

## Ruby trunk - Bug #5353

### TLS v1.0 and less - Attack on CBC mode

09/23/2011 01:14 AM - MartinBosslet (Martin Bosslet)

<b>Status:</b>	Closed	
<b>Priority:</b>	Normal	
<b>Assignee:</b>	MartinBosslet (Martin Bosslet)	
<b>Target version:</b>	2.0.0	
<b>ruby -v:</b>	-	
<b>Description</b>		<b>Backport:</b>
<p>A well-known vulnerability of TLS v1.0 and earlier has recently gained some attention:</p> <p><a href="http://www.theregister.co.uk/2011/09/19/beast_exploits_paypal_ssl/">http://www.theregister.co.uk/2011/09/19/beast_exploits_paypal_ssl/</a></p> <p>Although this has been known for a long time (<a href="http://www.openssl.org/~bodo/tls-cbc.txt">http://www.openssl.org/~bodo/tls-cbc.txt</a>), and a fix for this has been provided, in reality most applications seem to be working with</p> <p>SSL_OP_ALL</p> <p>which is a flag that enables some bug workarounds that were considered harmless.</p> <p>We, too, use this in <code>ossl_sslctx_s_alloc(VALUE klass)</code> in <code>ossl_ssl.c</code>. Unfortunately, this flag also includes</p> <p>SSL_OP_DONT_INSERT_EMPTY_FRAGMENTS</p> <p>which disables the fix for the "CBC vulnerability". Here is what a comment says about the flag (OpenSSL 1.0.0d)</p> <pre>/* Disable SSL 3.0/TLS 1.0 CBC vulnerability workaround that was added  * in OpenSSL 0.9.6d. Usually (depending on the application protocol)  * the workaround is not needed. Unfortunately some broken SSL/TLS  * implementations cannot handle it at all, which is why we include  * it in SSL_OP_ALL. */</pre> <p>If I understand <a href="http://www.openssl.org/~bodo/tls-cbc.txt">http://www.openssl.org/~bodo/tls-cbc.txt</a> correctly, the most notable implementation that does not play well with these empty fragments was (is?) IE - I don't know how this has evolved over time, I would have to research further.</p> <p>An easy fix for the situation would be to discard <code>SSL_OP_DONT_INSERT_EMPTY_FRAGMENTS</code>, but this would risk affecting existing installations.</p> <p>What do you propose? Should we solve this before the 1.9.3 release?</p> <p>(PS: The actual attack and fix are outlined in</p> <p><a href="http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.61.5887&amp;rep=rep1&amp;type=pdf">http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.61.5887&amp;rep=rep1&amp;type=pdf</a></p> <p>The attack to be presented by Thai Duong and Juliano Rizzo at</p> <p><a href="http://ekoparty.org/cronograma.php">http://ekoparty.org/cronograma.php</a> (caution: currently the site is victim to the "reddit effect")</p> <p>is very likely to be based on what was already known and should therefore hopefully require no further fixes.)</p>		

#### Associated revisions

Revision 3ff2f9f3 - 02/08/2012 05:27 AM - nahi (Hiroshi Nakamura)

- ext/openssl/ossl\_ssl.c: Add SSL constants and allow to unset SSL option to prevent BEAST attack. See [Bug #5353].

In OpenSSL, OP\_DONT\_INSERT\_EMPTY\_FRAGMENTS is used to prevent TLS-CBC-IV vulnerability described at <http://www.openssl.org/~bodo/tls-cbc.txt> It's known issue of TLSv1/SSLv3 but it attracts lots of attention these days as BEAST attack. (CVE-2011-3389)

Until now openssl sets OP\_ALL at SSLContext allocation and call SSL\_CTX\_set\_options at connection. SSL\_CTX\_set\_options updates the value by using |= so bits set by OP\_ALL cannot be unset afterwards.

This commit changes to call SSL\_CTX\_set\_options only 1 time for each SSLContext. It sets the specified value if SSLContext#options= are called and sets OP\_ALL if not.

To help users to unset bits in OP\_ALL, this commit also adds several constant to SSL such as OpenSSL::SSL::OP\_DONT\_INSERT\_EMPTY\_FRAGMENTS. These constants were not exposed in Ruby because there's no way to unset bits in OP\_ALL before.

Following is an example to enable 0/n split for BEAST prevention.

```
ctx.options = OP_ALL & ~OP_DONT_INSERT_EMPTY_FRAGMENTS
```

- test/openssl/test\_ssl.rb: Test above option exists.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@34482 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

#### Revision 34482 - 02/08/2012 05:27 AM - nahi (Hiroshi Nakamura)

- ext/openssl/openssl.c: Add SSL constants and allow to unset SSL option to prevent BEAST attack. See [Bug #5353].

