
TOP REASONS TO TURN YOUR TEAM OF
DEVELOPERS INTO SECURITY CHAMPIONS

Limited Amount of Security Engineers in the Industry

Rise of sophisticated cyber security attacks on:

- Businesses
- End Users
- Software Applications

Costly problem for hiring managers and team leads

- High demand for jobs, can equal increased expenses to hire quality software engineers
- Lengthy search process
- Lengthy on-boarding process / time to adapt to company culture

"For every 245 software engineers, only one individual was a security expert"
- BSIMM 2016 Survey

How can we address this issue?

By turning your current team of software engineers into Security Champions

Equip your team with the know-how to ensure code is secure *and* functional. This can:

- Mitigate threats
- Decreases risks
- Stop data breaches from occurring
- Prevents costly law suits

Methods to turn engineers into security champions

1. Implement the use of Secure Software Development Life Cycle methodologies.
2. Utilizing secure coding best practices, and using engineering models such as DevOps (i.e. integrating security in all stages of building and deployment).
3. Presenting live demos or attending webinars.
4. Using a mentor program to engage your team in development and security assessments.
5. Researching the security features of different platforms and frameworks.
6. Attending security conferences, meetup groups, and online training.


WHAT MAKES A SECURITY CHAMPION?

Software Engineer Turned Security Champion

The modern software engineer is trained to quickly develop efficient, functional, feature-rich applications to meet the goals of corporate organizations and end-users.

Skills of a software engineer should include:

- Knowledge of multiple programming languages (e.g. Java, C#, JavaScript, Python),
- Development models (e.g. Agile, Waterfall, DevOps), frameworks (e.g. Node.js, Spring, MVC)
- Methodologies

Skills of a security champion should include those skills, plus:

- Security best practices
- Can quickly identify insecure coding flaws or bugs in the source code
- Can remediate security issues using secure coding techniques

A Security Champion Keeps Code Security at Front of Mind

During the design and implementation phase, it's imperative for your software engineer to start thinking like a security champion. They can do this by "shifting security left".

During all phases of the development life cycle, a security champion will have security best practices at the forefront.

This includes collaborating with the project team to ensure they understand what security framework features, access control, validation, and encoding libraries are required.

Taking this a step further, security champions might be responsible for writing the high-risk sections of code (e.g. password management or cryptography libraries)

A Security Champion Recognizes Issues When They See Them

Common vulnerabilities a security champion will identify:

- Command Injection
- Cross-site scripting
- Weak access control logic
- Catch security bugs early in the life cycle before reaching the test environment

When caught and fixed early, you can save your company time and money.

More time to implement revenue generating projects.


3 POWERFUL RESOURCES TO TURN A DEVELOPER
TO A SECURITY CHAMPION

Security Training

Training in secure coding, cryptography, web, mobile, cloud and SDLC security is a necessity. Divide training in three levels:

1. **Awareness** - The first level requires the trainee to learn about the OWASP Top 10, secure coding methodologies, and basic threat awareness.
2. **Skilled** - The second level requires the trainee to learn about the Secure Software Development Life Cycle, relevant development framework security features, and modern security in web, mobile, and cloud platforms.
3. **Champion** - The third level requires the trainee to learn how to run scanning tools, verify scan results, perform ethical hacking/penetration testing on applications, and perform security-focused code reviews.

Willingness to Learn

Adapting to rapid changes and learning new technologies are critical for a security champion to defend against new vulnerabilities

Willingness to Teach

Mentoring and passing knowledge along to colleagues and team members is critical to raising security awareness.

Five-Step Approach to Level Up a Software Engineer

Mentor / Mentee Program

- Assign a mentor to the trainee
- Weekly meeting to test new vulnerabilities

Trainee Education

- Learn secure testing methods
- Research security aspects of different frameworks and platforms

Learn about secure coding

- In specific languages
- Web App Security
- Mobile App Security
- Cloud and Modern Framework

