RSAConference2015
San Francisco | April 20-24 | Moscone Center

SESSION ID: ASD-R02

A Case Study in Building an AppSec Program: 0-60 in 12 months

Robb Reck

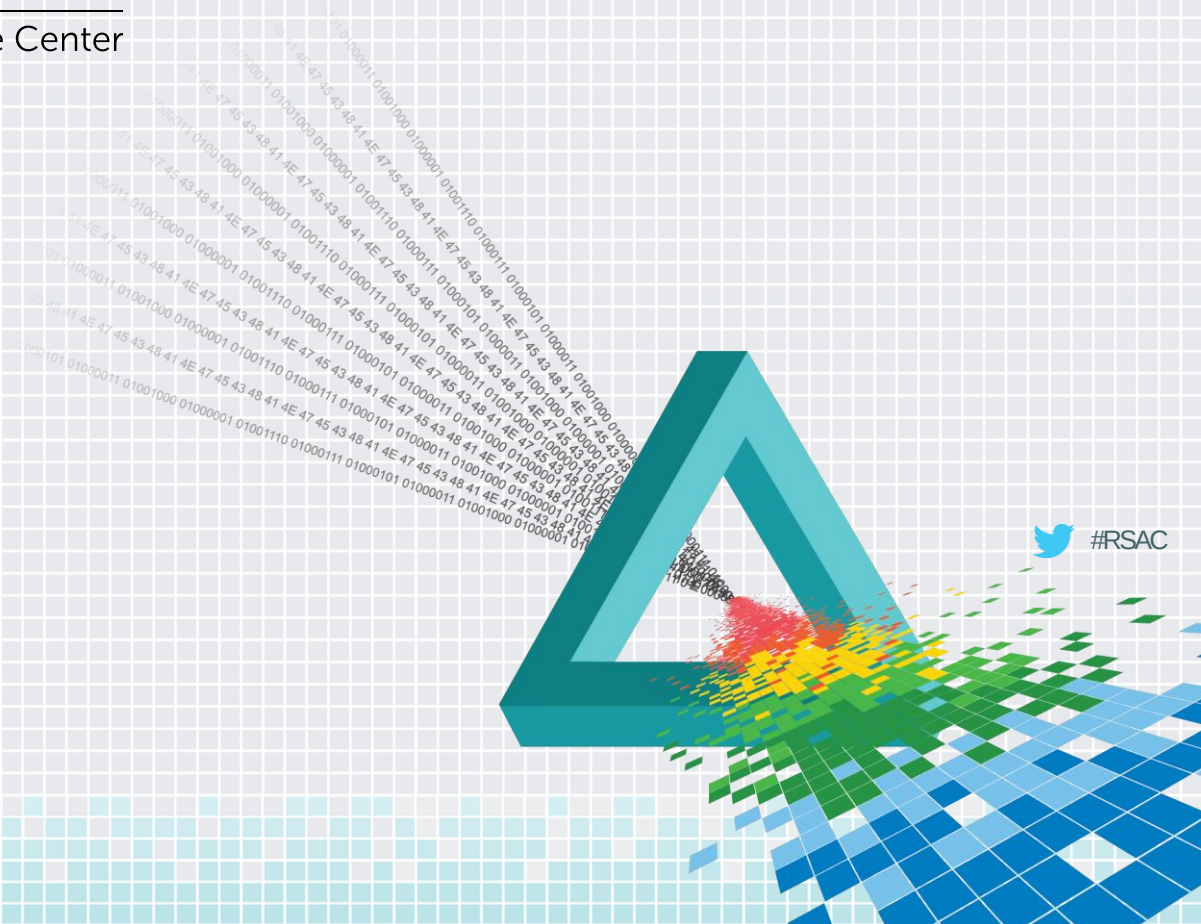
VP, CISO
Pulte Financial Services
@robbreck – robbreck@gmail.com

