



build | integrate | secure

Turning the Battleship: How to Build Secure Software in Large Organizations

Dan Cornell
May 11th, 2006

Overview

- Background and key questions
- Quick review of web application security
- The web application security scanner: a tool not a panacea
- To create secure software, build a process that embraces security
- Navigating organizational boundaries still a challenge
- Conclusion and Q&A

Key Learning Points

- Building a comprehensive application security program is not as easy as running an app vulnerability scanner or installing an application security firewall
- In order to build secure software, you need to have a software development lifecycle that considers security implications at every step
- You must overcome certain organizational, cultural, and business realities that prevent a large organization from building secure software on a consistent basis

Denim Group Background

- SA-based consultancy that builds and secures large-scale web applications
- Application development experience provides valued perspective on all aspects of software development process
- Application security services include:
 - *Black-box and white-box assessments*
 - *Secure application development and remediation*
 - *Application security training for developers, security professionals, and auditors*
 - *Software development lifecycle development (SDLC) consulting*
 - *Application security tool development*
- Sponsors local Open Web Application Security Project Chapter

Myself

- Dan Cornell
- Founder of/Partner in Denim Group
- Application developer
 - J2EE
 - .NET
 - LAMP (Linux, Apache, MySQL, Perl/PHP)
 - ASP, VB, ColdFusion, Python, C++, ...
- Started doing work in application security area after working on a number of e-commerce and other high-security-required applications

Key Questions

- Why is it that serious web application vulnerabilities still exist in organizations that have been conducting network and host-based assessments for years?
- How do information security professionals reduce the risk that Internet-facing applications represent to the enterprise?
- How can they quantify the risk when application security scanners identify only 30% of the most serious flaws that exist in large-scale web software systems?

OWASP Top 10 Critical Web Application Security Vulnerabilities

- Unvalidated Input
- Broken Access Control
- Broken Authentication and Authorization
- Cross Site Scripting (XSS)
- Buffer Overflows
- Injection Flaws
- Improper Error Handling
- Insecure Storage
- Denial of Service
- Insecure Configuration Management

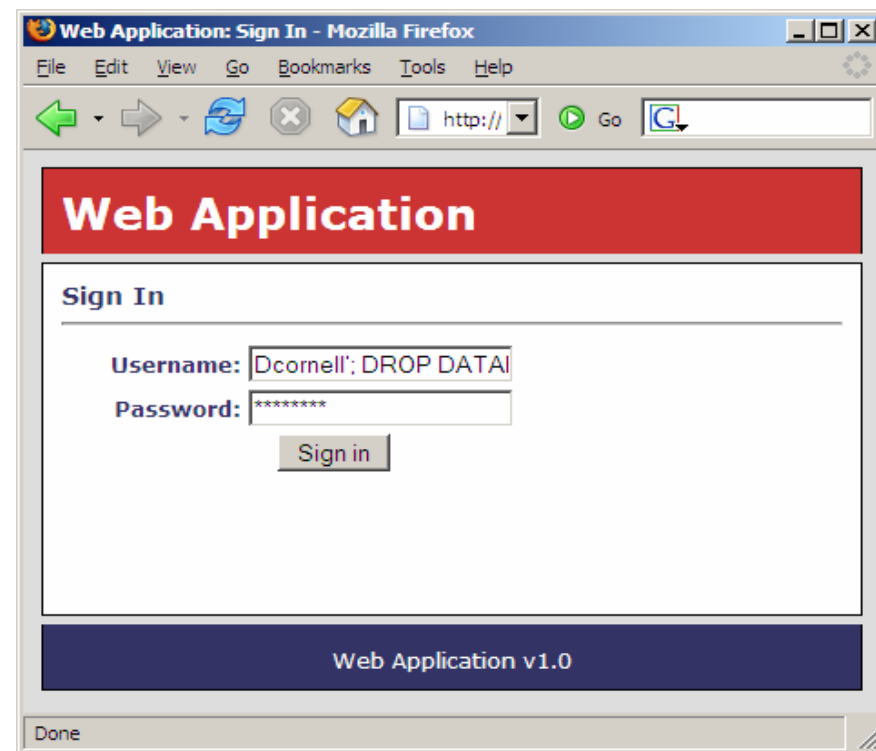
<http://www.owasp.org/documentation/topten.html>

