STR31-PL. Do not pass string literals to functions expecting regexes

Many built-in functions accept a regex pattern as an argument. Furthermore, any subroutine can accept a string yet treat it as a regex pattern. This could be done, for example, by passing the string to the match operator (m/). Because regex patterns are encoded as regular strings, it is tempting to assume that a string literal will be treated as if a regex that matched only that string literal were supplied. Unexpected function behavior can result if the string contains characters that have special meanings when the string is treated as a regex pattern. Therefore, do not pass strings that are not clearly regex patterns to a function that takes a regex.

Noncompliant Code Example

This code example appears to split a list of names.

```perl
my $data = 'Tom$Dick$Harry';
my @names = split( '$', $data);
```

But the first argument to `split()` is treated as a regex pattern. Because $ indicates the end of the string, no splitting occurs.

Compliant Solution

This compliant solution passes a regex pattern to `split()` as the first argument, properly specifying $ as a raw character. Consequently, `@names` is assigned the three names Tom, Dick, and Harry.

```perl
my $data = 'Tom$Dick$Harry';
my @names = split( m/$/, $data);
```

Exceptions

STR31-PL-EX0: A string literal may be passed to a function if it normally takes a regex pattern but provides special behavior for that string. For example, the `perflib` manpage [Wall 2011] says, regarding `PATTERN`, the first argument to `split()`:

> As a special case, specifying a PATTERN of space (' ') will split on white space just as "split" with no arguments does. Thus, "split(' ')" can be used to emulate awk's default behavior, whereas "split(/\ )" will give you as many initial null fields (empty string) as there are leading spaces.

Risk Assessment

<table>
<thead>
<tr>
<th>Recommendation</th>
<th>Severity</th>
<th>Likelihood</th>
<th>Remediation Cost</th>
<th>Priority</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>STR31-PL</td>
<td>Low</td>
<td>Likely</td>
<td>Low</td>
<td>P9</td>
<td>L2</td>
</tr>
</tbody>
</table>

Automated Detection

<table>
<thead>
<tr>
<th>Tool</th>
<th>Diagnostic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perl::Critic</td>
<td>BuiltInFunctions::ProhibitStringySplit</td>
</tr>
</tbody>
</table>

Bibliography

- [CPAN] Elliot Shank, Perl-Critic-1.116 BuiltInFunctions::ProhibitStringySplit
- [Wall 2011] perlfunc