Working with strings can lead to various security flaws and errors in software development using the C++ language.

What are the common string manipulation errors that can be encountered?

Some types of string operations in C++ such can pose a security threat such as buffer overflow. Buffer overflow can occur when memory space is insufficient to store a string, resulting in overwriting data already stored in memory. This can happen for very large string values retrieved from user input, command line arguments and retrieving information from databases.

A buffer overflow can be prevented by ensuring that sufficient memory is available and through limiting data copies using truncation. Buffer overflows can also be avoided by using a string object rather than an array of characters.

Buffer overflow can be detected by attempting to intentionally enter more data that is required. If the program accepts this data, a buffer overflow may occur. For example, I have tested the following code provided by

It turns out that although buf is an array of characters with size 12, it can actually accept more than 12 characters. If I understand correctly, truncation can be used in this instance to remove extra characters with removing the last null value which indicates the end of the string.

How can these errors be resolved and/or limited?

What tips can be utilized to identify security vulnerabilities related to strings in C++?

Be sure to provide an appropriate source code example to illustrate your points.