Software Security: State of the Practice 2010

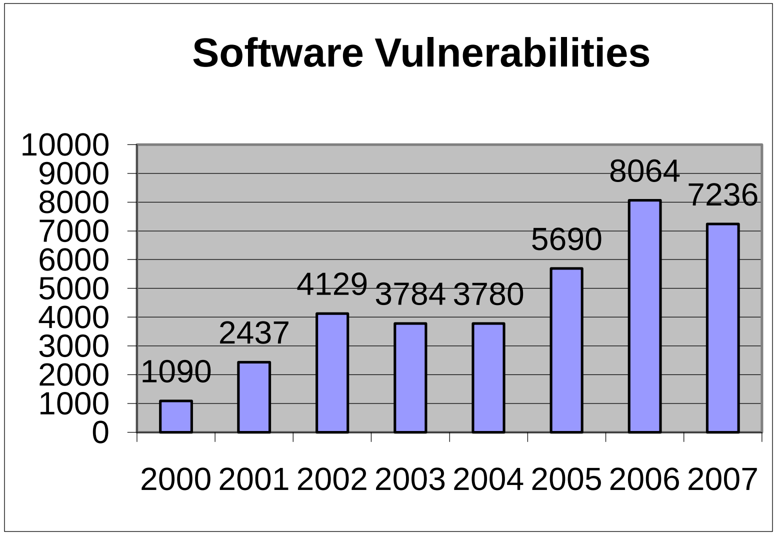
Gary McGraw, Ph.D.
Chief Technology Officer, Cigital

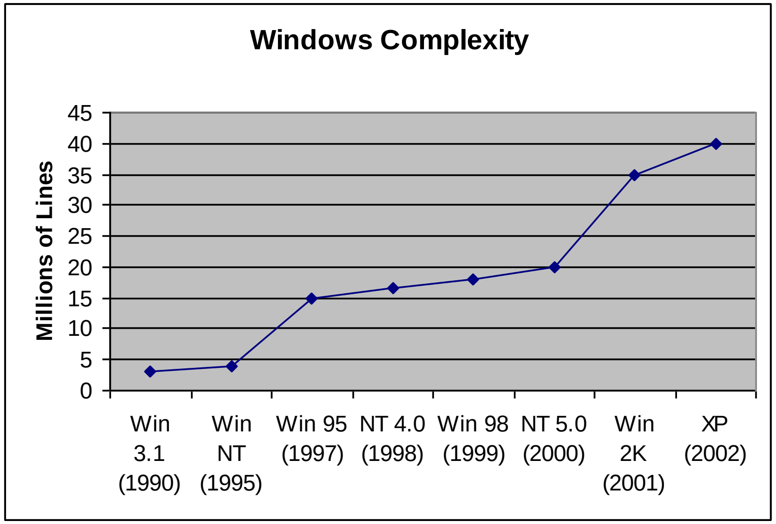
Cigital

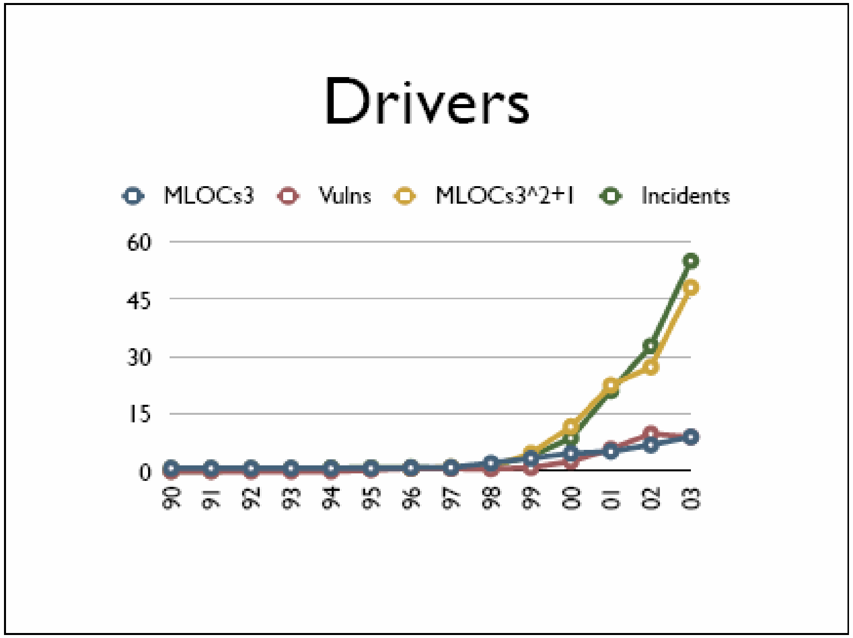
- Founded in 1992 to provide software security and software quality professional services
- Recognized experts in software security and software quality
  - Widely published in books, white papers, and articles
  - Industry thought leaders

Awareness

More code, more bugs







Security as a differentiator

- Apple sells iMac and MacBook with security
- Firefox sells browser with security

Diversity works

- We see both .NET and J2EE
- We see Oracle, SQL, and DB2
- We see Unix, Linux, AIX, Windows, OSX
- All in the same location

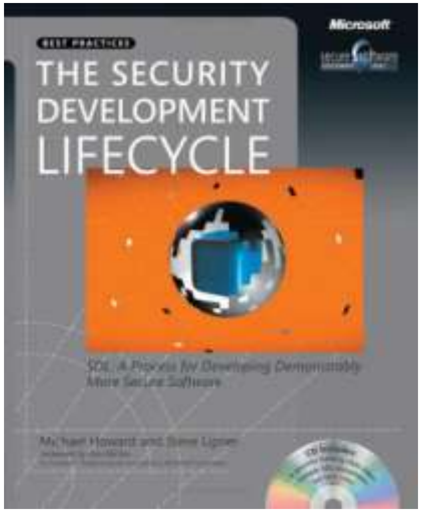
The rise of the software security group

- Cigital SSG turns ten
- Microsoft adopts the Secure Development Lifecycle
- Many companies have a group devoted to software security (58)