Example App Vulnerability: SQL Injection

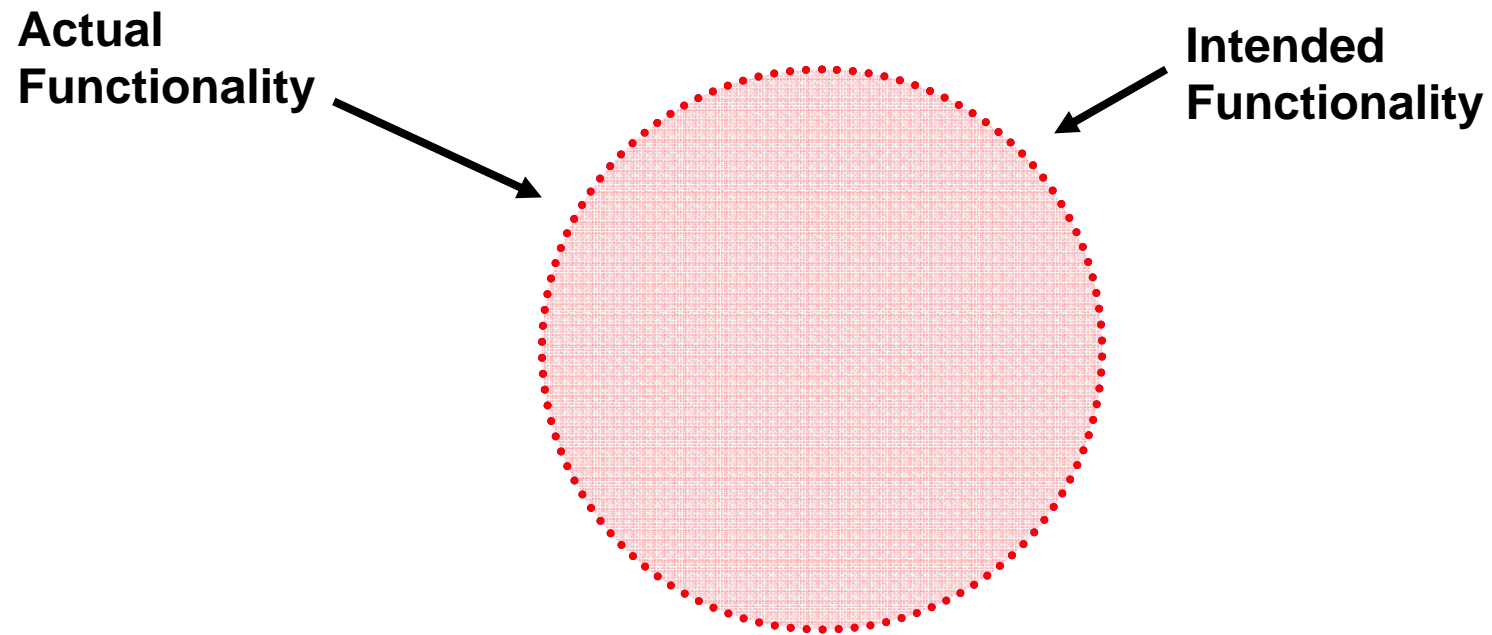
- SQL statements are created from a combination of static text and user inputs
- Assumption: users will enter well-formed inputs
- Attacker crafts a custom input to hijack control of the SQL interpreter and execute arbitrary code
- Very common flaw with tremendous security implications

Example App Vulnerability: SQL Injection

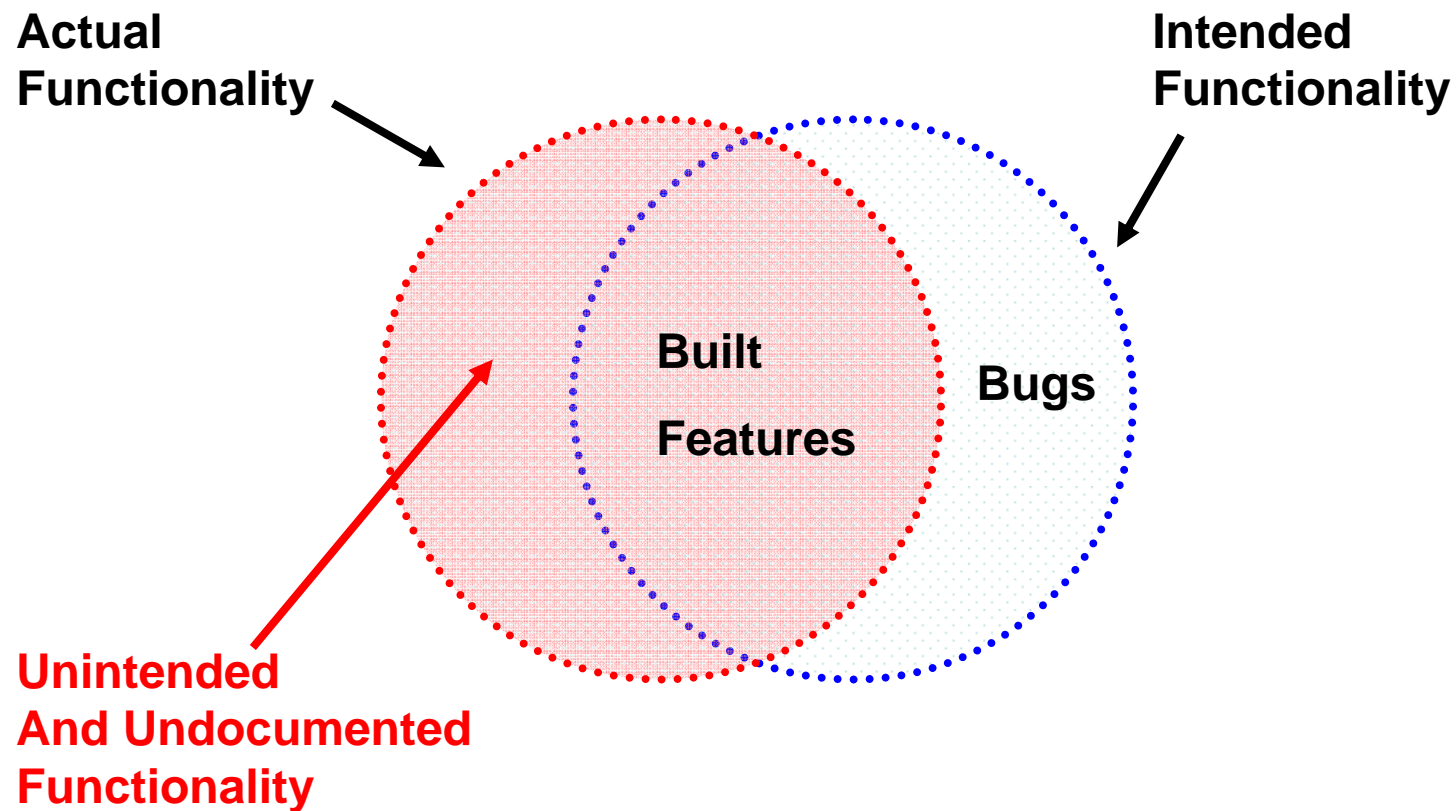
- Specially crafted input contains SQL control characters
- Malicious user sends in a username parameter of:
Dcornell'; DROP DATABASE Ecommerce; --
- Attacker can execute arbitrary database queries with the same permissions as the application
 - *View sensitive data*
 - *Modify data*
 - *Destroy data*



