Ruby master - Bug #4579

SecureRandom + OpenSSL may repeat with fork

04/15/2011 11:46 AM - normalperson (Eric Wong)

Status:	Closed		
Priority:	Normal		
Assignee:	akr (Akira Tanaka)		
Target version:	2.0.0		
ruby -v:	-	**Backport:**	

Description

=begin
This could arguably be a bug in OpenSSL or the openssl extension, but I think it's easier to fix in Ruby right now.

The PRNG in OpenSSL uses the PID to seed the PRNG. Since PIDs get recycled over time on Unix systems, this means independent processes over a long time span will repeat random byte sequences. This has security implications, but fortunately very little software forks very frequently. I am not a security expert.

I am using OpenSSL 0.9.8g-15+lenny11 (Debian Lenny)

Attached is a script that reproduces the issue (takes a while to run).
It'll output two identical lines to illustrate the issue.

=end

Associated revisions

Revision 32122 - 06/16/2011 10:32 AM - nahi (Hiroshi Nakamura)

- test/test_securerandom.rb: Add testcase. This testcase does NOT aim to test cryptographically strongness and randomness. It includes the test for PID recycle issue of OpenSSL described in #4579 but it's disabled by default.

Revision 32122 - 06/16/2011 10:32 AM - nahi (Hiroshi Nakamura)

- test/test_securerandom.rb: Add testcase. This testcase does NOT aim to test cryptographically strongness and randomness. It includes the test for PID recycle issue of OpenSSL described in #4579 but it's disabled by default.

Revision 32122 - 06/16/2011 10:32 AM - nahi (Hiroshi Nakamura)

- test/test_securerandom.rb: Add testcase. This testcase does NOT aim to test cryptographically strongness and randomness. It includes the test for PID recycle issue of OpenSSL described in #4579 but it's disabled by default.

Revision 32122 - 06/16/2011 10:32 AM - nahi (Hiroshi Nakamura)

- test/test_securerandom.rb: Add testcase. This testcase does NOT aim to test cryptographically strongness and randomness. It includes the test for PID recycle issue of OpenSSL described in #4579 but it's disabled by default.

Revision 32122 - 06/16/2011 10:32 AM - nahi (Hiroshi Nakamura)

- test/test_securerandom.rb: Add testcase. This testcase does NOT aim to test cryptographically strongness and randomness. It includes the test for PID recycle issue of OpenSSL described in #4579 but it's disabled by default.

Revision 32122 - 06/16/2011 10:32 AM - nahi (Hiroshi Nakamura)

- test/test_securerandom.rb: Add testcase. This testcase does NOT aim to test cryptographically strongness and randomness. It includes the test for PID recycle issue of OpenSSL described in #4579 but it's disabled by default.

History

#1 - 04/15/2011 02:54 PM - naruse (Yui NARUSE)

=begin
SecureRandom.random_bytes is just a wrapper of OpenSSL::Random.random_bytes(n) on systems with openssl.
And OpenSSL::Random.random_bytes is a wrapper of RAND_bytes(3).

You know its result depends pid and openssl ext should use RAND_add(3) or something.
http://www.openssl.org/docs/crypto/RAND_add.html
=end

#2 - 04/15/2011 06:19 PM - nahi (Hiroshi Nakamura)

=begin
I don't have an idea how OpenSSL can be 'fork-safe' for your purpose...

Call OpenSSL::Random.load_random_file("/dev/urandom" or "/dev/random" or ENV["RANDFILE"]) after each fork.
=end

#3 - 04/16/2011 12:00 AM - kosaki (Motohiro KOSAKI)

- File ossl_rand.patch added

=begin
Usually openssl read /dev/urandom only once. But RAND_cleanup() lead to read /dev/urandom again. Thus attached patch fixes this issue.

This is better patch than PAND_add(/dev/urandom) because openssl can use other entropy source internally.
=end

#4 - 04/16/2011 12:14 AM - kosaki (Motohiro KOSAKI)

- File ossl_rand2.patch added

=begin
More paranoia patch is here.
=end

#5 - 04/16/2011 05:27 AM - normalperson (Eric Wong)

=begin
I think RAND_cleanup() is enough and simpler. I'm also bringing this up on the openssl-dev mailing list.
=end

#6 - 04/16/2011 07:37 AM - nahi (Hiroshi Nakamura)

=begin
Motohiro: I don't know you're serious or not about using pthread_atfork(), we should ask to change OpenSSL's "1 time initialization by RAND_poll() per process when using built-in MD based RPNG engine" strategy if we really want.

Eric: I saw your post at openssl-dev[1]. Let's see how they treat this. I don't know OpenSSL can be 'fork-safe' for your purpose as I wrote. There must be other per-process initializations in it.

