

Web Application Security

A Holistic Approach to Application Security

ISACA Denver Chapter: March 31, 2011



Agenda

- ▶ Introductions
- ▶ Vulnerability trends
- ▶ Secure SDLC
- ▶ Web application firewalls
- ▶ Questions

Introductions

- ▶ Tushar Padhiar
Senior Manager
Ernst & Young

- ▶ Raghav Dube
Manager
Ernst & Young



DATA BREACH TRENDS

Current threats

- ▶ Threat agents
 - ▶ External
 - ▶ Internal
 - ▶ Trusted 3rd Parties
 - ▶ State Sponsored
- ▶ Threat vectors
 - ▶ Infrastructure
 - ▶ Applications
 - ▶ People

Verizon's 2009 Data Breach Report

- ▶ 90 breaches with 285,000,000 compromised records
 - ▶ 74% of data breaches result from external sources (+1%)
 - ▶ 64% of data breaches result from hacking (+4%)
 - ▶ 69% of breaches were not discovered by the victim (-6%)
 - ▶ 83% of attacks are not highly difficult (No change)
 - ▶ 87% were considered avoidable through reasonable controls (No change)
 - ▶ 93% of the breaches would have still occurred if systems had been fully patched (+15)

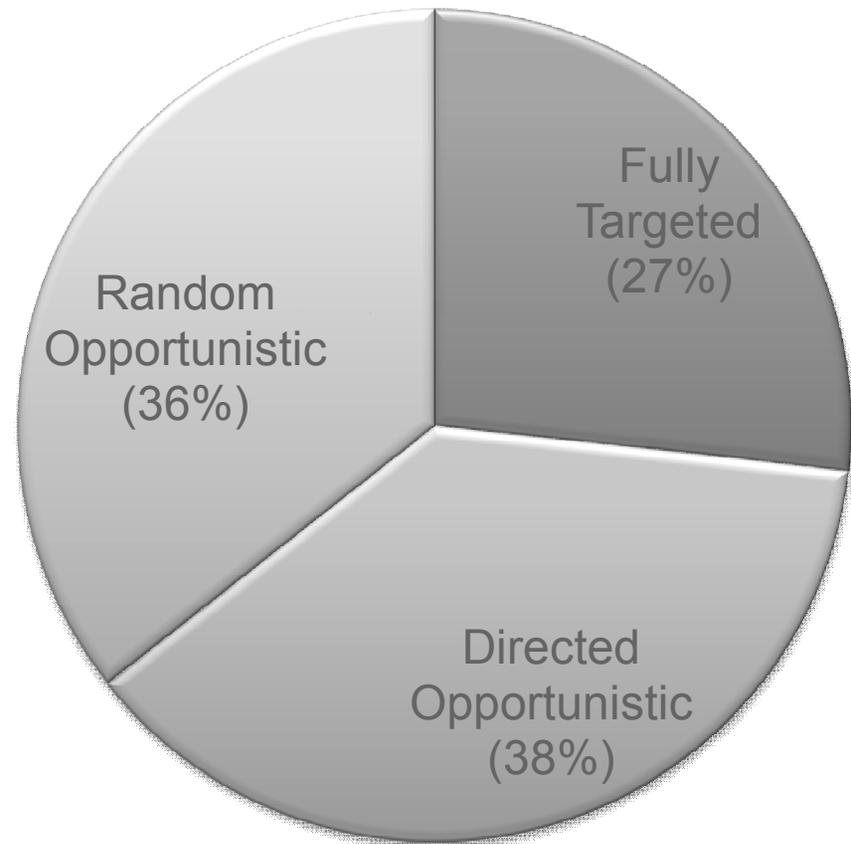
Verizon's 2010 Data Breach Report

- ▶ 141 breaches with 143,000,000 compromised records
 - ▶ 70% of data breaches result from external sources (-9%)
 - ▶ 40% of data breaches result from hacking (-24%)
 - ▶ 61% of breaches were not discovered by the victim (-8%)
 - ▶ 85% of attacks are not highly difficult (+2%)
 - ▶ 96% were considered avoidable through reasonable controls (+9%)
 - ▶ 100% of the breaches would have still occurred if systems had been fully patched (+8%)

Verizon's 2010 Data Breach Report

Where should mitigation efforts be focused?

- ▶ Ensure essential controls are met
- ▶ Find, track, and assess data
- ▶ Collect and monitor event logs
- ▶ Audit user accounts and credentials
- ▶ Test and review web applications



Targeted vs Opportunistic

EY ASC engagement data collection

- ▶ Over the last three years, we captured data from 551 tests with 4,200 individual findings
- ▶ ASC identified an average of 68.5 instances of issues across all tests
- ▶ More than 37,755 instances of findings
- ▶ More than 15,156 instances (40%) of high-risk findings
- ▶ 88% of our tests have at least one high-risk finding
- ▶ 58% of all high-risk issues require a low level of effort to exploit
- ▶ 54% of all identified issues require only a low level of effort to remediate

Application vulnerability metrics

- ▶ EY ASC engagement metrics
 - ▶ 93% of our application tests have at least one high-risk finding
 - ▶ 70% of the high-risk issues require a low level of effort to exploit
 - ▶ 46% of high-risk issues require only a low level of effort to remediate
 - ▶ 34% of all high-risk issues identified during application testing could be prevented by properly validating user input

- ▶ General trends
 - ▶ Cross site scripting and SQL injection most common
 - ▶ Business logic flaws

- ▶ Root causes
 - ▶ Limited understanding of input validation concepts
 - ▶ Security through obscurity
 - ▶ Complex development process with minimal focus on security
 - ▶ Lack of secure development training



SECURE SDLC

Application Security Goals

- ▶ Measure and Reduce **Risk**
 - ▶ Revenue
 - ▶ Brand
- ▶ Regulatory Compliance
 - ▶ **Remember:** Be secure and it will be easy to gain compliance. This formula does not work the other way around
- ▶ Do so **cheaply** and effectively

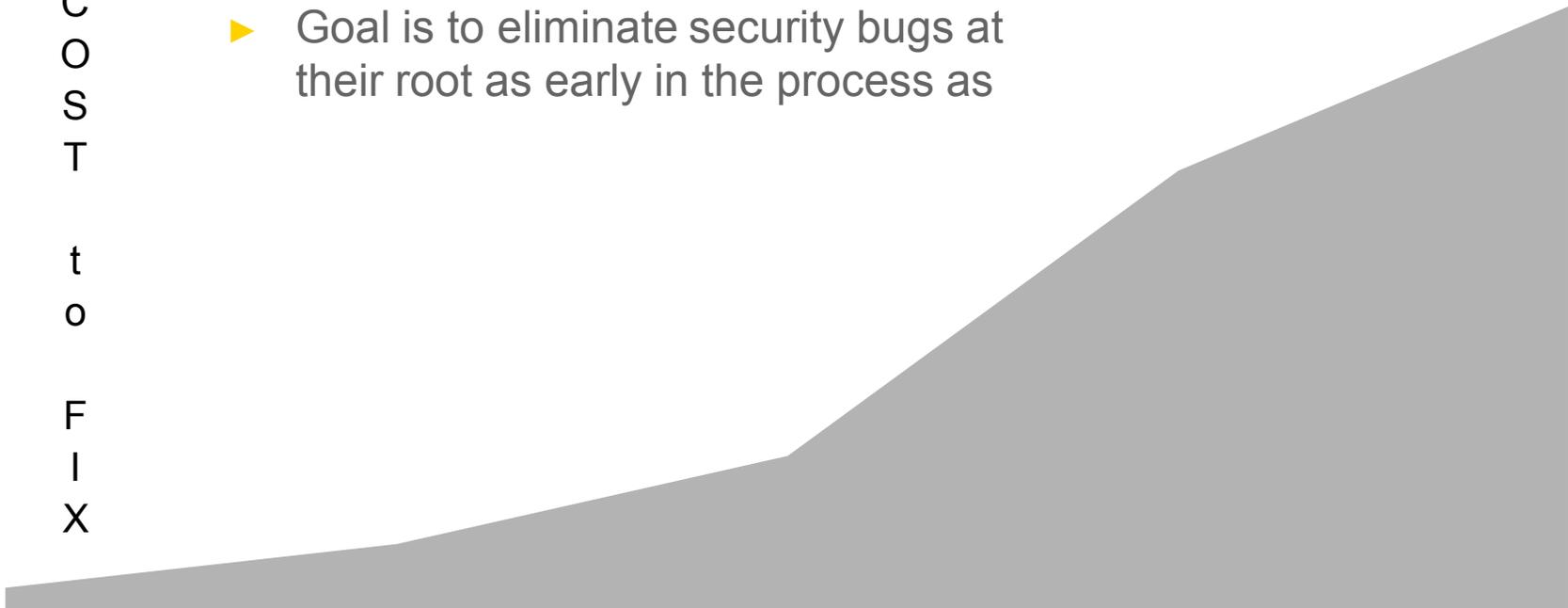
Security SDLC: “Push Left”