Software Implementation – Perfect World



Software Implementation – Real World



Application Security Scanners Background

- Very adept at identifying technical flaws in applications via black-box approach
- Automated crawling of large applications essential
- Through 2010, enterprise that scan their web applications will experience a 70% reduction in security incidents in these apps*
- By 2008, at least 40% of enterprises will have adopted web scanning tools as part of dev process*
- Best of breed tools include SPI Dynamic's WebInspect and Watchfire's AppScan; Acunetix getting market attention

– Gartner 2006

Application Security Scanners Background

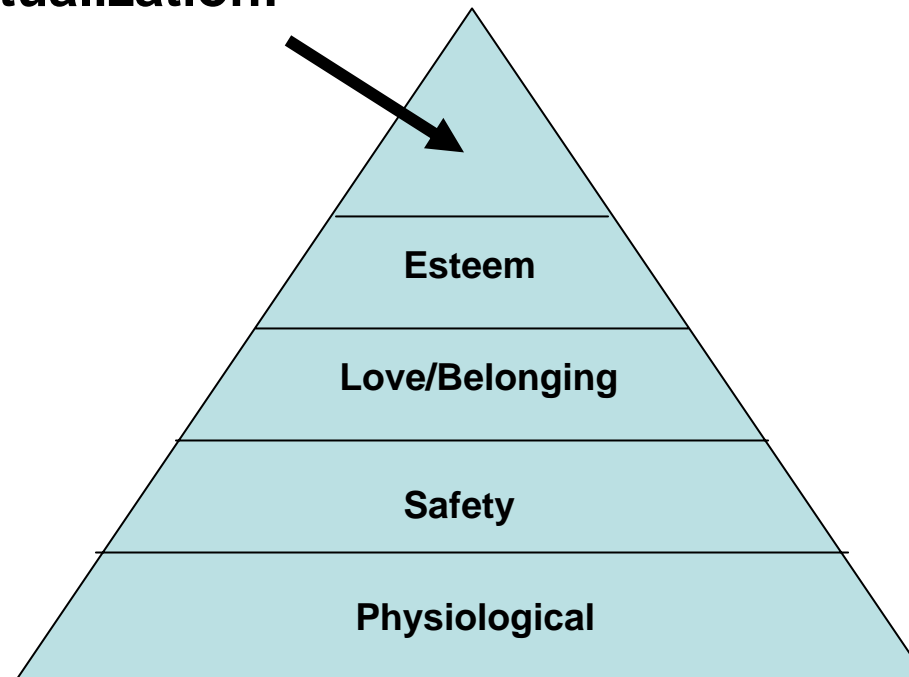
- Conventional wisdom is that scanners only get 30% of types of vulnerabilities
- Scanners are almost powerless to identify logical errors
 - *Usually these are the scariest vulnerabilities*
 - *Authentication, authorization, trust assumptions, session management*
- In the hands of most networks security professionals, results difficult to interpret
 - *Even tougher to provide recommendations to developers*
 - *Numerous examples of security groups “spinning their wheels”*
- Application security scanners identify vulnerabilities that need to be remediated, not patched
- Ultimately, as a standalone process, scanning alone creates a significant false sense of security

