How to Create a Software Security Practice

Jack Danahy

Ryan Berg

IBM Security Systems Division

Session ID: ASEC-303

Session Classification: Intermediate

RSACONFERENCE2012

So Why Listen to Us?

- **Jack Danahy** - Director/Advanced Security
- **Ryan Berg** – Senior Security Architect
  - You will note that we are **NOT** Service Providers

- **Founders**
  - Qiave Technologies (Acquired by Watchguard: 2001)
  - Ounce Labs (Acquired by IBM: 2009)

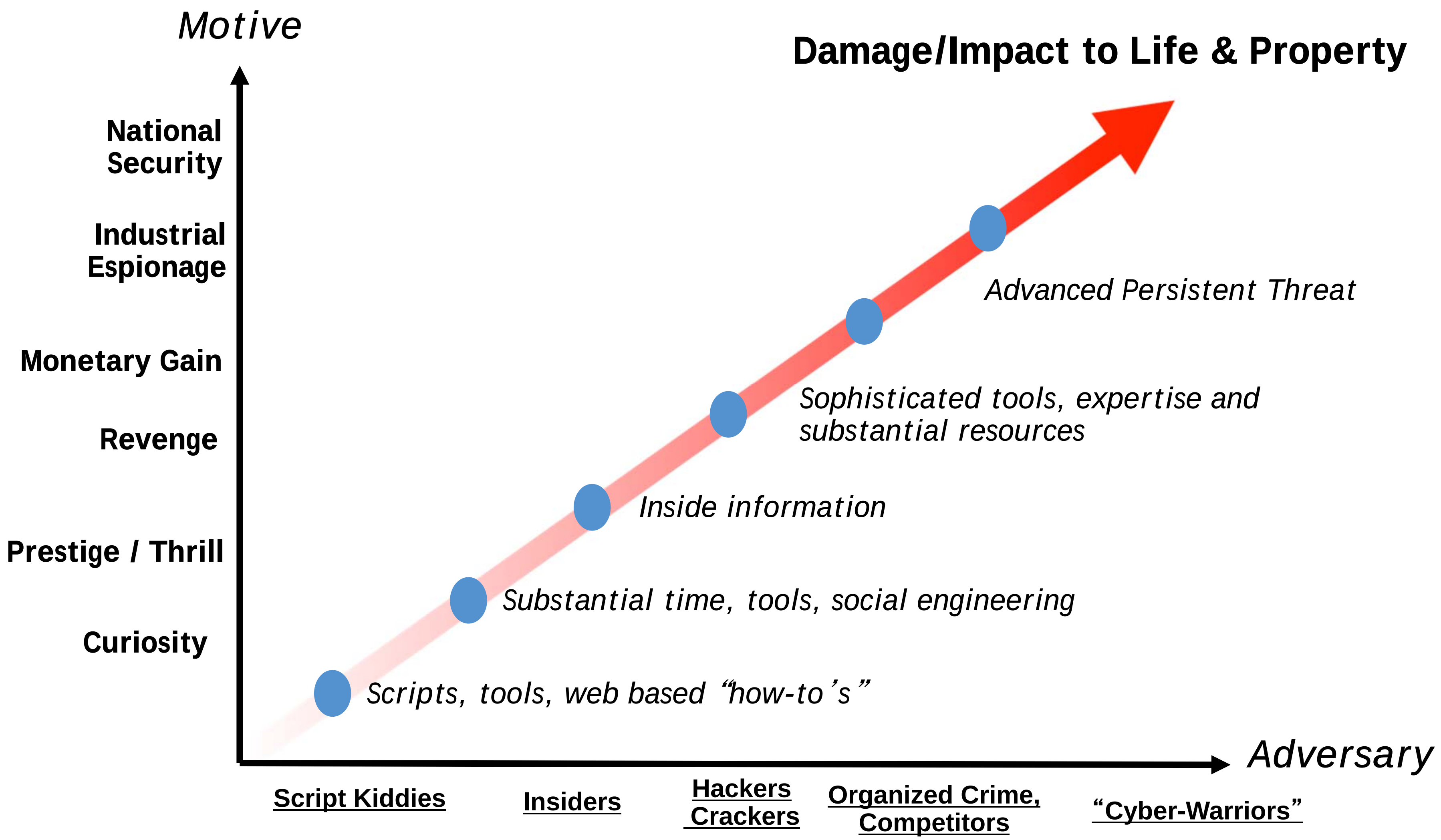
We **have** helped many organizations to create successful software security programs.

