

A Perl::Critic Complaint

Mark Senn

July 10, 2007

Perl::Critic

Perl::Critic critiques Perl source code for best practices.

Perl::Critic

Perl::Critic critiques Perl source code for best practices.

Module

search.cpan.org/~thaljef/Perl-Critic-1.06/lib/Perl/Critic.pm

Perl::Critic

Perl::Critic critiques Perl source code for best practices.

Module

search.cpan.org/~thaljef/Perl-Critic-1.06/lib/Perl/Critic.pm

CGI program that runs on uploaded file

www.perlcritic.com

test.pl

```
#!/usr/local/bin/perl
```

```
# The following line of code eliminates  
# Code before strictures are enabled.  
# See page 429 of PBP. Severity: 5  
# Perl::Critic message.  
use strict;
```

```
# The following line of code eliminates  
# Code before warnings are enabled.  
# See page 431 of PBP. Severity: 4  
# Perl::Critic message.  
use warnings;
```

test.pl (page 2)

```
# A contrived example.
my $passwd = -f '/etc/passwd';
my $shadow = -f '/etc/shadow';

# The following line of code causes a
#     Mixed high and low-precedence booleans
# Perl::Critic message.
#
# Perl::Critic goes on to say
#     Conway advises against combining the low-precedence
#     booleans ( and or not ) with the high-precedence
#     boolean operators ( && || ! ) in the same expression.
#     Unless you fully understand the differences between
#     the high and low-precedence operators, it is easy
#     to misinterpret expressions that use both. ...
($passwd && $shadow) and print "ok\n";
```

Complaint

The offending line of code is

```
($passwd && $shadow) and print "ok\";
```

Complaint

The offending line of code is

```
($passwd && $shadow) and print "ok\";
```

From Larry Wall's talk, "[Present Continuous, Future Perfect](#)":

Complaint

The offending line of code is

```
($passwd && $shadow) and print "ok\";
```

From Larry Wall's talk, "[Present Continuous, Future Perfect](#)":
Different things should look different.

Complaint

The offending line of code is

```
($passwd && $shadow) and print "ok\";
```

From Larry Wall's talk, "[Present Continuous, Future Perfect](#)":

Different things should look different.

Similar things should look similar.

Complaint

The offending line of code is

```
($passwd && $shadow) and print "ok\";
```

From Larry Wall's talk, "[Present Continuous, Future Perfect](#)":

Different things should look different.

Similar things should look similar.

The "&&" and "and" look different and are different.

Complaint

The offending line of code is

```
($passwd && $shadow) and print "ok\";
```

From Larry Wall's talk, "[Present Continuous, Future Perfect](#)":

Different things should look different.

Similar things should look similar.

The "&&" and "and" look different and are different.

I'm using "&&" to separate variables and "and" to separate a condition from an action. *Perl::Critic* may not share this mental model, though.

Complaint

The offending line of code is

```
($passwd && $shadow) and print "ok\";
```

From Larry Wall's talk, "[Present Continuous, Future Perfect](#)":

Different things should look different.

Similar things should look similar.

The "&&" and "and" look different and are different.

I'm using "&&" to separate variables and "and" to separate a condition from an action. *Perl::Critic* may not share this mental model, though.

Plus, the expression is parenthesized to remove ambiguity. Two spaces surround "and" to emphasize its low precedence.

Complaint

The offending line of code is

```
($passwd && $shadow) and print "ok\";
```

From Larry Wall's talk, "[Present Continuous, Future Perfect](#)":

Different things should look different.

Similar things should look similar.

The "&&" and "and" look different and are different.

I'm using "&&" to separate variables and "and" to separate a condition from an action. *Perl::Critic* may not share this mental model, though.

Plus, the expression is parenthesized to remove ambiguity. Two spaces surround "and" to emphasize its low precedence.

Rhetorical question: Why is *Perl::Critic* warning me about this?