VERACODE

DevSecOps @ Veracode: Security Champions

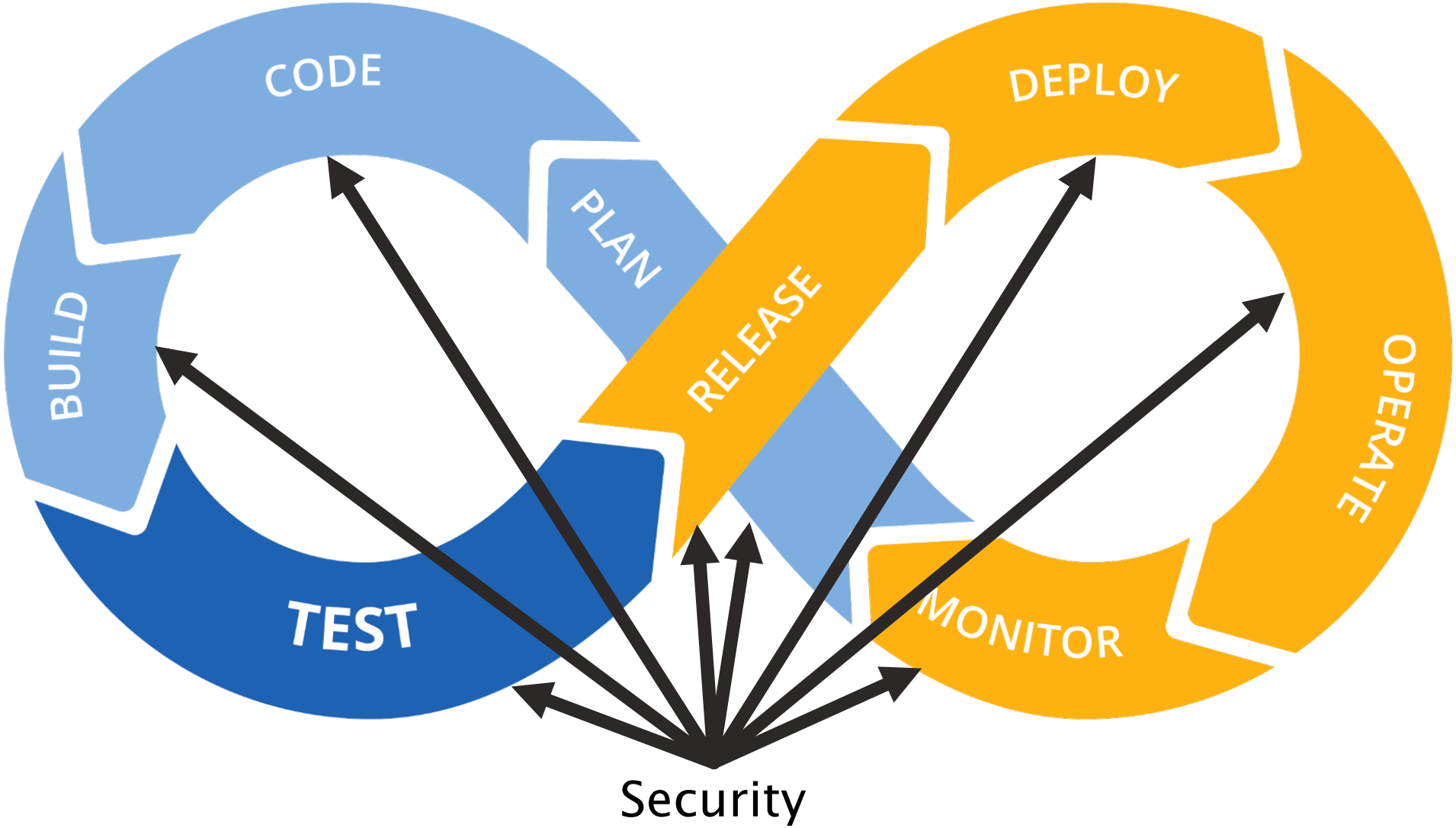
Chris Eng
Vice President, Research

Chris Eng, VP of Research

- 15+ years focused solely on application security, both offense and defense
- Leads a team responsible for integrating security expertise into all of Veracode's products; also product security and SDLC
- Frequent conference speaker and media spokesperson on a range of security-related topics
- Hates the term "thought leadership" (see http://tiny.cc/thoughtleader)