What is an Application Security Practice?

Welcome

- An Application Security Practice is:
  - An internal or external center of excellence in assessment and improvement of application security
- Why is There a Need?
  - ***"Forty percent of 678 Fortune 500 and popular websites contain client-side JavaScript vulnerabilities"*** *– X-Force 2011 Trend Report*
  - In 2011, there were 535 reported data breaches, involving 30.4 million records – *privacyrights.org*
    - Applications are the most common target for attack
    - Insecure applications played a role in 5 of top 6 breaches
  - Demand for trained resources far exceeds supply

Marketplace of Attacks is Evolving



Motive

Damage/Impact to Life & Property

National Security

Industrial Espionage

Monetary Gain

Revenge

Prestige / Thrill

Curiosity

Advanced Persistent Threat

Sophisticated tools, expertise and substantial resources

Inside information

Substantial time, tools, social engineering

Scripts, tools, web based “how-to’s”

Adversary

Script Kiddies

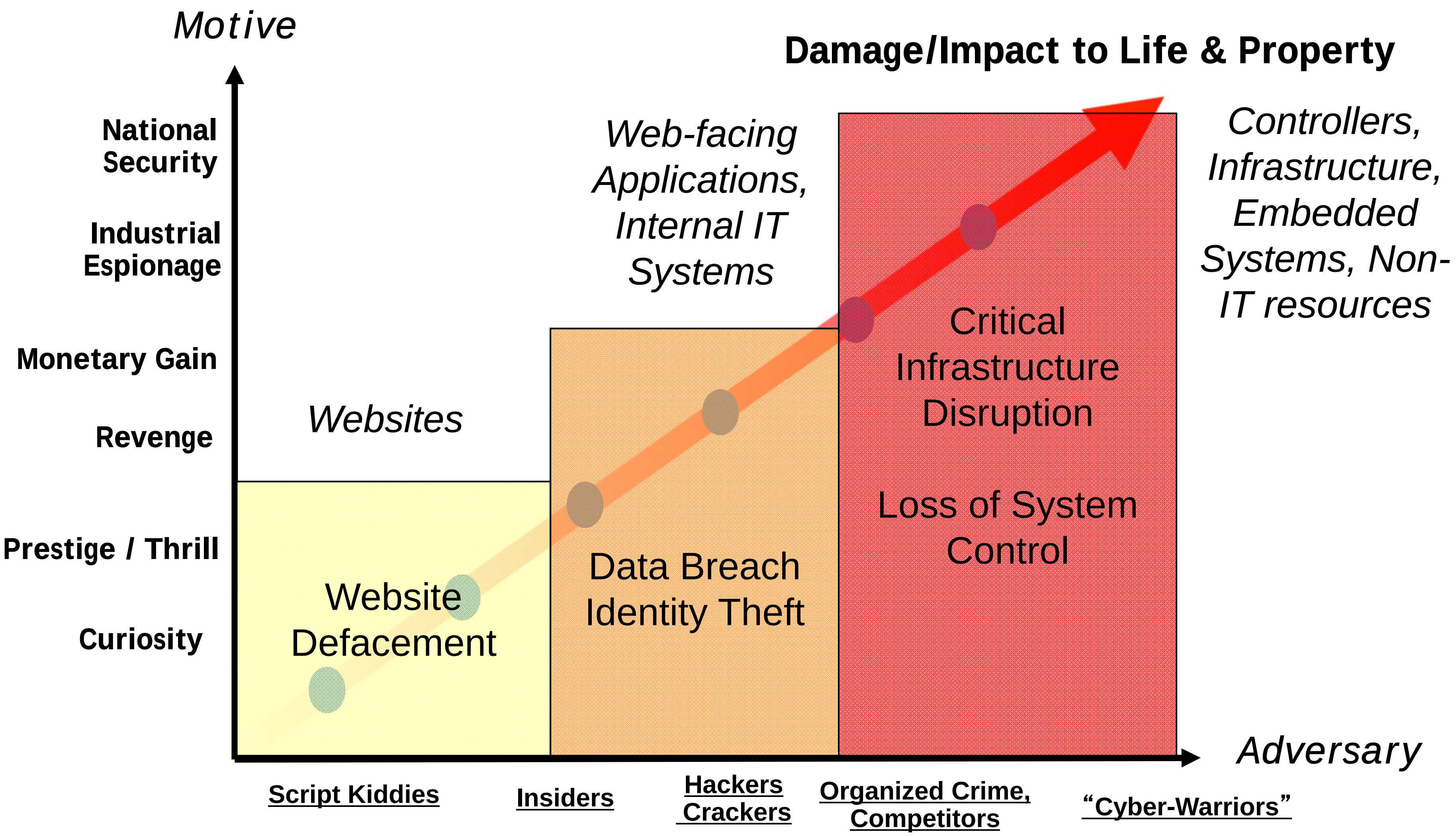
Insiders

Hackers Crackers

Organized Crime, Competitors

“Cyber-Warriors”

Motives, Impacts, and Adversaries



Motive

National Security

Industrial Espionage

Monetary Gain

Revenge

Prestige / Thrill

Curiosity

Damage/Impact to Life & Property