Application Security Scanning Recommendations

- Recommendations
 - *Conduct qualitative risk ranking of applications deployed*
 - Internet-facing and business critical applications first!
 - *Scan applications in black-box mode*
 - *Perform focused white-box code review of “hot spots”*
 - *Provide remediation recommendations based upon trade-offs*
 - *Consider integrating results into tracking or QA systems*
 - Ultimately application security should become part of application quality
 - *Augment your team with internal or external resources that are web development savvy*
 - Auditors consider auditing scanning process and not applications themselves

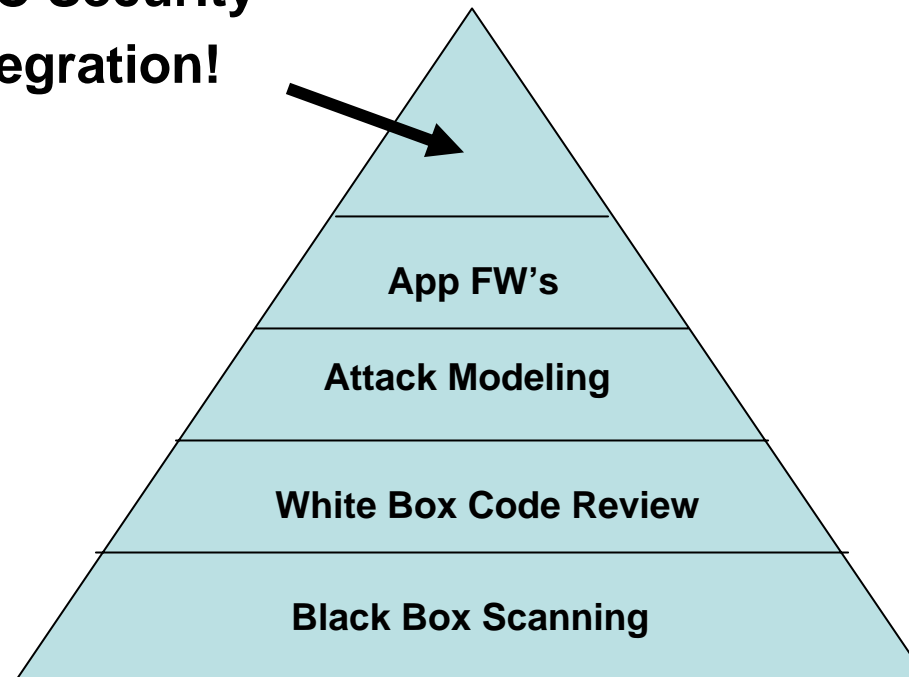
Maslow's Hierarchy of Human Needs

Actualization!



