Backport193 - Backport #5233

OpenSSL::SSL::SSLSocket has problems with encodings other than "ascii"

08/26/2011 08:45 PM - niklasb (Niklas Baumstark)

Status:	Closed	
Priority:	Normal	
Assignee:	MartinBosslet (Martin Bosslet)	

Description

The attached script shows the issue. It expects a combined cert/private key in the file server.pem under the current directory (also attached).
Under Ruby 1.9.2p290, the script prints "str.size is 50848 (expecting 100000)".
As a workaround the string encoding can be forced to "ascii" before the write.

Associated revisions

Revision 65ca601b - 10/19/2011 08:05 PM - emboss

- lib/openssl/buffering.rb: Force multi-byte strings to be treated as binary data.
- test/openssl/test_ssl.rb: Add test for it.

Thanks to Niklas Baumstark for reporting the issue!

[Ruby 1.9 - Bug #5233] [ruby-core:39120]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@33485 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 33485 - 10/19/2011 08:05 PM - emboss

- lib/openssl/buffering.rb: Force multi-byte strings to be treated as binary data.
- test/openssl/test_ssl.rb: Add test for it.

Thanks to Niklas Baumstark for reporting the issue!

[Ruby 1.9 - Bug #5233] [ruby-core:39120]

Revision 33485 - 10/19/2011 08:05 PM - emboss

- lib/openssl/buffering.rb: Force multi-byte strings to be treated as binary data.
- test/openssl/test_ssl.rb: Add test for it.

Thanks to Niklas Baumstark for reporting the issue!

[Ruby 1.9 - Bug #5233] [ruby-core:39120]

Revision 33485 - 10/19/2011 08:05 PM - emboss

- lib/openssl/buffering.rb: Force multi-byte strings to be treated as binary data.
- test/openssl/test_ssl.rb: Add test for it.

Thanks to Niklas Baumstark for reporting the issue!

[Ruby 1.9 - Bug #5233] [ruby-core:39120]

Revision 33485 - 10/19/2011 08:05 PM - emboss

- lib/openssl/buffering.rb: Force multi-byte strings to be treated as binary data.
- test/openssl/test_ssl.rb: Add test for it.

Thanks to Niklas Baumstark for reporting the issue!

[Ruby 1.9 - Bug #5233] [ruby-core:39120]

Revision 33485 - 10/19/2011 08:05 PM - emboss

- lib/openssl/buffering.rb: Force multi-byte strings to be treated as binary data.
- test/openssl/test_ssl.rb: Add test for it.

Thanks to Niklas Baumstark for reporting the issue!

[Ruby 1.9 - Bug #5233] [ruby-core:39120]

Revision 33485 - 10/19/2011 08:05 PM - emboss

- lib/openssl/buffering.rb: Force multi-byte strings to be treated as binary data.
- test/openssl/test_ssl.rb: Add test for it.

Thanks to Niklas Baumstark for reporting the issue!

[Ruby 1.9 - Bug #5233] [ruby-core:39120]

Revision 2c5d6bae - 02/10/2012 05:27 PM - naruse (Yui NARUSE)

merge revision(s) 33485:

```
    * lib/openssl/buffering.rb: Force multi-byte strings to be treated as
      binary data.

    * test/openssl/test_ssl.rb: Add test for it.
    Thanks to Niklas Baumstark for reporting the issue!
    [Ruby 1.9 - Bug #5233] [ruby-core:39120]
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_1_9_3@34534 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 34534 - 02/10/2012 05:27 PM - naruse (Yui NARUSE)

merge revision(s) 33485:

```
* lib/openssl/buffering.rb: Force multi-byte strings to be treated as
  binary data.

