Refers to the practice of checking for malignant input that can exploit a system.

1 | `eval()`

Eval takes user input and runs it as code, and this could easily be used to .

2 | Cross-site scripting

If a user gives a website HTML code and the website doesn't sanitiize it, the user could embed JS that could even run on other users computers (i.e. like in a comment where the JS is loaded when it's viewed)?

3 | Database Injection

Semicolons end SQL commands, so if you're reading user input to make a SQL query they could put a quote and semicolon in their input followed by arbitrary SQL lookup.

4 | Buffer Overflows

By writing past the length of a buffer in memory you can overwrite other memory that a user could own.

5 | Fixes

- Always check user input
- ~~Don't use languages (like C) that don't have protections against unsafe memory access~~ INCORRECT!
- Always use C++ string/vector objects and prefer the STL
- Always use string/array/list functions that take size as an argument