In OpenSSL, OP\_DONT\_INSERT\_EMPTY\_FRAGMENTS is used to prevent TLS-CBC-IV vulnerability described at <http://www.openssl.org/~bodo/tls-cbc.txt> It's known issue of TLSv1/SSLv3 but it attracts lots of attention these days as BEAST attack. (CVE-2011-3389)

Until now openssl sets OP\_ALL at SSLContext allocation and call SSL\_CTX\_set\_options at connection. SSL\_CTX\_set\_options updates the value by using |= so bits set by OP\_ALL cannot be unset afterwards.

This commit changes to call SSL\_CTX\_set\_options only 1 time for each SSLContext. It sets the specified value if SSLContext#options= are called and sets OP\_ALL if not.

To help users to unset bits in OP\_ALL, this commit also adds several constant to SSL such as OpenSSL::SSL::OP\_DONT\_INSERT\_EMPTY\_FRAGMENTS. These constants were not exposed in Ruby because there's no way to unset bits in OP\_ALL before.

Following is an example to enable 0/n split for BEAST prevention.

```
ctx.options = OP_ALL & ~OP_DONT_INSERT_EMPTY_FRAGMENTS
```

- test/openssl/test\_ssl.rb: Test above option exists.

#### Revision 34482 - 02/08/2012 05:27 AM - nahi (Hiroshi Nakamura)

- ext/openssl/openssl.c: Add SSL constants and allow to unset SSL option to prevent BEAST attack. See [Bug #5353].

In OpenSSL, OP\_DONT\_INSERT\_EMPTY\_FRAGMENTS is used to prevent TLS-CBC-IV vulnerability described at <http://www.openssl.org/~bodo/tls-cbc.txt> It's known issue of TLSv1/SSLv3 but it attracts lots of attention these days as BEAST attack. (CVE-2011-3389)

Until now openssl sets OP\_ALL at SSLContext allocation and call SSL\_CTX\_set\_options at connection. SSL\_CTX\_set\_options updates the value by using |= so bits set by OP\_ALL cannot be unset afterwards.

This commit changes to call SSL\_CTX\_set\_options only 1 time for each

SSLContext. It sets the specified value if SSLContext#options= are called and sets OP\_ALL if not.

To help users to unset bits in OP\_ALL, this commit also adds several constant to SSL such as OpenSSL::SSL::OP\_DONT\_INSERT\_EMPTY\_FRAGMENTS. These constants were not exposed in Ruby because there's no way to unset bits in OP\_ALL before.

Following is an example to enable 0/n split for BEAST prevention.

```
ctx.options = OP_ALL & ~OP_DONT_INSERT_EMPTY_FRAGMENTS
```

- test/openssl/test\_ssl.rb: Test above option exists.

#### Revision 34482 - 02/08/2012 05:27 AM - nahi (Hiroshi Nakamura)

- ext/openssl/openssl.c: Add SSL constants and allow to unset SSL option to prevent BEAST attack. See [Bug #5353].

In OpenSSL, OP\_DONT\_INSERT\_EMPTY\_FRAGMENTS is used to prevent TLS-CBC-IV vulnerability described at <http://www.openssl.org/~bodo/tls-cbc.txt> It's known issue of TLSv1/SSLv3 but it attracts lots of attention these days as BEAST attack. (CVE-2011-3389)

Until now openssl sets OP\_ALL at SSLContext allocation and call SSL\_CTX\_set\_options at connection. SSL\_CTX\_set\_options updates the value by using |= so bits set by OP\_ALL cannot be unset afterwards.

This commit changes to call SSL\_CTX\_set\_options only 1 time for each SSLContext. It sets the specified value if SSLContext#options= are called and sets OP\_ALL if not.

To help users to unset bits in OP\_ALL, this commit also adds several constant to SSL such as OpenSSL::SSL::OP\_DONT\_INSERT\_EMPTY\_FRAGMENTS. These constants were not exposed in Ruby because there's no way to unset bits in OP\_ALL before.

Following is an example to enable 0/n split for BEAST prevention.

```
ctx.options = OP_ALL & ~OP_DONT_INSERT_EMPTY_FRAGMENTS
```

- test/openssl/test\_ssl.rb: Test above option exists.

#### Revision 34482 - 02/08/2012 05:27 AM - nahi (Hiroshi Nakamura)

- ext/openssl/openssl.c: Add SSL constants and allow to unset SSL option to prevent BEAST attack. See [Bug #5353].