* test/openssl/test_ssl.rb: Add test for it.
Thanks to Niklas Baumstark for reporting the issue!
[Ruby 1.9 - Bug #5233] [ruby-core:39120]
```

History

#1 - 09/02/2011 08:31 AM - MartinBosslet (Martin Bosslet)

- Category set to ext

- Status changed from Open to Assigned

- Assignee set to MartinBosslet (Martin Bosslet)

- Target version set to 1.9.4

#2 - 10/19/2011 01:37 AM - MartinBosslet (Martin Bosslet)

The problem is in lib/openssl/buffering.rb:

```
def do_write(s)
  @wbuffer = "" unless defined? @wbuffer
  @wbuffer << s
  @sync ||= false
  if @sync or @wbuffer.size > BLOCK_SIZE or idx = @wbuffer.rindex($/)
    remain = idx ? idx + $/.size : @wbuffer.length
    nwritten = 0
    while remain > 0
      str = @wbuffer[nwritten,remain]
      begin
        nwrote = syswrite(str)
      rescue Errno::EAGAIN
        retry
      end
      remain -= nwrote
      nwritten += nwrote
    end
    @wbuffer[0,nwritten] = ""
  end
end
```

remain gets initialized with @wbuffer.length, the string length in characters,
but nwrote receives the actual number of bytes written, so less bytes than
actually available are written.

A fix for this would be treating @wbuffer strictly as binary data by forcing
its encoding to BINARY. I'm not sure, does anyone see a more elegant way or
would this solution suffice?

#3 - 10/19/2011 02:53 AM - normalperson (Eric Wong)

Martin Bosslet Martin.Bosslet@googlemail.com wrote:

> The problem is in lib/openssl/buffering.rb:
>
> ```
> def do_write(s)
>   @wbuffer = "" unless defined? @wbuffer
>   @wbuffer << s
>   @sync ||= false
>   if @sync or @wbuffer.size > BLOCK_SIZE or idx = @wbuffer.rindex($/)
>     remain = idx ? idx + $/.size : @wbuffer.length
>     nwritten = 0
>     while remain > 0
>       str = @wbuffer[nwritten,remain]
>       begin
>         nwrote = syswrite(str)
>       rescue Errno::EAGAIN
>         retry
>       end
>       remain -= nwrote
>       nwritten += nwrote
>     end
>     @wbuffer[0,nwritten] = ""
>   end
> end
> ```
>
> remain gets initialized with @wbuffer.length, the string length in characters,
> but nwrote receives the actual number of bytes written, so less bytes than
> actually available are written.
>
> A fix for this would be treating @wbuffer strictly as binary data by forcing
> its encoding to BINARY. I'm not sure, does anyone see a more elegant way or
> would this solution suffice?

I use an "-- *encoding: binary* --" comment at the top of all Ruby
source files where I initialize string literals for storing binary data.
It's cleaner than setting Encoding::BINARY on every string I create
(and nearly all my code works exclusively on binary data).

Also, all of the Ruby (non-SSL) *Socket objects have Encoding::BINARY by
default anyways, so I think SSLSocket should be the same.

#4 - 10/19/2011 11:08 AM - MartinBosslet (Martin Bosslet)

> I use an "-- *encoding: binary* --" comment at the top of all Ruby
> source files where I initialize string literals for storing binary data.
> It's cleaner than setting Encoding::BINARY on every string I create
> (and nearly all my code works exclusively on binary data).

I'm afraid this had no effect, or I did it wrong, or I might also have
misunderstood you. The incoming string s already has UTF-8 encoding, so