[1] http://marc.info/?l=openssl-dev&m=130289811108150&w=2

=end

#7 - 04/16/2011 01:23 PM - kosaki (Motohiro KOSAKI)

- ruby -v changed from ruby 1.9.3dev (2011-04-14 trunk 31267) [x86_64-linux] to -

=begin
Hi

> Motohiro: I don't know you're serious or not about using pthread_atfork(), we should ask to change OpenSSL's "1 time initialization by RAND_poll() per process when using built-in MD based RPNG engine" strategy if we really want.

It's ruby's spec. We already decided random seed should reinitialize per fork.

void rb_thread_atfork(void)
{
rb_thread_atfork_internal(terminate_atfork_i);
GET_THREAD()->join_list_head = 0;

```
 /* We don't want reproduce CVE-2003-0900. */
```

```
rb_reset_random_seed();
```

}

Now, SecureRandom is insecure than normal random from fork issue. It's
rather than unhappy.
=end

#8 - 04/20/2011 11:57 AM - nahi (Hiroshi Nakamura)

=begin
I think you're confusing SecureRandom's spec and ext/openssl (formerly ruby-pki) spec. ext/openssl aims to wrap OpenSSL that user's using so if OpenSSL is not 'fork-safe' as Eric expected, so ruby-pki doesn't.

So if OpenSSL can't change this behavior (I bet they can't at least in the near future), why don't we change lib/securerandom.rb?

The reason why I think you're not serious is adding ptherad_atfork() in ext is too ad-hoc-ish. We can't do it from Ruby world if I understand correctly. Adding atfork hook first?

To change (OK, 'fix') this behavior, it could be good to abandon using OpenSSL::Random.random_bytes(). We cannot use OpenSSL's engine which could provide physical random number but OpenSSL::Random.random_bytes must be enough for such user.

Tanaka-san, what do you think about the change?

=end

#9 - 04/20/2011 05:23 PM - normalperson (Eric Wong)

=begin
Hiroshi NAKAMURA wrote:

> I think you're confusing SecureRandom's spec and ext/openssl (formerly ruby-pki) spec. ext/openssl aims to wrap OpenSSL that user's using so if OpenSSL is not 'fork-safe' as Eric expected, so ruby-pki doesn't.

I hope everything in Ruby (including 3rd-party extensions/gems) can be made fork-safe by default (if they run on a system with fork) one day.
I don't agree with blindly mimicking OpenSSL upstream behavior if Ruby can be made easier-to-use.

> So if OpenSSL can't change this behavior (I bet they can't at least in the near future), why don't we change lib/securerandom.rb?

Yes, I confirmed OpenSSL can't change the current behavior:
http://marc.info/?l=openssl-dev&m=130298304903422&w=2

I'm still hoping to get a list of things that need to be reinitialized in OpenSSL after fork() from openssl-dev...

> The reason why I think you're not serious is adding ptherad_atfork() in ext is too ad-hoc-ish. We can't do it from Ruby world if I understand correctly. Adding atfork hook first?

I would be 100% in favor of making something analogous to pthread_atfork() available in Ruby. It would make it much easier to manage various resources in a multi-process situation

No comment on the appropriateness of pthread_atfork() inside an ext.

--
Eric Wong
=end

#10 - 06/11/2011 05:02 PM - ko1 (Koichi Sasada)

- Status changed from Open to Assigned

- Assignee set to nahi (Hiroshi Nakamura)

#11 - 06/11/2011 06:36 PM - nahi (Hiroshi Nakamura)

- File securerandom_opensslfree.diff added

- Assignee changed from nahi (Hiroshi Nakamura) to akr (Akira Tanaka)

Attached is the patch which removes OpenSSL dependency. Tanaka-san, aside from how OpenSSL.random_bytes should behave, can you accept this change?

#12 - 06/11/2011 10:38 PM - kosaki (Motohiro KOSAKI)

Eeek. I dislike to remove OpenSSL dependency from SecureRadom. Because /dev/urandom is less secure than OpenSSL.

#13 - 06/12/2011 08:07 PM - akr (Akira Tanaka)

This issue seems a OpenSSL issue.

Does someone reported to OpenSSL project?

#14 - 06/12/2011 10:48 PM - kosaki (Motohiro KOSAKI)

Yes, comment#9 said,

> Yes, I confirmed OpenSSL can't change the current behavior:
> http://marc.info/?l=openssl-dev&m=130298304903422&w=2

#15 - 06/13/2011 01:11 AM - akr (Akira Tanaka)

- File securerandom-openssl-pid-recycle.patch added

Hm.

I don't like pthread_atfork because the hook is run even if we don't need random functions in the child process.
(Remember only async signal safe functions are safe in forked child process)

We should delay modifying PRNG state until we really need it.

securerandom-openssl-pid-recycle.patch do it.
It should work until we have very fast machine which pid is recycled in a nano second.