- microsoft
- dtcc
- emc
- fidelity
- adobe
- wells fargo
- goldman sachs
- google
- qualcomm
- morgan stanley
- usaf
- dell
- pershing
- the hartford
- barclays capital
- bank of tokyo
- ups
- bank of montreal
- sterling commerce
- time warner
- cisco
- bank of america
- walmart
- finra
- vanguard
- college board
- oracle
- state street
- omgeo
- motorola
- general electric
- lockheed martin
- intuit
- vmware
- amex
- bank of ny mellon
- harris bank
- paypal
- symantec
- visa europe
- thomson/reuters
- BP
- SAP
- nokia
- ebay
- mckesson
- ABN/amro
- ING
- telecom italia
- swift
- standard life
- cigna
- AON
- coke
- mastercard
- apple
- AOL
- CA

A shift from philosophy to HOW TO

- Integrating best practices into large organizations
  - Microsoft’s SDL
  - Cigital’s touchpoints
  - OWASP adopts CLASP


Microsoft
BEST PRACTICES
THE SECURITY DEVELOPMENT LIFECYCLE
SDL: A Process for Developing Demonstrably More Secure Software
Michael Howard and Steve Lipner

What works: BSIMM

- Building Security In Maturity Model
- Real data from real initiatives
- Descriptive (not prescriptive)
- http://bsi-mm.com

cigital

FORTIFY

The software security market grows (2006-7-8)

Code Review [$55M→95.4M→126.9M]

- Fortify [$15.9M→29.2M→41M]
- Secure Software (Fortify) [$2M]
- Ounce Labs [$3.1M→9.5M→9.1M]
- Coverity [$18M→27.2M→35.36M]
- Klokwork[$16M→26.0M→36.4M]

Web Application Black Box [$53.9M→55.1M→74.13M]

- IBM/Watchfire [$26M→24.1M→32.13M]
- HP/SPI Dynamics [$21.2M→22.3M→25M]
- Cenzic, Codenomicon, Whitehat, … [$12.5M→17M]

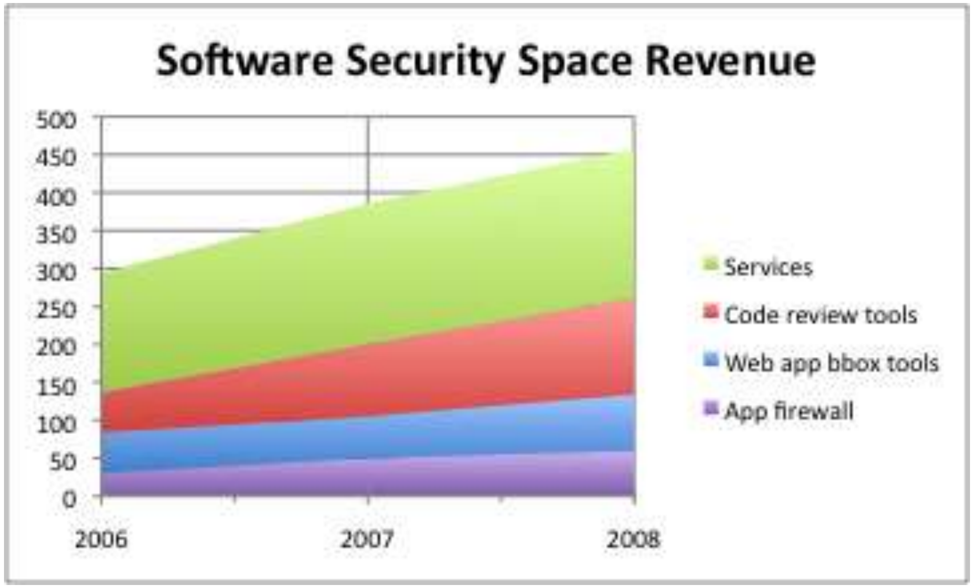
- Application firewalls [$30M→50M →60M]

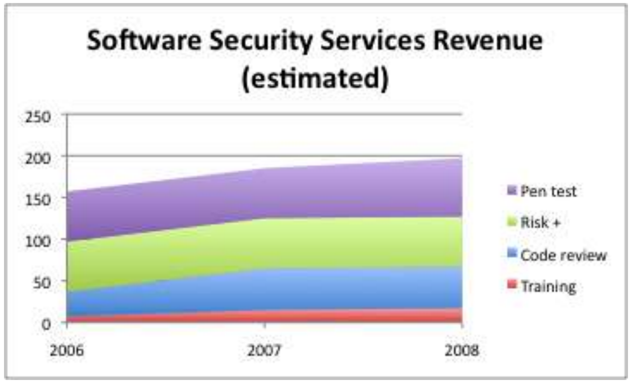
- Software security services both around tools and other assessments [$157M→185M→197M]
  - Cigital, Foundstone, E&Y, IBM, Cybertrust

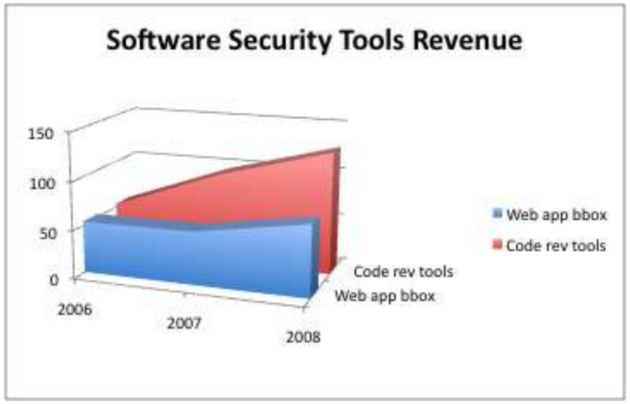
- Total estimate = $295M→385M→458M
- http://www.informit.com/articles/article.aspx?p=1338343