In OpenSSL, OP\_DONT\_INSERT\_EMPTY\_FRAGMENTS is used to prevent TLS-CBC-IV vulnerability described at <http://www.openssl.org/~bodo/tls-cbc.txt> It's known issue of TLSv1/SSLv3 but it attracts lots of attention these days as BEAST attack. (CVE-2011-3389)

Until now openssl sets OP\_ALL at SSLContext allocation and call SSL\_CTX\_set\_options at connection. SSL\_CTX\_set\_options updates the value by using |= so bits set by OP\_ALL cannot be unset afterwards.

This commit changes to call SSL\_CTX\_set\_options only 1 time for each SSLContext. It sets the specified value if SSLContext#options= are called and sets OP\_ALL if not.

To help users to unset bits in OP\_ALL, this commit also adds several constant to SSL such as OpenSSL::SSL::OP\_DONT\_INSERT\_EMPTY\_FRAGMENTS. These constants were not exposed in Ruby because there's no way to unset bits in OP\_ALL before.

Following is an example to enable 0/n split for BEAST prevention.

```
ctx.options = OP_ALL & ~OP_DONT_INSERT_EMPTY_FRAGMENTS
```

- test/openssl/test\_ssl.rb: Test above option exists.

#### Revision 34482 - 02/08/2012 05:27 AM - nahi (Hiroshi Nakamura)

- ext/openssl/openssl.c: Add SSL constants and allow to unset SSL option to prevent BEAST attack. See [Bug #5353].

In OpenSSL, OP\_DONT\_INSERT\_EMPTY\_FRAGMENTS is used to prevent TLS-CBC-IV vulnerability described at <http://www.openssl.org/~bodo/tls-cbc.txt> It's known issue of TLSv1/SSLv3 but it attracts lots of attention these days as BEAST attack. (CVE-2011-3389)

Until now openssl sets OP\_ALL at SSLContext allocation and call SSL\_CTX\_set\_options at connection. SSL\_CTX\_set\_options updates the value by using |= so bits set by OP\_ALL cannot be unset afterwards.

This commit changes to call SSL\_CTX\_set\_options only 1 time for each SSLContext. It sets the specified value if SSLContext#options= are called and sets OP\_ALL if not.

To help users to unset bits in OP\_ALL, this commit also adds several constant to SSL such as OpenSSL::SSL::OP\_DONT\_INSERT\_EMPTY\_FRAGMENTS. These constants were not exposed in Ruby because there's no way to unset bits in OP\_ALL before.

Following is an example to enable 0/n split for BEAST prevention.

```
ctx.options = OP_ALL & ~OP_DONT_INSERT_EMPTY_FRAGMENTS
```

- test/openssl/test\_ssl.rb: Test above option exists.

#### Revision 34482 - 02/08/2012 05:27 AM - nahi (Hiroshi Nakamura)

- ext/openssl/openssl.c: Add SSL constants and allow to unset SSL option to prevent BEAST attack. See [Bug #5353].

In OpenSSL, OP\_DONT\_INSERT\_EMPTY\_FRAGMENTS is used to prevent TLS-CBC-IV vulnerability described at <http://www.openssl.org/~bodo/tls-cbc.txt> It's known issue of TLSv1/SSLv3 but it attracts lots of attention these days as BEAST attack. (CVE-2011-3389)

Until now openssl sets OP\_ALL at SSLContext allocation and call SSL\_CTX\_set\_options at connection. SSL\_CTX\_set\_options updates the value by using |= so bits set by OP\_ALL cannot be unset afterwards.

This commit changes to call SSL\_CTX\_set\_options only 1 time for each SSLContext. It sets the specified value if SSLContext#options= are called and sets OP\_ALL if not.

To help users to unset bits in OP\_ALL, this commit also adds several constant to SSL such as OpenSSL::SSL::OP\_DONT\_INSERT\_EMPTY\_FRAGMENTS. These constants were not exposed in Ruby because there's no way to unset bits in OP\_ALL before.

Following is an example to enable 0/n split for BEAST prevention.

```
ctx.options = OP_ALL & ~OP_DONT_INSERT_EMPTY_FRAGMENTS
```

- test/openssl/test\_ssl.rb: Test above option exists.

