The theory and implementation of InputValidator: A semi-automated value-level bypass testing tool

J. Miller, University of Alberta, Canada
L. Zhang, University of Alberta, Canada
E. Ofuonye, University of Alberta, Canada
M. Smith, University of Calgary, Canada

ABSTRACT

The construction and testing of Web-based systems has become more complex and challenging because of continual innovations in technology. Security is a major concern, particularly for the deployment of mission critical applications. One of the principal vulnerabilities in Web-based systems revolves around insufficient and inappropriate input validation, a deficiency that can be exploited by attacks that bypass client-side checking. This article describes a partially automated mechanism, the tool InputValidator, which seeks to address this issue by sending test data directly to the server to test the robustness and security of the back-end software. The tool allows a user to construct, execute, and evaluate a number of test cases through a form-filling exercise instead of writing bespoke test code.

Keywords: software testing; tool support; security threats; Web based applications; Web technologies

INTRODUCTION

The usage of Web-based applications has significantly expanded in recent years, and now affects our daily working lives with many organizations setting up sites for online trading, retail, planning, scheduling, and so forth. Constructing an effective Web-based system
has become increasingly complex and challenging with systems typically run on distributed hardware and containing both client-side and server-side components. Incompatibility and associated security issues abound on the client side from the variety of browsers. The problems change with each new software release (Nguyen, 2001), and are compounded with the countless combinations of hardware configurations. On the server side, there is equal complexity derived from the deployment of miscellaneous environments to support Web applications. The challenges to the testing of Web-based systems have increased given that the server-side software for many companies and global corporations has to be distributed over a number of physical servers, or hosted by third-party Web service providers. Vulnerabilities concerning network reliability, accessibility, security, and compatibility are made worse by the simple fact that, of necessity, most Web-based applications are exposed to an unidentified worldwide set of (untrustworthy) users.

Another challenge to testing such systems is the “management factor.” The competitiveness of software development and IT industry has pushed companies to shorten their software development life cycle to design, code, test, and deliver products rapidly using development processes such as extreme programming and test-driven development, among others (Beck, Beedle, van Bennekom, Cockburn, Cunningham, Fowler, et al., 2008). However, this has placed increased pressure on testing and quality assurance activities.

Due to the complexity introduced by the environment and technology factors, as well as the pressure from management, the testing of a Web-based system must be automated to be successful. While some testing tools have been adapted to accommodate Web-based systems (Automated Testing Specialists, 2006; Hower, 2008), these tools tend to be rather generic in nature and do not cover the full spectrum of issues that are unique to Web-based systems.

This article introduces a new test tool developed to help testing engineers to automatically parse form parameters, generate test cases according to users’ input data, and provide an interface which implements bypass testing (Ofutt, Wu, Du, & Huang, 2004). The remainder of this article is as follows. In the second section, testing methods are defined for two of the most common vulnerabilities: structured query language (SQL) injection and invalid input. In the third section, bypass testing is introduced as a technique to solve these issues, together with how the new tool (the main topic of the article) can be efficiently used to implement bypass testing. The fourth section illustrates this tool-based testing approach on a real Web site, followed by the final section, the conclusion.

**CURRENT STATE OF THE ART**

There is no common wisdom on how to categorize the very wide range of
Related Content

**A Conceptual Tool for Usability Problem Identification in Website Development**
[www.igi-global.com/chapter/conceptual-tool-usability-problem-identification/53065?camid=4v1a](www.igi-global.com/chapter/conceptual-tool-usability-problem-identification/53065?camid=4v1a)

**Engineering Conceptual Data Models from Domain Ontologies: A Critical Evaluation**
[www.igi-global.com/article/engineering-conceptual-data-models-domain/2624?camid=4v1a](www.igi-global.com/article/engineering-conceptual-data-models-domain/2624?camid=4v1a)
E-Learning and the Semantic Web: A Descriptive Literature Review
[www.igi-global.com/chapter/e-learning-and-the-semantic-web/140794?camid=4v1a](www.igi-global.com/chapter/e-learning-and-the-semantic-web/140794?camid=4v1a)

Disambiguating the Twitter Stream Entities and Enhancing the Search Operation Using DBpedia Ontology: Named Entity Disambiguation for Twitter Streams