European Scholar Journal (ESJ)
Available Online at: https://www.scholarzest.com
Vol. 3 No.10, October 2022
ISSN: 2660-5562

FUNDAMENTAL PRINCIPLES OF SOFTWARE SECURITY

Erejepbaev Bekzat
Nukus Branch of Tashkent University of Information Technologies
3rd year student

Article history:

Received: 20th August 2022
Accepted: 20th September 2022
Published: 30th October 2022

Abstract:

This article provides a brief overview of several fundamental principles of software security.

Keywords: software security, security standards, security policies, metrics, haphazard security.

3 FUNDAMENTALS OF A SOFTWARE SECURITY INITIATIVE

The best software security initiative is tuned to fit your organization and built to scale. Three SSI fundamentals are standards, policies, and metrics.

You take calculated risks every day. Just this morning, say, you might have decided to cross an empty street against the light because you were late for work. But if you had been with your child, you would have made a different decision.

We rely on our experiences, and those of people we trust, to set the bar for the risks we take on. Some risks are acceptable and some aren't. Software security is no different. As a security practitioner, you're in charge of assessing security risks that have an impact on your customers' trust and your business's reputation. So you can't make arbitrary decisions.

A key finding from the annual BSIMM study is that many firms focus on high-risk applications, thinking this is enough to mitigate their risk of attack. But medium- and low-risk apps are also part of the attack surface. How do you decide which applications to secure and how to secure them?

You could do nothing and just hope your software and systems are secure. You could waste resources performing haphazard security testing on random applications. Or you could create a software security initiative (SSI), a program that helps you balance available resources against unacceptable risks.

THE CRITICAL NEED FOR A SOFTWARE SECURITY INITIATIVE

Compliance and regulatory requirements are increasing, and high-profile breaches are raising awareness of software security. In response, organizations are investing in approaches to reduce risk, such as application security testing regimes. But these approaches vary widely. Some organizations perform penetration testing on a handful of apps once a year to meet minimal compliance requirements. Others have installed wireless application protocols (WAP) and monitoring programs to bolster firewall protection.

These approaches are valid. But they are not strategic.

"Proactive security saves time and money, but it is not going to be enough. A security program is what you need to put in place to lower your exposure across the board," says Tyler Shields, senior analyst at Forrester Research, Inc.

A comprehensive software security initiative has multiple benefits. It will help you:

- Get your team up to speed on security requirements more quickly
- Improve communication and alignment across teams
- Provide consistency for everyone in your software supply chain

- Measure and communicate success
- Make sure you address unacceptable risk

A SOFTWARE SECURITY INITIATIVE IS A BLUEPRINT

The most effective software security initiative is tuned to fit your organization and built to scale. It helps you "show your work" by creating a methodology for lowering your risk and explaining how you have made investment decisions.

The best way to develop a software security initiative is a three-pronged approach that includes security standards, security policies, and security metrics. Standards and policies drive cooperation toward a defined and shared goal. And metrics are crucial to ensure that you are achieving success with your security program.

1ST FUNDAMENTAL: SECURITY STANDARDS

Security standards provide developers and application testers with guidance on what your company will accept and what it won't. They are essential to maintaining consistency across your supply chain.

When security standards are documented and widely communicated, developers understand rules for the type of code they may use (e.g., COTS, open source, libraries) and the security requirements they must incorporate in their programs (e.g., specific crypto algorithms they must use or coding practices they must avoid).

2ND FUNDAMENTAL: SECURITY POLICIES

Security policies ensure that everyone involved shares a common definition of terms, understands roles and responsibilities, and has a set of operating procedures and governance rules to follow. Creating security policies paves the way for your team to follow the standards defined by your software security initiative.

Security policies typically cover:

- Application testing. Define risk classifications. Determine which applications must be tested and which gates they must pass.
- Remediation. Set expectations for fixing bugs and flaws.
- Network security. Determine protocols and authorization levels.
- Data security. Protect your valuable IP and sensitive customer data.
- Physical security. Govern access control and secure your infrastructure.
- Disaster recovery. Determine steps to take in the event of an attack, including reporting, recording, and resolution.

3RD FUNDAMENTAL: SECURITY METRICS

To demonstrate the results of your software security initiative and track your progress over time, you must establish a defined set of metrics.

Some examples of strategic and operational metrics:

- Amount and cost of resources to ensure application security
- Number of critical applications that must undergo in-depth testing
- Number of tested applications
- Time to run test per application
- Time from design to launch to create a secure application
- Number of applications that meet or exceed compliance requirements
- Number of bugs in code that reach production

RIGHT-SIZE YOUR SECURITY STRATEGY TO MATCH YOUR BUSINESS

Your software security initiative must be customized to match your organization, portfolio, environment, and culture. It must be scalable so it can grow with you. And of course you want the program to be cost-effective and address your business's specific level of unacceptable risk.

Whether you have a stand-alone software security group with security oversight or you embed responsibility for security within each engineering group or business unit, you can adapt this three-pronged model to create or evolve your software security initiative, and lower your risk of a security breach.

CONCLUSION

You take calculated risks every day. Just this morning, say, you might have decided to cross an empty street against the light because you were late for work. But if you had been with your child, you would have made a different decision.

We rely on our experiences, and those of people we trust, to set the bar for the risks we take on. Some risks are acceptable and some aren't. Software security is no different. As a security practitioner, you're in charge of assessing security risks that have an impact on your customers' trust and your business's reputation. So you can't make arbitrary decisions.

REFERENCES

1. Mathias Payer. *Software Security: Principles, Policies, and Protection*. July 2021 (version 0.37), updated regularly at this link.

2. Mark Dowd, John McDonald, Justin Schuh, *The Art of Software Security Assessment: Identifying and Preventing Software Vulnerabilities* (2007, Addison-Wesley).
3. Viega and McGraw, *Building Secure Software* (2001, Addison-Wesley).
4. Howard and LeBlanc, *Writing Secure Code, second edition* (2002, Microsoft Press). Web security, mobile code security, malicious code:
5. Michal Zalewski, *The Tangled Web: A Guide to Securing Modern Web Applications* (2011, No Starch Press).
6. OWASP project online resources.
7. McGraw and Felton, *Securing Java: Getting Down to Business with Mobile Code* (1999, Wiley). First edition (1997): *Java Security*, open online web edition.
8. Lincoln Stein, *Web Security: A Step-By-Step Reference Guide* (1998, Addison-Wesley).
9. Rubin, Geer and Ranum, *Web Security Sourcebook: A Complete Guide to Web Security Threats and Solutions* (1997, Wiley).
10. Avi Rubin, *White-Hat Security Arsenal* (2001, Addison-Wesley).