devmio blog

Improve your security know-how

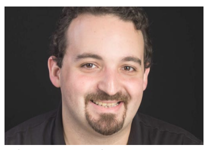
4 Common Software Security Development Issues & How to Fix Them

Security

Tim Jarrett

02. Jul 2021

As software has become the backbone of modern business, cyberattacks have become an ever-present threat, making application security a critical necessity to ensure business continuity. This article examines four commonly found software security development issues and how to address them.



The process of managing and maintaining secure software can pose unexpected roadblocks to developers who seek to deliver features as quickly as possible. Research shows 59% of companies now deploy code multiple times a day, once a day, or once every few days. However, as software has become the backbone of modern business, cyberattacks have become an ever-present threat, making application security a critical necessity to ensure business continuity.

The shift left movement—conducting security testing and fixing flaws earlier in the development process—has increased the need for developers to play a role in application security, but there remains a large skills gap in security trained developers. Developers interested in improving their security know-how can start by understanding some of the common DevSec issues.

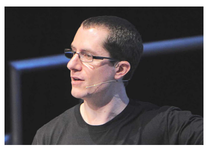
SEE ALSO: Evaluating application security in the age of cloud-native

card balance, are now much more expensive to address than when they were introduced. To avoid adding to security debt, developers can implement automated scanning and testing.

The more automation, the better: In our annual State of Software Security (SoSS) report, we found organizations that pair Dynamic Analysis (DAST) with Static Analysis (SAST) fix 50 percent of their security flaws 24.5 days faster on average.

Another way to find and fix new flaws faster is to scan more often. More frequent scanning enables organizations to reach that halfway point 22.5 days faster and running SAST scans through API decreases the time to remediate 50 percent of flaws by 17.5 days.

Research also shows a steady scanning cadence can help your team see meaningful change in the proportion of flaw types and reduce security debt over time. Think of security testing as a marathon, rather than a sprint: you don't prepare for a marathon by only running 50 miles in the week before the event.



Challenge #2: Introduction of common code security flaws

Understanding which flaws pose the greatest risk to your applications and how they're introduced is key to preventing damaging cyberattacks that these common flaws enable. Our SoSS report found information leakage (65.9 percent), CRLF injection (65.4 percent), cryptographic issues (63.7 percent), and code quality (60.4 percent) are the most common flaws found in applications. To address these common flaws, developers should consider the following:

- For information leakage, lean on secure coding best practices and implement security testing procedures as you write code.
- To prevent CRLF injection, don't trust user input, sanitize user-supplied data with proper validation and encoding, and be sure to correctly encode output in HTTP headers.
- Cryptographic vulnerabilities can be prevented with good secure coding practices. Also, most major languages inherently support good cryptographic practices, and concerns over incorrect implementation typically only arises on a case-by-case basis.
- Prevent poor code quality issues by utilizing consistent coding patterns, automating security testing in your SDLC, and keeping up to date through effective training.

It's worth noting that these same four flaws consistently place in the top 10 of the report year after year, indicating a gap in awareness and training among developers. In fact, security training for developers is potentially the biggest challenge of all. Not only is secure coding not taught regularly in university, on-the-job training is equally hard to get since most application security falls to the security team. To enable devel-

Challenge #3: Reliance on open source libraries, but only scanning application code written in-house

Open source code is in use nearly everywhere. And when you consider that many open source libraries are not chosen directly by developers — 46.6 percent of insecure open source libraries in applications are transitive, brought into the application by another library in use – it's easy to understand how open source code expands the attack surface within applications. In fact, our research found 71% of applications have a flaw in an open source library on initial scan.

Integrating a scanning tool like Software Composition Analysis (SCA) can help detect open source vulnerabilities with greater accuracy. And since 74 percent of open source flaws can be fixed with a patch, revision, or major/minor version update, this process enables efficient mitigation.

Making use of the right tools to stay on top of code is key to reducing risk and ensuring you can use open source libraries with confidence.

SEE ALSO: SolarWinds hack and security – What is a software bill of materials?

Challenge #4: Surplus of high and very high severity flaws in code

No matter which software language you prefer, understanding the flaws that impact them most will help you prevent errors before they become bigger problems. Our data shows that some languages carry more high-risk flaws than others, which means code written in specific languages should be crafted and tested thoughtfully. Some examples include:

- **C++ applications:** Nearly 60 percent of applications have high and very high severity flaws; common flaws include error handling, buffer management errors, numeric errors, and directory traversal flaws.
- **PHP applications:** 52.6 percent of PHP applications have high and very high severity flaws; the flaws found most frequently include cross-site scripting (XSS), cryptographic issues, directory traversal bugs, and information leakage vulnerabilities.
- **Java applications:** Java leads with CRLF injection flaws, code quality issues, information leakage, and cryptographic issues; Java applications are 97 percent third-party code and carry greater unseen risk.