badness-ometers lead to awareness

Growth of market segments


Software Security Space Revenue

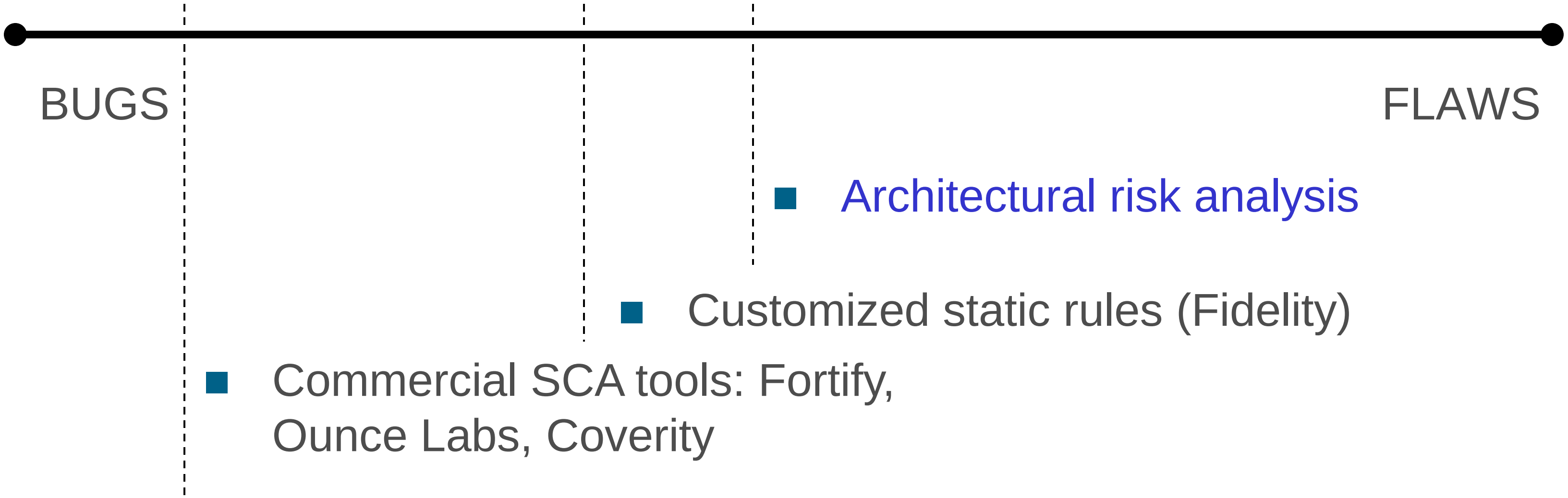

Software Security Services Revenue (estimated)


Software Security Tools Revenue

The bugs/flaws continuum

gets()

attacker in the middle



BUGS

FLAWS

- Architectural risk analysis

- Customized static rules (Fidelity)

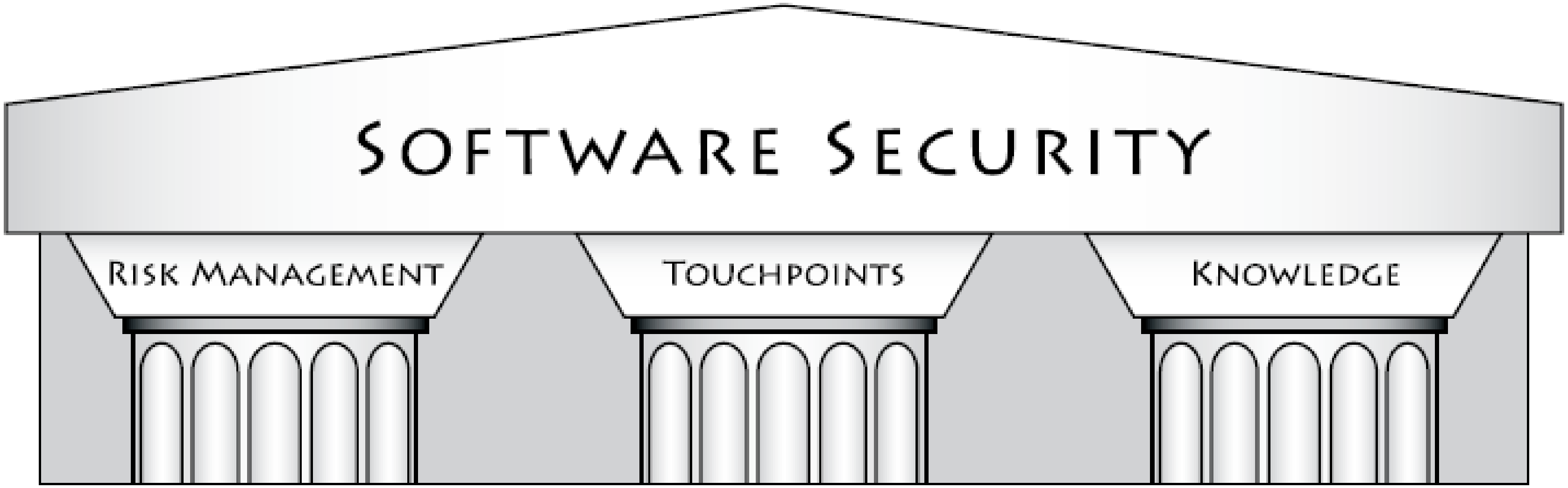
- Commercial SCA tools: Fortify, Ounce Labs, Coverity

- Open source tools: ITS4, RATS, grep()

Software security common sense

- Software security is more than a set of security functions
  - Not magic crypto fairy dust
  - Not silver-bullet security mechanisms
- Non-functional aspects of design are essential
- Bugs and flaws are 50/50
- Security is an emergent property of the entire system (just like quality)
- To end up with secure software, deep integration with the SDLC is necessary

Three Pillars of Software Security



Three pillars of software security

- Risk management framework
- Touchpoints
- Knowledge

Risk Management Framework

Why risk management?

- Business understands the idea of risk, even software risk
- Technical perfection is impossible
  - There is no such thing as 100% security
  - Perfect quality is a myth
- Technical problems do not always spur action
  - Answer the “Who cares?” question explicitly
- Help customers understand what they should *do* about software risk
- Build better software

Who cares?