Agenda

- The Problem
- Implementation Steps
- Control Points
- Implementation Hurdles
- Summary – Lessons Learned
- Apply It

Presenter's Company Logo – replace on master slide

The problem

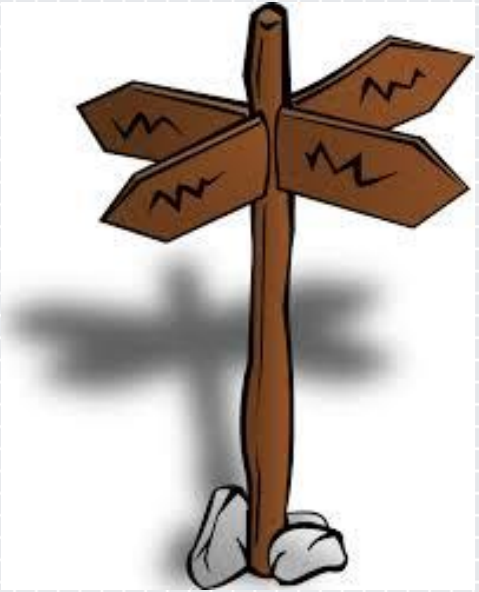


Can you image?

- Security department viewed as a roadblock, and avoided whenever possible
- Software that is used to process millions of banking transactions developed without an application security program in place
- Regulatory pressure to have an AppSec program in place yesterday

Presenter's Company Logo – replace on master slide

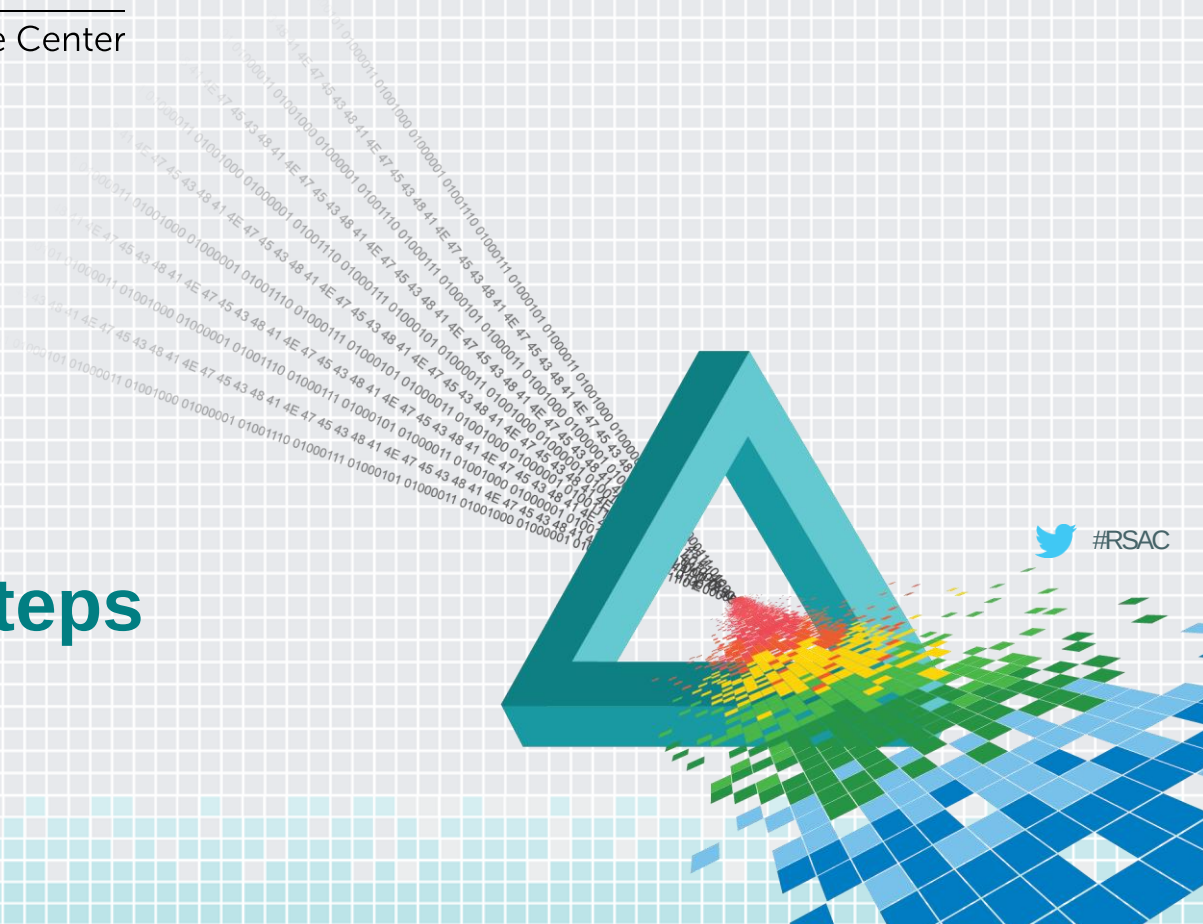
What would you do?