#16 - 06/13/2011 11:00 AM - kosaki (Motohiro KOSAKI)

The patch looks good to me.

Thank you.

#17 - 06/13/2011 12:23 PM - normalperson (Eric Wong)

Motohiro KOSAKI kosaki.motohiro@gmail.com wrote:

> Eeek. I dislike to remove OpenSSL dependency from SecureRadom. Because
> /dev/urandom is less secure than OpenSSL.

Is that only the case with poor urandom implementations in some
systems?

I'm fine with securerandom-openssl-pid-recycle.patch but I like
securerandom_opensslfree.diff since it removes OpenSSL dependency.
(I personally never liked OpenSSL)

#18 - 06/13/2011 05:06 PM - akr (Akira Tanaka)

I think securerandom_opensslfree.diff is too radical for this issue.
It may decrease working platforms.

However concrete advantage/disadvantage between OpenSSL and /dev/urandom is interesting.
(portability, randomness quality, performance, ...)

#19 - 06/14/2011 08:39 AM - akr (Akira Tanaka)

- Status changed from Assigned to Closed

- % Done changed from 0 to 100

#20 - 06/16/2011 08:05 PM - nahi (Hiroshi Nakamura)

- File securerandom.rb.diff added

Attached is the patch for 1.8.7. Urabe-san, can I apply it to ruby_1_8_7?

#21 - 06/16/2011 08:23 PM - nahi (Hiroshi Nakamura)

On Mon, Jun 13, 2011 at 17:07, Akira Tanaka akr@fsij.org wrote:

> I think securerandom_opensslfree.diff is too radical for this issue.
> It may decrease working platforms.

Agreed. Your fix is nice. We should have been aware of that. Thanks.

> However concrete advantage/disadvantage between OpenSSL and /dev/urandom is interesting.
> (portability, randomness quality, performance, ...)

On Linux, /dev/urandom seems to return the values which are
"theoretically vulnerable to a cryptographic attack on the algorithms
used by the driver" (from man page). I though it returns shorter bytes
than expected. I was wrong.

And using /dev/urandom every time consumes too much entropy that OS
has, so /dev/random users would not like it. We should avoid using
/dev/urandom every time on the env w/o OpenSSL in the future.

Regards,
// NaHi

#22 - 06/16/2011 09:42 PM - shyouhei (Shyouhei Urabe)

Hmm, OK. Go ahead.

#23 - 06/16/2011 10:53 PM - nahi (Hiroshi Nakamura)

> Hmm, OK. Â Go ahead.

OK.

And I found that Tanaka-san already fixed it at ruby_1_8 as well as
trunk. :) I'll push it to ruby_1_8_7 with tests.

Regards,
// NaHi

#24 - 06/23/2011 08:13 AM - akr (Akira Tanaka)

NaHi:

> We should avoid using
> /dev/urandom every time on the env w/o OpenSSL in the future.

I'd like to say "Please install OpenSSL" for such request.

Cryptographic algorithms should be implemented/maintained by cryptographic experts but I am not a cryptographic expert.

#25 - 06/23/2011 03:53 PM - nahi (Hiroshi Nakamura)

Hi,

On Thu, Jun 23, 2011 at 08:15, Akira Tanaka akr@fsij.org wrote:

> > We should avoid using
> > /dev/urandom every time on the env w/o OpenSSL in the future.
>
> I'd like to say "Please install OpenSSL" for such request.

Reasonable. Why don't you do so? I mean that removing /dev/urandom

fallback from securerandom.rb and letting simply warn "Please install OpenSSL".

> Cryptographic algorithms should be implemented/maintained by cryptographic experts but I am not a cryptographic expert.

You wrote securerandom.rb. I think it's too late. :-) :-) Joking aside, since there's no cryptography expert around us, delegating PRNG thing to OpenSSL is good I think.

Regards,
// NaHi

#26 - 06/23/2011 07:55 PM - akr (Akira Tanaka)

I don't understand why /dev/urandom fallback should be removed.

Is your reason the "theoretically vulnerable to a cryptographic attack on the algorithms used by the driver" (from Linux man page)?

Files

test_fork_random.rb	292 Bytes	04/15/2011	normalperson (Eric Wong)
ossl_rand.patch	848 Bytes	04/16/2011	kosaki (Motohiro KOSAKI)
ossl_rand2.patch	923 Bytes	04/16/2011	kosaki (Motohiro KOSAKI)
securerandom_opensslfree.diff	2.69 KB	06/11/2011	nahi (Hiroshi Nakamura)
securerandom-openssl-pid-recycle.patch	543 Bytes	06/13/2011	akr (Akira Tanaka)
securerandom.rb.diff	4.82 KB	06/16/2011	nahi (Hiroshi Nakamura)