DENIM GROUP

the leading secure software development firm

Basics of Application Security

Dan Cornell
CTO, Denim Group
@danielcornell

My Background

- Dan Cornell, founder and CTO of Denim Group
- Software developer by background (Java, .NET, etc)
- OWASP San Antonio, Global Membership Committee

Denim Group Background

- Secure software services and products company
  - *Builds secure software*
  - *Helps organizations assess and mitigate risk of in-house developed and third party software*
  - *Provides classroom training and e-Learning so clients can build software securely*
- Software-centric view of application security
  - *Application security experts are practicing developers*
  - *Development pedigree translates to rapport with development managers*
  - ***Business impact: shorter time-to-fix application vulnerabilities***
- Culture of application security innovation and contribution
  - *Develops open source tools to help clients mature their software security programs*
    - *Remediation Resource Center, ThreadFix*
  - *OWASP national leaders & regular speakers at RSA, SANS, OWASP, ISSA, CSI*
  - *World class alliance partners accelerate innovation to solve client problems*

Application Security in the News

- Heartland Payment Systems – Financial Data Compromise
  - http://voices.washingtonpost.com/securityfix/2009/01/payment_processor_breach_may_b.html
- PayPal – Cross Site Scripting
  - http://news.netcraft.com/archives/2006/07/20/paypal_xss_exploit_available_for_two_years.html
- T-Mobile – SQL Injection
  - http://www.pcworld.com/article/119851/paris_hilton_victim_of_tmobiles_web_flaws.html
- IKEA – Database Downloaded
  - http://news.cnet.com/2100-1017-245372.html