Websites

Website Defacement

Web-facing Applications, Internal IT Systems

Data Breach Identity Theft

Critical Infrastructure Disruption

Loss of System Control

Controllers, Infrastructure, Embedded Systems, Non-IT resources

Adversary

Script Kiddies

Insiders

Hackers Crackers

Organized Crime, Competitors

"Cyber-Warriors"

Benefits to Creating a Practice

- Internal Practice
  - Cost savings from multiple avenues
    - Decreased remediation costs
    - Decreased likelihood of vulnerability exploit
    - Simplified reporting and compliance
  - Increased positive visibility for resources
  - Beneficial center of gravity for expertise
- External Practice
  - Constant demand for trained resources
  - Full life-cycle engagement
  - Premium service and resource returns

Resources and Skills Needed

- Technical Skills Required
  - Knowledge of application vulnerability types and causes. Multiple sources for skill improvement
  - Familiarity with dynamic testing methodologies and toolkits for deployed application testing
  - Programming, Release/Integration experience for static testing during development process
- Non-technical Skills Required
  - Organized triage/project management
  - Client briefing delivery and prioritization

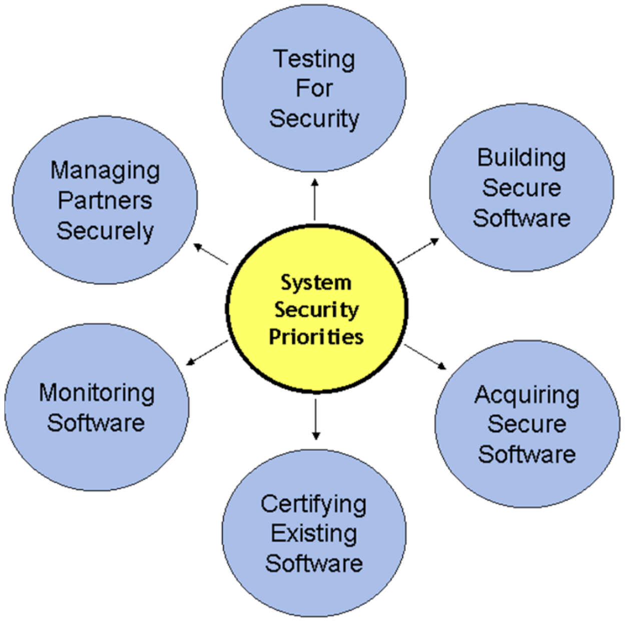
What activities are involved?