- ▶ “Push Left” to achieve highest ROI
- ▶ Goal is to eliminate security bugs at their root as early in the process as

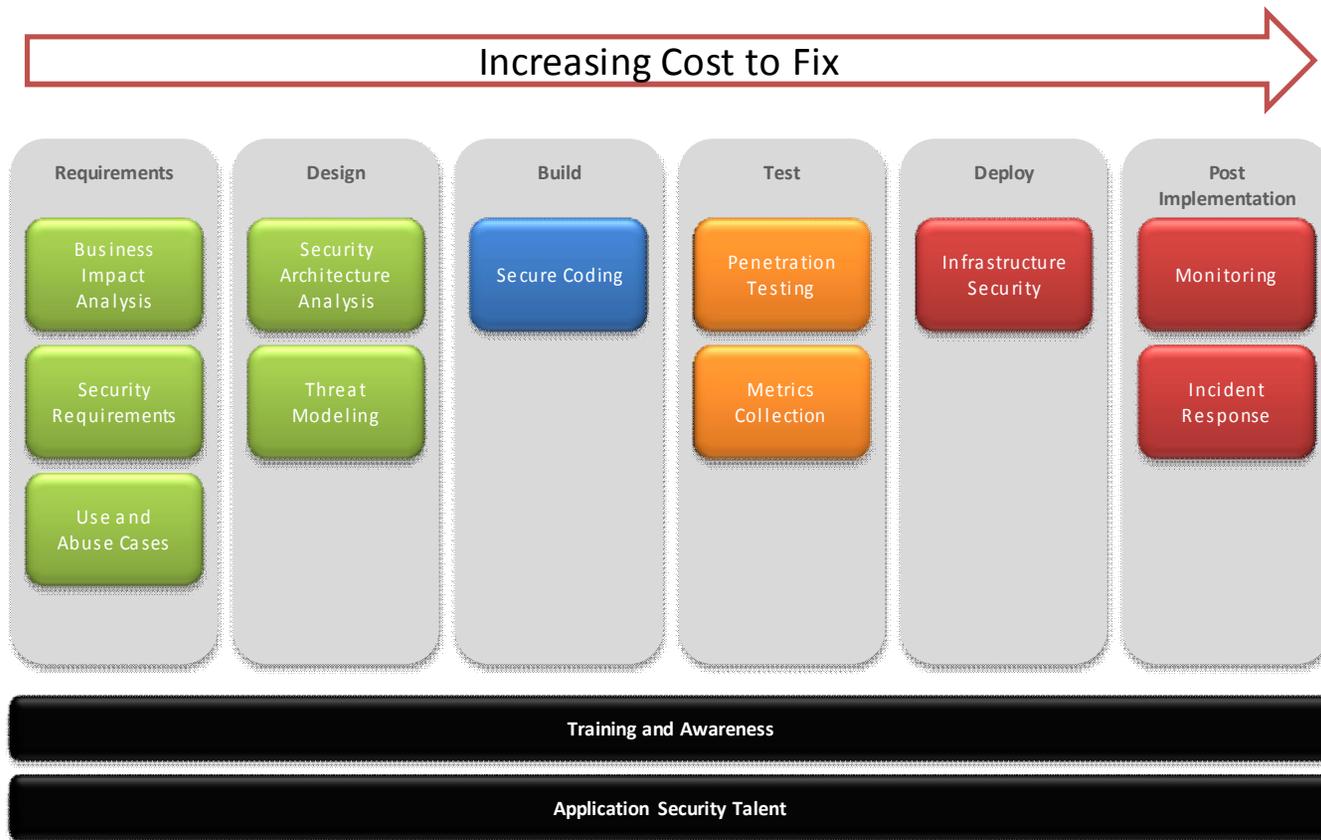
C
O
S
T

t
o

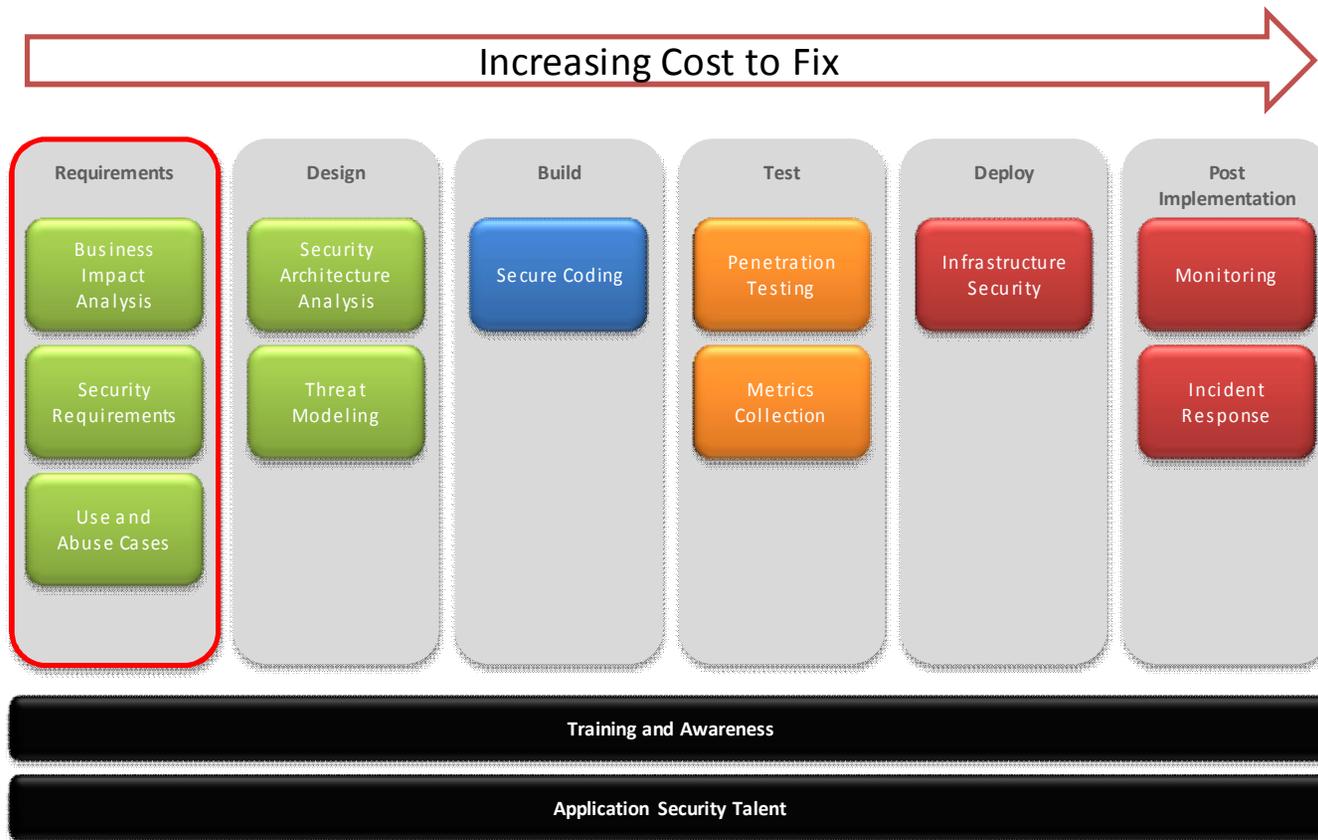
F
I
X



Security SDLC



Requirements Phase



Business Impact Analysis

- ▶ Interview owner of Technology to construct application inventory
- ▶ Calculate Business Impact, regulatory requirements to formulate a Risk based effort. Example:
 - ▶ High: Entire SDLC
 - ▶ Medium: A La Carte (Example: Threat Modeling & Penetration Testing)
 - ▶ Low: Automated