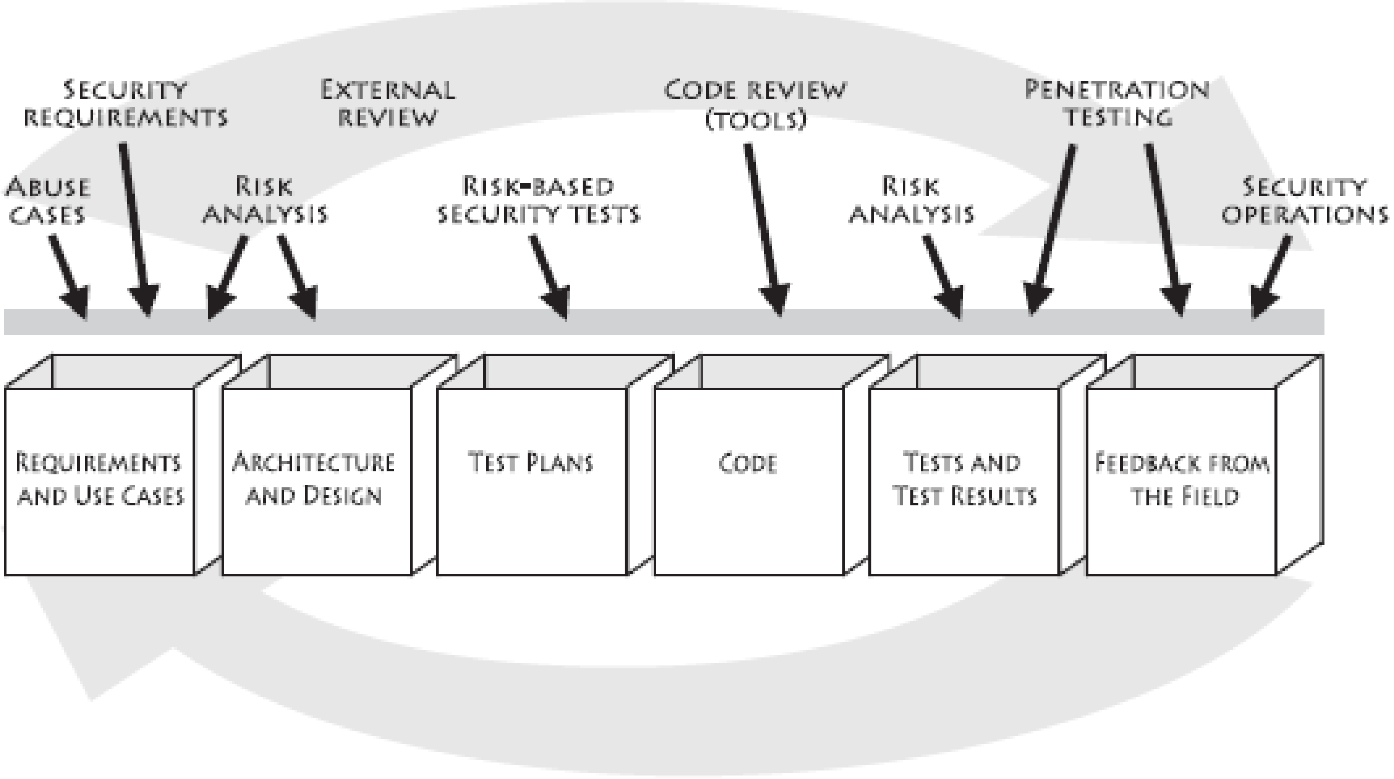
Financial vertical leads the pack

- All major investment banks have a Chief Risk Officer
  - SOX caused banks to realize their software risk
  - Software security initiatives resulted
- Credit card consortiums recognize software security in PCI standards
- Software vendors and high tech companies have a much harder time connecting to business

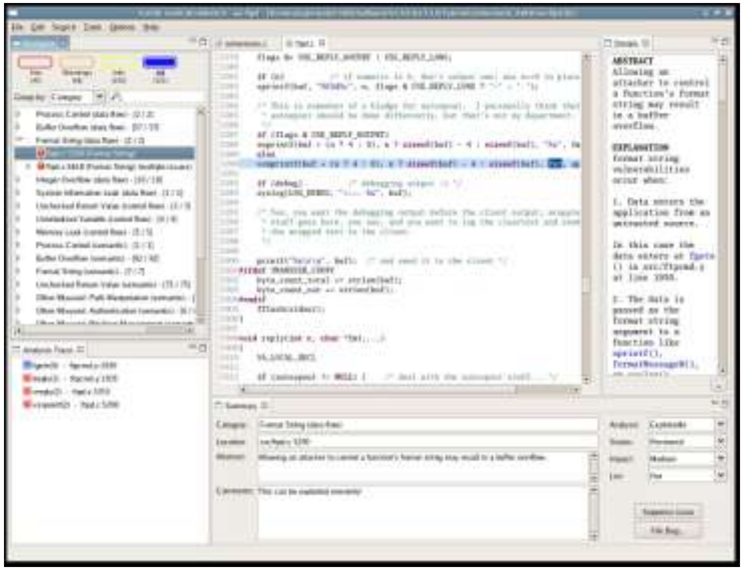
PCi Security Standards Council™

Software Security Touchpoints

Software security touchpoints



Touchpoint: Code review (with a tool)



ABSTRACT
Allowing an attacker to control a function's format string may result in a buffer overflow.

EXPLANATION
Format string vulnerabilities occur when:

1. Data enters the application from an untrusted source.

In this case the data enters at fgets() in src/ftpcmd.y at line 1958.

2. The data is passed as the format string argument to a function like sprintf(), FormatMessageW(), ...

Abstract: Allowing an attacker to control a function's format string may result in a buffer overflow.