- Activities Driving an Application Security Practice
  - **Application Inventory** : Assisting the organization to identify all applications
  - **Asset Prioritization** : Developing a rationale around the value/impact/importance of each application
  - **Application Assessment and Analysis** : Performing the actual assessments of individual applications
  - **Application Vulnerability Remediation** : Suggesting remediation plans and process to address issues
  - **Application Security Integration** : Models to make application security a regular lifecycle component

What is an Application Inventory?

Why an Application Inventory?

- Most organizations have an incomplete awareness of applications
- Software and Systems have added capability quickly
- There typically exist disconnects in governance and provenance
- Constrained resources and critical threats demand prioritization


Testing For Security
Building Secure Software
Acquiring Secure Software
Certifying Existing Software
Monitoring Software
Managing Partners Securely
System Security Priorities

> *"It happens, indeed, to be the case that a thing to which movement this way and that is equally (in)appropriate is obliged to remain at the center."*
>
> - **Aristotle**
> *De Caelo*, Book II

Steps to an Application Inventory

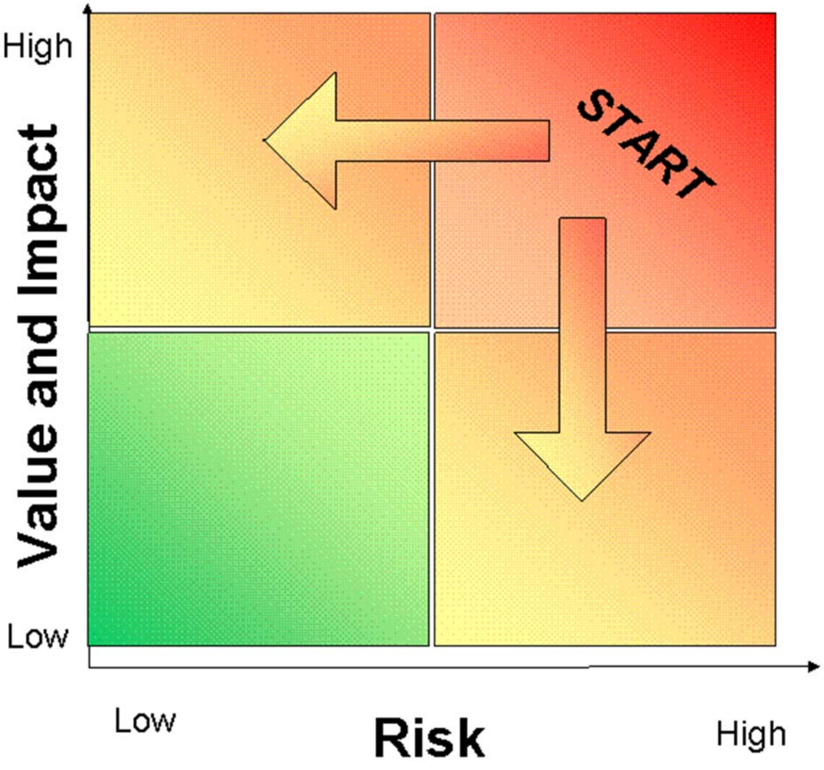
- Outline Scope for Inventory
  - Commonly segregated by Source, Purpose, or Business Unit
- Identify Providers of Insight within Scope
  - Group Interaction for Communication and Awareness
  - Functional Group Insight to Improve Accuracy
- Populate inventory with demographic data
  - Application business function
  - Application budget owner
  - Application operational management
  - Application development/project team
  - Any application security resources

Steps in Application Inventory (2)