Incorporate

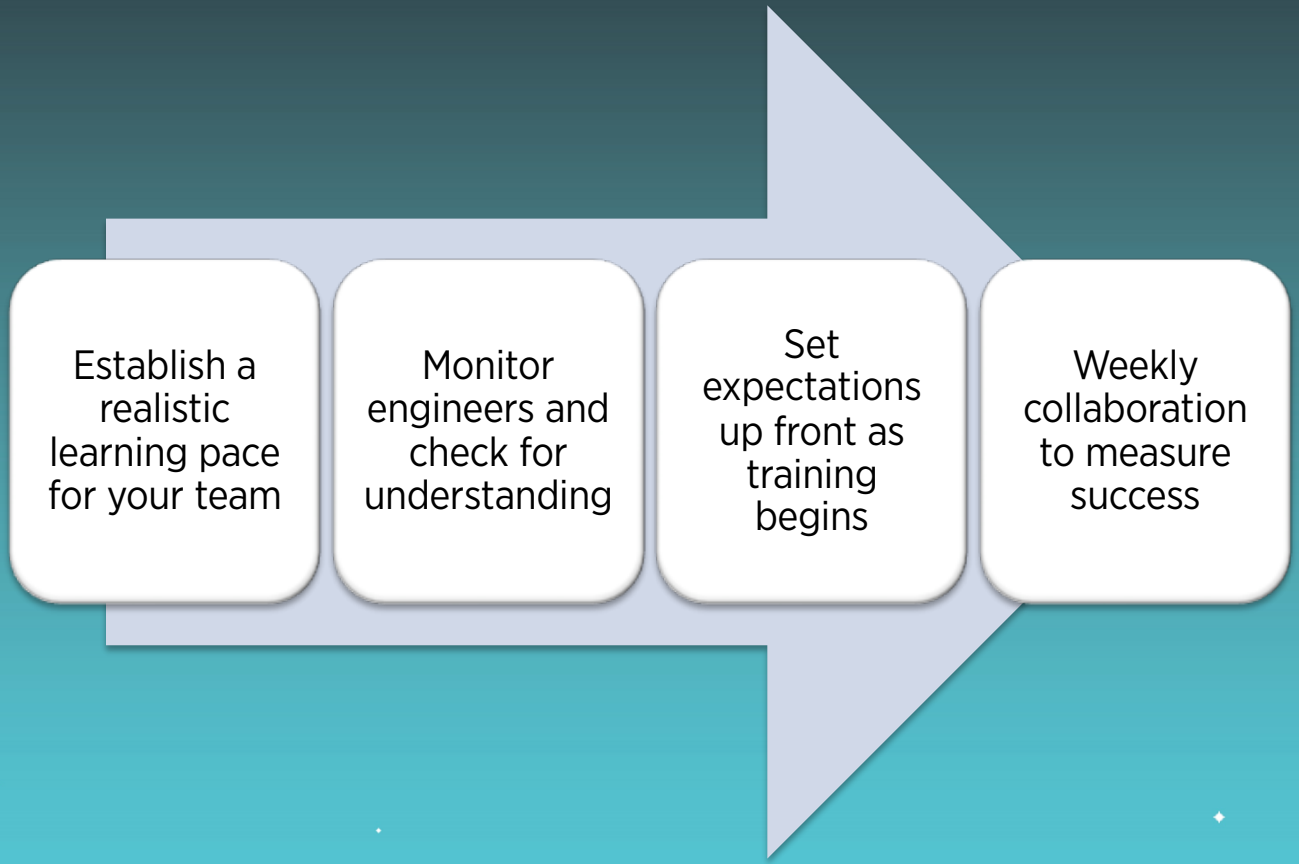
- Put training into the development life cycle
- Create security requirements, abuse cases and other functional test cases to enforce security throughout the dev pipeline

AppSec Security Scanning

- Static analysis
- Secure code review
- Dynamic analysis of run-time applications
- Scanning third-part libraries for known vulnerabilities

TIMELINE FOR TRAINING AND DEVELOPMENT

Security Training



When is my software engineer ready to take on or lead security projects?

- Indications you are moving in the right direction:
- Writing secure code, identifying security issues during their security-focused peer reviews, and mentoring security engineers in training.
- Ensuring code does not contain issues identified in the OWASP Top 10 is a place to start, for more in-depth analysis refer to ASVS (Application Security Verification Standards)
- No high-risk findings are found when scanning code with static and dynamic tools.
- Results from in-depth security reviews by the security team or external security consultants no longer turn up high risk findings.

Gradual Process

- The goal is not to turn security champions into full time security testers!
- Change comes when best practices are put into place throughout the software development life cycle methodology and how application development is approached.
- Strongly encourage your firm to invest in the right scanning tool, to help guide teams as they navigate through various security risks.

Security champions must continue to work as part of the project team and be an advocate for secure product development going forward. Implementing security features into the source code, building unit and function security tests, and performing continuous security reviews, acts as a powerful pre-emptive action to mitigate security issues before they start.

Substantial Impact on Software Security

1. Building applications with security top-of-mind creates **a secure framework, secure architectural design, and decreased attack surfaces**, all of which help to reduce the risk of running high-risk applications in production.

2. Having security champions as software engineers **enables security coverage throughout the development life cycle.**

3. Vulnerabilities that are considered to be “low-hanging fruit” are **easily caught and mitigated before their release to production**, which prevents hackers from finding issues that are trivial to exploits.

About Cypress Data Defense:

Our goal is to help organizations secure their IT development and operations using a pragmatic, risk-based approach. The diverse background of our founders allows us to apply security controls to governance, networks, and applications across the enterprise.

Contact us to learn more!

https://www.cypressdatadefense.com/contact/

Email: info@cypressdatadefense.com

Phone Number: 720.588.8133

REFERENCES:

"Top Reasons to Turn Your Team of Developers Into Security Champions"

https://www.cypressdatadefense.com/technical/upskilling-software-engineer-security-champion/

"2015 Trustwave Global Security Report"
Retrieved from: https://www2.trustwave.com/2015-Global-Security-Report-Landing-Page_2015-Global-Security-Report-Success-Page.html?aliId=79544114