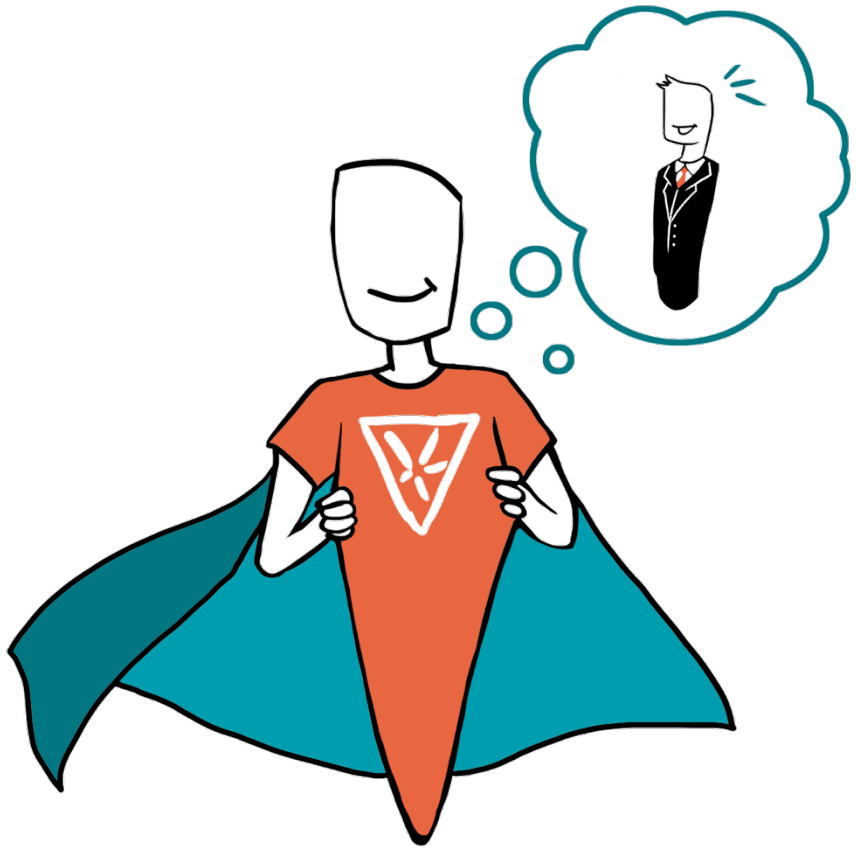
@chriseng

DevOps – process: where is security?



Security champions

- Your security team does not scale indefinitely!
- Build and train a team to take on specific tasks and to be the security “conscience” on their respective teams



Pick the right people

Start strong

Empower, within limits

How to pick the right people

- 1-2 members from every product team
- Volunteers are best
- Influencers
  - Respected or influential team members
  - Doesn’t have to be a developer

How *not* to pick the right people

- New employee
- New to team or product
- Already responsible for an existing Scrum role
  - Product Owner
  - Scrum Master
  - etc.

Start strong

- Start with formal training in security fundamentals
- Reinforce with eLearning
- Use CTFs and other opportunities to learn in the wild

Empower, within limits

- Security grooming within guidelines
- Security review guidelines
- Know when, and how, to escalate

Empower – security grooming

- New feature introductions
  - New UI elements
  - New API endpoints
- New architectures
- New security controls
- New forms or actions
- Fix for pen test finding

Empower – security grooming

- AuthN, AuthZ
- Crypto
- Data validation
- Encoding
- Error handling
- Session management
- Cache management

Empower – code reviews

Limited topics based on security controls they have proven they understand:

- Data validation
- Encoding
- Parameterization
- Logging
- Error Handling

The conscience of security

Keeping momentum

Measuring and managing

CAUTION

You can’t improve what you don’t measure.

And you shouldn’t measure what you don’t manage.

Measuring and managing

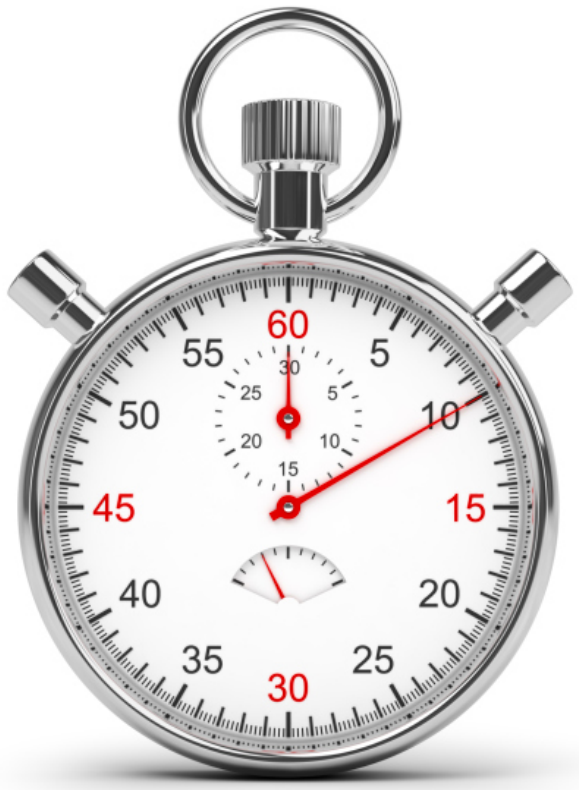
- Baseline security maturity
- Code review certifications
- Individual and team goals

