Chapter 12
Towards Automated Bypass Testing of Web Applications

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 with continual innovations in technology. One major concern particularly for the deployment of mission critical applications is security. In Web-based systems, the principal vulnerabilities revolve around deficient input validation. This chapter describes a partially automated mechanism, the tool InputValidator, which seeks to address this issue through bypassing client-side checking and 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.

INTRODUCTION
The usage of Web-based applications has significantly expanded and affects our daily lives in a multitude of ways. Internet usage statistics show that as of June 2008 (Internet World Stats, 2008), over 1.4 billion individuals have used the World Wide Web (WWW) for various undertakings ranging from communication (mail, telecommunication), business (buying and selling, stock trading), social events (multimedia, virtual hang-outs, gaming), and information gathering (weather data, news); all from the convenience of their homes and offices. Constructing an effective Web-based system to satisfy this rising dependence and very demanding non-functional requirements has become increasingly complex and challenging with
systems typically running 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 has 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 (un-trustworthy) 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, etc (Beck 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 Web-based systems must be automated to be successful. While some testing tools have been adapted to accommodate Web-based systems (Hower, 2008; Automated Testing Specialists, 2006), 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 paper 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 (Offutt, 204). The remainder of this paper is as follows. In section 2, testing methods and issues arising from them is defined for two of the most common vulnerabilities: SQL injection and invalid input. In section 3, bypass testing is introduced as a technique to solve these issues; together with how the new tool (the main topic of the paper) can be efficiently used to implement bypass testing. Section 4 illustrates this tool-based testing approach on a real Web site; followed by the final section, Section 5, the conclusion.

**CURRENT STATE OF THE ART**

There is no common wisdom on how to categorize the very wide range of potential risks or prevention mechanisms found in Web-based systems. Attempting this categorization, Gerrard and Thompson (2002) identified 24 distinct test types; grouped into seven categories. Nguyen (2001) categorizes tests into 14 distinct types. Vijayaraghavan and Kaner (2002) summarize bugs that can happen on (virtual) “shopping carts” into no less than 60 categories! However, no matter how the testing of Web-based systems is sorted, security concerns and their associated testing have drawn much attention and have become extremely critical for Web-based systems; especially for e-commerce applications. The Open Web Application Security Project’s (2006) top ten vulnerabilities that need to be addressed through security testing are, in order of importance, invalidated input, broken access control, broken authentication, cross-site scripting, buffer overflows, injection flows, improper effort handling, insecure storage, application denial of service, and insecure configuration management. Although categorized separately, many of the listed vulnerabilities, for example cross-site scripting, injection flows, and buffer overflows, are indirectly a result of insufficient input validation. Thus, validating inputs can be identified as