Demonstration

- How do attackers view your web applications?
- RiskE Utility site

DENIM GROUP

build | integrate | secure

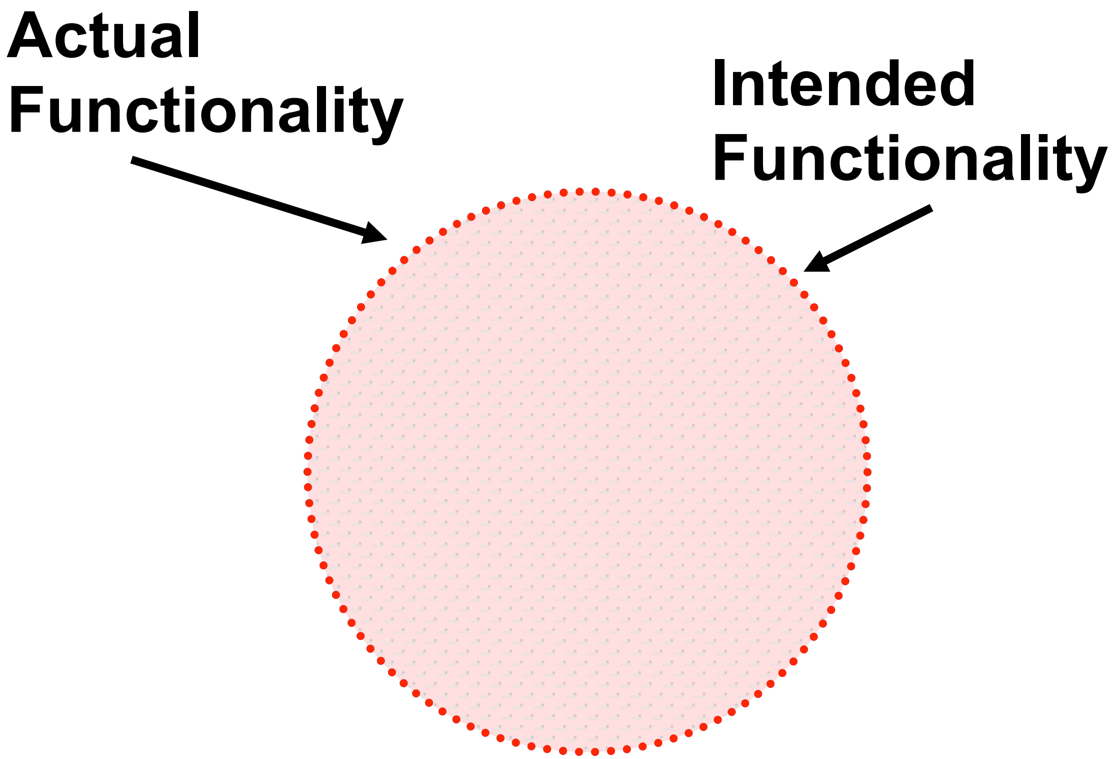
Application Security Defined

Application Security Defined

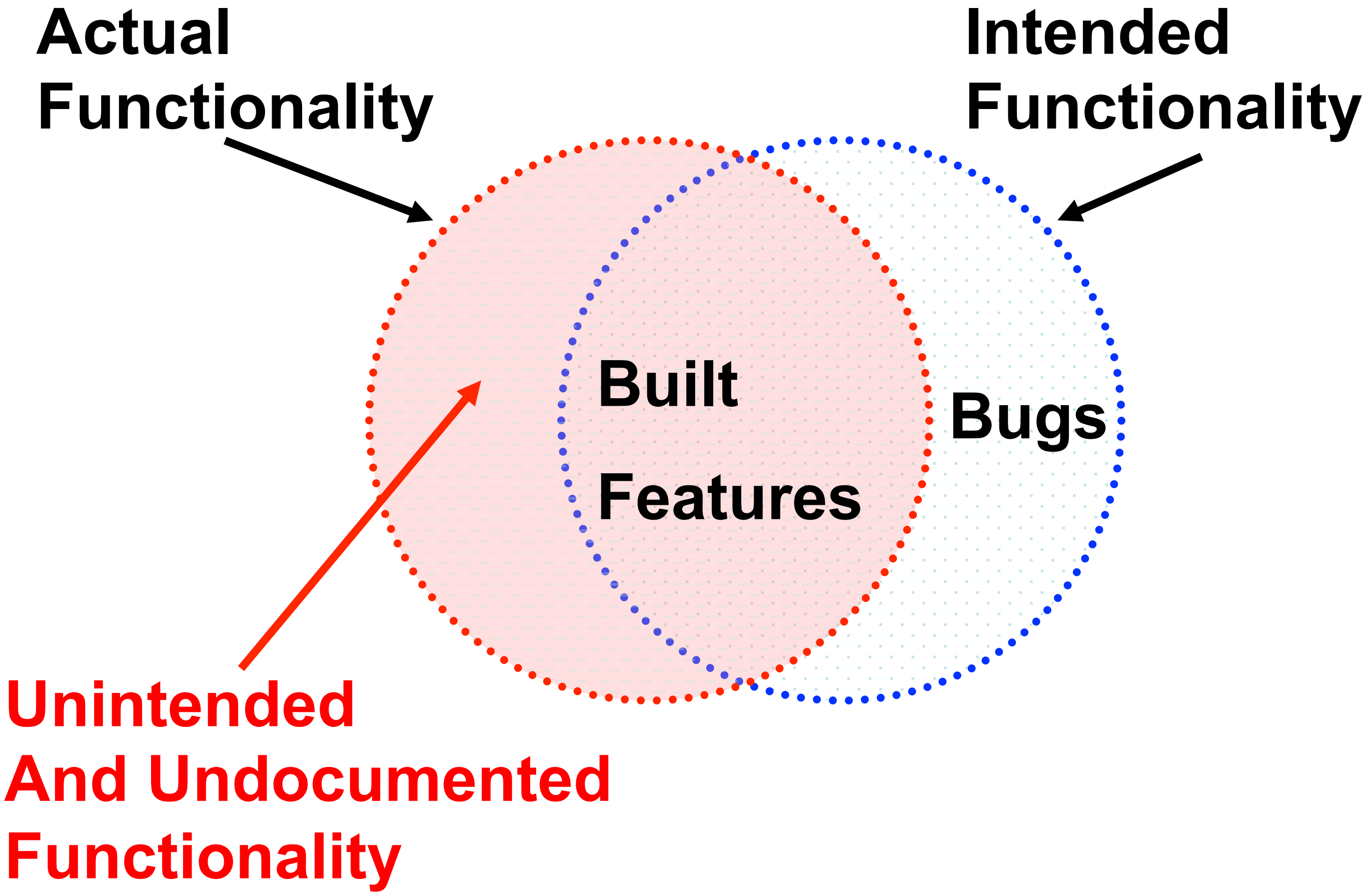
- A common definition – "Ensuring that an attacker cannot compromise an application's resources or data".
  - *Too narrow*
  - *Not very actionable*

- A better definition – "Ensuring that custom application code performs as expected under the entire range of possible inputs"

Software Implementation – Perfect World



Software Implementation – Real World



Application Security is Different

- Quality Assurance?
  - *The processes are similar*
  - *The goals are not*
- Traditional Information Security? Network Security?
  - *The goals are similar*
  - *The processes are not*

Quality Assurance vs. Security Assurance

- Both are evolving practices
  - *Tools and techniques are continually improving*
- Quality and Security Assurance both require continual effort
  - *You cannot declare software 100% bug-free*
  - *You similarly cannot declare an application 100% secure*
- Both are often managed by dedicated teams in addition to the development team

