Engineering Excellence in Security on an Agile Smorgasbord

Ravi Krishnan M
Security Architect – Web Application Security
@mrsrivikris
Introduction

- This presentation is about:
  - Few key methodologies for adapting security to Agile/DevOps world.
  - Vendor neutral.
  - Plausible guideline.

- This presentation is **NOT** about:
  - A case study of secure development life cycle (SDLC) implementation in specific organization.
  - Sales pitch.
  - Silver bullet – Do you own research, YMMV. 😊

The views expressed in this presentation are my own, not those of my employer and/or any of its affiliates.
The “Agile Security Problem”
“Don’t go chasin’ waterfalls”. Dev would like to complete security review in a hour/day to meet modern agile needs. @msravikris #RSAC 2017 #APJ
YOU WENT FULL AGILE, MAN

NEVER GO FULL AGILE

image source: https://www.pinterest.com/pin/255297872603473563/
From Waterfall to Agile

- Scrum Master
- User story
- Epic
- Product Owner
- Continuous Integration/Continuous Deployment (CI/CD)
- Stand-ups

“Ideas don’t come fully formed. They only become clear as you work on them.” - Zuckerberg @msravikris #RSAC 2017 #APJ
Agile Manifesto

- Yes, it is Agile!
  - Scrum
  - Kanban
  - Extreme Programming (XP)

- It may be Agile!
  - Dynamic System Development Method (DSDM)
  - Disciplined Agile Delivery (DAD)
  - Lean software development

“It’s not enough to do your best; you must know what to do, and then do your best” - Deming @msravikris #RSAC 2017 #APJ

Useful Resource: http://agilemanifesto.org/
Most of the Agile methods are compatible so it becomes very easy to pick and chose at will, like Swedish Smorgasbord. @msravikris #RSAC 2017 #APJ

Image source: https://www.pinterest.com/pin/19914423324469728/
Sample Agile in action. It can vary from team to team in same org.

Source/Credits: Spotify Labs
Multiple coding languages.

Teams spread across time zones with different priorities.

Lack of complete documentation.

Changes occur multiple orders of magnitude faster than before.

Decentralized ownership of deployment with DevOps model.
New realities of InfoSec in Agile world

- The gap...
  - Show stoppers!
  - The Department of NO

- It widens more now...
  - Hey, I’ve already deployed this change. You can still do security review.
  - There’s only couple of days before deployment to security test this release/change.
AppSec Dilemma

Image source: 9GAG
Journey towards a DevSecOps world
On a brighter note

- Security issues can get fixed in hours/days rather than months/years!

Agile security is in the eyes of beholder, one needs to adapt few changes to look for goods. @msravikris #RSAC 2017 #APJ #DevSecOps

Image source: https://katkinnie.wordpress.com/2013/05/11/there-is-no-good-or-bad/
Traditional SDLC components and focus areas today

- **Security Requirements**
- Security architecture and design reviews
- **Threat Modeling**
- **Static Analysis**
- Dynamic Analysis
- Pen-testing
- **Security Tracking and Visibility**
- Developer Training

Note: Bold bullets are primary discussion points today. Other bullets are important too, but just not in scope for discussion today.
Key Change #1: Security Requirements to include business threat analysis

- Scrum teams can be prone to being silo and look at implementation-level threats, not necessarily business level threats.

- Enable product (business) owners to create security stories or features, along with user stories.
  - This can be a simple survey based tool hooked into release/sprint planning to suggest security indicators to product owners or based on interactions with security team.
  
  - *Ex: User story* - Implement feedback form for customers to provide feedback.
  
  - *Ex: Security story/feature* – To meet legal requirement, use TLS for feedback form to protect PII submitted in feedback forms.
  
  - *Ex: Misuse story* – As an attacker, I must not be able to read the data submitted in feedback form nor submit numerous feedbacks in short time. [Leave technical implementation downstream]

Make user stories and security stories as equal partners.

@msravikris #RSAC 2017 #APJ #DevSecOps
Key Change #2: Threat Modeling as planned technical threat analysis activity

- Identify high risk features and add threat analysis as a “sprint spike” task across related feature, in agile project management tool.