- Highlight Application Lifecycle Phase
  - Under Discussion : Lowest cost to add Security
  - Under Development : Capacity to influence SDLC
  - In Test : Opportunity to add Security to test matrix
  - In Deployment
- Describe Application Architecture
  - Monolithic
  - Composite
  - Cloud-based
  - Heterogeneous in platform, language, and/or provider
  - Supporting technologies and mitigating controls

Mapping Value vs Exposure

Goal – Understand value, breach impact, and exposure for applications



Value and Breach Impact

- Capture the value to the organization of the application
- Revenues derived from application
- Investment in system to date
- Impact costs if system were corrupted or compromised

Risk: Audience and Exposure

- Characterize the expected user interaction and exposure of the application
- Capture the profile of users and authorization activity
- Capture the network exposure and compartmentalization
- Identify any inter-system/inter-application connection points

Prioritizing Application Assessment Efforts

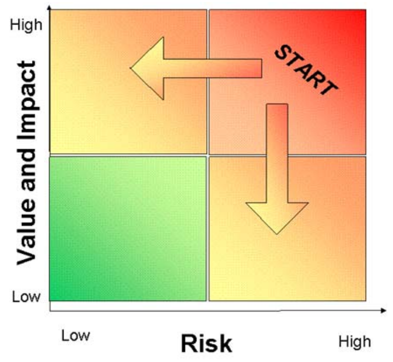
Prioritizing Your Workflow

Activity
Define your risk
Know your Enemy
Prioritize

Define Risk

- Every Organization has a different way of categorizing application risk.
  - Internet vs Intranet
  - PII vs credit card data
  - Recipe for Secret sauce

- It is important to define application risk in terms the business can understand (high med, low, not good enough)



Know your Enemy

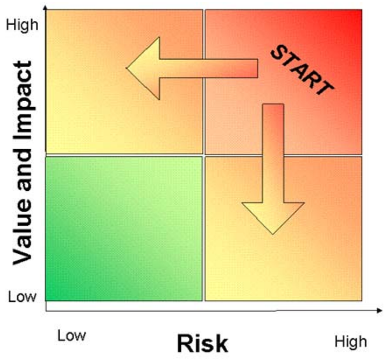
- Each type of attacker has a different motivation
- Define the cope of the potential thread
  - Internal vs External
  - All users vs authenticated users
- Leverage threat modeling
  - No one side fits all, but pick one

Image: chanpipat / FreeDigitalPhotos.net

https://www.owasp.org/index.php/Threat_Risk_Modeling#Alternative_Threat_Modeling_Systems

Prioritize

- Prioritize



  - Each Application identified in the inventory needs to me mapped against the risk
  - Prioritize based on identified risk and scope of the potential threats
    - Don’t lose the forest in the trees

Performing the Assessment

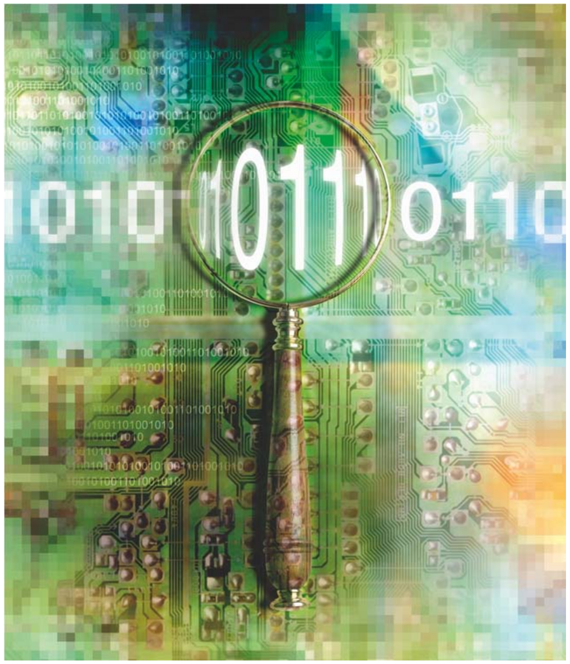
Assessing your inventory

Activity
Discovery
Technical Team kickoff
Initial Assessment and Planning
Assessment
Final Report

