SAFECode
Software Assurance Forum for Excellence in Code
Driving Security and Integrity

Reference tables from Section 2b of
Practical Security Stories and Security Tasks for Agile Development Environments
www.safecode.org/publications/SAFECode_Agile_Dev_Security0712.pdf

Operational Security Tasks

No.	Operational Security Task	Requirement/Recommendation
1	Configure bug tracking to track security vulnerabilities	Requirement for software development team
2	Verify security POCs and plan for fixes	Recommendation for software development team
3	Use latest compiler versions	Recommendation for new code
4	Resolve critical and high severity issues identified by static code analysis tools	Requirement for new code and recommendation for existing code
5	Keep track of patches/fixes to third party dependencies	Requirement for new as well as existing code
6	Keep track of patches/fixes to OS components	Requirement for new as well as existing code
7	Perform stricter code review of 'risky' code **	Requirement for new as well as existing code
8	Use appropriate security-related flags for compiler	Requirement for new as well as existing code
9	Continuously verify coverage of static code analysis tools	Requirement for new as well as existing code
10	Perform (and add to testing cycle) automated vulnerability scanner (OS and web as appropriate)	Requirement for software development team
11	Perform (and add to release cycle) automated malware scanner on released binaries	Requirement for software development team
12	Use secure versions of communication protocols	Requirement for new code and recommendation for existing code
13	Ensure inclusion of security patches/fixes applied in previous release(s)	Requirement for software development team
14	Ensure all developers have obtained secure coding training	Requirement for software development team
15	Ensure all QA engineers have obtained secure testing training	Requirement for software development team
16	Ensure security fixes are verified by security experts before committing them	Requirement for software development team
17	Periodically check to ensure that your SSL certificates (and all certificates above it in its chain of trust) have not expired or been revoked	Requirement for operational team

** Type of code that fits this category is as follows:

No.	Risky code category requiring a focused code review
1	Windows services and *nix daemons listening on network connections
2	Windows services/applications running as SYSTEM/Admin or *nix daemons running as root
3	Code listening on unauthenticated network ports connections
4	ActiveX controls
5	User (direct or indirect) input validation code
6	setuid root applications on *nix
7	Code that parses data from non-admin/non-root writeable files, email attachments, /temp directory, web downloaded files, log viewer code, event viewer code, report generators, file paths values, files that use UNC, lock files, application critical files such as config. files, dll load paths
8	Code that interfaces with third-party module(s)
9	Code that interfaces with C/C++/any ordinary executable files
10	Code dealing with authentication/authorization/encryption/any other core security features
11	Code that implements proprietary protocol OR handles proprietary implementation of standard protocols

For more information, please visit www.safecode.org.

SAFECode
Software Assurance Forum for Excellence in Code
(p) +1 781-876-8833 (f) +1 781-224-1239
feedback@safecode.org
www.safecode.org
Twitter: @SAFECodeForum
Facebook: www.facebook.com/SAFECode