Dickson's Hierarchy of AppSec Needs

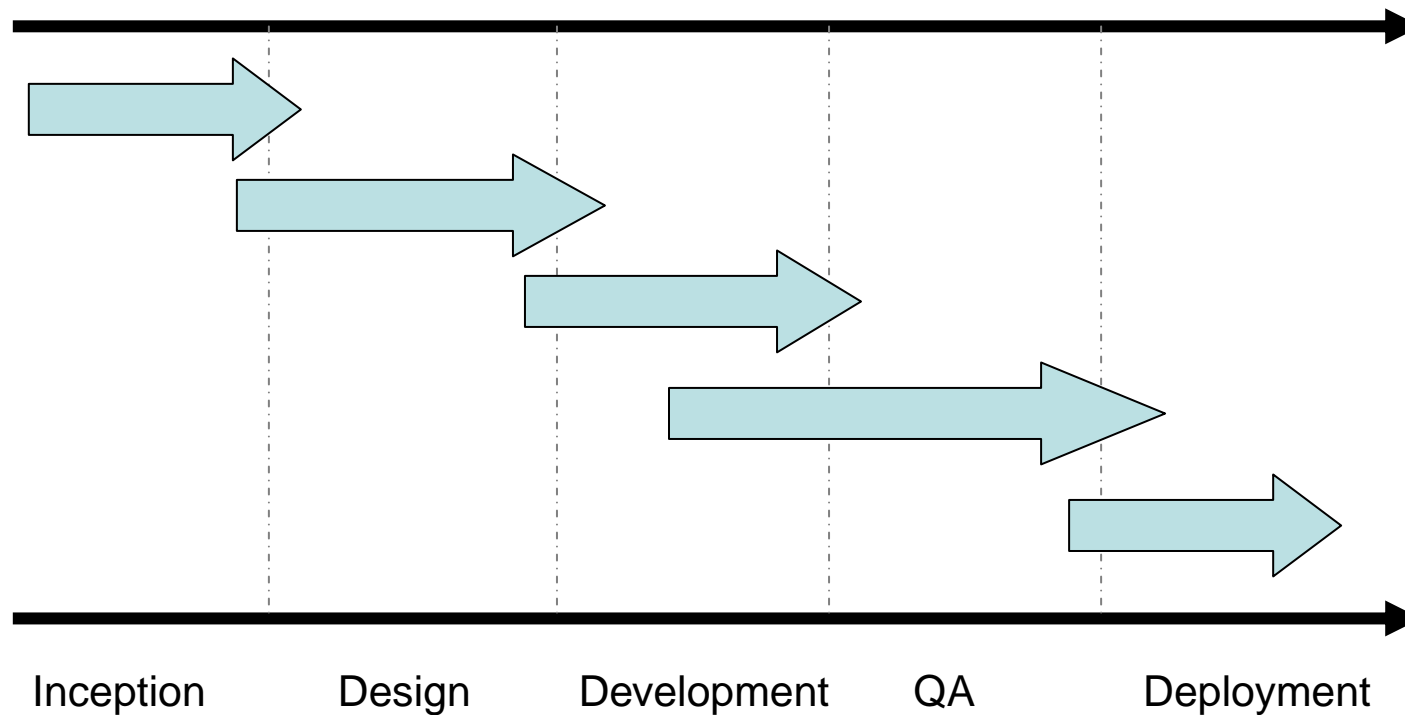
**SDLC Security
Integration!**



Building a Better and More Secure SDLC

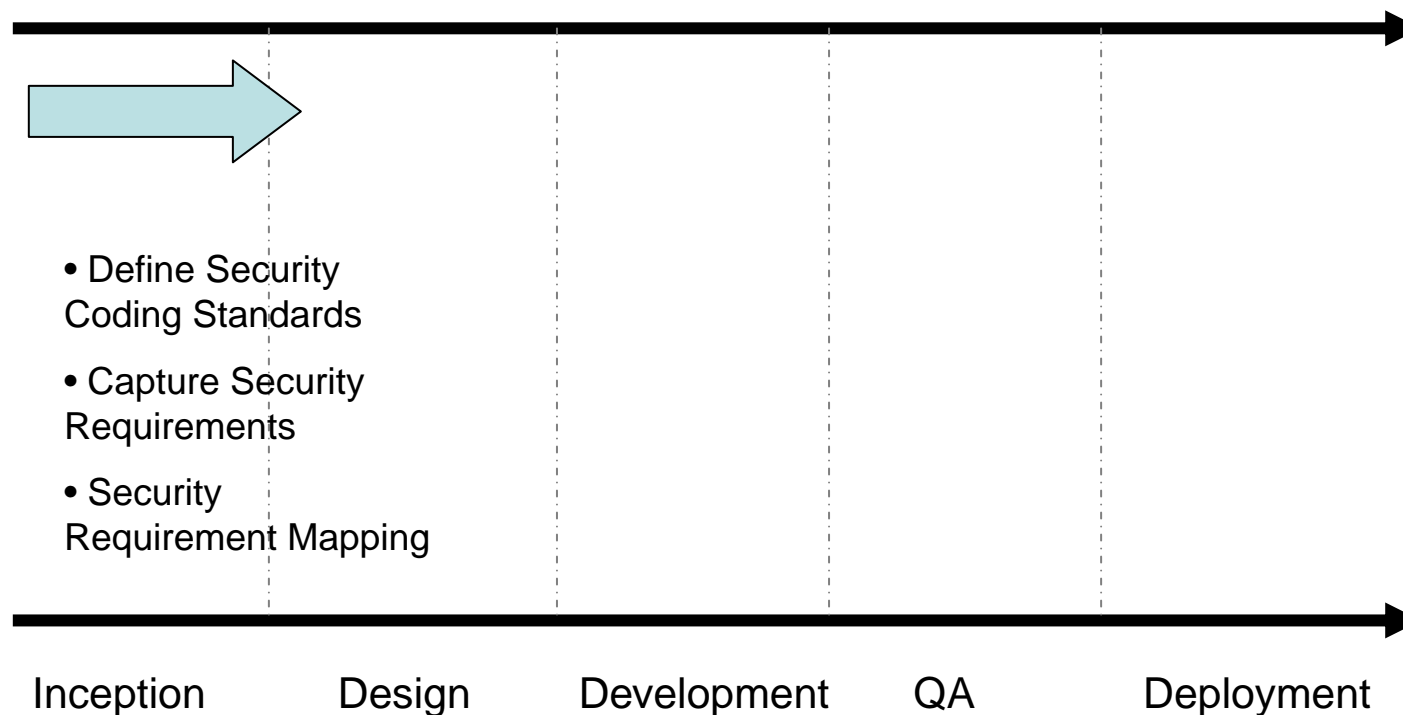
- The iterative nature and rapid development of web software drives security throughout the process
- Different players (audit, security, architecture, app dev, and PM) need to know when they enter and exit the process
- Security professionals need a more fundamental understanding of their organization's development processes
 - *MSF vs. waterfall?*
 - *Scrum vs. XP*
- Different security concepts apply to different points of the SDLC
 - *Inception, design, development, QA, and deployment*