Quality Assurance vs. Security Assurance

- QA, even excellent QA, does not account for security
- QA essentially compares an application to its "intent", its requirements
  - *Is the functionality there?*
  - *Is it reliable in corner cases?*
  - *Is the performance acceptable?*
- Attackers are interested in what the application DOES that it is NOT SUPPOSED to do
  - *I can access my transaction data. Can I access someone else's as well?*
  - *I can enter a data query string. Can I twist it into a data tampering command?*
  - *I can upload documents. Can I also upload server pages? Overwrite their content?*

Traditional Security vs. Application Security

- Traditional Information Security shares the same goals
  - *Confidentiality*
  - *Integrity*
  - *Availability*
- Network and application security experts must continually keep up with the latest threats

Traditional Security vs. Application Security

- Traditional Information Security has a "measure and maintain" culture
  - *Track servers, workstations, devices*
  - *Manage advisories, patches, configurations*
  - *Monitor the systems in operation*
- Application development has a "build" culture
  - *Create something that did not exist before*
  - *Get it working on time and within budget*
- Application threats are as unique as the applications themselves

Why Does Application Security Matter?

- Critical Systems are Internet-facing
- Most applications have serious design or coding flaws
- Laws and Regulations

Critical Systems are Internet Facing

- More and more business have moved to online commerce
  - *Hard goods, soft goods*
  - *Flight check-in*
  - *Personals*
  - *Pizza Delivery*
- This has tremendous advantages
  - *Cost of doing business goes down*
  - *Market barriers are lower*

Critical Systems are Internet Facing

- What are the drawbacks?
  - *Systems no longer have an "air" gap, personal interaction*
  - *Physical security and personal scrutiny matter less*
- Imagine an ATM machine in the desert…

Most applications have serious flaws

- 70%+ according to studies performed by @Stake and Foundstone
- Too many development teams treat application security as a "check box"
  - *"This site is certified secure" labels on web pages*
- Too few development teams regard security as fundamental as design or QA

Laws and Regulations

- New laws and regulations govern how data is stored and made available
  - *HIPAA*
  - *Sarbanes Oxley*
  - *California SB-1386*
  - *PCI*
- Failing to comply can have legal repercussions and damage trust from partners

Application Security Goals

- Confidentiality
- Integrity
- Availability

- A flaw can be considered a security vulnerability when one of the goals is compromised

Confidentiality

- Ensuring that information is accessible only to those authorized to have access

- Compromises
  - *Spoofing Identity*
  - *Direct Object Reference*
  - *Forced Browsing*
  - *Database compromise*
  - *Packet Sniffing*

- This is not limited to information the application directly manages
  - *What about phishing?*
  - *An attacker can use an application to manipulate users*

Integrity

- Information should only be modified by those users authorized to modify it

- Compromises
  - *Injection*
  - *Direct Object Reference*
  - *Malicious File Execution*
  - *Cross Site Request Forgery*

- There is a lot of crossover with Confidentiality, but many threats to Integrity are unique

Availability

- The system is online and responding to user requests for valid users at all times it is supposed to

- Compromises
  - *Malicious File Execution*
  - *Buffer Overflow*
  - *Lockout Exploits*

- Threats are not limited to "bringing down" an application server
  - *What about forcing an exception?*
  - *What about saturating sockets between web and application servers?*

What Goes Wrong?

- Failure in Design
- Failure in Implementation

Causes of Application Security Vulnerabilities

- Failure in Design
  - *Poor decisions about trust*
  - *Unspoken assumptions*
  - *Not accounting for failure*

- Failure in Implementation
  - *Insecure coding techniques*
  - *Insecure configuration*
  - *Poor deployment practices*

Types of Vulnerabilities

- Logical Vulnerabilities
  - *Surface due to insecure program logic*
  - *Typically due to poor decisions about trust*
  - *Most "scanner" tools are powerless to find logical vulnerabilities*
  - *Remediation: architecture and design changes*

- Technical Vulnerabilities
  - *Surface due to insecure programming techniques*
  - *Typically due to poor input handling and input validation*
  - *Most "scanner" tools primarily find technical vulnerabilities*
  - *Remediation: coding changes*

Common Application Vulnerabilities

- Logical
  - *Poor Authentication*
  - *Direct Object References*
  - *Unchecked Input*
- Technical
  - *Cross Site Scripting*
  - *Injection Flaws*
  - *Insecure Communications*
- Logical or Technical
  - *Information Leakage*
  - *Poor Cryptographic Storage*
  - *Poor Configuration Management*

Software Assurance Maturity Model (OpenSAMM)