Business Impact Analysis Continued

<i>Question</i>	<i>Options</i>	<i>Answer</i>	<i>Scoring Guidance</i>	<i>Score</i>
Confidentiality				
Q1. What is the effect on reputation if confidentiality of data is compromised?	High Medium Low		High = 4 Medium = 2 Low = 1	
Q2. What is the internal cost to investigate and repair the problem?	> \$1M \$100K - \$1M <\$100K No Impact		> \$1M = 4 \$100K - \$1M = 2 <\$100K = 1 No Impact = 0	
Q3. What is the impact if security controls does not meet legal requirements?	Criminal Prosecution Government Fine Client Lawsuit No Impact		Criminal Prosecution = 4 Government Fine = 2 Client Lawsuit = 1 No Impact = 0	

Security Requirement

- ▶ Define application security requirements:
 - ▶ Security control requirements
 - ▶ Secure coding requirements and best practices (i.e., PCI)

- ▶ Goal of requirement
 - ▶ Communicate to business
 - ▶ SLAs with vendors

- ▶ Typical categories
 - ▶ Authentication
 - ▶ Access control
 - ▶ Session management
 - ▶ Input validation
 - ▶ Data protection
 - ▶ Output handling
 - ▶ Error handling & logging
 - ▶ Encryption
 - ▶ Communication security
 - ▶ HTTP security

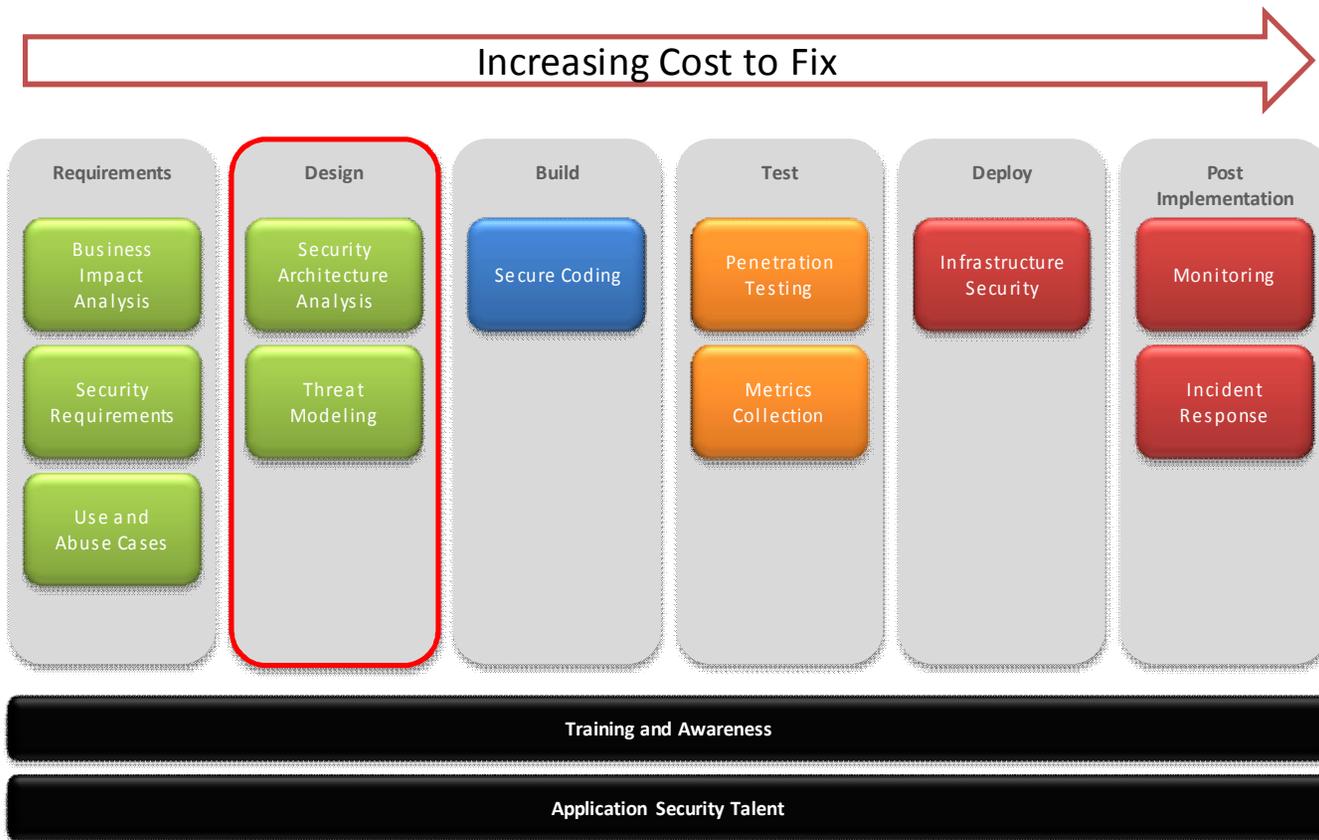
Security Requirement Continued

ID	Requirement	PCI	BIA Score	Technology Layer
	Authentication			
AUTH-1	The application enforce that all pages and resources require authentication except those specifically intended to be public.		H,M,L	Application
AUTH-2	The application should be partitioned to separate user functionality/interfaces from administrator functionality/interfaces		H,M	Application
AUTH-3	Account passwords must be encrypted		H,M,L	Application
AUTH-4	Verify that re-authentication is required before any application-specific sensitive operations are permitted		H,M	Application
AUTH-5	User authentication events should be logged		H,M,L	Application, Infrastructure
AUTH-6	Account passwords must meet applicable Sony password complexity requirements		H,M	Application,Infrastructure, OS, Network
AUTH-7	After a maximum number of authentication attempts is exceeded, the account should be locked for a period of time long enough to deter brute force attacks		H,M	Application,Infrastructure, OS, Network
AUTH-8	Communication during authentication must be encrypted		H,M	Application,Infrastructure, OS, Network
AUTH-9	In response to failed login attempts, the application should use generic error messages that do not indicate if the failure was due to incorrect username or incorrect password		H,M	Application
AUTH-10	All authentication credentials for accessing services external to the application are encrypted and stored in a protected location (not in source code).		H,M,L	Application,Infrastructure, OS, Network

Use & Abuse Cases

ABUSE CASES	USE CASES			ACCOUNT MANAGEMENT			ACCOUNT MANAGEMENT					
	Access Management	Anonymous user logs-in to the application	Anonymous user creates account	User logs out of the application	User view login information	User edit login information	User view profile	User edits profile	User views credit card information	User edits credit card information	User views PayPal information	User edits PayPal information
Access Management												
1 Malicious entity guesses credentials	x											
2 Malicious user able to prevent legitimate users from logging in	x											
3 Malicious user able to capture legitimate users' credentials	x											
4 Malicious entity creates too many accounts		x										
5 Malicious user creates account with same user-id		x										
6 Malicious user hijacks an improperly terminated session			x									
Account Management												
7 Malicious user views legitimate user's profile					x		x					
8 User views another user's profile					x		x					
9 Malicious user edits legitimate user's profile						x		x				
10 User edits another user's profile						x		x				
11 Malicious user inputs illegal characters						x		x	x		x	
12 Anonymous user able to view legitimate user's billing information									x		x	
13 Malicious user able to view another user's billing information									x		x	
14 Anonymous user able to edit legitimate user's billing information										x		x
15 Malicious user able to edit another user's billing information										x		x
16 Anonymous user able to view legitimate user's address book												