Security Integration Points within the SDLC



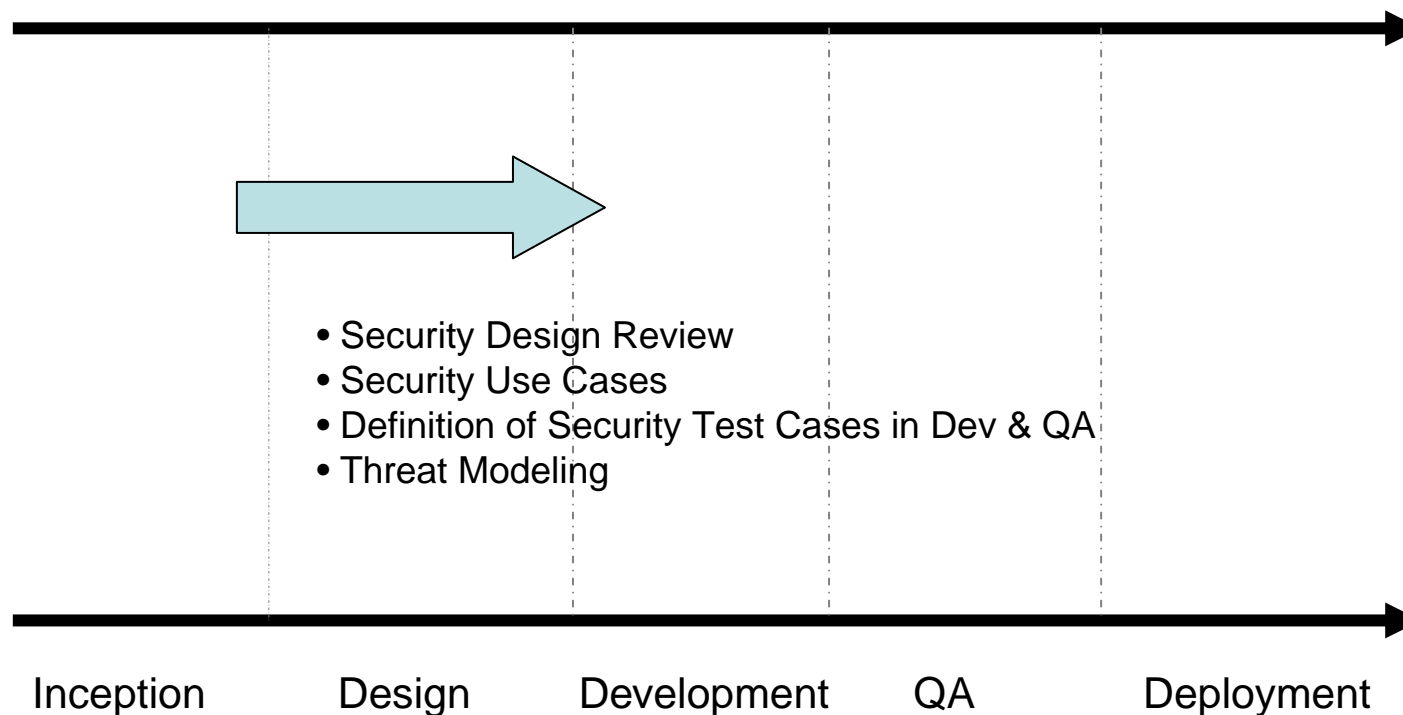
Source: Gartner (February 2006)