- This sprint spike will then be carried out during the sprint cycle, when the feature design pans out.

---

Identify and slot dedicated time for threat analysis by adding sprint spikes to a release.

@msravikris #RSAC 2017 #APJ #DevSecOps
Key Change #2: Threat Modeling as planned technical threat analysis activity

- It would be ideal to do this identification activity during:
  - Sprint/Release planning phase
  - Backlog grooming phase (for larger multi-sprint features)

- And perform technical threat analysis *(only on critical items!)* during the sprint.

- Few example indicators to identify critical items to perform threat analysis sprint spike include any backlog item that: process data from different network, authentication mechanism implementation or changes, external user input, new database.
Key Change #3: Static Analysis (Waterfall)

- Traditional static analysis is heavy weight:
  - Ran on very large codebase (every month/weeks), hence typically involves many hours or day to complete.
  - Large false positives: Typically requires deep search knowledge to triage and separate false positive.

- These would not be acceptable in *time constrained* agile development model.
Key Change #3: Do we have fast but insecure development?

- Lack of proper documentation has lead to Internet resources, like Stack Overflow being increasingly used by developers to implement quick solutions, including solutions for security problems.

Figure 4. Resource type's effect on use of secure solutions, by task and condition. The resource used had the opposite effect on security than functionality: participants restricted to Stack Overflow were least likely to achieve secure solutions.

Insecure #stackoverflow answers aren’t unusual – Yasemin et. al @msravikris #RSAC 2017 #APJ
Key Change #3: Static Analysis (Agile)

- Hook security source code scans into the CI/CD process and encourage daily scans.
- Focus on enabling velocity for security issues with increased training and code samples or official documentation (instead of stack overflow).
- Identify certain implementations that could kick-start conversations with security teams, like hashing, encryption, etc.
  - Old approach: Realize after multiple releases - “MD5 is vulnerable, use SHA256”
  - New approach: We’re implementing hashing function, can we chat on protection mechanism – is hashing the right approach or encryption?
- Implement notifications mechanisms to detect code changes to critical features of product, like hashing, encryption, etc.
TrustMe
This is science

Key Change #4: Security Tracking and Visibility

- It will be fixed only if it is measured and tracked.
- Short release cycles would mean less time to report and share security issues across infosec and product teams.
- Team spread across different geo location globally could result in communication and tracking issues.
Creating a highly visible dashboard is critical to track the findings:

- True to the agile models, the dashboard should provide rapid feedback and reduce variability.
- Also, it must be decentralized for use by security and development teams to keep pace with agile speeds.

This would also help to avoid “backlog pit of despair” and prioritize security findings for fix in the sprints.
Key Change #4: Security Tracking and Visibility

Credits: Jira Risk Workflow by Dinis Cruz
Summary: Lessons Learned
## In a nutshell

<table>
<thead>
<tr>
<th>Waterfall</th>
<th>Agile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security as a non-functional requirement, maintained in ~100+ PDF docs.</td>
<td>Security as stories, continuous and part of functional requirement (user stories).</td>
</tr>
<tr>
<td>Threat model everything in design phase.</td>
<td>Threat model critical items as a sprint spike in each release.</td>
</tr>
<tr>
<td>Static code scan during test phase.</td>
<td>Static code scans throughout sprints and during code check-in to CI/CD.</td>
</tr>
<tr>
<td>Track findings in a centralized repo.</td>
<td>Track findings in decentralized repo with clear workflow to reduce variability.</td>
</tr>
</tbody>
</table>
Finally, one key thing!

To adapt security to modern application development and align with DevSecOps manifesto, it is important to shift:

- **From:** Focusing on gatekeeping controls at each development stage to find and eliminate vulnerabilities.
- **To:** Everyone is responsible for security mindset, with distributed, early, and continuous engagement of AppSec experts, process and tools to achieve higher security standards in deployments.
Questions?

rmuthuk@ncsu.edu
@msravikris