Ruby - Bug #466

test_str_crypt(TestM17NComb) failed

08/21/2008 08:19 PM - znz (Kazuhiro NISHIYAMA)

Status:	Closed		
Priority:	Normal		
Assignee:	znz (Kazuhiro NISHIYAMA)		
Target version:			
ruby -v:	1.9.0	**Backport:**	

Description

```
$ ruby-trunk -v
ruby 1.9.0 (2008-08-21 revision 18753) [powerpc-darwin9.4.0]
```

の環境でtest_str_crypt(TestM17NComb)がFailureになります。

```
$ ruby-trunk test/ruby/test_m17n_comb.rb -v -n /crypt/
Loaded suite test/ruby/test_m17n_comb
Started
test_str_crypt(TestM17NComb): F

Finished in 0.03673 seconds.

  1) Failure:
test_str_crypt(TestM17NComb)
    [test/ruby/test_m17n_comb.rb:800:in `block in test_str_crypt'
     /Users/nishiyamakazuhiro/wc/ruby-lang/trunk/test/ruby/allpairs.rb:83:in `block in each'
     /Users/nishiyamakazuhiro/wc/ruby-lang/trunk/test/ruby/allpairs.rb:75:in `block in each_index'
     /Users/nishiyamakazuhiro/wc/ruby-lang/trunk/test/ruby/allpairs.rb:46:in `block in make_large_
block'
     /Users/nishiyamakazuhiro/wc/ruby-lang/trunk/test/ruby/allpairs.rb:26:in `block (2 levels) in
make_basic_block'
     /Users/nishiyamakazuhiro/wc/ruby-lang/trunk/test/ruby/allpairs.rb:21:in `times'
     /Users/nishiyamakazuhiro/wc/ruby-lang/trunk/test/ruby/allpairs.rb:21:in `block in make_basic_
block'
     /Users/nishiyamakazuhiro/wc/ruby-lang/trunk/test/ruby/allpairs.rb:20:in `times'
     /Users/nishiyamakazuhiro/wc/ruby-lang/trunk/test/ruby/allpairs.rb:20:in `make_basic_block'
     /Users/nishiyamakazuhiro/wc/ruby-lang/trunk/test/ruby/allpairs.rb:45:in `make_large_block'
     /Users/nishiyamakazuhiro/wc/ruby-lang/trunk/test/ruby/allpairs.rb:71:in `each_index'
     /Users/nishiyamakazuhiro/wc/ruby-lang/trunk/test/ruby/allpairs.rb:82:in `each'
     test/ruby/test_m17n_comb.rb:118:in `combination'
     test/ruby/test_m17n_comb.rb:794:in `test_str_crypt']:
"".force_encoding("ASCII-8BIT").crypt("\xE0\xA0\xA1".force_encoding("UTF-8")).
<"\xE0\xA0fT7zdRv9Y7A"> expected but was
<"\xE0\xA0swiH3o6yAu2">.

1 tests, 55 assertions, 1 failures, 0 errors
$

$ ruby-trunk -ve '3.times{p "".crypt("\xE0\xA0")}'                              ruby 1.9.0 (2008-0
8-21 revision 18753) [powerpc-darwin9.4.0]
"\xE0\xA0X8NBuQ4l6uQ"
"\xE0\xA0fT7zdRv9Y7A"
"\xE0\xA0fT7zdRv9Y7A"
$
```

のように2回目以降で結果が違うのが原因のようです。

直接crypt(2)を同じ引数で呼んでも同じ結果になります。

```
$ cat a.c

#include <stdio.h>
```

```
#include <unistd.h>

int main()
{
         printf("%s\n", crypt("", "\xE0\xA0"));
         printf("%s\n", crypt("", "\xE0\xA0"));
         printf("%s\n", crypt("", "\xE0\xA0"));
         return 0;
}

$ gcc a.c
$ ./a.out |LANG=C cat -v
M-`M- X8NBuQ4l6uQ
M-`M- fT7zdRv9Y7A
M-`M- fT7zdRv9Y7A
```

Related issues:

Related to Ruby - Feature #4042: String#crypt shoud not accepted "\x00" as a ...	**Rejected**	**11/11/2010**

History

#1 - 09/03/2008 06:52 PM - ko1 (Koichi Sasada)

- Assignee set to naruse (Yui NARUSE)

#2 - 09/03/2008 07:01 PM - naruse (Yui NARUSE)

- Category set to core

- Assignee changed from naruse (Yui NARUSE) to znz (Kazuhiro NISHIYAMA)