#### Revision 0dea8a71 - 02/08/2012 05:57 AM - nahi (Hiroshi Nakamura)

Backport r34482 from trunk. See #5353

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby\_1\_8@34485 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

#### Revision 2cb7a6c0 - 02/08/2012 06:09 AM - nahi (Hiroshi Nakamura)

Backport r34482 from trunk. See #5353

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby\_1\_8\_7@34486 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

**Revision 0234bcfd - 02/09/2012 05:04 PM - emboss**

- backport r34482 from trunk
- ext/openssl/openssl.c: Add SSL constants and allow to unset SSL option to prevent BEAST attack. See [Bug #5353].

In OpenSSL, OP\_DONT\_INSERT\_EMPTY\_FRAGMENTS is used to prevent TLS-CBC-IV vulnerability described at <http://www.openssl.org/~bodo/tls-cbc.txt> It's known issue of TLSv1/SSLv3 but it attracts lots of attention these days as BEAST attack. (CVE-2011-3389)

Until now openssl sets OP\_ALL at SSLContext allocation and call SSL\_CTX\_set\_options at connection. SSL\_CTX\_set\_options updates the value by using |= so bits set by OP\_ALL cannot be unset afterwards. This commit changes to call SSL\_CTX\_set\_options only 1 time for each SSLContext. It sets the specified value if SSLContext#options= are called and sets OP\_ALL if not.

To help users to unset bits in OP\_ALL, this commit also adds several constant to SSL such as OpenSSL::SSL::OP\_DONT\_INSERT\_EMPTY\_FRAGMENTS. These constants were not exposed in Ruby because there's no way to unset bits in OP\_ALL before.

Following is an example to enable 0/n split for BEAST prevention.

```
ctx.options = OP_ALL & ~OP_DONT_INSERT_EMPTY_FRAGMENTS
```

- test/openssl/test\_ssl.rb: Test above option exists.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby\_1\_9\_3@34524 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

**Revision 85fd9aad - 02/09/2012 05:20 PM - emboss**

- backport r34482 from trunk
- ext/openssl/openssl.c: Add SSL constants and allow to unset SSL option to prevent BEAST attack. See [Bug #5353].

In OpenSSL, OP\_DONT\_INSERT\_EMPTY\_FRAGMENTS is used to prevent TLS-CBC-IV vulnerability described at <http://www.openssl.org/~bodo/tls-cbc.txt> It's known issue of TLSv1/SSLv3 but it attracts lots of attention these days as BEAST attack. (CVE-2011-3389)

Until now openssl sets OP\_ALL at SSLContext allocation and call SSL\_CTX\_set\_options at connection. SSL\_CTX\_set\_options updates the value by using |= so bits set by OP\_ALL cannot be unset afterwards. This commit changes to call SSL\_CTX\_set\_options only 1 time for each SSLContext. It sets the specified value if SSLContext#options= are called and sets OP\_ALL if not.

To help users to unset bits in OP\_ALL, this commit also adds several constant to SSL such as OpenSSL::SSL::OP\_DONT\_INSERT\_EMPTY\_FRAGMENTS. These constants were not exposed in Ruby because there's no way to unset bits in OP\_ALL before.

Following is an example to enable 0/n split for BEAST prevention.

```
ctx.options = OP_ALL & ~OP_DONT_INSERT_EMPTY_FRAGMENTS
```

- test/openssl/test\_ssl.rb: Test above option exists.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby\_1\_9\_2@34525 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

**Revision 84f1dae9 - 12/18/2012 02:02 AM - emboss**

- ext/openssl/lib/ssl.rb: Enable insertion of empty fragments as a countermeasure for the BEAST attack by default. The default options of OpenSSL::SSL::SSLContext are now: OpenSSL::SSL::OP\_ALL & ~OpenSSL::SSL::OP\_DONT\_INSERT\_EMPTY\_FRAGMENTS [Bug #5353] [ruby-core:39673]
- test/openssl/test\_ssl.rb: Adapt tests to new SSLContext default.
- NEWS: Announce the new default.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@38433 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

**Revision 38433 - 12/18/2012 02:02 AM - emboss**

- ext/openssl/lib/ssl.rb: Enable insertion of empty fragments as a countermeasure for the BEAST attack by default. The default options of OpenSSL::SSL::SSLContext are now: OpenSSL::SSL::OP\_ALL & ~OpenSSL::SSL::OP\_DONT\_INSERT\_EMPTY\_FRAGMENTS [Bug #5353] [ruby-core:39673]
- test/openssl/test\_ssl.rb: Adapt tests to new SSLContext default.
- NEWS: Announce the new default.

