Rule 05. Characters and Strings (STR)

- STR50-CPP. Guarantee that storage for strings has sufficient space for character data and the null terminator
- STR51-CPP. Do not attempt to create a std::string from a null pointer
- STR52-CPP. Use valid references, pointers, and iterators to reference elements of a basic_string
- STR53-CPP. Range check element access

The following rules from the SEI CERT C Coding Standard also apply in C++:

- STR30-C. Do not attempt to modify string literals
- STR31-C. Guarantee that storage for strings has sufficient space for character data and the null terminator
- STR32-C. Do not pass a non-null-terminated character sequence to a library function that expects a string
- STR34-C. Cast characters to unsigned char before converting to larger integer sizes
- STR37-C. Arguments to character-handling functions must be representable as an unsigned char
- STR38-C. Do not confuse narrow and wide character strings and functions

Information for Editors
To have a new guideline automatically listed above be sure to label it str and rule.

Risk Assessment Summary

<table>
<thead>
<tr>
<th>Rule</th>
<th>Severity</th>
<th>Likelihood</th>
<th>Remediation Cost</th>
<th>Priority</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>STR50-CPP</td>
<td>High</td>
<td>Likely</td>
<td>Medium</td>
<td>P18</td>
<td>L1</td>
</tr>
<tr>
<td>STR51-CPP</td>
<td>High</td>
<td>Likely</td>
<td>Medium</td>
<td>P18</td>
<td>L1</td>
</tr>
<tr>
<td>STR52-CPP</td>
<td>High</td>
<td>Probable</td>
<td>High</td>
<td>P6</td>
<td>L2</td>
</tr>
<tr>
<td>STR53-CPP</td>
<td>High</td>
<td>Unlikely</td>
<td>Medium</td>
<td>P6</td>
<td>L2</td>
</tr>
</tbody>
</table>