Ruby trunk - Bug #6928

SecureRandom.random_bytes: assume zero entropy for seed value

08/26/2012 01:58 AM - MartinBosslet (Martin Bosslet)

Status:	Closed		
Priority:	Normal		
Assignee:	akr (Akira Tanaka)		
Target version:	2.6		
ruby -v:	trunk	**Backport:**	

Description

If OpenSSL is available SecureRandom.random_bytes uses
OpenSSL::Random.random_bytes and the random generator is reseeded [1]
whenever the current pid changes (due to repeated values when a pid
is reused, cf. #4579).

Since this seeding is also called the first time the method is entered,
using OpenSSL::Random.seed is potentially dangerous. OpenSSL::Random.seed
is equal to using OpenSSL::Random.random_add where it is assumed that the
string passed to seed possesses full entropy. This is definitely not the
case for pid and time values. In fact, OpenSSL itself assumes an entropy
of 1.0 or even 0.0 when doing similar seeding in RAND_poll [2][3]. However,
this seems to have no impact so far, since the OpenSSL random generator
gathers enough entropy on startup even if we seeded with what it would
consider enough bytes of entropy (32 by default). So even if our seed
string is already 32 bytes or larger, OpenSSL's RAND_poll still seems to
collect 32 bytes of entropy on initialization regardless of what has been
added/seeded so far, which is a good thing in this case. Still, this could
change over time if OpenSSL for example changes internal behaviour and
would decide that enough entropy had been provided while seeding.

Therefore I believe using OpenSSL::Random.random_add with an assumed
entropy of 0.0 might be a more defensive choice. The forking test from
#4579 still passes with the attached patch. What do you think?

[1] https://github.com/ruby/ruby/blob/trunk/lib/securerandom.rb#L56
[2] https://github.com/plenluno/openssl/blob/master/crypto/rand/rand_unix.c#L179
[3] https://github.com/plenluno/openssl/blob/master/crypto/rand/rand_unix.c#L398

Associated revisions

Revision c3c4ffa9 - 04/02/2013 03:09 PM - akr (Akira Tanaka)

- lib/securerandom.rb (SecureRandom.random_bytes): Use OpenSSL::Random.random_add instead of OpenSSL::Random.seed and specify 0.0 as the entropy. [ruby-core:47308] [Bug #6928]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@40072 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 40072 - 04/02/2013 03:09 PM - akr (Akira Tanaka)

- lib/securerandom.rb (SecureRandom.random_bytes): Use OpenSSL::Random.random_add instead of OpenSSL::Random.seed and specify 0.0 as the entropy. [ruby-core:47308] [Bug #6928]

Revision 40072 - 04/02/2013 03:09 PM - akr (Akira Tanaka)

- lib/securerandom.rb (SecureRandom.random_bytes): Use OpenSSL::Random.random_add instead of OpenSSL::Random.seed and specify 0.0 as the entropy. [ruby-core:47308] [Bug #6928]

Revision 40072 - 04/02/2013 03:09 PM - akr (Akira Tanaka)

- lib/securerandom.rb (SecureRandom.random_bytes): Use OpenSSL::Random.random_add instead of OpenSSL::Random.seed and specify 0.0 as the entropy. [ruby-core:47308] [Bug #6928]

Revision 40072 - 04/02/2013 03:09 PM - akr (Akira Tanaka)

- lib/securerandom.rb (SecureRandom.random_bytes): Use OpenSSL::Random.random_add instead of OpenSSL::Random.seed and specify 0.0 as the entropy. [ruby-core:47308] [Bug #6928]

Revision 40072 - 04/02/2013 03:09 PM - akr (Akira Tanaka)

- lib/securerandom.rb (SecureRandom.random_bytes): Use OpenSSL::Random.random_add instead of OpenSSL::Random.seed and specify 0.0 as the entropy. [ruby-core:47308] [Bug #6928]

Revision 40072 - 04/02/2013 03:09 PM - akr (Akira Tanaka)

- lib/securerandom.rb (SecureRandom.random_bytes): Use OpenSSL::Random.random_add instead of OpenSSL::Random.seed and specify 0.0 as the entropy. [ruby-core:47308] [Bug #6928]

History

#1 - 08/27/2012 11:21 PM - nahi (Hiroshi Nakamura)

Agreed. We should fix it because the current usage of OpenSSL::Rand.seed in secrerandom.rb is not expected; OpenSSL::Rand.seed(bytes) is a wrapper for RAND_seed(), RAND_seed() is equivalent to RAND_add() when num == entropy, and the entropy for RAND_add() must be a lower bound of an estimate of entropy of the given seed. 'ary.to_s' clearly does not have an entropy of 30 bytes.

The patch looks good to me. Though the buf would have 5 bytes or so of entropy, we don't need to bother the exact lower bound I think. :-)

#2 - 12/21/2012 10:32 PM - tarui (Masaya Tarui)

- Status changed from Open to Assigned

#3 - 02/18/2013 11:50 PM - mame (Yusuke Endoh)

Martin, may I postpone this to next minor?
Or must it be fixed immediately?

--
Yusuke Endoh mame@tsg.ne.jp

#4 - 02/20/2013 04:24 PM - mame (Yusuke Endoh)

- Target version changed from 2.0.0 to 2.6

I assume that if this is so significant issue, Martin would have reported this to security@ruby-lang.org.
So I postpone this to next minor.

--
Yusuke Endoh mame@tsg.ne.jp

#5 - 02/25/2013 10:16 AM - MartinBosslet (Martin Bosslet)

mame (Yusuke Endoh) wrote:

> I assume that if this is so significant issue, Martin would have reported this to security@ruby-lang.org.
> So I postpone this to next minor.

Sorry for not responding in time. It is safe to move this to next minor - right now, the risk I mentioned is only hypothetical and would only affect us if OpenSSL decided to change their internals.

#6 - 04/03/2013 12:09 AM - akr (Akira Tanaka)

- Status changed from Assigned to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r40072.
Martin, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- lib/securerandom.rb (SecureRandom.random_bytes): Use OpenSSL::Random.random_add instead of OpenSSL::Random.seed and specify 0.0 as the entropy. [ruby-core:47308] [Bug #6928]

Files

securerandom.patch	454 Bytes	08/26/2012	MartinBosslet (Martin Bosslet)