Sample AppSec maturity model
(you don't have to read the text)

	Base	Beginner	Intermediate	Advanced	Expert
Training	• No formal security training	• Some team members (≥10%) have taken a basic secure development eLearning course	• **All** team members have taken a basic secure development eLearning course • Security Champions and Team Leads have taken additional advanced or domain specific training	• All team members take a secure development eLearning course **annually** • Security Champions and Team Leads have taken additional advanced or domain specific training • Security Champions and Team Leads conduct informal learning sessions for other team members	• All team members take a secure development eLearning course annually • Security Champions and Team Leads have taken additional advanced or domain specific training • Security Champions and Team Leads routinely conduct **formal and informal** learning sessions for other team members
Secure Design	• Security is not a design consideration	• Security requirements are generally defined after development has started or completed	• Threat modeling before major components or features • Security requirements are defined before major components or features	• Threat modeling before **all** components or features • Security requirements are defined before **all** components or features • Threat modeling is incorporated into the story planning/grooming process • Security requirements are defined in Story Acceptance Criteria of most (≥50%) relevant stories	• Threat modeling before all components or features • Security requirements are defined before all components or features • Threat modeling is incorporated into the story planning/grooming process • Security Acceptance Criteria defined for **all** relevant stories
Security Code Review	• No security specific code review	• Major components are reviewed by Security Team or 3rd party • Only the most critical findings are addressed	• Security team review of high risk stories • High and critical findings are addressed	• Some peer Security Review within teams • Security team review of high risk stories • Automated code checks • Most findings (critical, high, medium) are addressed within 30 days	• Peer Security Review of all pull requests • Security team review of high risk stories • Custom automated code checks • Holistic review of product by Security Team or 3rd party periodically • All findings are addressed rapidly (≤7 days)
Security Testing	• No security testing	• Annual 3rd party Pen. Test (where required by policy) • Only the most critical findings are addressed	• Annual 3rd party Pen. Test (where required by policy) • Ad hoc SAST and/or DAST • High and critical findings are addressed	• Annual 3rd party Pen. Test (where required by policy) • SAST and DAST on regular basis (e.g., per-release, monthly) • Test plans include security requirements • Most findings (critical, high, medium) are addressed within 30 days	• Annual 3rd party Pen. Test • Continuous SAST and DAST integrated into build and bug tracking systems • Security testing integrated into unit and feature tests • All findings are addressed rapidly (≤7 days)
Third Party	• Security is not a consideration when managing third party assets	• List of third party assets and versioning information is documented	• List of third party assets and versioning information is documented **using a repeatable scripted process** • Security track record is taken into account when choosing third party assets	• List of third party assets and versioning information is documented using a repeatable scripted process • Third party assets are chosen based on proven security track record • Team has setup alerts when new security events that effect the product become available and have a process defined for applying relevant patches or configuration changes	• List of third party assets and versioning information is documented **with no manual effort** • Third party assets are chosen based on proven security track record • Team has setup alerts when new security events that effect the product become available and have a process defined for applying relevant patches or configuration changes

Measuring and managing

- Goals for champions
  - Code review certification
  - Spot check grooming decisions
- Goals for teams
  - Against maturity model
  - Baseline and update
- Are you getting what you expect?

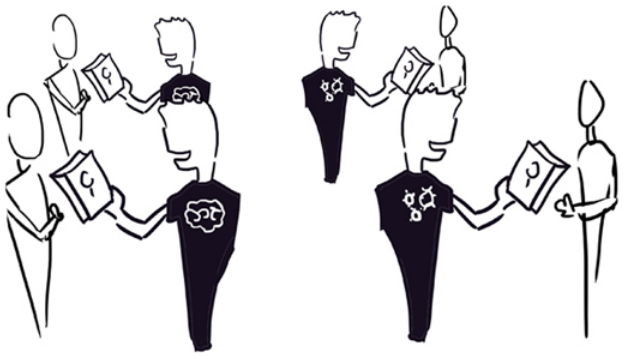


Learn about their world

- Read
  - The Phoenix Project
  - The DevOps Handbook
- Attend some scrum ceremonies
- Learn their tools
- Write security stories and/or code

Maintain high touch

- Support not abandonment
- Monthly group meetings to compare experiences and share information
- Slack channel, mailing list: however the developers prefer to communicate
- Periodic check-ins, e.g. quarterly PSMM check-ins
- Joint projects (e.g. VSSL)



Rewards and recognition

- Additional training opportunities
  - Internal (mentoring)
  - External (conferences)
- Teach them to hack
  - Internal CTF sessions
- Give them swag, badges, certifications

Conclusions

- Have empathy
- Overcommunicate
- Remember motivations
- Stay engaged and responsive
- Iterate

VERACODE

Thank You!