```
@wbuffer << s
```

ends up as UTF-8 regardless of the encoding I set for the .rb file, I
figured this was because "<<" calls rb_str_append which again calls
rb_enc_check which will determine a compatible encoding, in this case
UTF-8, for @wbuffer. But again, I might have misunderstood you.

> Also, all of the Ruby (non-SSL) *Socket objects have Encoding::BINARY by
> default anyways, so I think SSLSocket should be the same.

I'm sorry, I don't understand what you mean by the *Socket objects have binary encoding by default - do you mean it's binary data they are expecting to deal with for input and output? So a user would have to make sure to only pass already BINARY-encoded strings to any *Socket?

I quickly checked with a TCPServer and Net::HTTP client, there the aforementioned situation would work, when sending 100000 a-Umlauts you again receive the same amount, after enforcing the response to UTF-8 again, of course. That's why I thought that an SSLSocket should behave the same way.

#5 - 10/19/2011 11:53 AM - normalperson (Eric Wong)

Martin Bosslet Martin.Bosslet@googlemail.com wrote:

> I use an "-- *encoding: binary* --" comment at the top of all Ruby source files where I initialize string literals for storing binary data. It's cleaner than setting Encoding::BINARY on every string I create (and nearly all my code works exclusively on binary data).

> I'm afraid this had no effect, or I did it wrong, or I might also have misunderstood you. The incoming string s already has UTF-8 encoding, so
>
> ```
> @wbuffer << s
> ```
>
> ends up as UTF-8 regardless of the encoding I set for the .rb file, I figured this was because "<<" calls rb_str_append which again calls rb_enc_check which will determine a compatible encoding, in this case UTF-8, for @wbuffer. But again, I might have misunderstood you.

You're right. rb_str_append() modifies the empty @wbuffer to the encoding of "s" above :(

I suppose calling @wbuffer.force_encoding(Encoding::BINARY) after @wbuffer is necessary (unless you write the buffering code in C like io.c does).

> Also, all of the Ruby (non-SSL) *Socket objects have Encoding::BINARY by default anyways, so I think SSLSocket should be the same.

> I'm sorry, I don't understand what you mean by the *Socket objects have binary encoding by default - do you mean it's binary data they are expecting to deal with for input and output? So a user would have to make sure to only pass already BINARY-encoded strings to any *Socket?

For all newly-created *Socket objects, external_encoding is already ASCII-8BIT (binary) and the sockets should just pass the byte buffer of any underlying String objects given to it.

> I quickly checked with a TCPServer and Net::HTTP client, there the aforementioned situation would work, when sending 100000 a-Umlauts you again receive the same amount, after enforcing the response to UTF-8 again, of course. That's why I thought that an SSLSocket should behave the same way.

Yes, underlying IO#read/read_nonblock/sysread for the TCPSocket objects should return new ASCII-8BIT Strings. You needed to force them to UTF-8 yourself upon receipt.

#6 - 10/20/2011 05:05 AM - Anonymous

- Status changed from Assigned to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r33485.
Niklas, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- lib/openssl/buffering.rb: Force multi-byte strings to be treated as binary data.
- test/openssl/test_ssl.rb: Add test for it.

Thanks to Niklas Baumstark for reporting the issue!

[Ruby 1.9 - Bug #5233] [ruby-core:39120]

#7 - 10/20/2011 05:14 AM - MartinBosslet (Martin Bosslet)

Also thanks to Eric, for providing his thoughts on the topic!

#8 - 02/11/2012 02:22 AM - naruse (Yui NARUSE)

- Tracker changed from Bug to Backport

- Project changed from Ruby master to Backport193

- Category deleted (ext)

- Target version deleted (1.9.4)

#9 - 02/11/2012 07:22 AM - naruse (Yui NARUSE)

- Status changed from Closed to Assigned

This breaks tests on CentOS 5.6 (but not Ubuntu 10.04, FreeBSD 8, 9).
http://c5664.rubyci.org/~chkbuild/ruby-1.9.3/log/20120210T173209Z.diff.html.gz

Maybe more some commits are needed.

#10 - 02/11/2012 01:11 PM - naruse (Yui NARUSE)

- Status changed from Assigned to Closed

Backported r33508 and fixed this.

Files

openssl_write_failure.rb	892 Bytes	08/26/2011	niklasb (Niklas Baumstark)
server.pem	1.62 KB	08/26/2011	niklasb (Niklas Baumstark)