- I am not an application security guru
- Chances are, you're not an AppSec guru either
- Here is the story of what I did, and how it went

Presenter's Company Logo – replace on master slide

Implementation steps



It's all about the relationships

- Got buy in
- Different relationships with different levels in the organization
- Built support both high and wide.
  - Started with the highest level of support I could find
  - Got executive champions from multiple departments – development, legal, product development, and more
- Figured out how to turn the project into a 'win' for every individual and team involved

Presenter's Company Logo – replace on master slide

Prioritize the work

- Didn't try to do it all at once
- Risk ranked the applications (more on this later)
- Gathered input from as many stakeholders as possible – developers, infrastructure, legal, product management, finance, and many others
- Fed the plan back to the stakeholders – showed them what we created collaboratively, and what their role will be

Presenter's Company Logo – replace on master slide

Creating champions

- Champions keep program momentum while I'm looking the other way
- The senior technical folks are often excited for the new challenge
- Junior level developers may see this as a way to differentiate themselves versus their peers
- In either case, worked with their managers to get their security responsibilities formally recognized in their review process
- Success breeds success, and champions breed other champions

Presenter's Company Logo – replace on master slide

Putting the controls in place

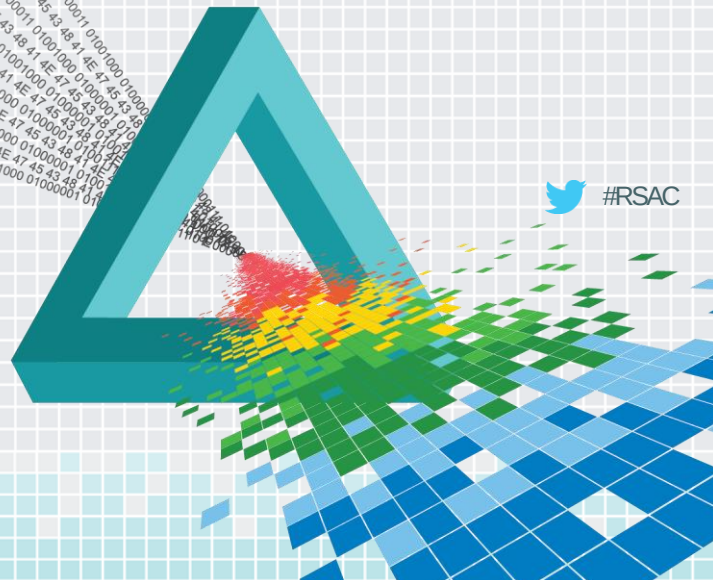
- One control at a time!
- Identified an associated effectiveness measurement from the very beginning
- Didn't move on to the next control until I had achieved success with the previous

