

Ruby - Feature #14225

untaint hash key strings

12/23/2017 02:08 AM - normalperson (Eric Wong)

Status:	Closed	
Priority:	Normal	
Assignee:		
Target version:		
Description Since we are working on deprecating and removing \$SAFE for [Feature #5455], I propose untainting all string keys used for hashes in Ruby 2.6. It will make implementing [Feature #13725] (fstring dedupe of hash keys) easier. Furthermore, Perl (which I assume is the influence for tainting in Ruby) does not taint hash keys. In fact, perlsec(1) manpage states: "Hash keys are never tainted" cf. http://perldoc.perl.org/perlsec.html		

History

#1 - 12/27/2017 11:16 PM - Eregon (Benoit Daloze)

I think we should remove tainting as a whole along with \$SAFE.
Untainting automatically seems bad practice and counter-intuitive.

#2 - 12/27/2017 11:41 PM - normalperson (Eric Wong)

eregon@p@gmail.com wrote:

Issue [#14225](#) has been updated by Eregon (Benoit Daloze).

I think we should remove tainting as a whole along with \$SAFE.

Agreed.

Untainting automatically seems bad practice and counter-intuitive.

It wouldn't untaint the actual non-frozen string; but the frozen copy which is auto-created when a non-frozen string is used as a hash key. In other words, it should become:

```
h = {}  
f = 'foo'.taint  
h[f] = :bar  
h.keys[0].taint? # => false (true in <= 2.5)
```

```
# In any version of Ruby, it'll stay:  
h.keys[0].object_id != f  
# unless f is already frozen
```

Anyways, I think the change to remove taint should be gradual (like \$SAFE removal) so people ahve time to adapt; and this is one step.

#3 - 12/30/2017 03:09 AM - normalperson (Eric Wong)

<https://bugs.ruby-lang.org/issues/14225>

Proposed patch here:

<https://80x24.org/spew/20171230025109.1946-1-e@80x24.org/raw>

#4 - 10/07/2021 08:45 PM - jeremyevans0 (Jeremy Evans)

- Status changed from Open to Closed