Software Engineering Institute | Carnegie Mellon University

Assembly, Integration, and Evolution Overview

Howard Lipson

September 2005

ABSTRACT: The objective of the Assembly, Integration & Evolution content area is to raise awareness about the essential technical, business, and individual user issues that must be addressed during assembly, integration, and evolution to achieve and maintain a high degree of system-wide assurance of security and survivability.

ACKNOWLEDGEMENT: My thanks to Gunnar Peterson of Cigital for his contribution to this overview.

The design, acquisition, and implementation decisions made during the assembly and integration of components and services into a larger system clearly have a profound impact on the security and survivability of the system as a whole. The objective of the Assembly, Integration & Evolution content area is to raise awareness about the essential technical, business, and individual user issues that must be addressed during assembly, integration, and evolution to achieve and maintain a high degree of system-wide assurance of the desired security properties.

During assembly and integration, the logical design assumptions for a system meet the physical, business, technical, organizational, and individual user realities of the target system environment. Current trends point toward a sharp increase in exploits based on assembly-integration design errors, architectural mismatches among components, insecure identity management and services, false assumptions about a component's properties, an over-reliance on perimeter-based network security mechanisms, and the use of components in contexts (environments) not envisioned by the components' designers. Business pressures for increased efficiency and flexibility are moving applications toward "just-in-time" service creation and delivery (for example, through dynamic assembly in a web services environment), and are therefore stressing the limits of security and survivability even further. User privacy concerns centering on what identifiable information will be used for tracking and tracing may create constraints and conflict with security goals. The system-wide effects of the emergent behavior of large numbers of (software) components and services are mapped onto the production infrastructure.

Software Engineering Institute
Carnegie Mellon University
4500 Fifth Avenue
Pittsburgh, PA 15213-2612

Phone: 412-268-5800
Toll-free: 1-888-201-4479

www.sei.cmu.edu

Unfortunately, how to compose the security and survivability properties of these components and services in a trustworthy manner is poorly understood by the software engineering and research communities. The problematic effects of emergent behavior are accentuated in the Service Oriented Architecture and web services paradigms, which rely on loose coupling and do not lend themselves to comprehensive end-to-end testing. One of the primary goals for this content area is to outline the fundamental software assurance challenges and design considerations posed by assembly, integration, and evolution for managers, software engineers, and researchers, and to thereby help promote improvement in the state of the practice.

The Assembly, Integration, and Evolution (AI&E) outline below describes the documents in this content area. As the BSI website continues to grow, new AI&E documents will be added over time and some of the existing documents will continue to evolve. Contributions from the software engineering community to the AI&E content area are most welcome.

- Assembly, Integration, and Evolution— Content Area Overview (this document)
- Trustworthy Composition: The System is Not Always the Sum of Its Parts: This document surveys several of the profound technical problems and challenges faced by practitioners in the assembly and integration of secure and survivable systems. "It is critical that the practitioner understands the limitations of current techniques and hence maintains a healthy skepticism about the assurance associated with a complex software-intensive system, as well as for any 'silver bullets' proposed to mitigate that complexity."
- Identity in Assembly and Integration: Securely integrating a shared service across highly distributed software systems presents a significant challenge at every phase of the software development life cycle. Moreover, there is a crucial need within the project team(s) for common abstractions and a common understanding of all the relevant aspects of a shared service. This document discusses the issues and necessary abstractions related to integrating identity services, which are particularly critical as the basis for granting or denying access to system resources and data.
- Assembly and Integration Case Study: Enterprise Patch Management: Successfully managing the inevitable changes to an enterprise-wide application is a key aspect of assembly and integration. "If these changes aren't properly managed across platforms and throughout each of the stages of the software development life cycle, production failures, including security problems, can result." This document presents a case study of a Fortune 500 company where deficiencies in patch management left many of the company's servers vulnerable to cyber attack and subsequent infection by the Slammer worm.

- Application Firewalls & Proxies— Introduction and Concept of Operations: Providing a secure operating environment for business-critical applications is among the most crucial steps in the assembly and integration process. This document describes one of the many potential topic areas involving the integration of business applications into a supporting IT security infrastructure. Application firewalls attempt to use application-specific knowledge to improve the perimeter defense that the security infrastructure provides.
- Evolutionary Design of Secure Systems—The First Step is Recognizing the Need for Change: A fundamental truth of system design is that, in the absence of countermeasures, a system's security and survivability will degrade over time. Changes in the environment or usage of a system, or changes to the elements that compose the system, often introduce new or elevated threats that the system was not designed to handle and is ill-prepared to defend itself against. The first step in evolving to meet new threats to your system's security and survivability is to recognize the need to modify your system—that is, to recognize changes in security and survivability risks that trigger the need to enter the evolution phase of the system development life cycle.
- Security Concepts, Challenges, and Design Considerations for Web Services Integration
- The emergence of web services as an integration pattern presents new threats and opportunities for a system's security. Beyond the initial hype, where web services were viewed as a security pandemic, lie both real risks and new security paradigms. Among the most severe security challenges in the web services world are message-level security, interoperability across security protocols, and how to deal with services and systems not under your direct control. Since message exchange is a core part of web services' architectural design, a high-level of security must be built into the messages from the outset, as well as into the services and systems.

This material is based upon work funded and supported by Department of Homeland Security under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center sponsored by the United States Department of Defense.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of Department of Homeland Security or the United States Department of Defense.

DM-0001120