Aborted (core dumped)
```

The digest extension, in the rb_digest_base_alloc, rb_digest_base_reset, and rb_digest_base_finish functions, is ignoring the return code of algo->init_func. If OpenSSL is present at build time, algo->init_func works out to be the MD5_Init function from OpenSSL. This function, according to its man page, returns a 1 for success or 0 for failure.

I see the problem under Ruby 1.8.7 as patched by Red Hat; I can't easily build the trunk on my system, but it looks like in r43668 the return value still isn't being checked in these three places:

- source:ext/digest/digest.c@43668#L551
- source:ext/digest/digest.c@43668#L589
- source:ext/digest/digest.c@43668#L627

Associated revisions

Revision 6046b9f1 - 07/15/2014 02:58 PM - nobu (Nobuyoshi Nakada)

digest.c: raise exception on init failure

- ext/digest/digest.c: expect digest init and finish functions to indicate success or failure; raise exception on failure. [ruby-core:61614] [Bug #9659]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@46826 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 46826 - 07/15/2014 02:58 PM - nobu (Nobuyoshi Nakada)

digest.c: raise exception on init failure

- ext/digest/digest.c: expect digest init and finish functions to indicate success or failure; raise exception on failure. [ruby-core:61614] [Bug #9659]

Revision 46826 - 07/15/2014 02:58 PM - nobu (Nobuyoshi Nakada)

digest.c: raise exception on init failure

- ext/digest/digest.c: expect digest init and finish functions to indicate success or failure; raise exception on failure. [ruby-core:61614] [Bug #9659]

Revision 46826 - 07/15/2014 02:58 PM - nobu (Nobuyoshi Nakada)

digest.c: raise exception on init failure

- ext/digest/digest.c: expect digest init and finish functions to indicate success or failure; raise exception on failure. [ruby-core:61614] [Bug #9659]

Revision 46826 - 07/15/2014 02:58 PM - nobu (Nobuyoshi Nakada)

digest.c: raise exception on init failure

- ext/digest/digest.c: expect digest init and finish functions to indicate success or failure; raise exception on failure. [ruby-core:61614] [Bug #9659]

Revision 46826 - 07/15/2014 02:58 PM - nobu (Nobuyoshi Nakada)

digest.c: raise exception on init failure

- ext/digest/digest.c: expect digest init and finish functions to indicate success or failure; raise exception on failure. [ruby-core:61614] [Bug #9659]

Revision 46826 - 07/15/2014 02:58 PM - nobu (Nobuyoshi Nakada)

digest.c: raise exception on init failure

- ext/digest/digest.c: expect digest init and finish functions to indicate success or failure; raise exception on failure. [ruby-core:61614] [Bug #9659]

Revision aadebb29 - 07/15/2014 02:59 PM - nobu (Nobuyoshi Nakada)

ext/digest: return values of init and final

- ext/digest: make built-in digest function implementations indicate success or failure of init and final functions. [ruby-core:61614] [Bug #9659]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@46827 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 46827 - 07/15/2014 02:59 PM - nobu (Nobuyoshi Nakada)

ext/digest: return values of init and final

- ext/digest: make built-in digest function implementations indicate success or failure of init and final functions. [ruby-core:61614] [Bug #9659]

Revision 46827 - 07/15/2014 02:59 PM - nobu (Nobuyoshi Nakada)

ext/digest: return values of init and final

- ext/digest: make built-in digest function implementations indicate success or failure of init and final functions. [ruby-core:61614] [Bug #9659]

Revision 46827 - 07/15/2014 02:59 PM - nobu (Nobuyoshi Nakada)

ext/digest: return values of init and final

- ext/digest: make built-in digest function implementations indicate success or failure of init and final functions. [ruby-core:61614] [Bug #9659]

Revision 46827 - 07/15/2014 02:59 PM - nobu (Nobuyoshi Nakada)

ext/digest: return values of init and final

- ext/digest: make built-in digest function implementations indicate success or failure of init and final functions. [ruby-core:61614] [Bug #9659]

Revision 46827 - 07/15/2014 02:59 PM - nobu (Nobuyoshi Nakada)

ext/digest: return values of init and final

- ext/digest: make built-in digest function implementations indicate success or failure of init and final functions. [ruby-core:61614] [Bug #9659]

Revision 46827 - 07/15/2014 02:59 PM - nobu (Nobuyoshi Nakada)

ext/digest: return values of init and final

- ext/digest: make built-in digest function implementations indicate success or failure of init and final functions. [ruby-core:61614] [Bug #9659]

Revision 2d33fc97 - 07/15/2014 02:59 PM - nobu (Nobuyoshi Nakada)

md5ossl.c: indicate the result

- ext/digest/md5/md5ossl.c: use OpenSSL EVP API instead of MD5 API to perform MD5 hashes using OpenSSL in ext/digest. [ruby-core:61614] [Bug #9659]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@46828 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 46828 - 07/15/2014 02:59 PM - nobu (Nobuyoshi Nakada)