Presenter's Company Logo – replace on master slide

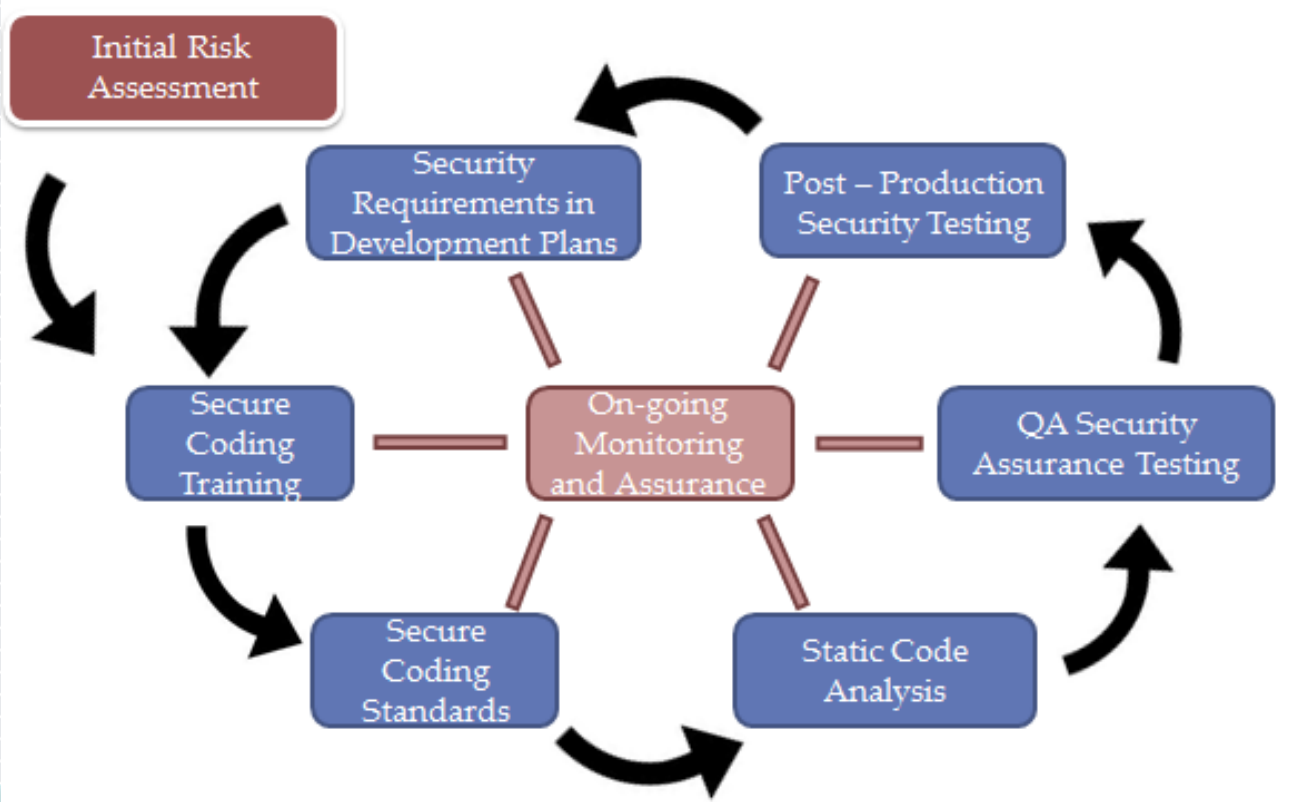
RSAConference2015

San Francisco | April 20-24 | Moscone Center

Control Points



Security in the SDLC



Presenter's Company Logo – replace on master slide

Initial Risk Assessment

- Performed initial risk assessment of all enterprise applications to tier them
- This assessment was used to determine which applications should get focus first
- Included numerous elements (many non-technical) to determine relative risk

