Security Injections @Towson

Security Injections, Java CS1 - Integer Error

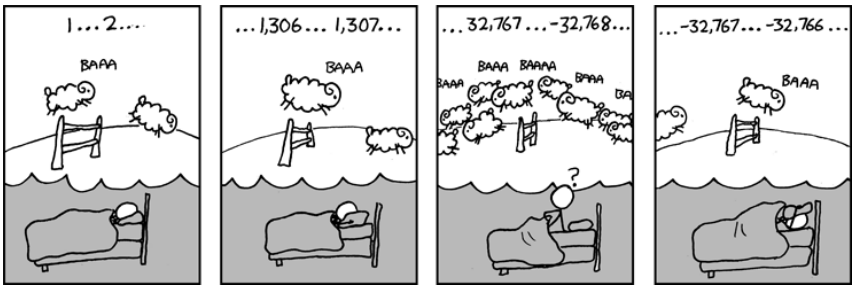
1. Background | 2. Code Responsibly | 3. Laboratory Assignment
4. Security Checklist | 5. Discussion Questions

Integer Error –"You Can't Count That High" - CS1

Background

Summary:

Integer values that are too large or too small may fall outside the allowable bounds for their data type, leading to unpredictable problems that can both reduce the robustness of your code and lead to potential security problems.



Description:

Declaring a variable as type **int** allocates a fixed amount of space in memory. Most languages include several integer types, including **short**, **int**, **long**, etc. , to allow for less or more storage. The amount of space allocated limits the range of values that can be stored. For example, a 32-bit **int** variable can hold values from -2^{31} through $2^{31}-1$.

Input or mathematical operations such as addition, subtraction, and multiplication may lead to values that are outside of this range. This results in an integer error or overflow, which causes undefined behavior and the resulting value will likely not be what the programmer intended. Integer overflow is a common cause of software errors and vulnerabilities.

Risk – How Can It Happen?

An integer error can lead to unexpected behavior or may be exploited to cause a program crash, corrupt data, lead to incorrect behavior, or allow the execution of malicious software.

Example of Occurrence:

1. There is a Facebook group called "If this group reaches 4,294,967,296it might cause an integer overflow." This value is the largest number that can fit in a 32 bit unsigned integer. If the number of members of the group exceeded this number, it might cause an overflow. Whether it will cause an overflow or not depends upon how Facebook is implemented and which language is used – they might use data types that can hold larger numbers. In any case, the chances of an overflow seem remote, as roughly 2/3 of the people on earth would be required to reach the goal of more than 4 billion members.

2. Many Unix operating systems store time values in 32-bit signed (positive or negative) integers, counting the number of seconds since midnight on January 1, 1970. On Tuesday, January 19, 2038, this value will overflow, becoming a negative number. Although the impact of this problem in 2038 is not yet known, there are concerns that software that projects out to future dates – including tools for mortgage payment and retirement fund distribution – might face problems long before then. Source: Year 2038 Problem" http://en.wikipedia.org/wiki/Year_2038_problem

Answer the following questions:

Question 1:

Declaring a variable as type integer:

- Allocates an infinite amount of storage
- ✓ Allocates a fixed amount storage

(Hint: read summary and description sections to answer this question.)

Question 2:

An integer error in C++ or Java causes:

- a syntax error
- the program to correct itself
- ✓ unexpected behavior

(Hint: read the risk section above to answer this question.)

Go To Next Section

This project is supported by the National Science Foundation under grants DUE-1241738 and DUE -0817267. Any opinions, findings, conclusions, or recommendations expressed are those of the authors and do not necessarily reflect the views of the National Science Foundation.