Design Phase



Security Architecture Analysis

- ▶ Provide a high-level view of the security architecture of the application and infrastructure under consideration
- ▶ Evaluate security capabilities within all tiers of an architecture against established security requirements
 - ▶ End Client Tier
 - ▶ Presentation Layer
 - ▶ Business Logic Layer
 - ▶ Integration Tier
 - ▶ Data tier
- ▶ Categorize security requirements into layers
 - ▶ Application – Application code and corresponding libraries
 - ▶ Infrastructure – Support applications and services necessary for application to function
 - ▶ Operating System – Platform on which a given application resides
 - ▶ Network – Responsible for communication taking place within the application

Security Architecture Analysis Continued

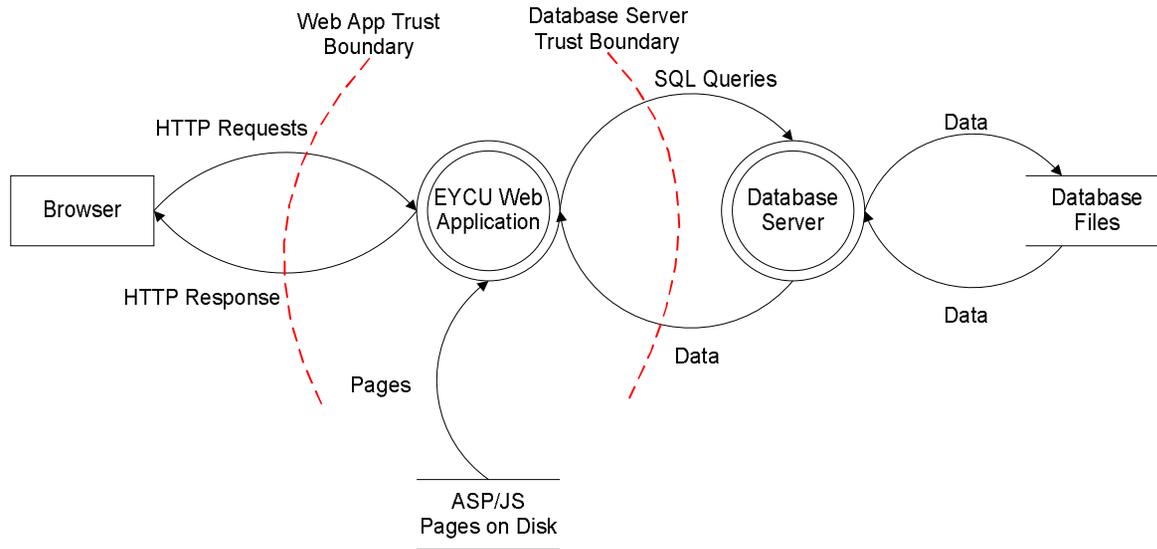
► Presentation Layer

Technical Security Category	Detail	Gaps Present / Action
Authentication	<p>Application: The user provides a username and password to the Ruby on Rails Forms authentication interface. The initial authentication is passed to the Business Logic tier to authenticate the username and password against the User store within WebSphere Commerce Server. After the initial authentication, a user's subsequent requests are authenticated by secure cookies.</p> <p>Infrastructure: Authentication to infrastructure components is handled via local user accounts on the Operating System</p> <p>Operating System: The underlying OS is Solaris Password expiration and strength rules are enforced.</p> <p>Network: The machine authenticates to the network via IP address</p>	No gaps noted
Access Control	<p>Application: Users are only presented interfaces to authorized functionality. Additionally, there is only 1 role assigned to authenticated users (Registered users)</p> <p>Infrastructure: A non administrator-level account is used for access to the user store and for access to backed SAP systems, the Apache server runs as a user with low privileges</p> <p>Operating System: The underlying Linux OS was hardened and access control lists (ACLs) provide a restricted environment for infrastructure components</p>	No gaps noted
Session Management	<p>Application: User's sessions are tied to an authenticated session cookie; the activity code within the cookie changes on each use. Upon logout the session cookie is destroyed. Users are presented with a logout option on each page.</p>	Minor gap noted; inactive HTTP sessions are not destroyed.

Threat Modeling

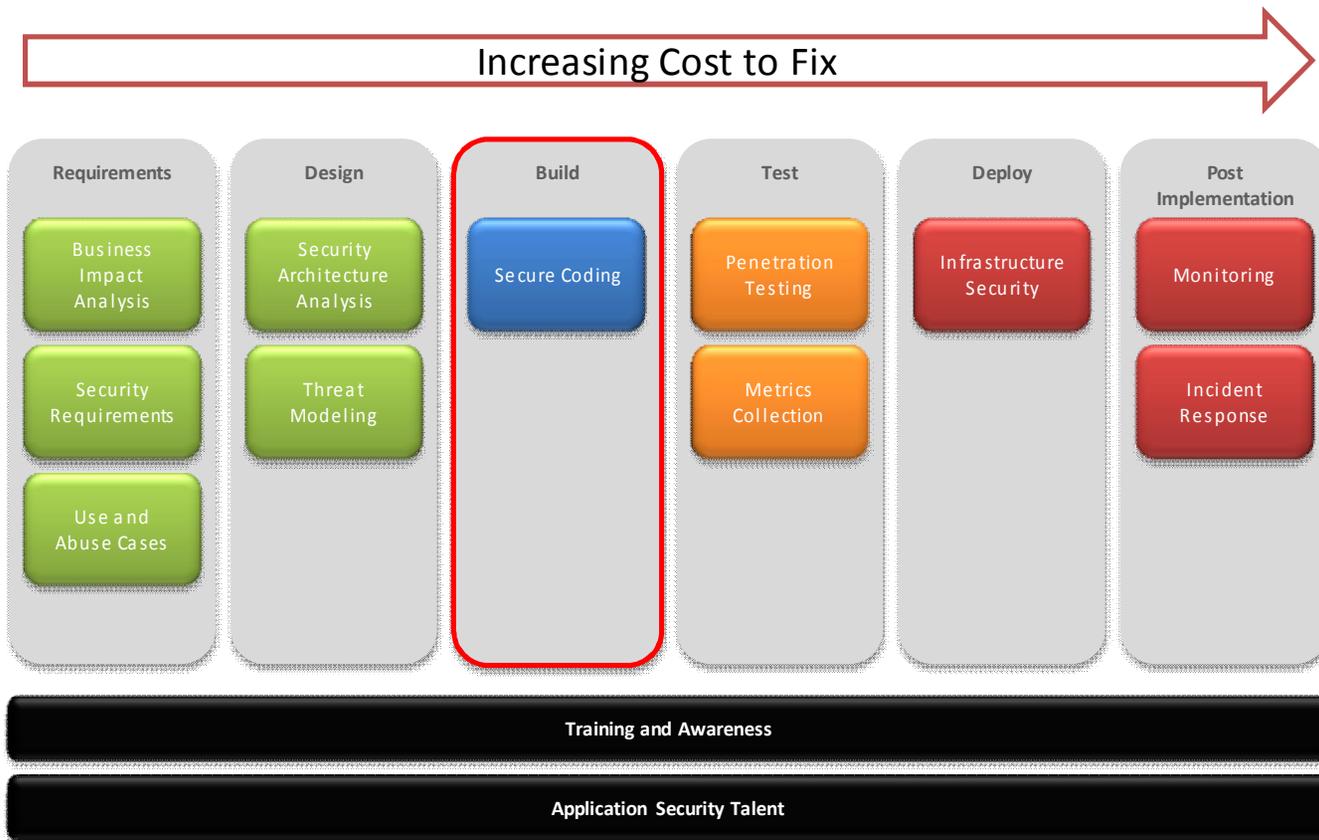
- ▶ Enumeration of Hackers' goals even if they are known to be mitigated
- ▶ Forces developers and business to look at design from a Hacker mindset
- ▶ Tied to real business risks
- ▶ Living document: Security, Technology, and Business BUs
- ▶ Measures good efforts in addition to improvement areas
- ▶ Facilitates the SDLC