Comments: This can be exploited remotely

FORTIFY

coverity

OUNCE LABS

- Code scanning catches on
  - Demand for manual services up
  - Tool adoption proceeding apace (being measured)
- Tools (finally) handle large code bases
  - Don't fail to grep()
  - Simple enforcement is no longer useful
- Customization pays off royally
  - Fidelity
  - DTCC
- Training courses about bugs and tools widespread

Fidelity leads the pack

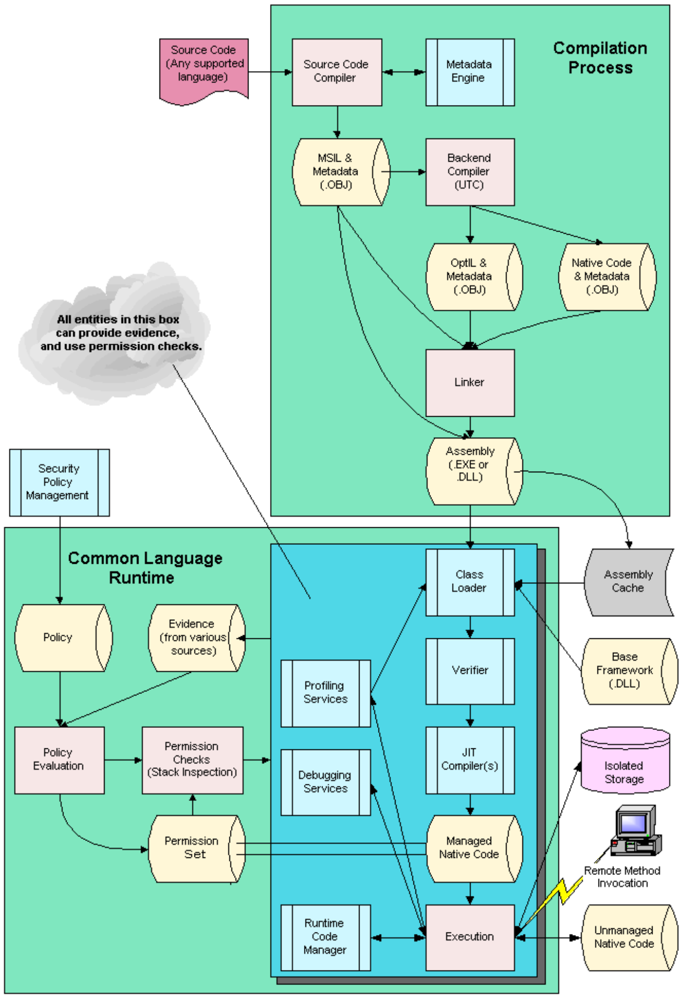
- Corporate-wide adoption of the tool
- Creation of rules
  - Corporate standards enforcement (DES vs 3DES)
  - Custom rules push past the tool’s natural limits
  - Custom rules look at more constraints surrounding a particular code construct (false positives drop)
- Application assessment factory
  - Code that builds in
  - Actionable bugs out
  - Hide the assembly line behind an API for better management

  - http://www.informit.com/articles/article.aspx?p=1231818

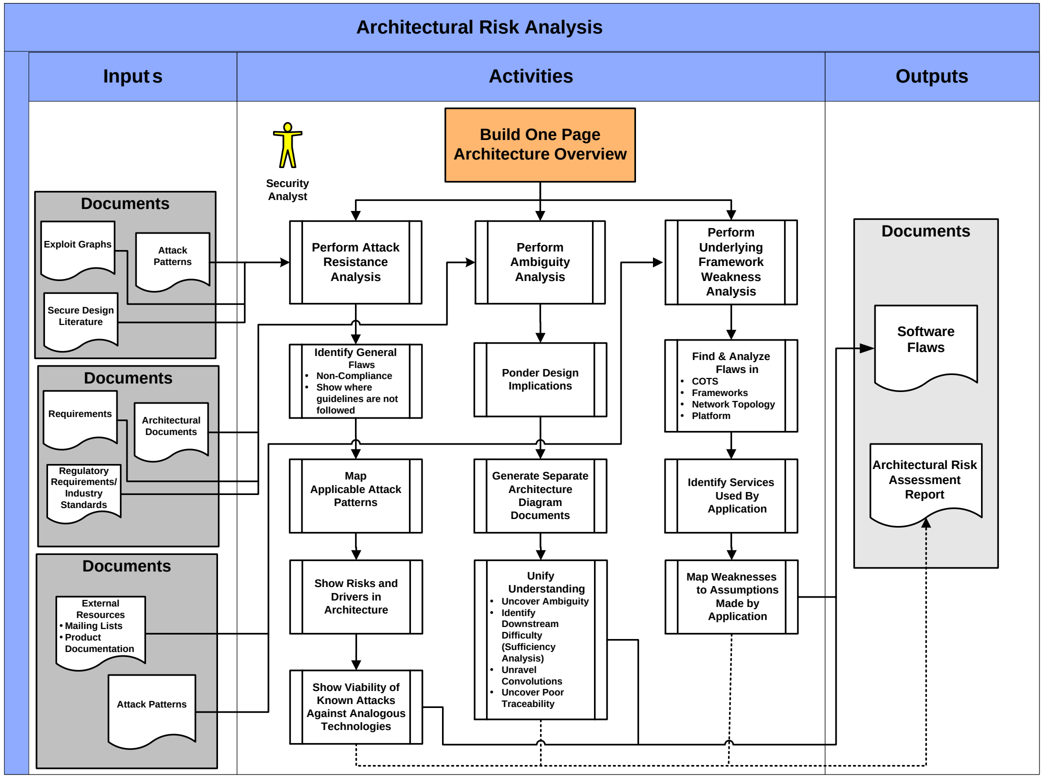
Touchpoint: Architectural risk analysis

- More common to find customers with a handle on software architecture
- Widespread use of common components
  - Spring
  - Hibernate
  - Log4J
  - OpenSSL
  - “ripple effect”
- Design patterns help

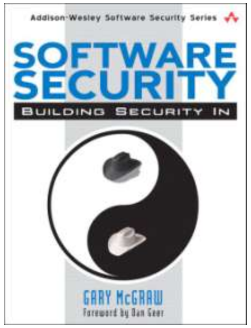
- High-expertise work is still hard to teach
- Training courses about ARA just being adopted