**Revision 38433 - 12/18/2012 02:02 AM - emboss**

- ext/openssl/lib/ssl.rb: Enable insertion of empty fragments as a countermeasure for the BEAST attack by default. The default options of OpenSSL::SSL::SSLContext are now: OpenSSL::SSL::OP\_ALL & ~OpenSSL::SSL::OP\_DONT\_INSERT\_EMPTY\_FRAGMENTS [Bug #5353] [ruby-core:39673]
- test/openssl/test\_ssl.rb: Adapt tests to new SSLContext default.
- NEWS: Announce the new default.

**Revision 38433 - 12/18/2012 02:02 AM - emboss**

- ext/openssl/lib/ssl.rb: Enable insertion of empty fragments as a countermeasure for the BEAST attack by default. The default options of OpenSSL::SSL::SSLContext are now: OpenSSL::SSL::OP\_ALL & ~OpenSSL::SSL::OP\_DONT\_INSERT\_EMPTY\_FRAGMENTS [Bug #5353] [ruby-core:39673]
- test/openssl/test\_ssl.rb: Adapt tests to new SSLContext default.
- NEWS: Announce the new default.

**Revision 38433 - 12/18/2012 02:02 AM - emboss**

- ext/openssl/lib/ssl.rb: Enable insertion of empty fragments as a countermeasure for the BEAST attack by default. The default options

of OpenSSL::SSL::SSLContext are now:  
OpenSSL::SSL::OP\_ALL & ~OpenSSL::SSL::OP\_DONT\_INSERT\_EMPTY\_FRAGMENTS  
[Bug #5353] [ruby-core:39673]

- test/openssl/test\_ssl.rb: Adapt tests to new SSLContext default.
- NEWS: Announce the new default.

#### Revision 38433 - 12/18/2012 02:02 AM - emboss

- ext/openssl/lib/ssl.rb: Enable insertion of empty fragments as a countermeasure for the BEAST attack by default. The default options of OpenSSL::SSL::SSLContext are now:  
OpenSSL::SSL::OP\_ALL & ~OpenSSL::SSL::OP\_DONT\_INSERT\_EMPTY\_FRAGMENTS  
[Bug #5353] [ruby-core:39673]
- test/openssl/test\_ssl.rb: Adapt tests to new SSLContext default.
- NEWS: Announce the new default.

#### Revision 38433 - 12/18/2012 02:02 AM - emboss

- ext/openssl/lib/ssl.rb: Enable insertion of empty fragments as a countermeasure for the BEAST attack by default. The default options of OpenSSL::SSL::SSLContext are now:  
OpenSSL::SSL::OP\_ALL & ~OpenSSL::SSL::OP\_DONT\_INSERT\_EMPTY\_FRAGMENTS  
[Bug #5353] [ruby-core:39673]
- test/openssl/test\_ssl.rb: Adapt tests to new SSLContext default.
- NEWS: Announce the new default.

## History

---

### #1 - 09/23/2011 09:29 AM - Anonymous

-----BEGIN PGP SIGNED MESSAGE-----  
Hash: SHA1

(2011/09/23 1:14), Martin Bosslet wrote:

A well-known vulnerability of TLS v1.0 and earlier has recently gained some attention:

[http://www.theregister.co.uk/2011/09/19/beast\\_exploits\\_paypal\\_ssl/](http://www.theregister.co.uk/2011/09/19/beast_exploits_paypal_ssl/)

I think the thread here would be better than media articles.  
<http://www.ietf.org/mail-archive/web/tls/current/msg08032.html>

My current BEAST understanding is: "TLS/SSL CBC IV chaining + victim/attacker multiplexed onto a single TLS/SSL connection on Browser (SSL client side) + CPA(Chosen-plaintext Attack)" but we should wait the conference session today. Done already?

For existing TLS/SSL + CBC IV vuln issue, I rarely set SSL\_OP\_DONT\_INSERT\_EMPTY\_FRAGMENTS since clients I write don't allow CPA by attacker. In openssl, when an attacker can have the same SSLSession object with a victim, the attacker can sniff plaintext easier in another way. I do the same for servers.

But yeah, using this option correctly must be hard for Ruby users. It would be better to turn the SSL\_OP\_DONT\_INSERT\_EMPTY\_FRAGMENTS bit off by default. We might get some claims, but we can explain the reason.

What do you propose? Should we solve this before the 1.9.3 release?