Security Integration Points within the SDLC



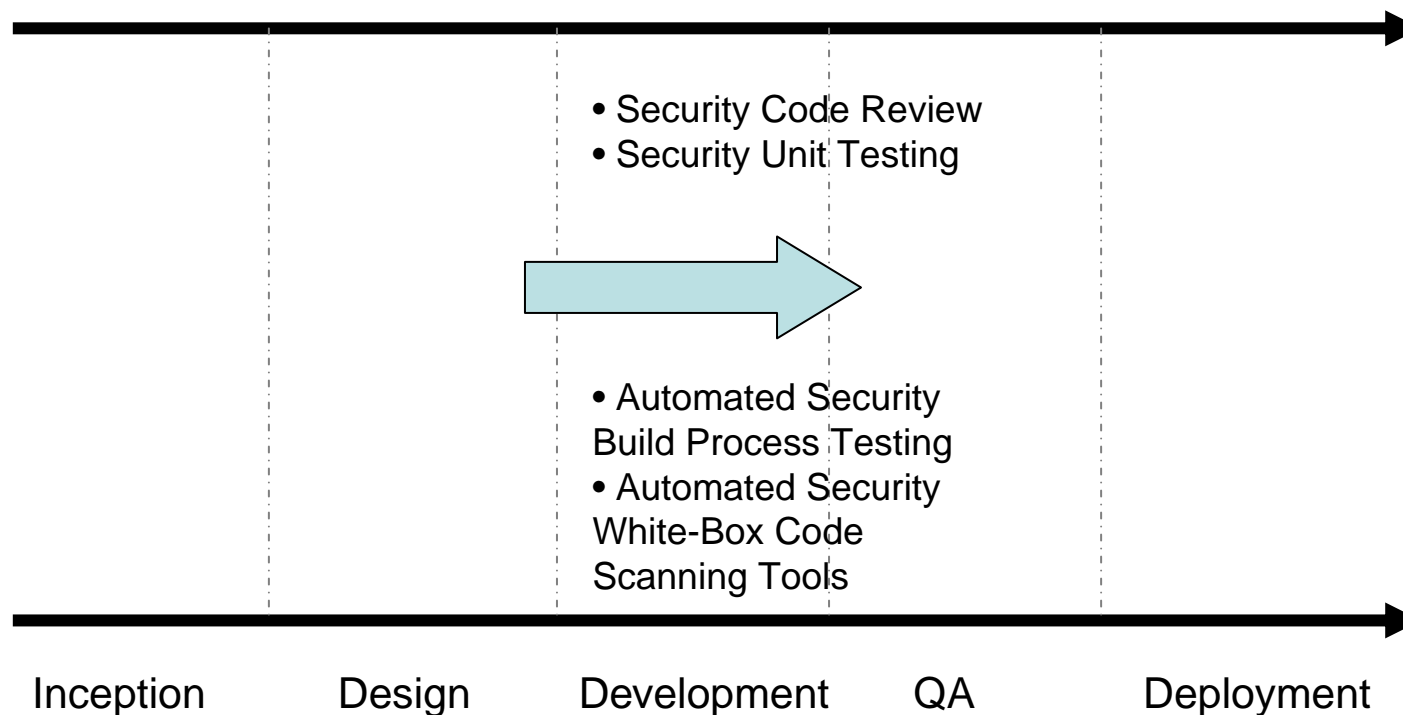
Source: Gartner (February 2006)

Security Integration Points within the SDLC



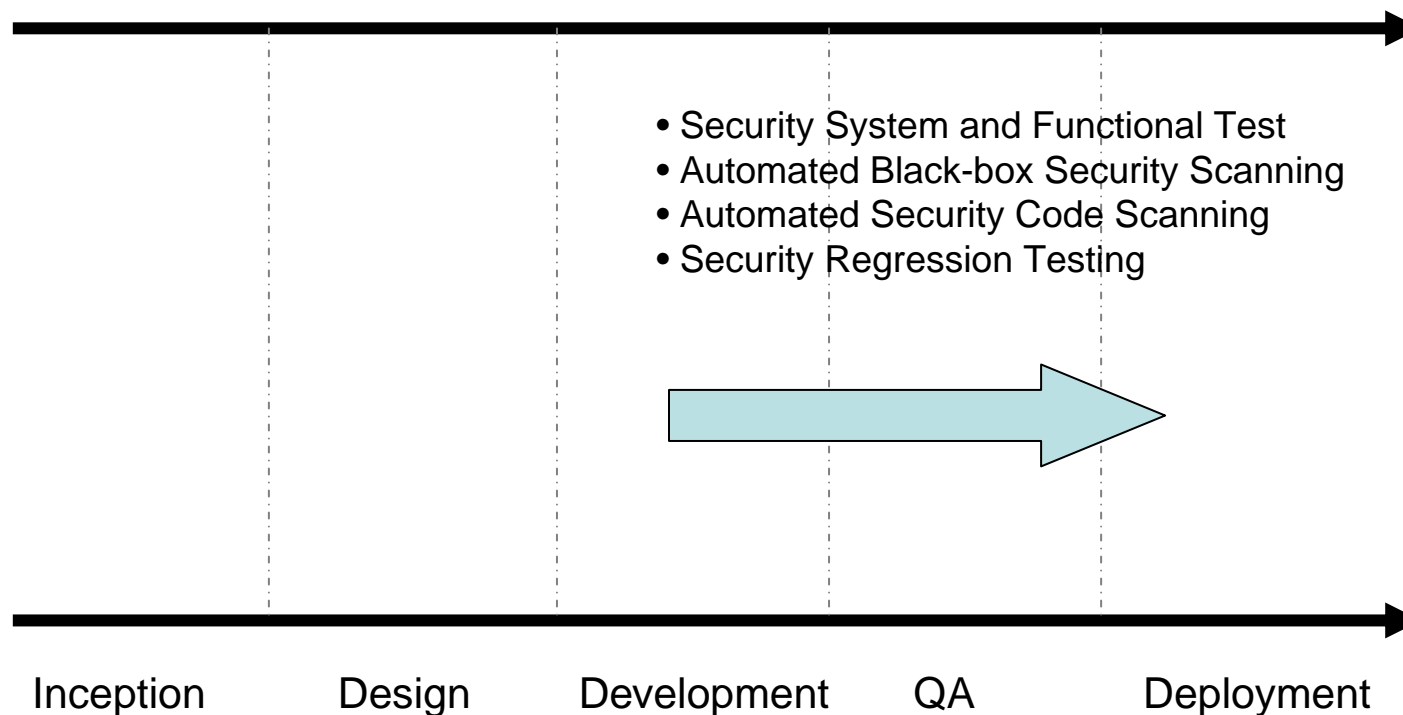
Source: Gartner (February 2006)