M17N でなくライブラリ側の問題というのと、妥当な解決策の判断がわたしにはつけられないので、
とりあえず担当を西山さんに預けます。

#3 - 09/07/2008 12:45 PM - znz (Kazuhiro NISHIYAMA)

以下のようにconfigureでチェックしてしまうのがいいかと思ったのですが、
どうでしょうか？

影響範囲がよくわからなかったので、実際に値が変わってしまうことがあった
darwinの時だけチェックするようなパッチにしてみました。

```
Index: configure.in
===================================================================
--- configure.in (revision 19208)
+++ configure.in (working copy)
@@ -523,6 +523,26 @@
       AC_DEFINE(BROKEN_SETREUID, 1)
       AC_DEFINE(BROKEN_SETREGID, 1)
       ])
+  ac_cv_lib_crypt_crypt=no
+                AC_CACHE_CHECK(for broken crypt with 8bit chars, rb_cv_broken_crypt,
+                    [AC_TRY_RUN([
+#include <stdio.h>
+#include <unistd.h>
+#include <string.h>
+int
+main()
+{
+   char buf[256];
+   strcpy(buf, crypt("", "\xE0\xA0"));
+   return strcmp(buf, crypt("", "\xE0\xA0"));
+}
+],
+             rb_cv_broken_crypt=no,
+             rb_cv_broken_crypt=yes,
+             rb_cv_broken_crypt=yes)])
+                if test "$rb_cv_broken_crypt" = yes; then
+                  AC_DEFINE(BROKEN_CRYPT, 1)
+                fi
   ;;
 hpux*)  LIBS="-lm $LIBS"
   ac_cv_c_inline=no;;
Index: string.c
```

```
===================================================================
--- string.c (revision 19208)
+++ string.c (working copy)
@@ -5862,6 +5862,10 @@
     extern char *crypt(const char *, const char *);
     VALUE result;
     const char *s;
+#ifdef BROKEN_CRYPT
+    VALUE salt_8bit_clean;
+    rb_encoding *enc;
+#endif

     StringValue(salt);
     if (RSTRING_LEN(salt) < 2)
@@ -5869,7 +5873,18 @@

     if (RSTRING_PTR(str)) s = RSTRING_PTR(str);
     else s = "";
+#ifdef BROKEN_CRYPT
+    salt_8bit_clean = rb_str_dup(salt);
+    enc = rb_ascii8bit_encoding();
+    str_modifiable(salt_8bit_clean);
+    rb_enc_associate(salt_8bit_clean, enc);
+    salt_8bit_clean = rb_str_tr(salt_8bit_clean,
+                                rb_enc_str_new("\x80-\xFF", 3, enc),
+                                rb_usascii_str_new("\x00-\x7F", 3));
+    result = rb_str_new2(crypt(s, RSTRING_PTR(salt_8bit_clean)));
+#else
     result = rb_str_new2(crypt(s, RSTRING_PTR(salt)));
+#endif
     OBJ_INFECT(result, str);
     OBJ_INFECT(result, salt);
     return result;
```

#4 - 09/07/2008 01:05 PM - matz (Yukihiro Matsumoto)

まつもと ゆきひろです

In message "Re: [ruby-dev:36193] [Bug #466] test_str_crypt(TestM17NComb) failed"
on Sun, 7 Sep 2008 12:39:11 +0900, Kazuhiro NISHIYAMA redmine@ruby-lang.org writes:

> 以下のようにconfigureでチェックしてしまうのがいいかと思ったのですが、
> どうでしょうか？
>
> 影響範囲がよくわからなかったので、実際に値が変わってしまうことがあった
> darwinの時だけチェックするようなパッチにしてみました。

コミットしてくださいませんか?

#5 - 09/07/2008 06:07 PM - Anonymous

- *Status changed from Open to Closed*

- *% Done changed from 0 to 100*

Applied in changeset r19213.

#6 - 09/08/2008 10:53 AM - nobu (Nobuyoshi Nakada)

なかだです。

At Sun, 7 Sep 2008 12:39:11 +0900,
Kazuhiro NISHIYAMA wrote in [ruby-dev:36193]:

> 以下のようにconfigureでチェックしてしまうのがいいかと思ったのですが、
> どうでしょうか？
>
> 影響範囲がよくわからなかったので、実際に値が変わってしまうことがあった
> darwinの時だけチェックするようなパッチにしてみました。

現在のライブラリではたまたま毎回結果が変わっていますが、つねにそ
うなるとは限らないのではないでしょうか。毎回同じ正しくない結果を
返すという可能性もあるわけで。

