**EXP33-PL. Do not invoke a function in a context for which it is not defined**

Perl functions can be invoked in two contexts: list and scalar. These contexts indicate what is to be done with the return value. Functions can return different values in list context than in scalar context. For instance, the `grep` function takes a list and a block or expression and filters out elements of the list for which the block or expression evaluates to false. The `grep` function returns the filtered list when called in list context, but when called in scalar context, it merely returns the size of this list. That is, it returns the number of elements for which the block or expression evaluates to true.

Some functions do not define what they return in list or scalar context. For instance, according to the `perldoc` manpage, the `sort` function "sorts the LIST and returns the sorted list value. In scalar context, the behavior of 'sort()' is undefined."

**Noncompliant Code Example (sort())**

This noncompliant code example inadvertently assigns a scalar to the result of the `sort` function.

```perl
sub ret {  
    my $list = shift;  
    my @list = @{$list};  
    # ...  
    return sort @list;  
}  
my @list = ( "foo", "bar", "baz" );  
my $result = ret @list;
```

The contents of `$result` are undefined because the `sort` function's return value is not defined in a scalar context.

**Compliant Solution (sort())**

This compliant solution guarantees that the `ret` function is called only in list context.

```perl
sub ret {  
    my $list = shift;  
    my @list = @{$list};  
    # ...  
    return sort @list;  
}  
my @list = ( "foo", "bar", "baz" );  
my @result = ret @list;
```

In this case, the `@result` array will contain the list `{"bar", "baz", "foo"}`.

**Risk Assessment**

Using an unspecified value can lead to erratic program behavior.

<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>EXP33-PL</td>
<td>Medium</td>
<td>Unlikely</td>
<td>Low</td>
<td>P6</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>Subroutines::ProhibitReturnSort</td>
</tr>
</tbody>
</table>

**Bibliography**

[Conway 2005]