Threat Modeling Continued



#1 Disclosure of Credentials	
Description	Attacker acquires the credentials of another user
STRIDE	Information Disclosure, Escalation of Privilege
Mitigated	No
Known Mitigation	None
Threat Tree	<ol style="list-style-type: none"> 1. Threat: Attacker acquires another user's credentials <ol style="list-style-type: none"> 1.1. Attacker gets valid username <ol style="list-style-type: none"> 1.1.1. Username enumeration in login page 1.1.2. Attacker coerces user to disclose username 1.1.3. Attacker uses SQL Injection to retrieve username (See #3) 1.1.4. Attacker has direct access to SQL server <ol style="list-style-type: none"> 1.1.1.1. Attacker has a valid SQL Login <ol style="list-style-type: none"> 1.1.1.1.1. Attacker views tables directly

Build Phase



Build Phase

- ▶ 5~50 coding mistakes are found in every 1000 lines of code
- ▶ Large applications could potentially have thousands of security vulnerabilities
- ▶ Security relies on
 - ▶ Knowledge of past issues
 - ▶ Good coding practice
 - ▶ Regular code review

Secure Coding

- ▶ Develop platform and language specific coding standards
 - ▶ Identify known vulnerabilities
 - ▶ Assess capability
- ▶ Utilize a software development kit (SDK)
 - ▶ Common code base established with detailed review
 - ▶ Encourage consistency and decrease effort required for secure coding
- ▶ Code review
 - ▶ Establish policies for manual / automated code review during development
 - ▶ Dedicated application testers separate from the development group

Secure Coding

- ▶ Typically what we see in security coding is tied to OWSAP top 10
 - ▶ Injection
 - ▶ Cross Site Scripting (XSS)
 - ▶ Broken Authentication and Session Management
 - ▶ Insecure Direct Object References
 - ▶ Cross Site Request Forgery (CSRF)
 - ▶ Security Misconfiguration
 - ▶ Failure to Restrict URL Access
 - ▶ Unvalidated Redirects and Forwards
 - ▶ Insecure Cryptographic Storage
 - ▶ Insufficient Transport Layer Protection

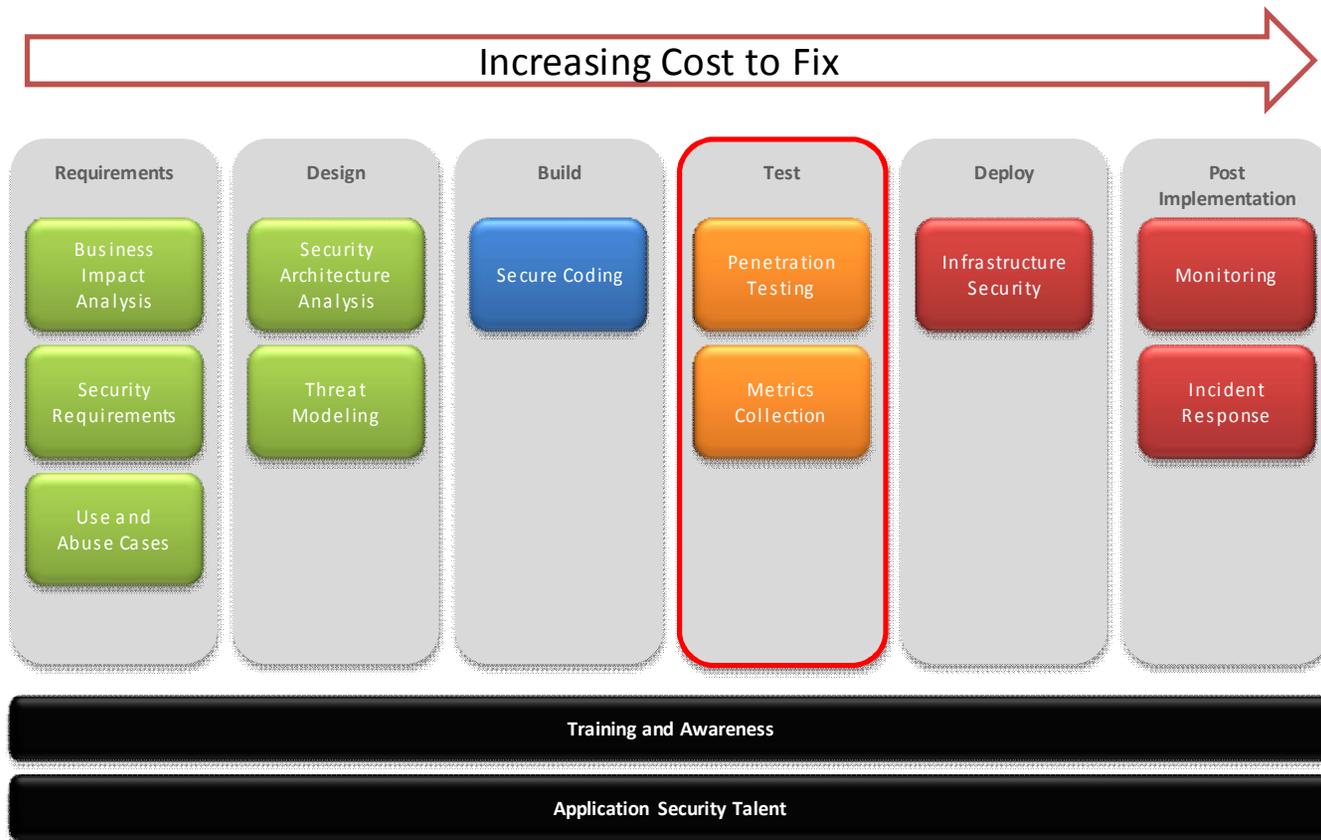
Secure Coding

Vulnerability Category	Summary
Injection Flaws	Injection flaws, particularly SQL injection, are common in web applications. Injection occurs when user-supplied data is sent to an interpreter as part of a command or query. The attacker's hostile data tricks the interpreter into executing unintended commands or changing data.
Cross Site Scripting (XSS)	XSS flaws occur whenever an application takes user supplied data and sends it to a web browser without first validating or encoding that content. XSS allows attackers to execute script in the victim's browser which can hijack user sessions, deface web sites, possibly introduce worms, etc.
Broken Authentication and Session Management	Account credentials and session tokens are often not properly protected. Attackers compromise passwords or authentication tokens to assume other users' identities.
Insecure Direct Object Reference	A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, database record, or kErnst & Young, as a URL or form parameter. Attackers can manipulate those references to access other objects without authorization.
Cross Site Request Forgery (CSRF)	A CSRF attack forces a logged-on victim's browser to send a pre-authenticated request to a vulnerable web application, which then forces the victim's browser to perform a hostile action to the benefit of the attacker. CSRF can be as powerful as the web application that it attacks.

Secure Coding

Vulnerability Category	Summary
Security Misconfiguration	Security settings for the application, framework, web server, application server and platform should be defined, implemented and maintained as many are not shipped with secure defaults.
Failure to Restrict URL Access	Frequently, an application only protects sensitive functionality by preventing the display of links or URLs to unauthorized users. Attackers can use this weakness to access and perform unauthorized operations by accessing those URLs directly.
Unvalidated Redirects and Forwards	Unvalidated redirects and forwards describe scenarios where websites use untrusted data to redirect and forward users to other pages and websites. Since the redirection code resides on a trusted domain, attackers can leverage this vulnerability to redirect victim to phishing sites or malware sites.
Insecure Cryptographic Storage	Web applications rarely use cryptographic functions properly to protect data and credentials. Attackers use weakly protected data to conduct identity theft and other crimes, such as credit card fraud.
Insecure Communications	Applications frequently fail to encrypt network traffic when it is necessary to protect sensitive communications.