Discovery

"Strategy without tactics is the slowest route to victory. Tactics without strategy is the noise before defeat."
Sun Tzu

- Discovery is the first and foundational step
  - Acquire all the code
  - Configuration Files
  - Third party dependencies
  - Buildable environment
  - Recursive deployed application directory
  - Design Documents
    - Architectural ducuments
    - Design patterns used
    - Frameworks
      - MVC, MVVM, Spring, Struts, EJB
  - Running test version of the application (matching the you you received as part of discovery)
    - Multiple test accounts for each entitlement

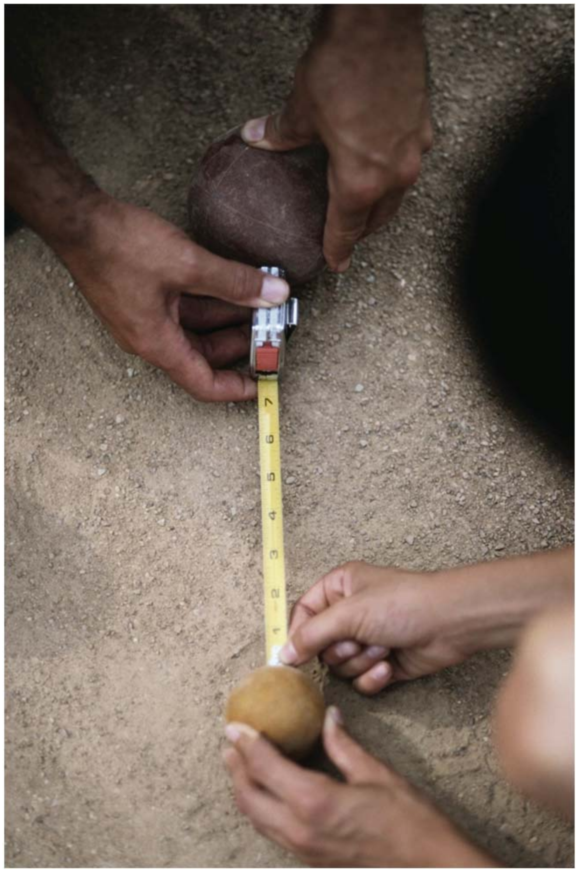


Estimation based on Discovery

- Too often estimation occurs prior to discovery.
  - At best assume 100,000 LOC per week

- Well executed discovery will influence and drive more predictable and accurate time estimates

- Things that drive estimates that can only be understood during discovery
  - Undocumented “features”
  - Custom frameworks
    - IOC anybody
  - Legacy System interfaces
  - Entitlements