Presenter's Company Logo – replace on master slide

Project reviews

- Formal involvement in all software development projects
- Utilized the Project Management Office for access
- Created a series of questions to understand the impact of the project
- Created list of security action items based on project review – action items managed by project manager in normal project scope
- Medium level of effort for the first project on given application
- Low level of effort for subsequent projects

Presenter's Company Logo – replace on master slide

Secure coding training

- Rolled out to highest risk development teams, based on the results of the risk assessment
- Partnered with a provider for a suite of CBT courses
- All developers, QA, and PMs were assigned a general secure coding principles CBT
- Developers were assigned a technology specific course
- QA were assigned testing specific course
- PMs were assigned an SDLC focused course

Presenter's Company Logo – replace on master slide

Secure coding standards

- Organization-wide standard for how code will be (securely) written
- Championed by security
- Written by the development department
- Technology agnostic, but addresses key principles
- Used CERT's "Top 10 Secure Coding Practices" as the starting point
- Deployed and trained all developers on the standard

Presenter's Company Logo – replace on master slide

Static code analysis

- Regularly recurring static application testing throughout the development process
- Integrated into the code repository, and tested automatically with a nightly build process
- Results integrated directly into the ticketing system
- All high flaws had to be remediated before the release could be approved

Presenter's Company Logo – replace on master slide

Security QA testing

- Eventually QA should be the ones who own all application flaws (security or otherwise) found in the QA environment
- Initially the security team had the expertise to run application testing tools
- Partnered security resources with QA resources to cross-train

Presenter's Company Logo – replace on master slide

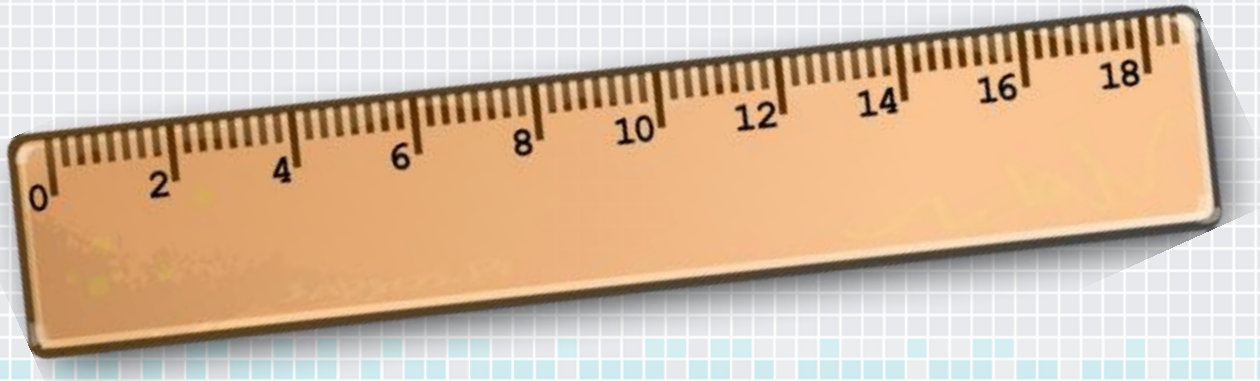
Dynamic testing in production

- Similar to the testing performed in QA
- Regularly scheduled scans of all production systems
- Designed to find anything that slipped through the cracks or newly discovered system level vulnerabilities

Presenter's Company Logo – replace on master slide

Metrics, monitoring and reporting

- What gets measured gets done
- Created a dashboard of application security metrics, showing the effectiveness of each control
- Promoted adoption among all teams



Presenter's Company Logo – replace on master slide

Example Metrics

Secure Code Training

Training Completed on Time:

Status: COMPLIANT

Date	Training 1 (Due 9/30)	Training 2 (Due 12/31)
12/1/2014	47 of 47 complete	43 of 47 complete

