Selecting Secure Web Applications Using Trustworthiness Benchmarking

Afonso Araújo Neto, University of Coimbra, Portugal
Marco Vieira, University of Coimbra, Portugal

ABSTRACT

The multiplicity of existing software and component alternatives for web applications, especially in open source communities, has boosted interest in suitable benchmarks, able to assist in the selection of candidate solutions, concerning several quality attributes. However, the huge success of performance and dependability benchmarking contrasts the small advances in security benchmarking. Traditional vulnerability/attack detection techniques can hardly be used alone to benchmark security, as security depends on hidden vulnerabilities and subtle properties of the system and its environment. A comprehensive security benchmarking process should consist of a two-step process: elimination of flawed alternatives followed by trustworthiness benchmarking. In this paper, the authors propose a trustworthiness benchmark based on the systematic collection of evidences that can be used to select one among several web applications, from a security point-of-view. They evaluate this benchmark approach by comparing its results with an evaluation conducted by a group of security experts and programmers. Results show that the proposed benchmark provides security rankings similar to those provided by human experts. In fact, although experts may take days to gather the information and rank the alternative web applications, the benchmark consistently provides similar results in a matter of few minutes.

Keywords: Benchmarking, Security, Trust-Based Metrics, Trustworthiness, Web Applications

INTRODUCTION

Due to the increasing diversity of web applications, systems and components, system administrators and developers have nowadays the chance to select the software that best fit their needs based on quality attributes such as performance, usability, and security (Barbacci, 2003). However, as several types and brands of operating systems, web servers, database management systems (DBMS), and other classes of applications become available, selecting the most appropriate one(s) becomes less and less trivial. In consequence, the interest in methods to accomplish fair and representative comparison among software and systems concerning these various attributes has grown considerably (Bondavalli, 2009).

A benchmark is a standard procedure that allows assessing and comparing systems or components according to specific characteris-
tics (e.g., performance, availability, security) (Gray, 1993). Computer industry holds a reputed infrastructure for performance evaluation, where the Transaction Processing Performance Council (TPC) (http://www.tpc.org) benchmarks are recognized as one of the most successful benchmarking initiatives of the overall computer industry. Furthermore, the concept of dependability benchmarking has gained ground in the last few years, having already led to the proposal of dependability benchmarks for operating systems, web servers, databases and transactional systems in general (Kanoun & Spainhower, 2005). Security, however, has been largely absent from previous efforts, in a clear disparity to performance and dependability. Theoretically, a security benchmark would provide a metric (or small set of metrics) able to characterize the degree to which security goals are met in the system under testing (Payne, 2006), allowing developers and administrators to compare alternatives and make informed decisions. No clear methodology to accomplish this has been proposed so far.

Traditional security metrics are hard to define and compute (Torgerson, 2007), as they involve making isolated estimations about the ability of an unknown individual (e.g., a hacker) to discover and maliciously exploit an unknown system characteristic (e.g., a vulnerability). While techniques to find, correct and prevent actual vulnerabilities flourish in the research community (Zanero, Carettoni, & Zanchetta, 2005), the lack of accurate and representative security metrics makes the conception of security benchmarking an extremely difficult task (Bondavalli, 2009).

An alternative way to tackle this problem is to look for metrics that systematize and summarize the trustworthiness that can be justifiably put in a system or application. Instead of quantifying absolute security factors, trust-based metrics are grounded on the idea of quantifying the evidences available regarding the trustworthiness that one can put in the assessed application. However, as trust does not necessarily provide guarantees, security benchmarking can only be accomplished as a twofold process, with trustworthiness being the metric used for selecting among non-obviously flawed alternatives. In other words, a reliable benchmarking approach should provide a set of security guarantees by forcing the systems under evaluation to pass a set of basic security assessments before considering the trustworthiness aspect to support the final selection (e.g., in a web application benchmarking campaign, no application should present actual vulnerabilities detectable during testing; the ones that do not present vulnerabilities are then ranked using a process like the one proposed in this paper).

Trust-based metrics allow characterizing “the degree to which security goals are met in the given system or component” by summarizing the amount of protection that it has in terms of security mechanisms, processes, configurations, procedures and behaviors. In the web context, these metrics can be actually used in several scenarios, including:

- **Comparing the trustworthiness of alternative web applications.** This is extremely useful for system administrators when selecting, from a set of alternative solutions that implement the same high-level requirements, the one that offers more guarantees in terms of security (based on the security evidences available).

- **Comparing the trustworthiness of alternative software components.** In a development environment, this is relevant for developers to select the most trustworthy software components to be integrated in an application, especially when considering component-based development (Crnkovic, Chaudron, & Larsson, 2006).

- **Redirecting the web application development effort.** By comparing the software components developed in a project, it is possible to identify the ones that require more attention (e.g., testing and rework) in terms of security. This allows effectively managing the development effort and is, obviously, of utmost importance in the context of large complex projects.
Related Content

A Context-Aware Model of Trust for Facilitating Secure Ad Hoc Collaborations
www.irma-international.org/chapter/context-aware-model-trust-facilitating/40784/

A Framework for Studying the Problem of Trust in Online Settings
www.irma-international.org/article/framework-studying-problem-trust-online/51600/

Human-Machine Trust Interaction: A Technical Overview
www.irma-international.org/article/human-machine-trust-interaction/51605/

Technology: A Bane or Boon for Cross-National Levels of Generalized Trust?
Blaine G. Robbins and Maria S. Grigoryeva (2010). *Trust and Technology in a Ubiquitous Modern Environment: Theoretical and Methodological Perspectives* (pp. 123-144).
www.irma-international.org/chapter/technology-bane-boon-cross-national/42904/

STRIDE: A Secure Framework for Modeling Trust-Privacy Tradeoffs in Distributed Computing Environments
www.irma-international.org/article/stride-secure-framework-modeling-trust/43582/