Interactive Support for Secure Programming Education

UNC CHARLOTTE
College of Computing and Informatics

Jun Zhu, Heather Lipford, Bill Chu, Michael Whitney
University of North Carolina at Charlotte

Software flaws often lead to serious security and privacy problems.

Secure programming must be threaded throughout the entire computing curriculum

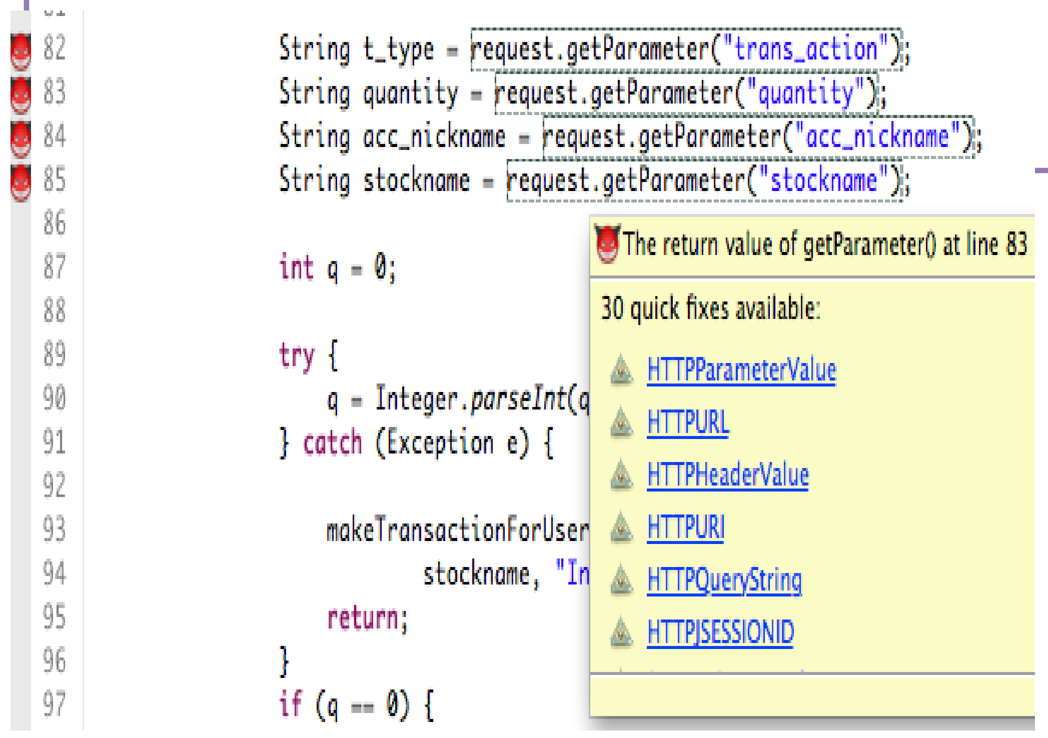
To provide training in, and support for, good secure programming practices as part of the tools that students use to program

Serves as a continuous educational opportunity that adds to or reinforces the students' secure programming training while they are performing their coding activities.

Eclipse plug-in for Java, named **ASIDE**

Integrate secure programming support into the IDE

Provides reminders and contextual education material

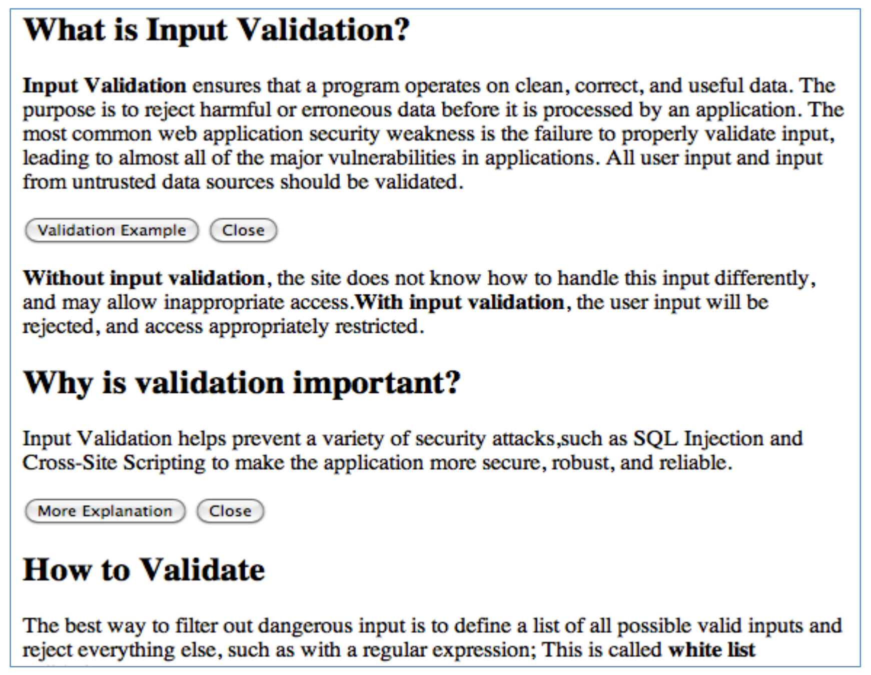


```
82              String t_type = request.getParameter("trans_action");
83              String quantity = request.getParameter("quantity");
84              String acc_nickname = request.getParameter("acc_nickname");
85              String stockname = request.getParameter("stockname");
86
87              int q = 0;
88
89              try {
90                  q = Integer.parseInt(q
91              } catch (Exception e) {
92
93                  makeTransactionForUser
94                          stockname, "In
95                  return;
96              }
97              if (q == 0) {
```

The return value of getParameter() at line 83

30 quick fixes available:

- HTTPParameterValue
- HTTPURL
- HTTPHeaderValue
- HTTPURI
- HTTPQueryString
- HTTPJSESSIONID



What is Input Validation?

Input Validation ensures that a program operates on clean, correct, and useful data. The purpose is to reject harmful or erroneous data before it is processed by an application. The most common web application security weakness is the failure to properly validate input, leading to almost all of the major vulnerabilities in applications. All user input and input from untrusted data sources should be validated.

Without input validation, the site does not know how to handle this input differently, and may allow inappropriate access.**With input validation**, the user input will be rejected, and access appropriately restricted.

Why is validation important?

Input Validation helps prevent a variety of security attacks,such as SQL Injection and Cross-Site Scripting to make the application more secure, robust, and reliable.

How to Validate

The best way to filter out dangerous input is to define a list of all possible valid inputs and reject everything else, such as with a regular expression; This is called **white list**

Study of 20 students using the the tool 3 hr. Test secure programming knowledge pre and post using ASIDE

Statistically Significant 10.3 points increase in test score (out of 100)
461 distinctive warnings generated. 70% clicked 47% resolved

Most students did spend several minutes reading explanations to learn more about the warnings

60% of the students having dynamic statement warnings successfully wrote prepared statements after reading the explanation (6/10)

Initial evaluation suggests that ASIDE can provide a learning opportunity and increase secure programming knowledge

It could integrate secure programming education across a computing curriculum to students at all levels

Future Work

Longer term studies to see whether these effects can be sustained.

Look at lower level CS courses

Secure Programming Education across curriculum with interactive IDE support

Supported by NSF Grants 1044745 and 0830624