Security Injections @Towson

Welcome, Marilyn Soyars!

Security Injections, Java CS1 - Input Validation

1. Background 2. Code Responsibly 3. Laboratory Assignment
4. Discussion Questions

Input Validation - "All Input is Evil" - CS1

Background

Summary:

Any program input--such as a user typing at a keyboard, a file or a network connection--can potentially be the source of security vulnerabilities and disastrous bugs. All input should be treated as potentially dangerous.

Description:

Most software packages rely upon external input. Although information typed at a computer might be the most familiar, networks and external devices can also send data to a program. Generally, this data is of a specific type: for example, a user interface that requests a person's name might be written to expect a series of alphabetic characters. If the correct type and form of data is provided, the program might work fine. However, if programs are not carefully written, attackers can construct inputs that can cause malicious code to be executed.

If video does not work, try refreshing the page:

Video by Summer Lagambi, Haley McComas, and Abbey Baker.

Risk – How Can It Happen?

Any data that can enter your program from an external source can be a potential source of problems. If external data is not checked to verify that it has the right type of information, the right amount of information, and the right structure of information, it can cause problems. Any program that processes data from external sources without adequate validation can be susceptible to security vulnerabilities.

Drawing used by permission of Dominik Joswig

Examples of Occurrence:

- In December 2005, a Japanese securities trader made a $1 billion typing error, when he mistakenly sold 600,000 shares of stock at 1 yen each instead of selling one share for 600,000 yen. A few lines of code may have averted this error. Fat fingered typing costs a trader's bosses £128m, *The Times Online,* December 09, 2005
- Web applications are highly vulnerable to input validation errors. Inputting the invalid entry "!@#$%^&*()" on a vulnerable e-commerce site may cause performance issues or denial of service on a vulnerable system or invalid passwords such as "pwd'" or "1=1— " may result in unauthorized access. http://www.processor.com/editorial/article.asp?article=articles%2Fp3112%2F32p12%2F32p12%2F32p12.asp&guid=&searchtype=&WordList=&bJumpTo=True
- A Norwegian woman mistyped her account number on an internet banking system. Instead of typing her 11-digit account number, she accidentally typed an extra digit, for a total of 12 numbers. The system discarded the extra digit, and transferred $100,000 to the (incorrect) account. A simple dialog box informing her that she had typed too many digits would have helped avoid this expensive error. Olsen, Kai. "The $100,000 Keying error" IEEE Computer, August 2008

Answer the following questions:

Question 1:

The following are sources of input for programs:

☑ ✓ Keyboard

☑ ✓ Network

☑ ✓ File

(HINT:Read summary and description sections to answer this question)

Question 2:

"Evil" input can occur from an error made by the user:

● ✓ True

○ False

(HINT:Read summary and description sections 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.

Powered by a modified version of Class2Go