Let's wait the session and see how other SSL clients (mainly Browsers) and SSL servers(OpenSSL project) reacts.

```
// NaHi
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.4.9 (Cygwin)

iQEcBAEBAgAGBQJOe9GyAAoJEC7N6P3yLb12yGUH/0BWS2Fvzpvuy22ul9uQPpyBC
Jocp+T+UeuJDZVxf0qzAbI7TLKCH8iVbA16nsy5LmH9Dq41mzJwPn8o0hmCaQXOu
UZh8MFp4T9VfDZlIF/3RwYB35amGrrSr5xc4IxQ60o2GhlutllrU6ZfrqUG7FJY
kEty4pnAba2e4fpwgVIA/1K7R+0QJe37fRhvzQ3DGIIxBNbGso3L8zfCmanck4N2
9hP2ftMyeFhb199+kaB9IKfyYzwKIPIKLRdmAxTORzllu0INRMzgnUoddHDIbixi
B6E1TV2B1Cfh0p07sP3gTZyykaZLQfNuEcXLA6PohHv3asnYEz3ddWZJGU1lxU=
=fwTo
-----END PGP SIGNATURE-----
```

**#2 - 09/23/2011 09:53 AM - Anonymous**

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1
```

(2011/09/23 9:25), Hiroshi Nakamura wrote:

For existing TLS/SSL + CBC IV vuln issue, I rarely set  
SSL\_OP\_DONT\_INSERT\_EMPTY\_FRAGMENTS since clients I write don't  
allow

Must be "I rarely unset", I meant "I always use SSL\_OP\_ALL". Using  
'NOT' in flag is harmful :)

And additional note: I'm not a cryptographer!

```
// NaHi
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.4.9 (Cygwin)