md5ossl.c: indicate the result

- ext/digest/md5/md5ossl.c: use OpenSSL EVP API instead of MD5 API to perform MD5 hashes using OpenSSL in ext/digest. [ruby-core:61614] [Bug #9659]

Revision 46828 - 07/15/2014 02:59 PM - nobu (Nobuyoshi Nakada)

md5ossl.c: indicate the result

- ext/digest/md5/md5ossl.c: use OpenSSL EVP API instead of MD5 API to perform MD5 hashes using OpenSSL in ext/digest. [ruby-core:61614] [Bug #9659]

Revision 46828 - 07/15/2014 02:59 PM - nobu (Nobuyoshi Nakada)

md5ossl.c: indicate the result

- ext/digest/md5/md5ossl.c: use OpenSSL EVP API instead of MD5 API to perform MD5 hashes using OpenSSL in ext/digest. [ruby-core:61614] [Bug #9659]

Revision 46828 - 07/15/2014 02:59 PM - nobu (Nobuyoshi Nakada)

md5ossl.c: indicate the result

- ext/digest/md5/md5ossl.c: use OpenSSL EVP API instead of MD5 API to perform MD5 hashes using OpenSSL in ext/digest. [ruby-core:61614] [Bug #9659]

Revision 46828 - 07/15/2014 02:59 PM - nobu (Nobuyoshi Nakada)

md5ossl.c: indicate the result

- ext/digest/md5/md5ossl.c: use OpenSSL EVP API instead of MD5 API to perform MD5 hashes using OpenSSL in ext/digest. [ruby-core:61614] [Bug #9659]

Revision 46828 - 07/15/2014 02:59 PM - nobu (Nobuyoshi Nakada)

md5ossl.c: indicate the result

- ext/digest/md5/md5ossl.c: use OpenSSL EVP API instead of MD5 API to perform MD5 hashes using OpenSSL in ext/digest. [ruby-core:61614] [Bug #9659]

History

#1 - 03/20/2014 09:43 PM - jared.jennings.ctr (Jared Jennings)

Now I see that rb_digest_hash_init_func_t (source:ext/digest/digest.h@43668#L20) is a typedef for a pointer to a function returning void. This complicates the patch: the typedef must be changed so init functions return an int, and the init functions in each digest algorithm implementation included in the digest extension must be changed slightly, to return a 1 for success or a 0 for failure, as the OpenSSL implementations they imitate claim to do.

#2 - 03/24/2014 11:00 PM - jared.jennings.ctr (Jared Jennings)

I changed the rb_digest_hash_init_func typedef from a return type of void to int, so that the return value of MD5_Init could be checked. I changed digest.c to check the return value of algo->init_func, which at the time of the crash seems to point at MD5_Init, and raise an exception if the function returns 0.

The interpreter still crashes. Running with gdb reveals that in my version of OpenSSL the MD5_Init function goes sort of like, { if (FIPS_mode() ...) { OpenSSLDie(..., "Digest MD5 forbidden in FIPS mode!"); } return private_MD5_Init(...); }. OpenSSLDie goes on to call abort. There's no returning 0 for failure in this case.

On a further look at md5(3), I see: "Applications should use the higher level functions EVP_DigestInit(3) etc. instead of calling the hash functions directly." Those functions should return a value to indicate failure rather than raising a signal: the openssl module was successfully modified to check their return value in #4944, to good effect.

#3 - 03/28/2014 12:28 AM - jared.jennings.ctr (Jared Jennings)

- File 002-builtin-indicate-digest-failure.patch added

- File 003-digest-openssl-md5-use-evp-api.patch added

- File 001-detect-digest-failure.patch added

Attached are three cumulative patches against source:/trunk@45452.

The first, 001-detect-digest-failure, changes the prototypes of digest initialization and finalization functions in the digest extension to return int instead of void; changes digest.c to check the return value of the initialization function and raise an exception in case of failure; and bumps the digest API version from 2 to 3.

The second, 002-builtin-indicate-digest-failure, changes the built-in digest implementations so that their initialization and finalization functions return an int, 1 for success or 0 for failure, as the OpenSSL functions return.

The third, 003-digest-openssl-md5-use-evp-api, changes the OpenSSL implementation of the md5 algorithm to use functions from openssl/evp.h rather than openssl/md5.h. The old, deprecated MD5_Init function calls abort(3) if used in FIPS-compliant mode, killing the interpreter; the EVP_DigestInit_ex function returns 0 to indicate initialization failure instead.

With these patches:

```
[vagrant@localhost ruby]$ OPENSSL_FORCE_FIPS_MODE= ruby -v -rdigest -e "puts Digest::MD5.hexdigest('hi')"
ruby 2.2.0dev (2014-03-27) [x86_64-linux]
-e:1:in `digest': Digest initialization failed. (RuntimeError)
    from -e:1:in `hexdigest'
    from -e:1:in `<main>'
```

I think further improvement is possible. Generally, it appears that the functions and types used in the builtin digest algorithm implementations are made to mirror the MD5_*, RIPEMD160_*, etc APIs from OpenSSL. Since I'm moving the ossl implementations to use the EVP_* API instead, I think the Right Thing to do here would be to change the builtins to mirror that newer API. If someone else agrees, I can produce the patches; until then, I have tried to make the smallest patches possible.