Security Integration Points within the SDLC



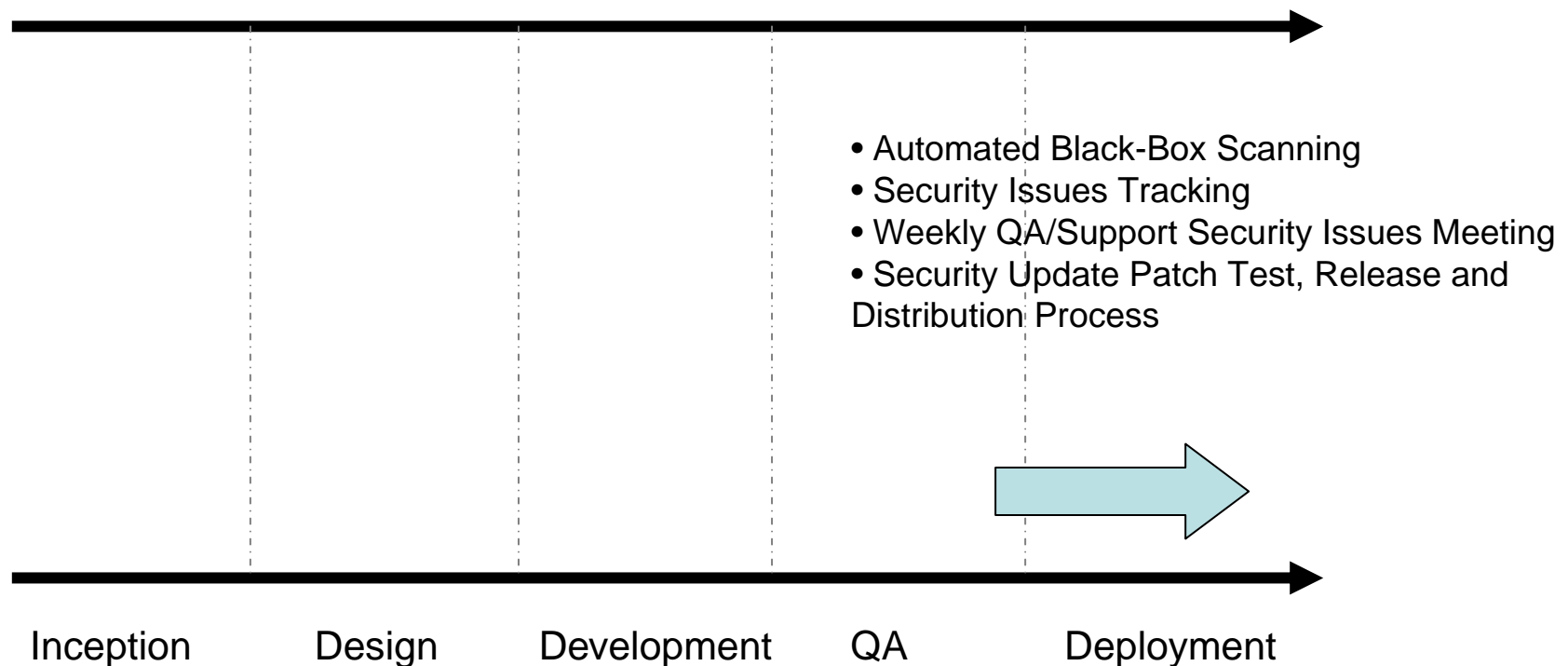
Source: Gartner (February 2006)

Security Integration Points within the SDLC



Source: Gartner (February 2006)

Security Integration Points within the SDLC



Source: Gartner (February 2006)

Bridge Cultural Gap Between Security and Developers

- A huge roadblock to implementing secure software
- Key Challenge: Build vs. Measure Cultures
- Application Development groups are building technical capabilities based upon evolving business requirements
- Corporate IS Security dept. in charge of ongoing security operations
- Although mostly security managers worry about secure software, ultimately it will be the development teams that solve the problem.
- Results of informal survey!

Conclusion

- Application security scanning is a first step to tackling the application security problem
- Ultimately, you need to help build a software development lifecycle that considers security implications at every step
- Organizational, cultural, and business sometimes are a bigger challenge than technical issues to fixing the problem

Questions & Answers

- Dan Cornell
 - dan@denimgroup.com, 210.572.4400

Web Site: www.denimgroup.com

Blog: denimgroup.typepad.com

- Jumpstart Document and OWASP “A Guide to Building Secure Applications and Secure Web Services” available upon request