Test Phase



Test Phase

- ▶ Locate remaining vulnerabilities from previous phases and issue remediation
- ▶ Key components
 - ▶ Clearly defined policies, procedures and methodology
 - ▶ Dedicated application security test team
 - ▶ Ongoing training and knowledge transfer
 - ▶ QA and business involvement

Penetration Testing

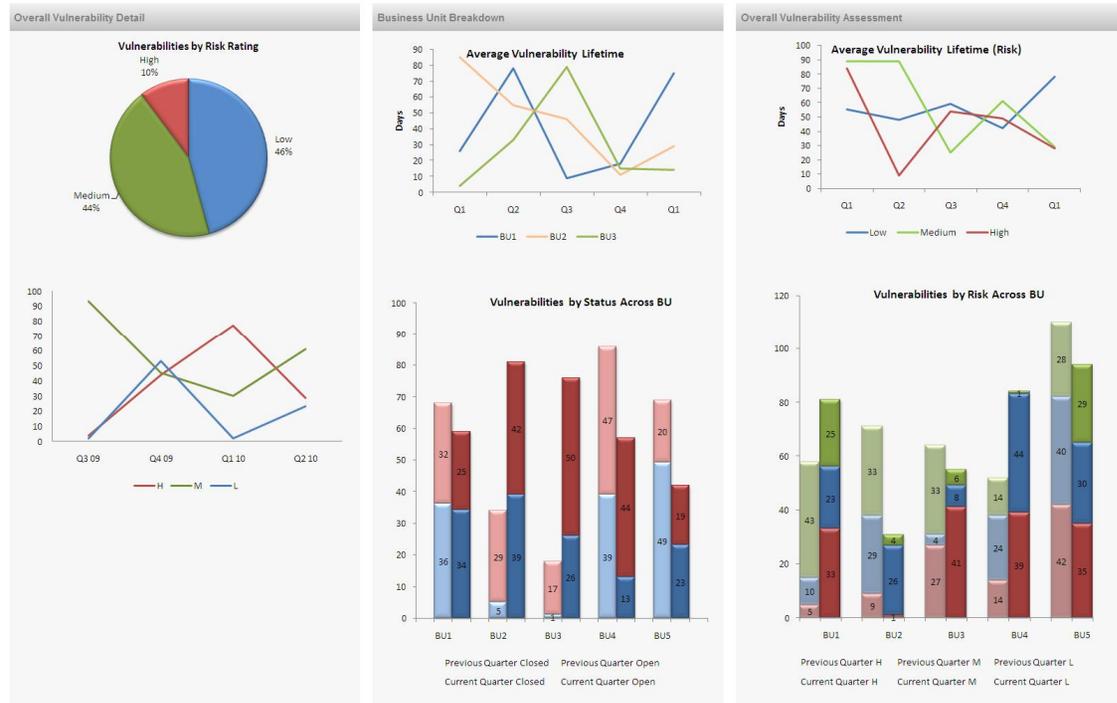
- ▶ Find defects in source code + Black Box assessment [Hacker Mindset]
- ▶ Higher assurance: Hard coded passwords, logic defects, backdoors, etc
- ▶ Win Credibility from Technology and Business

Metrics

- ▶ Identify reoccurring vulnerabilities
- ▶ Align education and awareness efforts
- ▶ Show *Risk Reduction* dashboard [compare BUs] to Senior Leadership: Healthy competition between the BUs

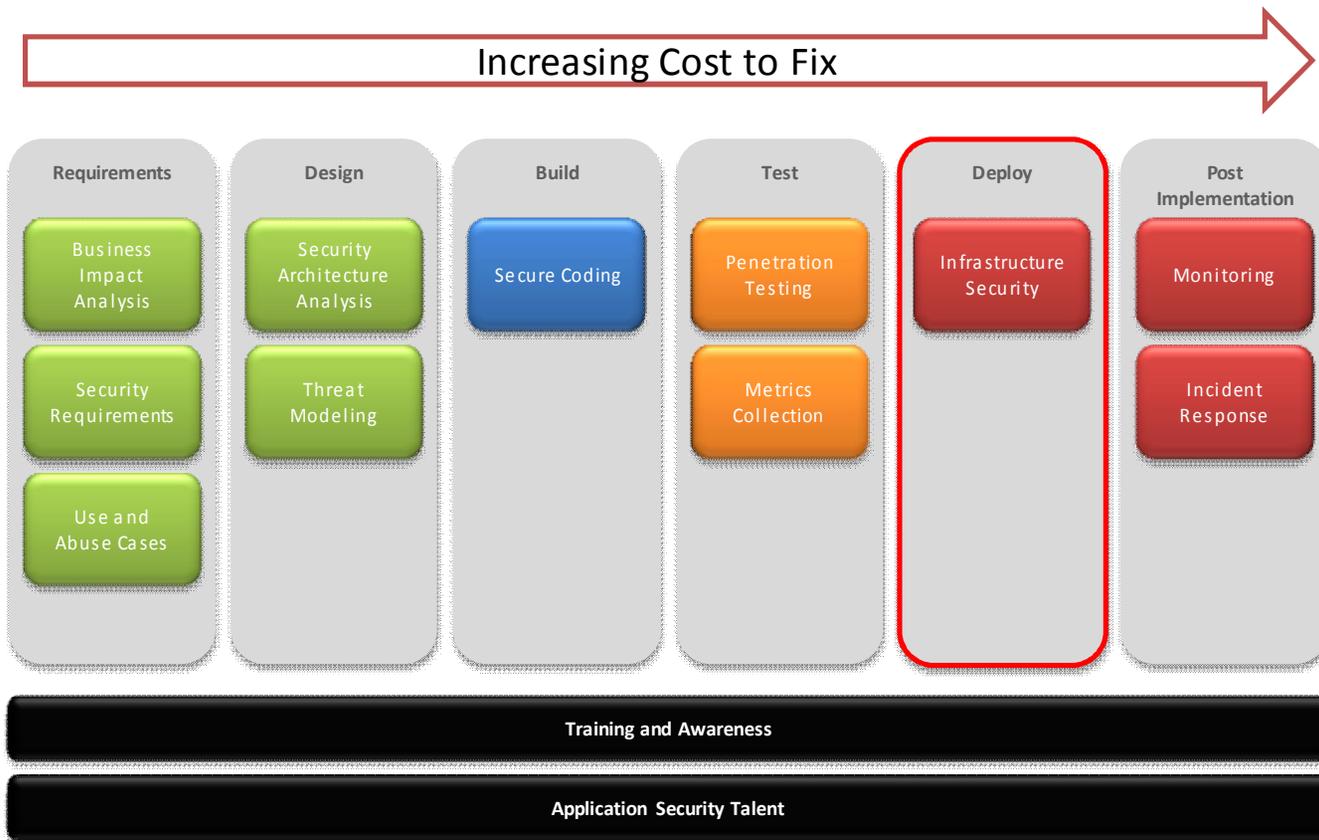
Metrics Continued

Executive Level Overview



Business Unit	Application Name	Risk Score	Vulnerability	Vulnerability Category	Vulnerability Description	Application Risk Assessment Score	Impact	Likelihood	Days Outstanding	Date Discovered	Date Closed	Risk
IT	App1	120.0	blah	XSS		11	15.0	8.0	83	2/19/2010	5/13/2010	High
IT	App2	11.0	blah	CSRF		28	11.0	1.0	81	2/19/2010	5/11/2010	Medium
Finance	App 3	28.0	blah	SQL INJ		20	14.0	2.0	60	2/19/2010	4/20/2010	Medium
Finance	App 4	28.0	blah	SQL INJ		11	7.0	4.0	86	2/19/2010	5/16/2010	Medium
HR	App 5	20.0	blah	AUTH		25	5.0	4.0	69	2/19/2010	4/29/2010	Medium
HR	App 5	35.0	blah	XSS		1	5.0	7.0	74	2/19/2010	5/4/2010	Medium
HR	App 6	21.0	blah	AUTH		29	7.0	3.0	67	2/19/2010	4/27/2010	Medium
HR	App 7	12.0	blah	AUTH		2	4.0	3.0	24	2/19/2010	3/15/2010	Low