App 1 Dynamic Testing

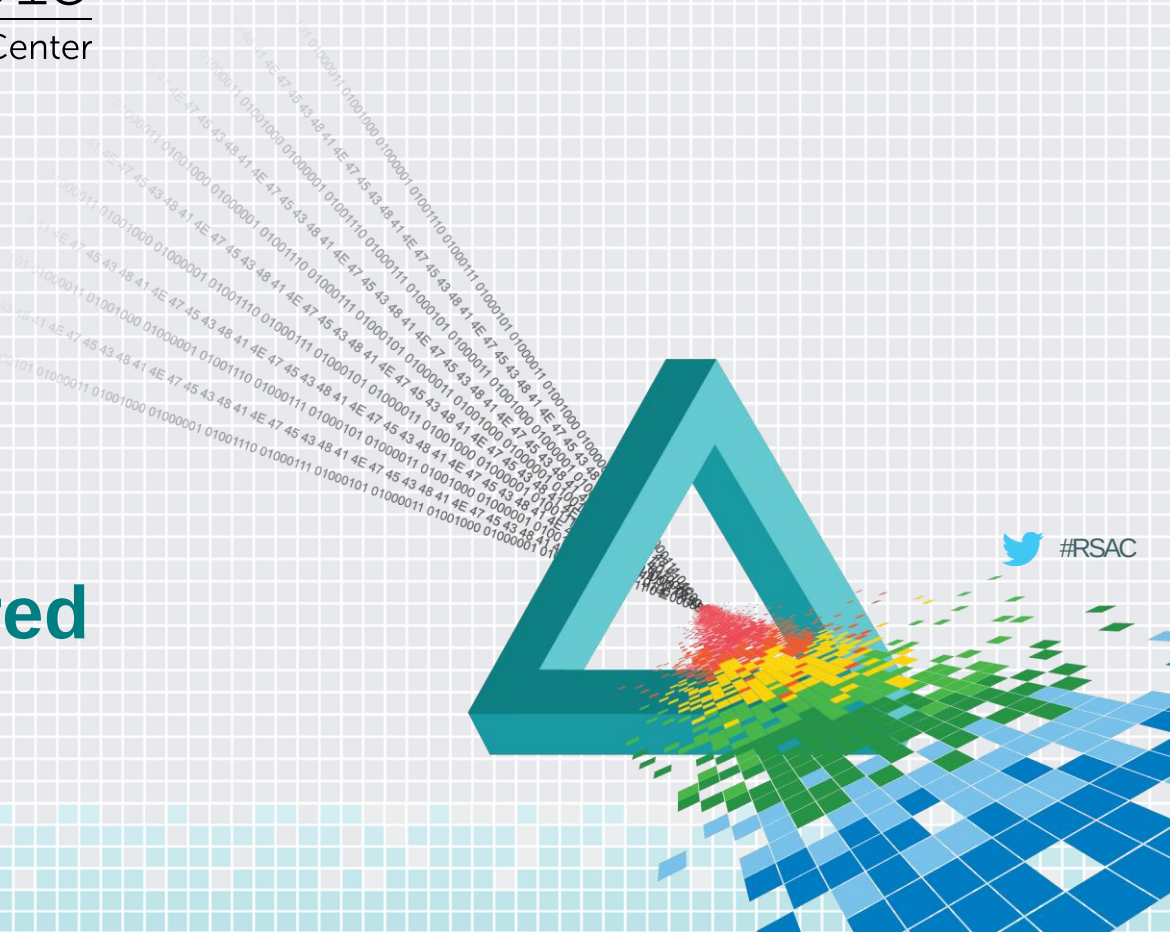
Flaws Fixed within SLO:

NON-COMPLIANT

Date	Critical	Outside SLO	High	Outside SLO	Medium	Outside SLO
12/1/14	0	0	1	1	7	0

Presenter's Company Logo – replace on master slide

Hurdles encountered



Resistance to security

- Legacy of being “The Department of No”
- Perception of security being an infrastructure problem (“Aren’t they the firewall guys?”)
- The need to create relationships in new areas