Touchpoint: Architectural risk analysis

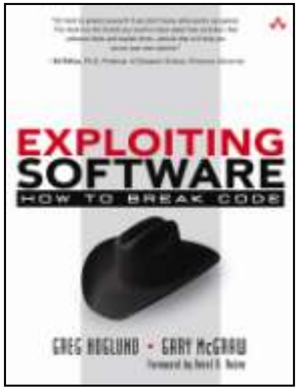

Architectural Risk Analysis
Inputs
Activities
Outputs
Security Analyst
Build One Page Architecture Overview
Documents
Exploit Graphs
Attack Patterns
Secure Design Literature
Documents
Requirements
Architectural Documents
Regulatory Requirements/ Industry Standards
Documents
External Resources
• Mailing Lists
• Product Documentation
Attack Patterns
Perform Attack Resistance Analysis
Identify General Flaws
• Non-Compliance
• Show where guidelines are not followed
Map Applicable Attack Patterns
Show Risks and Drivers in Architecture
Show Viability of Known Attacks Against Analogous Technologies
Perform Ambiguity Analysis
Ponder Design Implications
Generate Separate Architecture Diagram Documents
Unify Understanding
• Uncover Ambiguity
• Identify Downstream Difficulty (Sufficiency Analysis)
• Unravel Convolutions
• Uncover Poor Traceability
Perform Underlying Framework Weakness Analysis
Find & Analyze Flaws in
• COTS
• Frameworks
• Network Topology
• Platform
Identify Services Used By Application
Map Weaknesses to Assumptions Made by Application
Documents
Software Flaws
Architectural Risk Assessment Report

- Start by building a one-page overview of your system
- Then apply the three-step process we will describe more fully later
  - Attack resistance
  - Ambiguity analysis
  - Weakness analysis



Touchpoint: Penetration testing

- Penetration testing finds its place
  - Badnessometer (helpful for booting program)
  - Solutions more important than finding problems
- Focus on final software environment
  - Configuration
  - Context
- Clients no longer rely on pen tests exclusively



Touchpoint: Security testing

- Test security functionality
  - Cover non-functional requirements
  - Security software probing

- Risk-based testing
  - Use architectural risk analysis results to drive scenario-based testing
  - Concentrate on what “you can’t do”
  - Think like an attacker
  - Informed red teaming

- Training on security testing begins
- SQE offers public training courses
- Keynotes at major testing conferences on security

Touchpoint: Abuse cases

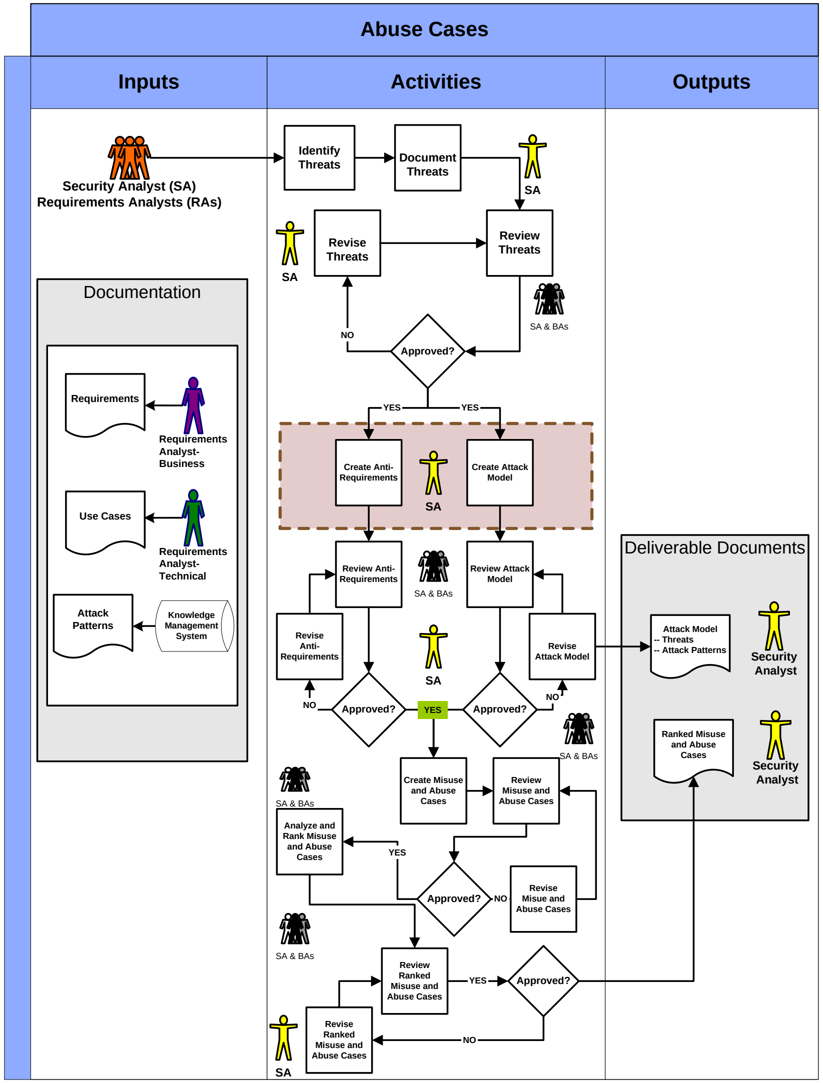
- Abuse cases used in DARPA work to drive requirements of advanced security system
- The problem of “implicit requirements” remains widespread