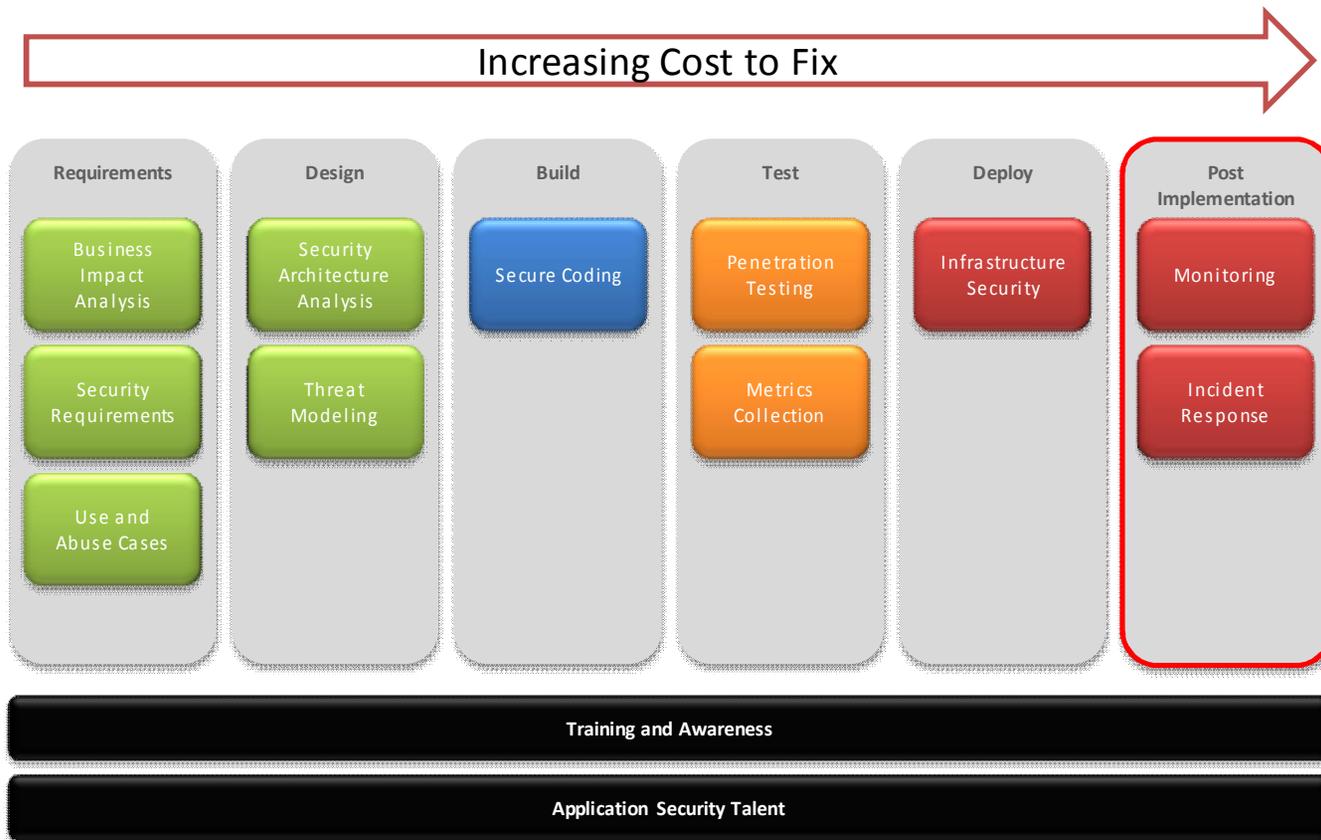
Deployment Phase



Infrastructure Security

- ▶ Assess and remediate potential infrastructure security issues
- ▶ Performed through policies and procedures that tailor to the information security requirement and security architecture phases
 - ▶ Standardized server build and configuration standards
 - ▶ Firewall deployment standards
 - ▶ SLA / vendor agreements

Post Implementation



Post Implementation

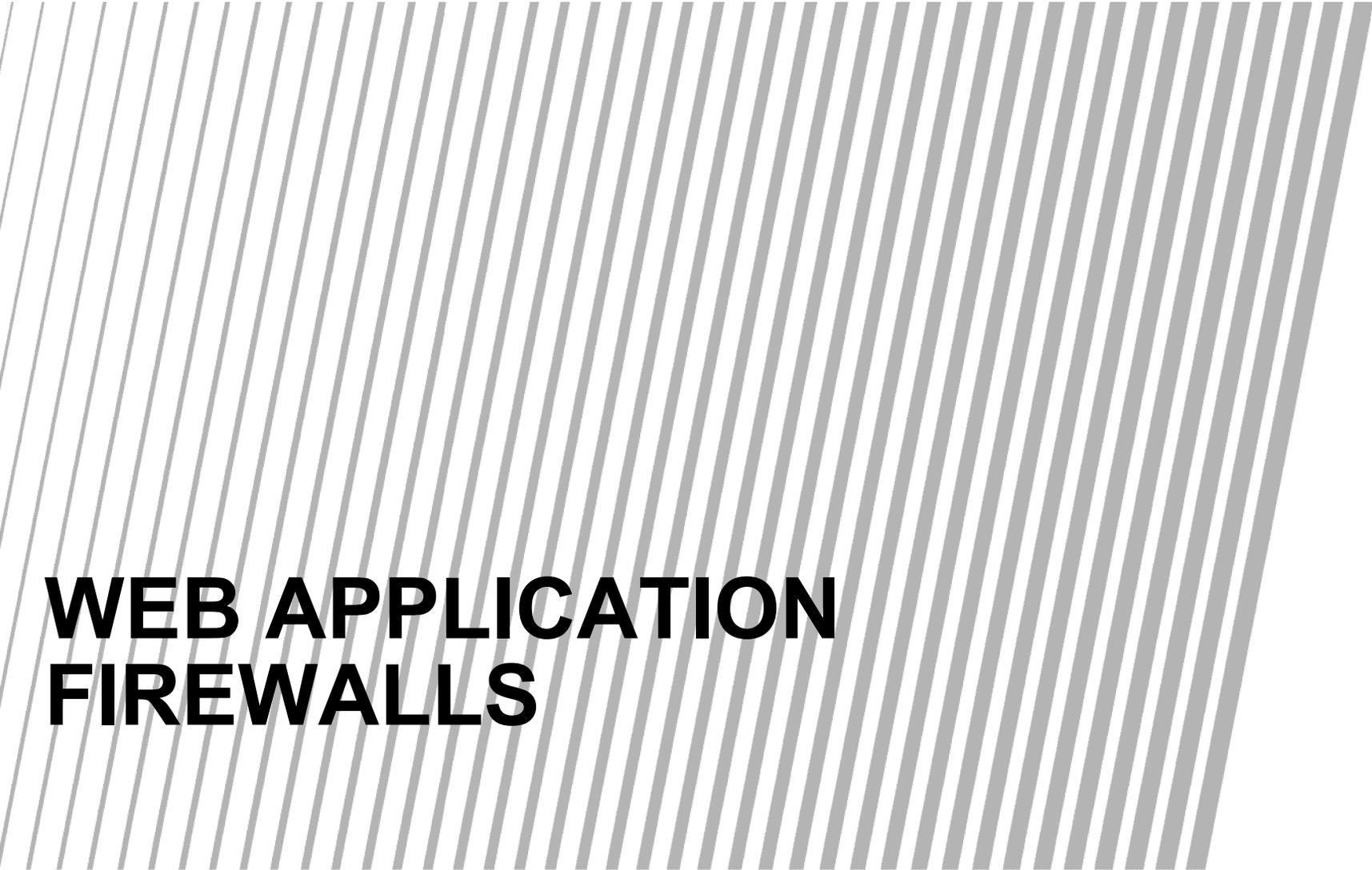
- ▶ Keep up with the latest attack vectors
- ▶ Keep track with inventory
- ▶ Continuously enhance Application Security efforts (processes, implementation, training, communication)
- ▶ Evaluate emerging solutions (Example: Web Application Firewalls)
- ▶ Targeted training:
 - ▶ Top Web Attacks
 - ▶ Based on Grey-box / Black-box Findings
 - ▶ Based on Metrics