Technical Team Kickoff

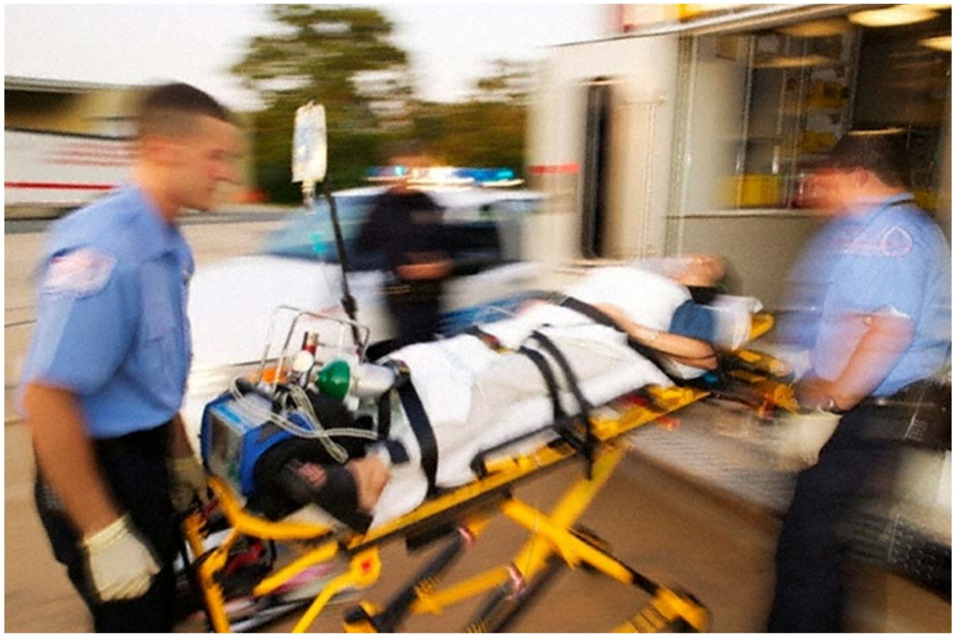
- Discovery should take 1-2 weeks (depending on size and complexity of the code base)
- The kickoff is to sync between the team doing the assessment and the application development team.
  - Ask outstanding questions about key artifacts found during discovery
  - Have an application walk through of key features (live demo)
  - Gain an understanding of any secure development practices already implemented and how they are utilized
  - Outline the assessment process and identify development points of contact

Initial Assessment and Planning

- Initial assessment
  - Tools, Tools, Tools
- Leveraging both SAST, DAST the initial assessment is to gain initial understanding of the application typically using default tool configurations
- Focus on building scan assurance
  - Do you have all the expected data flows in the application
  - Is your black box scan complete
- The goal of the assessment is building high assurance, defensible results so ensuring that the tools are giving the most complete picture possible is critical
- Create plan for handling tool gaps
  - Manual analysis, pen testing, code review

Assessment

- Triage, Triage, Triage
- Divide and conquer
  - Identify common insecure patterns
  - Validate findings against test web site
  - Communicate with development on critical findings

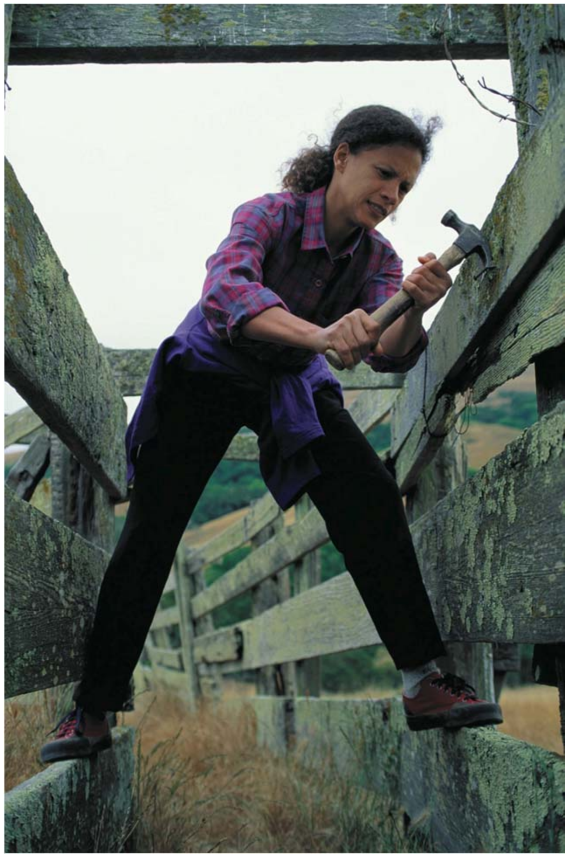


Remediating Found Issues

RSA CONFERENCE 2012

Remediating Found Issues

- Nobody wants to read a 500 page PDF
  - Leverage tooling to provide access to detailed findings
  - Summary report should contain
    - representative examples of the most critical findings
    - Architectural findings
      - Lack of standardized encoding
      - Lack of consistent use of secure practices
    - Risk Ranking for the findings
      - DREAD,CVSS (pick one but be consistent)
  - Meet with development team and walk through the high level findings and major areas of concern.



