



**Black Box versus White Box:
Different App Testing Strategies**
John B. Dickson, CISSP

- Learning objectives for today's session
 - *Understand what a black box and white box assessment is and how they differ*
 - *Identify tools that support black and white box testing*
 - *Understand testing coverage and limitation of automated black and white box tools*

• Denim Group Background

- *Professional services firm that builds & secures enterprise applications*
- *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 identity management enablement
- *Competencies in the following areas:*
 - PCI pre-assessment readiness
 - Secure agile development
 - Threat modeling

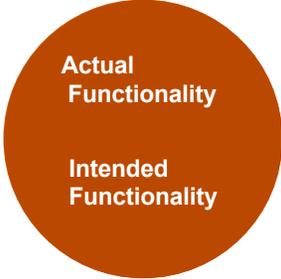
• Personal Background

- *15-year information security consultant background*
- *Principal at Denim Group*
- *Ex-Air Force security analyst at AFCERT*
- *Trident Data Systems, KPMG, SecureLogix, and Denim Group information security consultant*
- *Works with CISOs to help them develop and deploy more secure systems and applications*
- *CISSP since 1998*

- **Key Challenges**

- *Why is it that serious web application vulnerabilities still exist in organizations what 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 when they have little control over development efforts?*
- *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?*

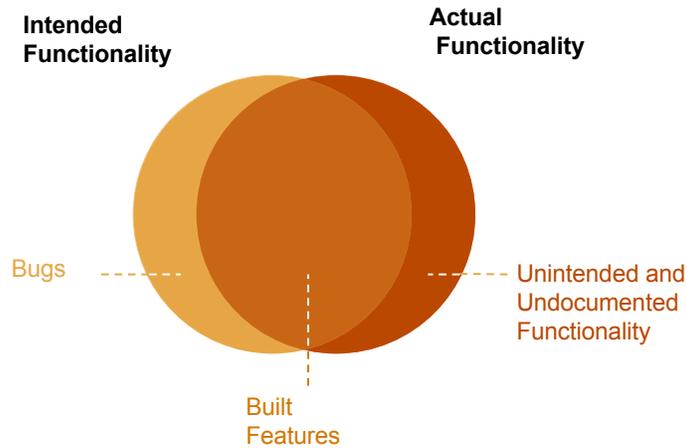
Software Implementation – Perfect World



**Actual
Functionality**

**Intended
Functionality**

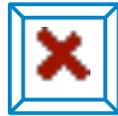
Software Implementation – Real World



- Nature of Application Security Problem

- *Most security professionals do not have a development background*
- *Security managers do not control application development*
- *Security requirements rarely are central to development priorities*
- *Attackers are focusing more on web applications as network perimeters are more secure*
- *Fielded applications developed over the years are largely insecure*
- *Who gets fired first when penetration occurs via web application?*

1998 Network Security Question?



Firewall?

2008 Application Security Question?

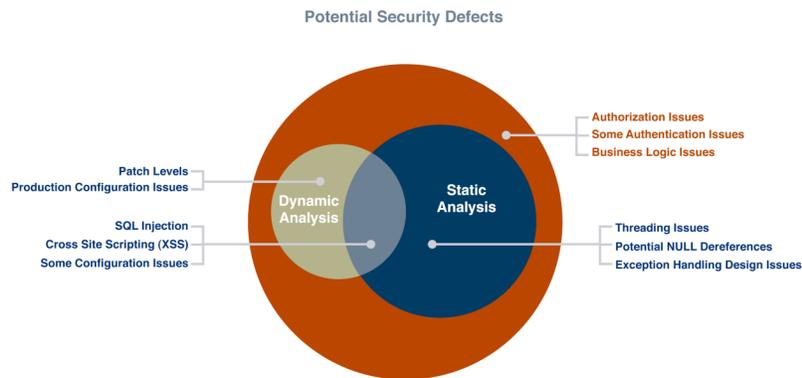


Automated Application
Scanner?