iQEcBAEBAgAGBQJOe9K0AAoJEC7N6P3yLb12ZNsH/0wYReyiGO/nolxMvzP5L6u
OI4gRhX3pdJMynXf5xCfSSVddVqKh9WfuuwT5OYa6wuxsoJNkR3fygBAsUmyCqo
+6B1ChN6o/lnpYcoLUky6yig8tzMRwrJFi+Q2IYwbngBWQhTYHI2OVC702/nwz57
CL+cn1kmZOXwSxc2D8phEOI5O3yvrhTjHoLCuLU22XAH52Lzdu99cjXvqYO6m8XK
mY/JX9E9quKc5lQcLwiCXtpbzZmC8Psw7l07ewW7cyQ7me0A3iMh+llwBHhvcL+
PieWB8kbFYCNIFYwf76X8cW07YySdWlsCqD+jQfzLbpfHpbxfWfuuXO4nC56ZSM=
=mgoe
-----END PGP SIGNATURE-----
```

**#3 - 09/24/2011 08:44 PM - MartinBosslet (Martin Bosslet)**

Some first reactions:

[http://www.educatedguesswork.org/2011/09/security\\_impact\\_of\\_the\\_rizzodu.html](http://www.educatedguesswork.org/2011/09/security_impact_of_the_rizzodu.html)  
<http://www.imperialviolet.org/2011/09/23/chromeandbeast.html>

From what I understand this is really sweet, instead of trying to guess a whole block at a time they play with block boundaries so that they effectively only have to guess one byte at a time instead of let's say 16.

And it looks like turning off SSL\_OP\_DONT\_INSERT\_EMPTY\_FRAGMENTS really does prevent this kind of attack, too. But then again, as nahi already hinted at, mounting this kind of attack requires quite some sophistication, usually there are often easier ways for an attacker.

An interesting approach that wouldn't break compatibility seems to be what is currently investigated for Chrome:

<http://codereview.chromium.org/7621002>

Instead of sending a totally empty first record they send one with exactly one byte to get the same effect of randomizing the IV.

Regards,  
Martin

PS: I would be really grateful if somebody got their hands on the original paper and could post a link here or send it to me!

**#4 - 09/26/2011 04:53 PM - nahi (Hiroshi Nakamura)**

- ruby -v changed from trunk to -

-----BEGIN PGP SIGNED MESSAGE-----

Hash: SHA1

On 09/24/2011 08:44 PM, Martin Bosslet wrote:

[http://www.educatedguesswork.org/2011/09/security\\_impact\\_of\\_the\\_rizzodu.html](http://www.educatedguesswork.org/2011/09/security_impact_of_the_rizzodu.html)  
<http://www.imperialviolet.org/2011/09/23/chromeandbeast.html>

From what I understand this is really sweet, instead of trying to guess a whole block at a time they play with block boundaries so that they effectively only have to guess one byte at a time instead of let's say 16.

Agreed. Wise and pragmatic :)

And it looks like turning off `SSL_OP_DONT_INSERT_EMPTY_FRAGMENTS` really does prevent this kind of attack, too. But then again, as nahi already hinted at, mounting this kind of attack requires quite some sophistication, usually there are often easier ways for an attacker.

Some fix needed especially for clients but for now it should be fixed at client side, and we should wait how OpenSSL treats this issue.

I would say that it's not a blocker for 1.9.3.

An interesting approach that wouldn't break compatibility seems to be what is currently investigated for Chrome:

<http://codereview.chromium.org/7621002>

Instead of sending a totally empty first record they send one with exactly one byte to get the same effect of randomizing the IV.

Yeah, if I understand the attack correctly, with this vulnerability, an attacker can try to guess a plain text only as the first block of CBC chain. And the above NSS patch reduces the range to 1 byte, and OpenSSL's empty fragment patch reduces it to 0 byte. It's wise and pragmatic, too. :) I wish the 1-byte patch is proven to be safe from compatibility point of view...

// NaHi

-----BEGIN PGP SIGNATURE-----

Version: GnuPG v1.4.11 (GNU/Linux)

iQEcBAEBAgAGBQJOgC8DAAoJEC7N6P3yLbI2Du8H/A88MBS3BCdDFjDzWtWgfntY  
5keNOMZZ+Z5syTKURtCLRqRHRMfvqzdfB83oSVsDXnkwTSacGW2OYKX59z6HezO  
Hf7rap9ozniFmXjUw0YsJOVUuNOL3NYbKzeK/O8Ycn//Yelw7ZQNPsb0vg4vgzwaZ  
RVaEps13WWRI3M0lfQ+wI9vHbCnL1kgJmc+Q+vYQ/cUW0k4RBEWrXZ9lQUk97+8  
42GS/ZRWI8nRK0VEVAYBY/zdD9oukdbwhW+cxol5Sx4blRgVyB6uoqpevd8rXliU  
h8jo7NEDx6o/HxgT4Jy/20CD5aHrT7N42ZumE8P0jgM0m5liR+6++IYfcMvznWg=  
=84SS

-----END PGP SIGNATURE-----

**#5 - 02/08/2012 02:27 PM - nahi (Hiroshi Nakamura)**

- Status changed from Open to Closed

- % Done changed from 0 to 100

This issue was solved with changeset [r34482](#).

Martin, thank you for reporting this issue.

Your contribution to Ruby is greatly appreciated.

May Ruby be with you.

- ext/openssl/openssl.c: Add SSL constants and allow to unset SSL option to prevent BEAST attack. See [Bug #5353].

In OpenSSL, OP\_DONT\_INSERT\_EMPTY\_FRAGMENTS is used to prevent TLS-CBC-IV vulnerability described at <http://www.openssl.org/~bodo/tls-cbc.txt>  
It's known issue of TLSv1/SSLv3 but it attracts lots of attention these days as BEAST attack. (CVE-2011-3389)

Until now openssl sets OP\_ALL at SSLContext allocation and call SSL\_CTX\_set\_options at connection. SSL\_CTX\_set\_options updates the value by using |= so bits set by OP\_ALL cannot be unset afterwards.

This commit changes to call SSL\_CTX\_set\_options only 1 time for each SSLContext. It sets the specified value if SSLContext#options= are called and sets OP\_ALL if not.

To help users to unset bits in OP\_ALL, this commit also adds several constant to SSL such as OpenSSL::SSL::OP\_DONT\_INSERT\_EMPTY\_FRAGMENTS. These constants were not exposed in Ruby because there's no way to unset bits in OP\_ALL before.

Following is an example to enable 0/n split for BEAST prevention.

