Seven Steps to an Ideal Secure Coding Training Program

Investing in secure coding training is not a one-time action but a commitment from organizations to prioritize long-term software security and resilience.

Based on customer feedback, our experts recommend that training programs should be multi-year commitments to achieve tangible results.

Our Ideal Secure Coding Training Program Guide outlines how to establish a multi-year program that systematically embeds secure coding practices into the core of your organizational culture.

1 Planning Your Program

Focusing on one or two program goals can help you measure key performance indicators and better determine the success of your program.

Your program goals are likely to include:

- Meeting Regulatory Compliance - such as PCI-DSS, OWASP, White House Executive Order, or others.
- Creating a Proactive Program - supporting a cultural shift towards security-mindedness.
- Recovering from an Incident or Vulnerability - as part of a response and to improve security moving forward.

2 Pulling Baseline Data

Collect information on vulnerabilities, types of tickets, time taken for remediation, etc., from tools such as SAST/DAST and internal processes like code reviews and logs.

By keeping track of these metrics, you can better understand what is effective and what requires improvement and even demonstrate the return on investment of your training initiatives.

3 Prioritizing Internal Communications

To gain internal support for a new program, focus on a strong rollout that engages both leadership and learners. Start with a detailed executive presentation outlining the program's goals, time requirements, and impact. Then, share a summary with the entire organization and conduct a live discussion to introduce the program directly to learners. Fun kickoff events and consistent communication will engage learners throughout the training program.

4 Building Your Program

A comprehensive secure coding training program should include multiple learning modalities to help increase learner knowledge retention. For example, video content that covers theories and security concepts paired with hands-on practice for developers.

These modalities can be used to educate a wide range of topics that your learners can benefit from, from basic security principles, compliance requirements, to specific languages and frameworks that your developers work in. Consider partnering with a secure coding training vendor that offers a wide range of topics for everyone within your SDLC.

Having a wide range of topics will help support your ability to run a multi-year training program with progressive learning structures (from foundational to advanced). Start with your industry's compliance requirements, then move onto role-based training with an emphasis on your organization's most relevant threats and vulnerabilities.

5 Incorporating Tournaments

Secure coding training tournaments provide a gamified approach to application security training for developers. In tournaments, developers compete to solve challenges involving identifying code vulnerabilities or writing secure code.

Program managers should consider running tournaments regularly, at least every 6 months, to boost learner engagement and to highlight milestones such as launching a new program or reinforcing skills during Cybersecurity Awareness Month.

6 Building Security Champions

To identify potential Security Champions, look for individuals who excel in training, show interest in security, and actively participate in communications.

Security Champions can advocate for security practices and help break down silos between teams. To keep them engaged, consider offering Security Champions advanced training opportunities, specialty curated Champion Learning Paths, and the freedom to tailor their learning experience.

7 Measuring Results

Accurately measuring the success of your program is crucial to its long-term success. Program managers should track learner progress, completion rates, knowledge gain, weekly training streaks, and points collected from activities.

To evaluate the program's overall impact every 6 months, compare this data to your initial baseline measurements. This analysis will highlight trends, inform necessary adjustments, and demonstrate the program's value to stakeholders.

Conclusion

Investing in secure coding training is essential for safeguarding your applications and data. By utilizing the right platforms, resources, and ongoing assessment, you build a strong foundation for software security within your organization.

We have developed an in-depth seven-step guide to building the ideal secure coding training program using Security Journey's AppSec Education Platform. To get the full guide or learn more about Security Journey's training, visit our website.

SECURITY JOURNEY