- Application Penetration test
 - Controlled test from the outside simulating a sophisticated attacker with limited information
 - Goal: exploit a vulnerability to gain system level access or obtain sensitive data
 - Somewhat a “capture the flag” exercise to prove a point – can potentially show you one route to gain access, not all possible approaches
 - Typically conducted to validate previous assessments or to prove a theory
- *Focus of the presentation will be assessments, and not penetration tests*

- Types of Application Vulnerabilities
 - *Technical*
 - Implementation flaws introduced at the keyboard
 - Straightforward to identify and mitigate
 - Most analogous to TCP vulnerabilities
 - Scanners best suited to identify technical flaws
 - *Logical*
 - Architectural or design flaws typically introduced before coding
 - Much harder to identify – potentially painful to mitigate
 - Fix might include an architectural re-write
 - Scanners deeply limited in ID'ing logical flaws

Dynamic, Static and Manual Testing



- **Black Box Assessments**
 - Automated application security testing that view the security state of an application from the outside looking in
 - *Mirrors the perspective of an outside attacker*
 - Infers that certain vulnerabilities exist by sending inputs to an application and analyzing outputs
 - Does not involve review of application source code

- Pro's for black box assessment approach
 - *Well understood by security professionals*
 - Network vulnerability analogy
 - *Measures security state of environment in which application resides*
 - *Can quantify security risks of third-party components or other resources outside the application*

- Con's for black box assessment approach
 - *Results tell you what vulnerabilities exist, not how or why they exist*
 - *Can only test the attack surface they identify*
 - May be additional endpoints with vulnerabilities
 - *Provides less input for remediation*

- White Box Assessments

- *Involve reviewing application source code to determine the difference between what security was designed in the system and what was built*
- *Typically complemented with an architectural design review to ID non-code problems*

- Pro's for white box assessment approach

- *Identifies exactly where vulnerabilities exist and why/how they occurred*
- *Tells you definitively whether code design is implemented in source code*
- *Easier to begin remediation because the exact location of the vulnerabilities has been identified*

- **Con's for white box assessment approach**
 - *Potentially can generate a large number of false positives (“noise”) if source code analyzer is not tuned well*
 - *Provides less feedback on environmental components that affect the security of an application*
 - *Likely the sole domain of developers – security staff are less trained to interpret results*
 - *Sometimes hard to identify context*

- **Black box automated assessment tools**
 - *HP (SPI Dynamics) WebInspect & DevInspect*
 - *IBM Rational (Watchfire) AppScan*
 - *Cenzic Hailstorm*
 - *NT Objectives NTO Spider*
 - *Acunetix Web Vulnerability Scanner*

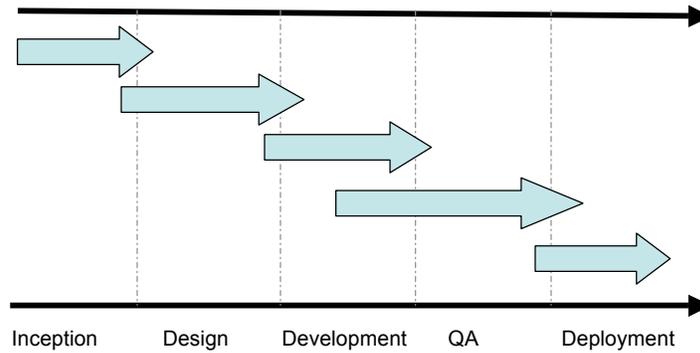
- White box assessment tools

- *Major product vendors:*
 - Fortify Source Code Analyzer
 - Ounce Labs
 - Coverity Prevent SQS
- *Attributes*
 - Licenses are often priced by LOC
 - Most web languages, some legacy languages

- Limitations of Automated Tools

- *Only find Technical flaws in applications*
 - What about Logical flaws?
- *Can require sophisticated users to drive them correctly*
- *Can provide a false sense of security*

Potential security points in SDLC



- OWASP Top 10 Critical Web Application Security Vulnerabilities

- *Cross Site Scripting (XSS)*
- *Injection Flaws*
- *Malicious File Execution*
- *Insecure Direct Object Reference*
- *Cross Site Request Forgery*
- *Information Leakage and Improper Error Handling*
- *Broken Authentication and Session Management*
- *Insecure Cryptographic Storage*
- *Insecure Communications*
- *Failure to Restrict URL Access*

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

Contact Information

John B. Dickson, CISSP
Principal
Denim Group, Ltd.
John.Dickson@denimgroup.com
(210) 572-4400