Deploying and Operating Secure Systems

Julia H. Allen

November 2006

ABSTRACT: This article provides a brief overview of deployment and operations security issues, notes to the reader to set expectations, and a recommended order for using the practices described in this content area.

CHALLENGES THAT ORGANIZATIONS FACE

Most organizations that are deploying and operating systems for customers and for themselves (including their own computing infrastructures) do not fully understand the discipline needed to ensure adequate software, computer, and information security (availability, confidentiality, integrity). Current trends indicate that IT operations departments are expending increasing effort to sustain existing system and security capabilities, making it exceedingly difficult to improve performance and add new features, services, products, and technologies.

Some of the factors contributing to this challenge include

- marketplace pressures for increased effectiveness, efficiency, and security that require more agile, timely, and responsive solutions for applications and systems.
- worldwide market factors such as globalization (including offshoring, outsourcing, unknown software provenance (who developed the software and where it was developed), and global supply chains)
- customer-driven requirements that are emergent based on use and their changing needs
- the increasing need for organizations to interconnect with their partners, customers, suppliers, and service providers
- rapidly evolving technology platforms, tools, and other solutions that must be integrated into the operational environment

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

- lack of education on what it takes to deploy and operate secure software, particularly more secure, web-based applications[1]

Physical security, authentication, and firewalls defend against external threats, but employees and contractors are authorized to bypass these measures. Current and former employees and contractors who have or had authorized access to their organization's system and networks are familiar with internal policies, procedures, and technology and can exploit that knowledge to facilitate attacks and even collude with external attackers. External and insider threats[2] need to be mitigated during deployment and operations.

The policies, procedures, processes, controls, and performance measures selected, enforced, and continuously improved during deployment and operation of systems often determine whether software and systems will function in a secure and survivable manner.

NOTES TO THE READER

Practices specific to the deployment and operations of software and systems that have been developed using a secure development life cycle approach are not well documented, given that this is a relatively new venture for most development organizations. A great deal is known about how to develop secure software (the Build Security In web site being a case in point) but sufficient time has not passed for the broad adoption of secure development practices or to capture and analyze how systems behave when they have been developed with security in mind.

Thus, this set of articles focuses on the current state of practice, which is providing an environment within which systems and software are more likely to operate securely. It is encouraging to note that many accepted standards, frameworks, and guidelines for providing a secure operating environment (described here) are now including practices for secure acquisition and development.

---

[1] Refer to efforts of the Open Web Application Security Project (http://www.owasp.org/) and the Web Application Security Consortium (http://www.webappsec.org/) for more on this subject.

[2] Research conducted by CERT since 2001 has focused on gathering data about actual malicious insider acts, including sabotage, fraud, theft of confidential or proprietary information, and potential threats to our nation's critical infrastructure. CERT's insider threat research is based on actual compromises and focuses on attack methods and tools, precursor activity, and how the insiders were detected and identified, including during the SDLC [CERT 08a].

HOW TO USE THE ARTICLES IN THIS CONTENT AREA

Articles 1 through 4 present a recommended order of practices to tackle. Of course, like every improvement life cycle, this recommended progression is iterative and can start with a small scope that expands over time:

- First, put an improvement cycle in place (even a modest one), confirm that prerequisite practices and the results they produce exist or work to do this, and make sure basic security hygiene practices are deployed. Consider and select a security practice implementation framework (Article 1: Plan, Do, Check, Act).
- Next, use a risk-centered approach to determine what assets (systems, software) are most critical to protect and the mitigating actions to protect these assets. This includes identifying what security practices to deploy and in what order (Article 2: Risk-Centered Practices).
- Then, ensure that appropriate security practices and controls are embedded within hopefully mature and well-defined IT operational processes. Learn from organizations that are doing this particularly well (Article 3: Integrating Security and IT, Article 4: Prioritizing IT Controls for Effective Security).
- Finally, review the range of available standards, frameworks, and guidelines on operational security practices for implementation guidance. These have all been in use for a number of years and are well vetted by their respective communities and market sectors (Article 5: Navigating the Security Practice Landscape). Several new efforts are emerging that begin to address the deployment and operation of software that has been developed using software security practices.[3]

Articles 1, 2, and 5 include extensive tables at the end of each article that provide supporting details and sources for the recommendations made in these articles. The tables are placed at the end to make the articles easier to read and follow.

---

[3] Microsoft's Security Development Lifecycle (SDL) Version 3.2 describes several relevant deployment practices (http://www.microsoft.com/downloads/details.aspx?familyid=2412C443-27F6-4AAC-9883-F55BA5B01814&displaylang=en). They are also starting to document SDL improvement results (http://msdn.microsoft.com/en-us/security/cc424866.aspx). The Software Assurance Forum for Excellence in Code (SAFECode; http://www.safecode.org) has published several reports that describe practices for secure software development. Gary McGraw and Brian Chess are working towards a maturity model for software security (http://www.informit.com/articles/article.aspx?p=1271382).

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.

DM-0001120