- Training: course development and delivery is nascent

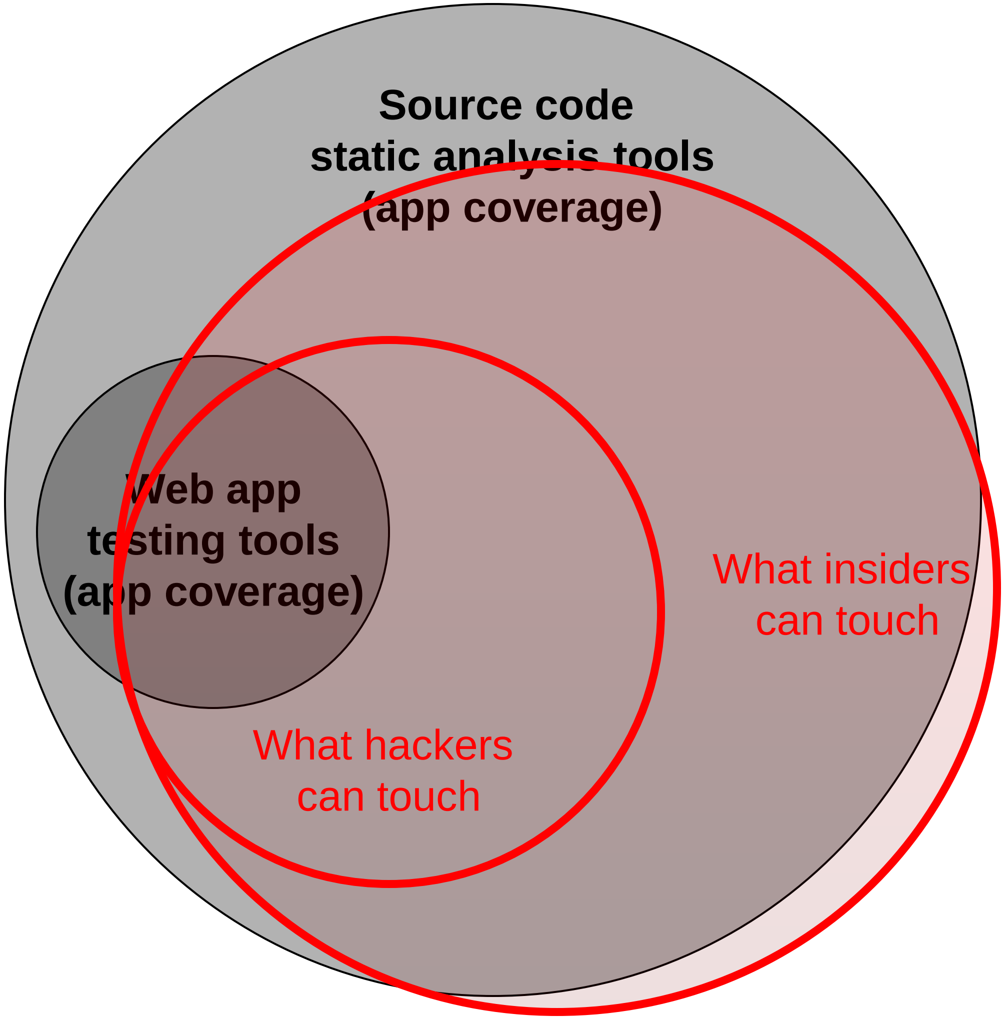
DARPA ITO
Information Technology Office.

Touchpoint: Abuse cases

- Starting with attack patterns, requirements and use cases
- Identify anti-requirements
- Build an attack model
- Determine misuse and abuse cases



Software security tools: app coverage



Source code static analysis tools (app coverage)

Web app testing tools (app coverage)

What insiders can touch

What hackers can touch

- Black box web testing tools only cover Web software
  - Useful for QA
- White box analysis tools cover a much larger set of software
  - Require clue about code

Knowledge

Knowledge catalogs

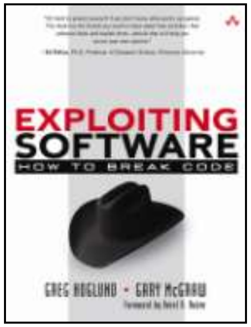
- Principles
- Guidelines
- Rules
- Attack patterns
- Vulnerabilities
- Historical Risks

Enterprise knowledge bases

- Corporate standards get smart
  - Written in code
  - Enforceable by tools
- Knowledge makes the round trip
  - What we see in scans
  - What goes into training
  - How we build code standards
  - What the tools enforce
- Fidelity identifies Common Vulnerability Patterns

Attack patterns

- Make the Client Invisible
- Target Programs That Write to Privileged OS Resources
- Use a User-Supplied Configuration File to Run Commands That Elevate Privilege
- Make Use of Configuration File Search Paths
- Direct Access to Executable Files
- Embedding Scripts within Scripts
- Leverage Executable Code in Nonexecutable Files
- Argument Injection
- Command Delimiters
- Multiple Parsers and Double Escapes
- User-Supplied Variable Passed to File System Calls
- Postfix NULL Terminator
- Postfix, Null Terminate, and Backslash
- Relative Path Traversal
- Client-Controlled Environment Variables
- User-Supplied Global Variables (DEBUG=1, PHP Globals, and So Forth)
- Session ID, Resource ID, and Blind Trust
- Analog In-Band Switching Signals (aka “Blue Boxing”)
- Attack Pattern Fragment: Manipulating Terminal Devices
- Simple Script Injection
- Embedding Script in Nonscript Elements
- XSS in HTTP Headers
- HTTP Query Strings
- User-Controlled Filename
- Passing Local Filenames to Functions That Expect a URL
- Meta-characters in E-mail Header
- File System Function Injection, Content Based
- Client-side Injection, Buffer Overflow
- Cause Web Server Misclassification
- Alternate Encoding the Leading Ghost Characters
- Using Slashes in Alternate Encoding
- Using Escaped Slashes in Alternate Encoding
- Unicode Encoding
- UTF-8 Encoding
- URL Encoding
- Alternative IP Addresses
- Slashes and URL Encoding Combined
- Web Logs
- Overflow Binary Resource File
- Overflow Variables and Tags
- Overflow Symbolic Links
- MIME Conversion
- HTTP Cookies
- Filter Failure through Buffer Overflow
- Buffer Overflow with Environment Variables
- Buffer Overflow in an API Call
- Buffer Overflow in Local Command-Line Utilities
- Parameter Expansion
- String Format Overflow in syslog()