Monitoring

- ▶ Performance, security, network and host intrusion events monitoring
- ▶ Identified process and procedures
- ▶ Dedicated monitoring team
- ▶ Effective and reliable monitoring tools

Incident Response

- ▶ Documented incident response methodology
- ▶ Promote awareness across organization
 - ▶ End users
 - ▶ Monitoring team
 - ▶ Production team
 - ▶ Development team
- ▶ Evidence of post mortem review and process improvement



WEB APPLICATION FIREWALLS

Web application firewall (WAF)

- ▶ A Web application firewall is an appliance, server plug-in, or filter that applies a set of rules to an HTTP conversation
- ▶ WAFs products can be software or hardware appliance
- ▶ Designed to compensate for insecure application coding practices
- ▶ Some WAFs look for attack signatures while others look for abnormal behavior
- ▶ Looks specifically for flaws in the application itself, ignores the traffic at the network level

WAF current state

- ▶ Ability to be configured to prevent specific problems, such as emergency patches
- ▶ Good alternative when application source code cannot be updated in a short time
- ▶ Quick updates to rules based on results from a security assessment
- ▶ Provide a positive security model
- ▶ Protection against known application vulnerabilities

WAF challenges

- ▶ Filtering challenges
 - ▶ SQL – use complex, but valid SQL statement
 - ▶ XSS – complex strings and encodings
 - ▶ Business logic flaws

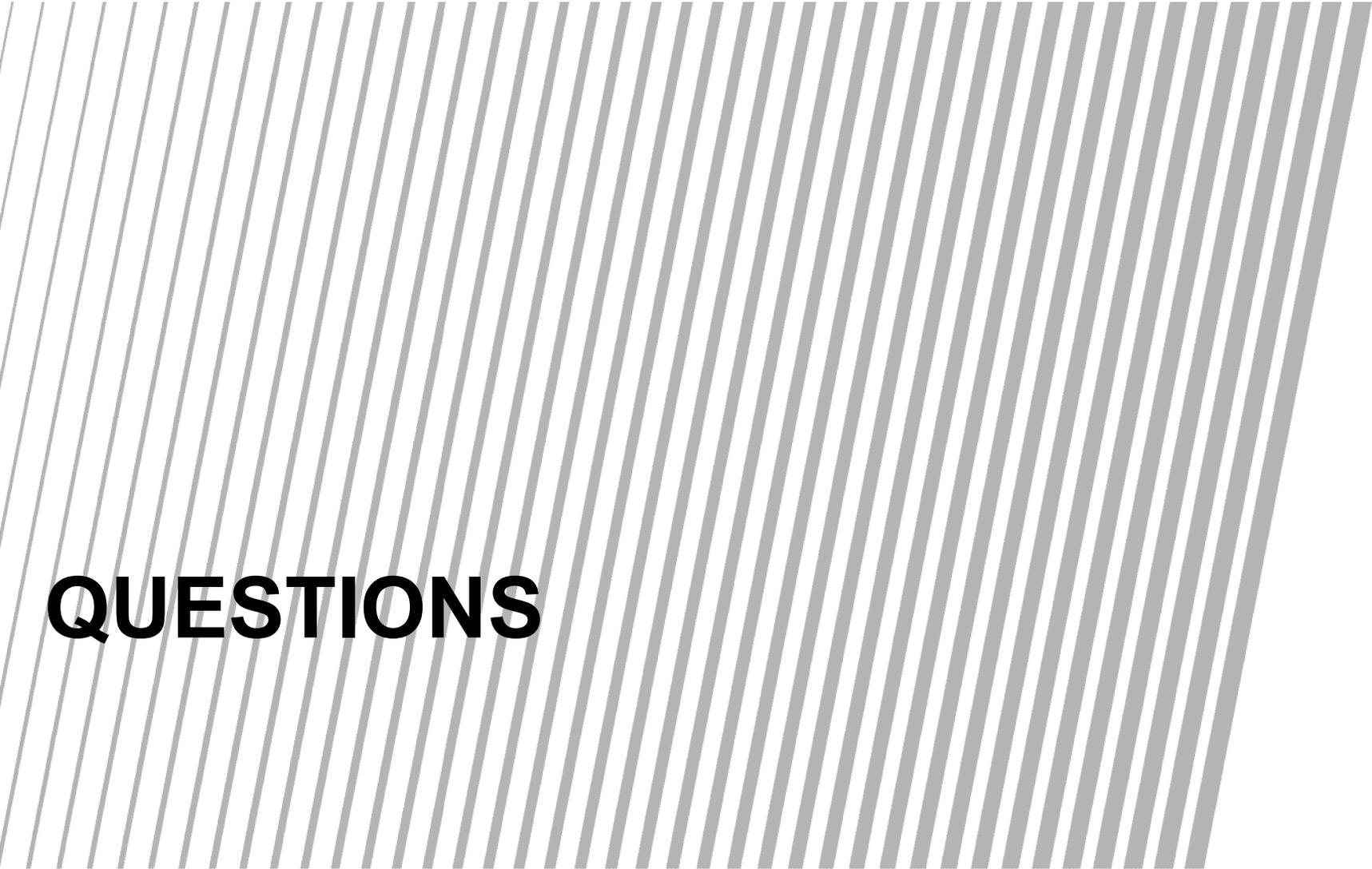
- ▶ Attacks on WAFs
 - ▶ WAF code vulnerabilities - Memory Corruption
 - ▶ Application vulnerabilities in WAFs themselves
 - ▶ Tools to detect if a WAF is in use

Different perspectives

- ▶ Use a WAF
 - ▶ Do not use a WAF / fix the code
 - ▶ Fix the code and use a WAF
 - ▶ Use a WAF until you can fix the code or on low risk apps
- ▶ "Before you spend your first dollar, consider whether you're in a position to remove vulnerabilities through a stronger system development lifecycle and by using tools such as source-code scanners" - Gartner
 - ▶ For most companies, it is sufficient to choose one or the other approach, although there is a small percentage of companies whose risk tolerance is so low that they'll want to use both.

Recommendations

- ▶ There are no patches for vulnerabilities in custom applications
- ▶ There are no silver bullets
- ▶ Do not shift responsibility from developer to a WAF manager
- ▶ Identified vulnerabilities should be imported as customized rules into WAF
- ▶ Use WAF as a stop-gap for backlogs or no access to vendor developed application code



QUESTIONS

Ernst & Young

Assurance | Tax | Transactions | Advisory

About Ernst & Young

Ernst & Young is a global leader in assurance, tax, transaction and advisory services. Worldwide, our 130,000 people are united by our shared values and an unwavering commitment to quality. We make a difference by helping our people, our clients and our wider communities achieve potential.

About Ernst & Young's Technology Risk and Security Services

Information technology is one of the key enablers for modern organizations to compete. It gives the opportunity to get closer, more focused and faster in responding to customers, and can redefine both the effectiveness and efficiency of operations. But as opportunity grows, so does risk. Effective information technology risk management helps you to improve the competitive advantage of your information technology operations, to make these operations more cost efficient and to manage down the risks related to running your systems. Our 6,000 information technology risk professionals draw on extensive personal experience to give you fresh perspectives and open, objective advice – wherever you are in the world. We work with you to develop an integrated, holistic approach to your information technology risk or to deal with a specific risk and security issue. And because we understand that, to achieve your potential, you need a tailored service as much as consistent methodologies, we work to give you the benefit of our broad sector experience, our deep subject matter knowledge and the latest insights from our work worldwide. It's how Ernst & Young makes a difference.

For more information, please visit www.ey.com.

© 2011 EYGM Limited. All Rights Reserved.

Proprietary and confidential. Do not distribute without written permission.

Ernst & Young refers to the global organization of member firms of Ernst & Young Global Limited, each of which is a separate legal entity. Ernst & Young Global Limited, a UK company limited by guarantee, does not provide services to clients.