Remediating Found Issues

- All issues that require remediation need to be tracked
  - Leverage defect tracking system but make sure you can identify security from non-security issues
- Prioritize based on Risk ranking
- Avoid spot fixing
  - XSS requires development of a consistent framework for proper handling.
- Avoid duplicate fixes for similar issue
- Leverage secure development framework (if you don't have one now is the time to invest)

An Integrated Assessment Effort

A Practical Cycle Described

Design Phase

- Consideration is given to security requirements of the application
- Issues such as required controls and best practices are documented on par with functional requirements

Development Phase

- Software is checked during coding for:
  - Implementation error vulnerabilities
  - Compliance with security requirements

Build & Test Phase

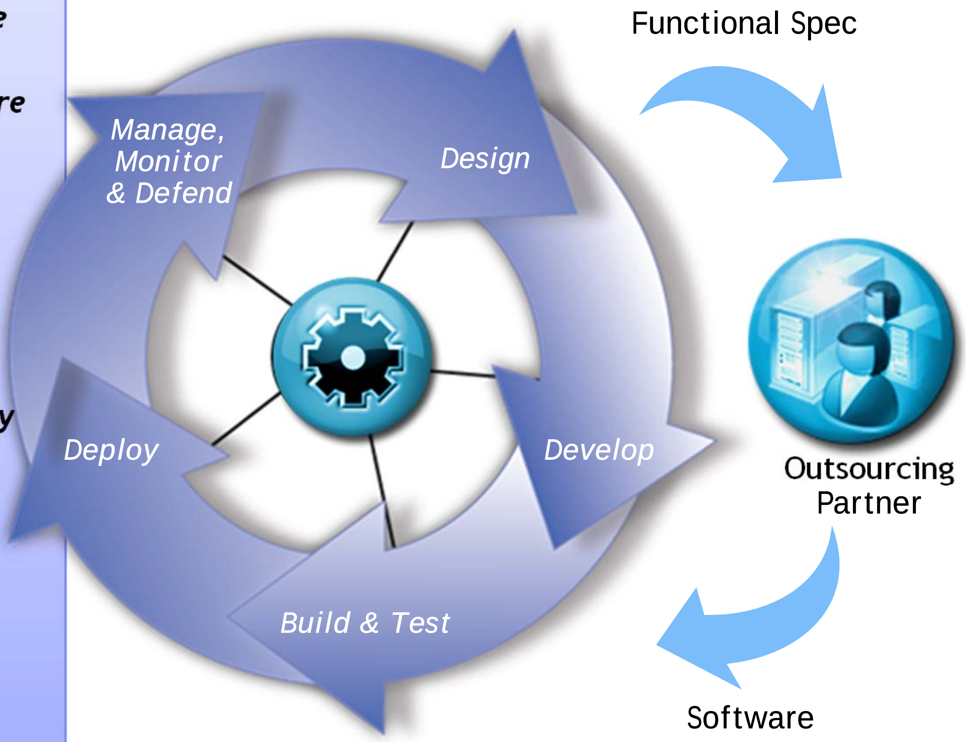
- Testing begins for errors and compliance with security requirements across the entire application
- Applications are also tested for exploitability in deployment scenario

Deployment Phase

- Configure infrastructure for application policies
- Deploy applications into production

Operational Phase

- Continuously monitor applications for appropriate application usage, vulnerabilities and defend against attacks



Applying All of This Information

- In the first three months following this conversation, you should go forth and:
  - Document your thoughts on organizational need or opportunity for a Software Security Practice
  - Evaluate your capability to provide these services, finding areas of necessary growth
  - Identify internal champions or external clients interested in working through the process
- In the first 6 months following this conversation, you should plan to:
  - Develop the necessary skills and take on your first project, documenting all the way along. Then drop us an email.

Questions?