About 001, I don't know the consequences of bumping the digest API version, and I didn't provide any migration code that will make code written against the version-2 API work with the version-3 API. Also I don't know if the exception raised in the case of digest failure is the right class of exception.

003 only changes the ossl implementation of MD5, not any of the other algorithms. To keep the patch size down, I hardcoded the digest and block size constants. This isn't very DRY. The larger changes I alluded to above could fix it.

I don't know if tests need to be added for this code, but there are none in the patches.

#4 - 03/28/2014 12:29 AM - jared.jennings.ctr (Jared Jennings)

If any credit is given for finding this problem, it belongs to Joseph Yaworski; see https://tickets.puppetlabs.com/browse/PUP-1840.

#5 - 03/28/2014 05:11 AM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Feedback

I can't reproduce that assertion failure, with openssl 0.9.8y and 1.0.1f.
OPENSSL_FIPS needs to be defined, perhaps?

#6 - 03/28/2014 05:54 PM - jared.jennings.ctr (Jared Jennings)

I've just compared the Debian and CentOS OpenSSL sources, and it looks like large parts of the FIPS functionality in OpenSSL that I've taken for granted are provided in CentOS/RHEL-specific patches. So you may not be able to duplicate the failure with stock OpenSSL, or on Debian or Ubuntu machines.

On my RHEL 6 machine, I needed the dracut-fips package installed, which contains the FIPS crypto module (sometimes it's called a "canister"); see https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Security_Guide/sect-Security_Guide-Federal_Standards_And_Regulations-Federal_Information_Processing_Standard.html. This was because the OpenSSL init function checked whether the FIPS module was installed, and it's distributed in this package. But the code to check this was part of the CentOS/RHEL patches.

#7 - 07/15/2014 10:06 AM - vo.x (Vit Ondruch)

Hi, can we please push this forward? Since the fixes proposed so far seems to break API/ABI, it would be nice to have fixes in upstream Ruby sooner than later. This would help incorporate this patch into future versions of RHEL/CentOS/Fedora or any other FIPS compliant system.

#8 - 07/15/2014 01:57 PM - nobu (Nobuyoshi Nakada)

- Backport changed from 2.0.0: UNKNOWN, 2.1: UNKNOWN to 2.0.0: REQUIRED, 2.1: REQUIRED

https://github.com/nobu/ruby/compare/Bug%239659-digest-failure

#9 - 07/15/2014 02:28 PM - knu (Akinori MUSHA)

The above set of patches looks good to me.

#10 - 07/15/2014 02:59 PM - nobu (Nobuyoshi Nakada)

- Status changed from Feedback to Closed

- % Done changed from 0 to 100

Applied in changeset r46826.

---

digest.c: raise exception on init failure

- ext/digest/digest.c: expect digest init and finish functions to indicate success or failure; raise exception on failure. [ruby-core:61614] [Bug #9659]

#11 - 07/15/2014 08:27 PM - vo.x (Vit Ondruch)

Thanks Nobu. Nonetheless, I don't think it is backportable (which was not necessarily the point :).

#12 - 07/15/2014 10:16 PM - nobu (Nobuyoshi Nakada)

- Backport changed from 2.0.0: REQUIRED, 2.1: REQUIRED to 2.0.0: DONTNEED, 2.1: DONTNEED

Is EVP API necessary?
I've reverted it because of segfaults on many platforms.

#13 - 07/16/2014 01:55 AM - usa (Usaku NAKAMURA)

- Status changed from Closed to Feedback

#14 - 10/21/2014 08:31 PM - jared.jennings.ctr (Jared Jennings)

Nobuyoshi Nakada wrote:

> Is EVP API necessary?

The EVP API has been recommended over the old digest-specific API for almost fifteen years. It seems that EVP might automatically use hardware acceleration where possible. And if EVP is not used, Ruby crashes on the secure systems used by banks and governments, with no indication of which Ruby code caused the problem.

Nobuyoshi Nakada wrote:

> I've reverted it because of segfaults on many platforms.

Since EVP is so old already, any problem is likely due somehow to my code. I'd like to fix this. Can you share any further details?

#15 - 01/05/2018 09:00 PM - naruse (Yui NARUSE)

- Target version deleted (2.2.0)

Files

002-builtin-indicate-digest-failure.patch	10.4 KB	03/27/2014	jared.jennings.ctr (Jared Jennings)
001-detect-digest-failure.patch	2.12 KB	03/27/2014	jared.jennings.ctr (Jared Jennings)
003-digest-openssl-md5-use-evp-api.patch	1.8 KB	03/27/2014	jared.jennings.ctr (Jared Jennings)