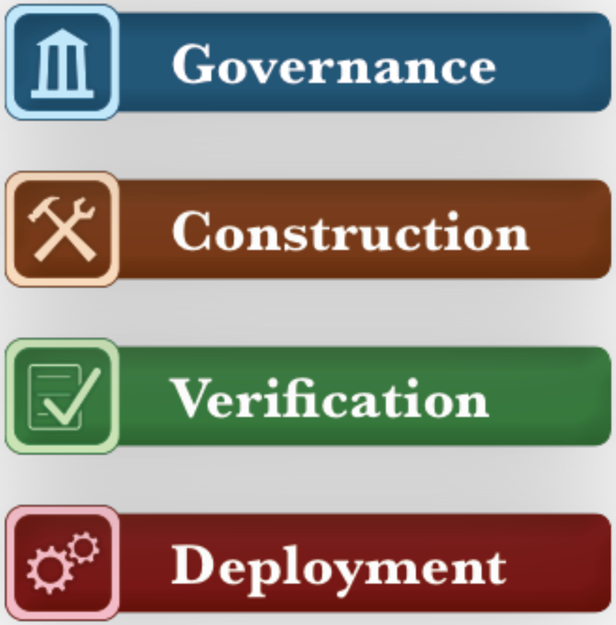
- Open framework to help organizations formulate and implement a strategy for software security that is tailored to the specific risks racing the organization

- Useful for:
  - *Evaluating an organization's existing software security practices*
  - *Building a balanced software security program in well-defined iterations*
  - *Demonstrating concrete improvements to a security assurance program*
  - *Defining and measuring security-related activities within an organization*

- Main website:
  - http://www.opensamm.org/

SAMM Business Functions

- Start with the core activities tied to any organization performing software development
- Named generically, but should resonate with any developer or manager



Governance

Construction

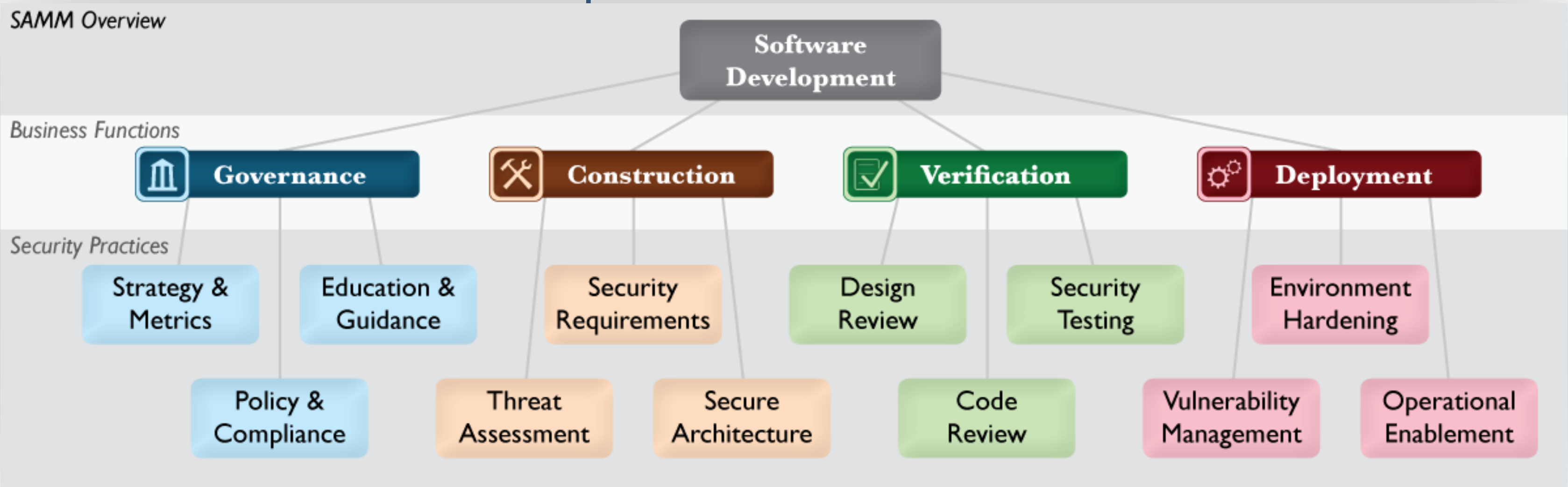
Verification

Deployment

[This slide content © Pravir Chandra]

SAMM Security Practices

- From each of the Business Functions, 3 Security Practices are defined
- The Security Practices cover all areas relevant to software security assurance
- Each one is a 'silo' for improvement



[This slide content © Pravir Chandra]

Conclusions / Questions

Dan Cornell
dan@denimgroup.com
Twitter: @danielcornell

www.denimgroup.com
www.denimgroup.com/threadfix
code.google.com/p/threadfix
(210) 572-4400