Software Security in Practice

March – April 2011

Summary

This article discusses the organizational commitment that is necessary to produce secure software. Over the past decade, software security has matured from a niche topic studied by academics and precious few in industry to a vital part of how software is built. Thus, security must be an essential part of the software development lifecycle. Although companies such as Microsoft and Adobe have created a “security development lifecycle” to implement these necessities, most large organizations’ software security teams end up performing activities which do not encompass their role.

There are numerous concerns which accompany attempting to produce secure software such as building a talented security team, maintaining legacy code, gaining organizational support for a budget, establishing an education and training program as well as standards and metrics, and handling breaches and vulnerabilities. To analyze the various concerns, the Building Security in Maturity Model (BSIMM) was developed. BSIMM is a study of 30 real-world software security initiatives that focuses on the behaviors of these organizations when building software. This study resulted in a framework which allows software security to succeed at the project level. The table on the next page illustrates BSIMM’s software security framework.
Below are the four domains of the BSIMM framework explained:

- **Governance**—practices that help organize, manage, and measure the initiative. Staff development is also a central governance practice.
- **Intelligence**—practices resulting in collections of corporate knowledge used to carry out software security activities throughout the organization. Collections include both proactive security guidance and organizational threat modeling.
- **Secure software development life cycle touchpoints**—practices associated with analysis and assurance of particular software development artifacts and processes. All software security methodologies include these practices.
- **Deployment**—practices that interface with traditional network security and software maintenance organizations. Software configuration, maintenance, and other environment issues directly affect software security.

**What I learned**

From reading this article, I learned about the growing need among organizations to implement an initiative to produce secure software. I now understand how secure software development has grown into such a major issue when compared to the past and what the current trends are for combating the problem. The information from this article will be beneficial in my career as a software developer since I not only need to be aware of the importance of producing secure software but also being aware of the precautions necessary. One question that I still had concerning this article was how expensive is it to really implement all of the measures noted in the BSIMM framework. I ask this since I feel that there are many widely standardized policies and frameworks in software, but due to the expense of resources and personnel, they are not always followed.
Bibliography