"$1"□"$2"□□□□salt□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□8bit□□□□□□□□□□□□□□□□□□□
□□□

```
Index: string.c
===================================================================
--- string.c (revision 19215)
+++ string.c (working copy)
@@ -5862,8 +5862,7 @@ rb_str_crypt(VALUE str, VALUE salt)
     extern char *crypt(const char *, const char *);
     VALUE result;
-    const char *s;
+    const char *s, *saltp;
 #ifdef BROKEN_CRYPT
-    VALUE salt_8bit_clean;
-    rb_encoding *enc;
+    char salt_8bit_clean[3];
 #endif

@@ -5872,18 +5871,14 @@ rb_str_crypt(VALUE str, VALUE salt)
  rb_raise(rb_eArgError, "salt too short (need >=2 bytes)");

-    if (RSTRING_PTR(str)) s = RSTRING_PTR(str);
-    else s = "";
-#ifdef BROKEN_CRYPT
-    salt_8bit_clean = rb_str_dup(salt);
-    enc = rb_ascii8bit_encoding();
-    str_modifiable(salt_8bit_clean);
-    rb_enc_associate(salt_8bit_clean, enc);
-    salt_8bit_clean = rb_str_tr(salt_8bit_clean,
-                                rb_enc_str_new("\x80-\xFF", 3, enc),
-                                rb_usascii_str_new("\x00-\x7F", 3));
-    result = rb_str_new2(crypt(s, RSTRING_PTR(salt_8bit_clean)));
-#else
-    result = rb_str_new2(crypt(s, RSTRING_PTR(salt)));
-#endif
+    s = RSTRING_PTR(str);
+    if (!s) s = "";
+    saltp = RSTRING_PTR(salt);
+    if (!ISASCII((unsigned char)saltp[0]) || !ISASCII((unsigned char)saltp[1])) {
+ salt_8bit_clean[0] = saltp[0] & 0x7f;
+ salt_8bit_clean[1] = saltp[2] & 0x7f;
+ salt_8bit_clean[2] = '\0';
+ saltp = salt_8bit_clean;
+    }
+    result = rb_str_new2(crypt(s, salts));
     OBJ_INFECT(result, str);
     OBJ_INFECT(result, salt);
```

--- □□□□Bug□□□□
--- □□□□□Bug□□□□□
□□ □□

#7 - 09/08/2008 07:30 PM - znz (Kazuhiro NISHIYAMA)

□□□□□□□

At Mon, 8 Sep 2008 10:46:45 +0900,
Nobuyoshi Nakada wrote:

> □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
> □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
> □□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□

> "$1"□"$2"□□□□salt□□□□□□□□□□□□□□□□□□□
> □□□□□□□□□□□□8bit□□□□□□□□□□□□□□□□□□□
> □□□

8□□□□□□□□□□□□□□□□□"$1"□"$2"□□□□□□□□□

ことを考えていなかったのですが、気にしなくてもいいのでしょうか?

```
+     result = rb_str_new2(crypt(s, salts));
```

このsaltsをsaltpに変更して、テストが通ることを確認しました。

--
|ZnZ(ゼット エヌ ゼット)
|西山和広(Kazuhiro NISHIYAMA)

#8 - 09/09/2008 02:13 AM - nobu (Nobuyoshi Nakada)

なかだです。

At Mon, 8 Sep 2008 19:23:47 +0900,
Kazuhiro NISHIYAMA wrote in [ruby-dev:36213]:

> > 現在のライブラリではたまたま毎回結果が変わっていますが、つねにそうなるとは限らないのではないでしょうか。毎回同じ正しくない結果を返すという可能性もあるわけで。
>
> 毎回同じ正しくない結果が返ってくるなら、それはそれで問題がおきなさそうな気がしますが。

問題がありそうなのは、他のシステムで暗号化されたものをもってきた場合でしょうね。

> > "$1"や"$2"で始まるsaltでアルゴリズムを選択するという拡張機能のあるシステムもありますが、8bit目を使った拡張というのはないと思われます。
>
> 8ビット目をクリアした結果がたまたま"$1"や"$2"になっていたときのことを考えていなかったのですが、気にしなくてもいいのでしょうか?

crypt(3)によるとsaltとして有効なのは[A-Za-z0-9./]ということだったので無視されるかと思ったのですが、glibcのcrypt()では"\xc1\xc1"と"\x41\x41"でも異なる結果を返すので、単にマスクしてしまうのも問題がありそうです。

--- 僕の前にBugはない。
--- 僕の後ろにBugはできる。
中田 伸悦

#9 - 12/22/2014 10:45 PM - nobu (Nobuyoshi Nakada)

- Description updated

- ruby -v set to 1.9.0