```
ctx.options = OP_ALL & ~OP_DONT_INSERT_EMPTY_FRAGMENTS
```

- test/openssl/test\_ssl.rb: Test above option exists.

#### #6 - 02/08/2012 02:30 PM - nahi (Hiroshi Nakamura)

- Status changed from Closed to Open

Should have written 'See #5353' not 'See [Bug #5353]'. I don't like machinery autoclosing. :(

#### #7 - 02/08/2012 03:10 PM - nahi (Hiroshi Nakamura)

Backported to ruby\_1\_8 and ruby\_1\_8\_7 by r34485 and r34486 respectively.

#### #8 - 02/10/2012 02:24 AM - MartinBosslet (Martin Bosslet)

Backported to ruby\_1\_9\_3 in r34524 and to ruby\_1\_9\_2 in r34525.

#### #9 - 02/10/2012 10:13 PM - nahi (Hiroshi Nakamura)

At first, I misunderstood the message from Martin that he just want to turn off the flag by default. I thought we can turn off the SSL\_OP\_DONT\_INSERT\_EMPTY\_FRAGMENTS flag if we want.

Based on Apple's report at January, I realized that we didn't offer the feature from the beginning (I confirmed it to Gotoyuzo, the author of original code.) So we added the feature. Please see the linked commit for more detail.

The original proposal from Martin, turning off the SSL\_OP\_DONT\_INSERT\_EMPTY\_FRAGMENTS bit by default, is still open.

#### #10 - 02/23/2012 08:42 PM - MartinBosslet (Martin Bosslet)

Hiroshi Nakamura wrote:

The original proposal from Martin, turning off the SSL\_OP\_DONT\_INSERT\_EMPTY\_FRAGMENTS bit by default, is still open.

Yes, to follow up on this: it remains to decide how this should be handled in libraries that use OpenSSL::SSL, such as Net::HTTP. In Net::HTTP's case (and I could imagine probably in most of the other cases, too), the SSLContext object is not directly accessible, so we can't configure 0/n splitting there now.

Two paths could be chosen to enable the functionality. Either patching each of the libraries by offering some way to configure 0/n splitting - or we could simply make 0/n splitting the default. The latter would only require one central change, but bears the potential to break existing installations.

Generally we are in favor of staying as compatible as possible for 2.0, but it would also mean that things

like the "BEAST" attack will remain feasible in the future.  
So should we make this the default in trunk? The time until 2.0 gets released should give incompatible setups enough time to patch their environment?

**#11 - 03/11/2012 03:51 PM - ko1 (Koichi Sasada)**

- Status changed from Open to Assigned
- Assignee set to nahi (Hiroshi Nakamura)

**#12 - 11/29/2012 11:14 PM - nahi (Hiroshi Nakamura)**

- Assignee changed from nahi (Hiroshi Nakamura) to MartinBosslet (Martin Bosslet)

=begin  
This could be an option:

Index: test/openssl/test\_ssl.rb

=====

```
--- test/openssl/test_ssl.rb (revision 37996)
+++ test/openssl/test_ssl.rb (working copy)
@@ -257,7 +257,7 @@
ctx = OpenSSL::SSL::SSLContext.new
ctx.set_params
assert_equal(OpenSSL::SSL::VERIFY_PEER, ctx.verify_mode)
```

- assert\_equal(OpenSSL::SSL::OP\_ALL, ctx.options)

- [REDACTED]

```
ciphers = ctx.ciphers
ciphers_versions = ciphers.collect{|_, v, _, _| v }
ciphers_names = ciphers.collect{|v, _, _, _| v }
```

```
@@ -397,6 +397,7 @@
end
```

```
def test_unset_OP_ALL
```

- # Can we safely assume every env has OP\_DONT\_INSERT\_EMPTY\_FRAGMENTS?  
ctx\_proc = Proc.new { |ctx|  
ctx.options = OpenSSL::SSL::OP\_ALL & ~OpenSSL::SSL::OP\_DONT\_INSERT\_EMPTY\_FRAGMENTS  
}

## Index: ext/openssl/lib/openssl/ssl.rb

```
--- ext/openssl/lib/openssl/ssl.rb (revision 37996)
+++ ext/openssl/lib/openssl/ssl.rb (working copy)
@@ -24,7 +24,9 @@
:ssl_version => "SSLv23",
:verify_mode => OpenSSL::SSL::VERIFY_PEER,
:ciphers => "ALL:!ADH:!EXPORT:!SSLv2:RC4+RSA:+HIGH:+MEDIUM:+LOW",
```

- [REDACTED]
- [REDACTED]
- [REDACTED]
- [REDACTED]

```
DEFAULT_CERT_STORE = OpenSSL::X509::Store.new
```

...but it causes connection problem for clients, that normally not affected by BEAST. I'll update WEBrick to disable the bit.

Martin, please close this issue if you're OK. WEBrick thing is a different problem.  
=end

**#13 - 12/18/2012 11:02 AM - Anonymous**

- Status changed from Assigned to Closed

This issue was solved with changeset [r38433](#).  
Martin, thank you for reporting this issue.  
Your contribution to Ruby is greatly appreciated.  
May Ruby be with you.

---

- ext/openssl/lib/ssl.rb: Enable insertion of empty fragments as a countermeasure for the BEAST attack by default. The default options of OpenSSL::SSL::SSLContext are now: OpenSSL::SSL::OP\_ALL & ~OpenSSL::SSL::OP\_DONT\_INSERT\_EMPTY\_FRAGMENTS [Bug [#5353](#)] [ruby-core:39673]
- test/openssl/test\_ssl.rb: Adapt tests to new SSLContext default.
- NEWS: Announce the new default.