Presenter’s Company Logo – replace on master slide

Consensus on the controls

- More stakeholders help get buy-in, but too many cooks…
- Many have worked with technologies in the past
- Nobody wants to be held accountable to a standard they didn't create
- Lengthy process testing controls across vastly different teams

Presenter's Company Logo – replace on master slide

Varying teams and SDLCs

- Waterfall, Agile, Ad Hoc and more
- .Net, Java, KOBOL, and more
- Multiple states, and even multiple countries
- Different release cycles
- Each difference required flexibility, but thrived when relationships were strong

Presenter's Company Logo – replace on master slide

Moving from compliance to security

- We started this as a compliance initiative, how do we turn it into a value-add?
- Finding the ‘win’ for every team
- Reduce security incidents, reduce bug fixes
- Increase productivity

Presenter’s Company Logo – replace on master slide

Summary: Lessons Learned

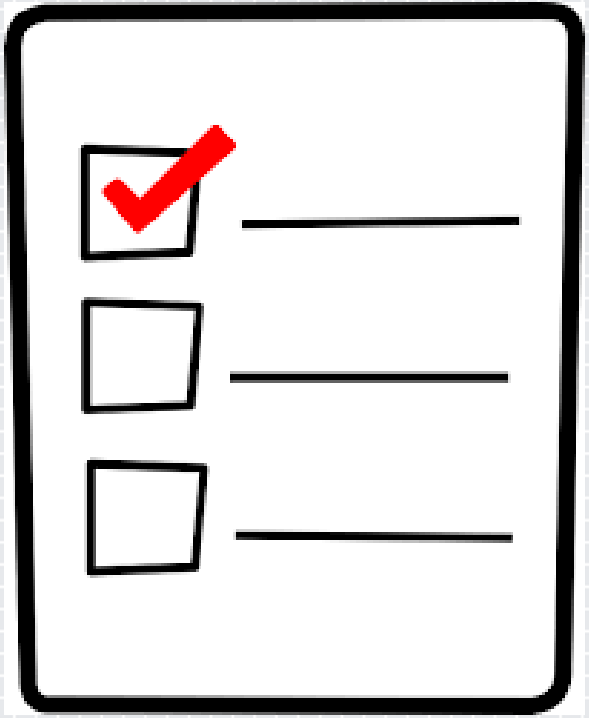
Recruit their smart guys

- Every team has a couple of key players
- Sit down with those people, and genuinely seek their help
- Let the others be swayed by their own leaders

Presenter's Company Logo – replace on master slide

Don’t tackle it all at once

- Triage!
- Focus on the highest risk first
- Work your way down the list until you’ve managed the risks appropriately



Presenter’s Company Logo – replace on master slide

Find the right technology partners

- Secure Code Training
- Static Application Analysis
- Dynamic Application Analysis
- Think flexibility and scalability
- Slaughter a few (figurative!) sacred cows

Presenter's Company Logo – replace on master slide

Key Lessons Learned

1. Recruit their smart guys
2. Don’t tackle it all at once
3. Find the right technical partners

Presenter’s Company Logo – replace on master slide

Apply It!

- **Week 1:**
  - Schedule times to talk with key stakeholders, determine importance of an AppSec program
- **Month 1:**
  - Assess the relative risks of your applications. Determine which should be handled first.
  - Begin vetting technical partners for:
    - Training
    - Static scanning
    - Dynamic scanning
    - Testing
    - Remediation work
- **Month 2+:**
  - Roll out the program to the highest risk applications
- **Month 3+:**
  - Track, report and repeat. Regularly scheduled check-ins with key stakeholders to measure progress

Presenter's Company Logo – replace on master slide

Questions / Contact me

Contact Robb:

robbreck@gmail.com

Follow Robb:

Inforeck.wordpress.com

Twitter: @robbreck

Presenter's Company Logo – replace on master slide