Addressing Malicious Code in COTS: A Protection Framework

Donald J. Reifer, Pranjali Baxi, Fabio Hirata,
Jonathan Schifman, and Ricky Tsao

Reifer Consultants, Inc.
Torrance, CA, USA
don@reifer.com
pranjali_baxi@yahoo.com
{fhirata, schifman, rtsao}@usc.edu

Abstract. The potential for problems due to malicious code increases in direct proportion with the number of COTS software used in a system. Because of this, many practitioners have used a variety of techniques to address potential attacks. Yet, little guidance has been offered as to which techniques work best, when, and under what conditions. To rectify this problem, we have created a framework that can be used to help those interested in addressing vulnerabilities with a solution. The framework matches defenses to attacks using a risk-based approach that focuses on providing cost-effective protection.

1 Introduction

The potential for malicious code within COTS (commercial-off-the-shelf) components has grown during the past few years as industry has used existing components to build their systems quicker, better and more cheaply. While many articles have been written discussing the security problems with COTS and potential solutions, little guidance has been offered in the literature as to what techniques to use, when, and under what conditions.

We launched a project early in 2004 to develop a framework to rectify this problem. The project's aim is to create a framework that practitioners could use to determine the most cost-effective defenses against potential attacks using risk management principles [1]. The framework by design addresses applications software. It seeks to protect software against the most common types of attacks using existing technology that is mature.

The purpose of this paper is to provide an overview of our proposed protection framework and discuss the rationale upon which is built. The framework is synthesized upon a combination of published approaches and also government approaches to protecting applications. Based on our trial-use experiments, this framework presents a useful and practical means for engineers to identify ways to mitigate security threats in new COTS software applications [2].

X. Franch and D. Port (Eds.): ICCBSS 2005, LNCS 3412, pp. 157 – 167, 2005.

2 Related Work

While other security frameworks have been developed in categorizing threats, few have addressed protection techniques for applications software. Most of the related work that we found seems to focus on classifying network-centric threats and approaches to mitigate them [3]. One of the few exceptions was Landwehr et al. [4] who provides a taxonomy for identifying and addressing security flaws in software applications during the system life cycle. This research focuses primarily on identifying a set of security flaws in applications by looking at how, when, and where the flaw is introduced. The research is a good start in threat identification, but it fails to focus on the security requirements of the target application. Therefore we felt that their taxonomy was not a practical tool for application protection.

We did find a hardware framework process that was developed by Battelle National Labs that seemed to provide a suitable model for what we were after [5]. Battelle's framework provides model for classifying hardware items defenses against piracy, tampering and reverse engineering. Their framework classifies the item to be protected, identifies the attacks and defenses to the item, and supports selection decision-making using risk management approaches. We followed Battelle's process to develop our framework because it seemed most applicable.

We diverged from the Battelle work when identifying vulnerabilities. Instead of using their approach, we employed an existing government standard, the Common Criteria [6], as our basis for determining vulnerabilities. The CC provides a comprehensive catalog of high level security requirements in the form of functional and assurance services that must be supported in applications. By using the CC, we were able to combine and expand on existing information instead of replicating it using uniquely determined characteristics as in the Battelle framework.

Our related work interest was to develop attack-defense mappings for security applications. The majority of the mapping research we found was in the field of network security, as in [7, 8, 9] which was not applicable to threats to applications software security. For example, Mirkovic et. al. [10] proposes a taxonomy of DDoS attack and defense mechanisms which is typical of this work.

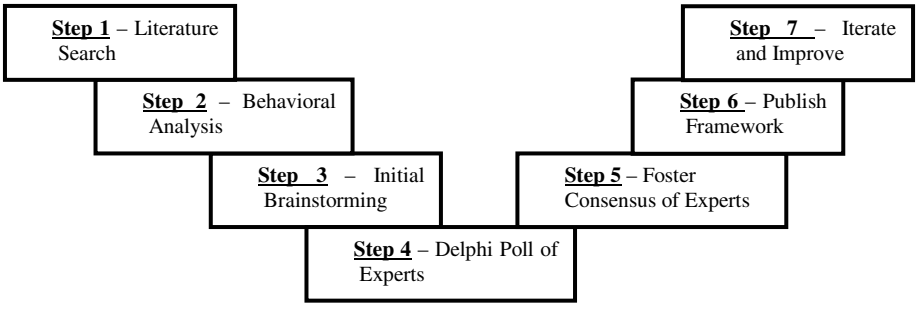

Fig. 1. Framework Development Process