EXPLOITING SOFTWARE
HOW TO BREAK CODE

GREG HOGLUND • GARY McGRAW

Enterprise Initiatives and the BSIMM

BSIMM

- Building Security In Maturity Model
- Real data from real initiatives
- 30 firms now in the study

The nine

EMC²
where information lives
Microsoft
Adobe
Google
DTCC
The Depository Trust & Clearing Corporation
WELLS FARGO
QUALCOMM

Two more unnamed financial services firms

BSIMM Europe (nine EU firms)

NOKIA
Connecting People

STANDARD LIFE®

THOMSON REUTERS

TELECOM
ITALIA

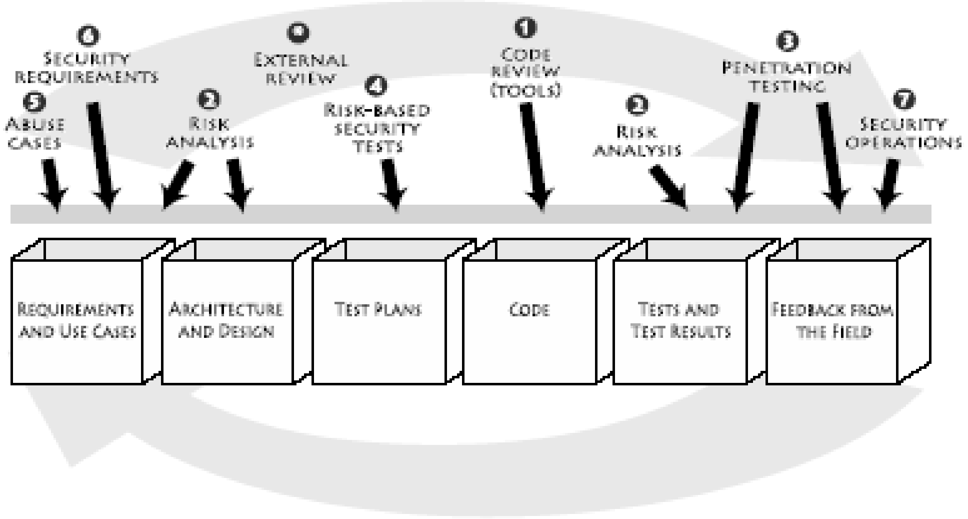
And four unnamed firms

Using BSIMM

- BSIMM released March 2009 under creative commons
  - http://bsi-mm.com (v1.5 includes Europe)
  - German and Italian translations are available
  - steal the data if you want
- BSIMM is a yardstick
  - Use it to see where you stand
  - Use it to figure out what your peers do
- BSIMM is growing
  - More BSIMM victims (30 and counting)
  - BSIMM Europe
  - BSIMM Begin
  - Statistics
  - Correlations

Touchpoints adoption

- Code review
  - Widespread
  - Customized tools
  - Training
- ARA
  - Components help
  - Apprenticeship
  - Training
- Pen testing
  - No longer solo
- Security testing
  - Training
- Abuse cases and security requirements
  - Training



Create an SSG

- Every BSIMM firm has a dedicated software security group
- BSIMM data show that SSG size should be 1.15% of the development group
- See informIT column “You Really Need a Software Security Group”

http://www.informit.com/articles/article.aspx?p=1434903

Where to Learn More

informIT & Justice League

- www.cigital.com/justiceleague
- In-depth thought leadership blog from the Cigital Principals
  - Scott Matsumoto
  - Gary McGraw
  - Sammy Migues
  - Craig Miller
  - John Steven

- www.informIT.com
- No-nonsense monthly security column by Gary McGraw

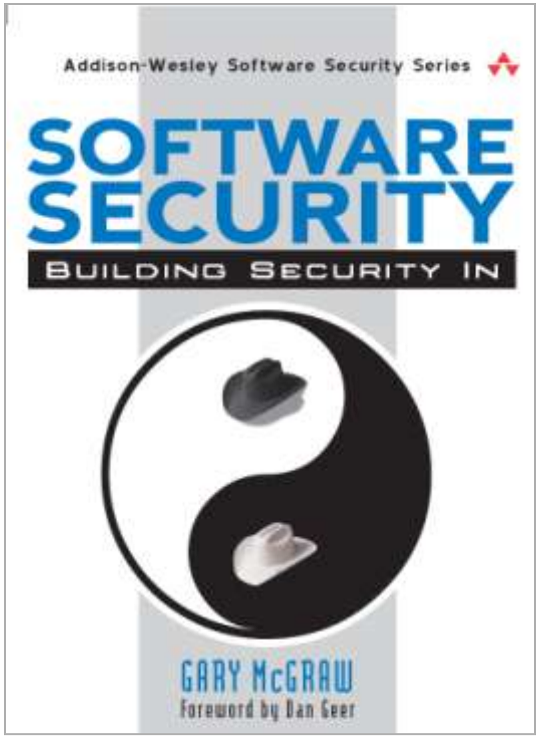
IEEE Security & Privacy Magazine + 2 Podcasts

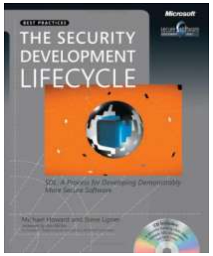
- Building Security In
- Software Security Best Practices column edited by John Steven
- www.computer.org/security/bsisub/

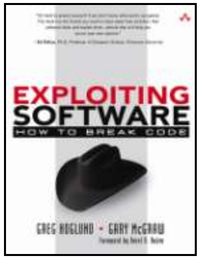
- www.cigital.com/silverbullet
- www.cigital.com/realitycheck

Software Security: the book

- How to DO software security
  - Best practices
  - Tools
  - Knowledge
- Cornerstone of the Addison-Wesley Software Security Series
- www.swsec.com







Addison
Wesley

For more

- Cigital's Software Security Group invents and delivers Software Quality Management
- See the Addison-Wesley Software Security series
- Send e-mail: gem@cigital.com

"*So now, when we face a choice between adding features and resolving security issues, we need to choose security.*"

-Bill Gates