# Ruby - Bug #5374

## Weird SecurityError with ruby1.9, File.stat/Dir.glob and $SAFE=1

09/28/2011 03:59 AM - 375gnu (Hleb  Valoshka)

| | | | |
|---|---|---|---|
| **Status:** | Closed | | |
| **Priority:** | Normal | | |
| **Assignee:** | mame (Yusuke Endoh) | | |
| **Target version:** | 2.0.0 | | |
| **ruby -v:** | ruby 1.9.2p180 (2011-02-18) [i386-mingw32] | **Backport:** | |

**Description**

Preface.

I've tried to find workaroud for one GetText-Ruby bug with untainted data from Dir.glob ( http://rubyforge.org//tracker/?func=detail&atid=3377&aid=28336&group_id=855).

Here it is (full text is in gettext-test.rb):

```
module GetText
class MOFile
alias :oldload :load
def load(arg)
arg = arg.dup.untaint if arg.kind_of? String
oldload(arg)
end
end
end
```

It works fine with ruby 1.8, but with 1.9 with debug enabled there is a
message about exception SecurityError:

```
Exception `SecurityError' at /usr/lib/ruby/1.9.1/gettext/runtime/mofile.rb:75 - Insecure operation - stat
```

The corresponding code is

```
74  begin
75    st = File.stat(arg)
76    @last_modified = [st.ctime, st.mtime]
77  rescue Exception
78  end
```

I've put line
warn "$SAFE == #{$SAFE}; arg.tainted? == #{arg.tainted?}"
before it, and it says:

```
$SAFE == 1; arg.tainted? == false
```

So why the exception is if arg is NOT tainted? Note: it was discovered on Debian
GNU/Linux box with 1.9.3preview1. Full log is in gettext-debian.log

Going further.

I've made very simple test program which mimics GetTExt-Ruby and workaround for
it, see test.rb in attachment.

This program was tested on Win32 box with 1.9.2-p180 and -p290.

Been run as "ruby -T1 test.rb u" output was clean. But been run as "ruby -T1
test.rb t" or "ruby -T1 test.rb t" is had an exception on files test1234.txt
and test12345.txt (see full test.log in attachment). 't' means "send tainted
object to function", 'u' means "send untainted", 'b' means "send tainted, then
untainted". But on Debian box it outputs NO error.

At last, I have run test for GetText on win box, and it failed with exception

Exception `SecurityError' at C:/fsc.tmp/gettext/runtime/locale_path.rb:90 - Insecure operation - glob

Log is in gettext-win.log

But whether Dir.glob is insecure with $SAFE==1?

## Associated revisions

**Revision d7444332257a6fb255300ee938a20dfa64cccf32 - 07/12/2012 02:44 AM - nobu (Nobuyoshi Nakada)**

rb_str_new_frozen: new object if tainted/untrusted unmatch

- string.c (rb_str_new_frozen): since the result object should have
  same tainted/untrusted bits with the original object, return new
  object if the shared object unmatch.  [ruby-core:39745][Bug #5374]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@36373 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

**Revision d7444332 - 07/12/2012 02:44 AM - nobu (Nobuyoshi Nakada)**

rb_str_new_frozen: new object if tainted/untrusted unmatch

- string.c (rb_str_new_frozen): since the result object should have
  same tainted/untrusted bits with the original object, return new
  object if the shared object unmatch.  [ruby-core:39745][Bug #5374]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@36373 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

## History

**#1 - 09/28/2011 04:02 AM - 375gnu (Hleb  Valoshka)**

*- File test.rb added*

*- File test.log added*

*- File gettext-test.rb added*

*- File gettext-debian.log added*

*- File gettex-win.log added*

**#2 - 03/11/2012 04:11 PM - ko1 (Koichi Sasada)**

*- Assignee set to mame (Yusuke Endoh)*

**#3 - 03/12/2012 10:24 AM - nobu (Nobuyoshi Nakada)**

*- Status changed from Open to Feedback*

Does this happen with recent versions?

**#4 - 07/12/2012 07:38 AM - 375gnu (Hleb  Valoshka)**

nobu (Nobuyoshi Nakada) wrote:

> Does this happen with recent versions?

Yes, at least with 1.9.3p194 (2012-04-20 revision 35410) [x86_64-linux].

And today I've made some investigations.

File.stat is rb_file_s_stat, which calls rb_get_path, which calls rb_get_path_check, which at very end calls rb_str_new4, which actually is rb_str_new_frozen.

As value passed to rb_str_new_frozen (filename) isn't frozen, it creates new string (see file string.c, line 679):
if (STR_SHARED_P(orig) && (str = RSTRING(orig)->as.heap.aux.shared)) {...} and that "new" string is returned to rb_get_path_check.

*This str is tainted!*

I also have found more simple way to reproduce the bug:

ruby -e '$SAFE=1;File.stat(("12345678901234567890123"+"4".taint).dup.untaint)'

Argument to stat sh'ld be at least 24 bytes on 64bit box.

I haven't checked again bug with glob on Win32, but suppose that it has the same nature but it didn't expressed itself on my Debian boxes cause they are 64bit. In few days I can check it on 32bit Debian.

**#5 - 07/12/2012 11:44 AM - nobu (Nobuyoshi Nakada)**

*- Status changed from Feedback to Closed*

*- % Done changed from 0 to 100*

This issue was solved with changeset r36373.
Hleb , thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

---

rb_str_new_frozen: new object if tainted/untrusted unmatch

- string.c (rb_str_new_frozen): since the result object should have same tainted/untrusted bits with the original object, return new object if the shared object unmatch. [ruby-core:39745][Bug #5374]

## Files

| | | | |
|---|---|---|---|
| test.rb | 891 Bytes | 09/28/2011 | 375gnu (Hleb Valoshka) |
| test.log | 4.77 KB | 09/28/2011 | 375gnu (Hleb Valoshka) |
| gettext-test.rb | 470 Bytes | 09/28/2011 | 375gnu (Hleb Valoshka) |
| gettext-debian.log | 1.73 KB | 09/28/2011 | 375gnu (Hleb Valoshka) |
| gettex-win.log | 1.12 KB | 09/28/2011 | 375gnu (Hleb Valoshka) |