By examining flaw frequency trends in various common languages, developers have a better understanding of the everyday risks they face while coding and can use that knowledge to get ahead of those flaws before they become a problem.

Implementing secure coding practices and utilizing hands-on training to increase know-how will help ensure the security of applications can keep up with modern development needs. When developers are empowered to not just find, but also fix flaws in their code, they'll be well on their way to becoming more security-savvy developers.

Tim Jarrett

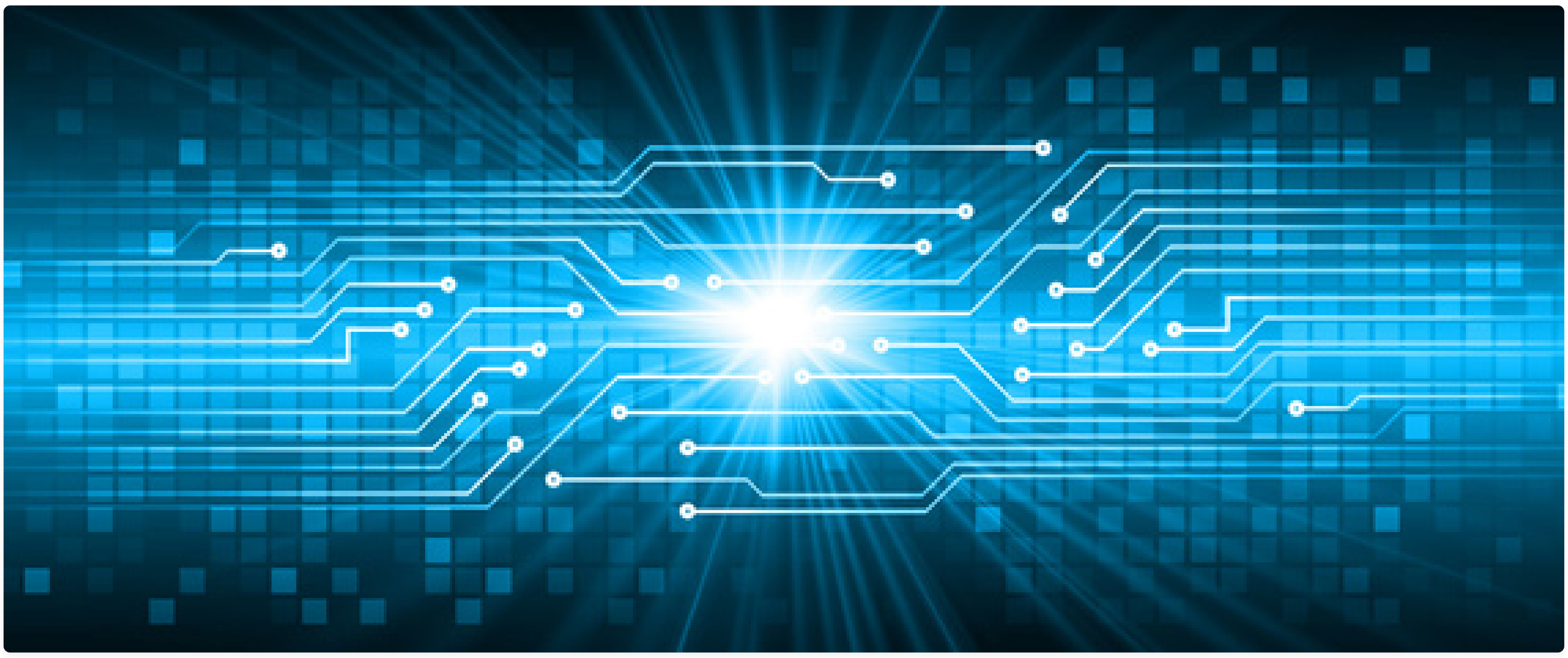
Tim Jarrett is Senior Director of Product Management at Veracode, where he is responsible for the strategy and roadmap for Veracode's integrated cloud platform and DevOps integrations. Prior to this, he spent time at iET Solutions and Microsoft as a Product Manager. He holds an MBA in New Product and Venture Development from MIT Sloan School of Management and a BS in Physics from University of Virginia.

More articles on this topic

Benefits and introduction

Beyond Shift Left, It's Time to Add Shift Left Security

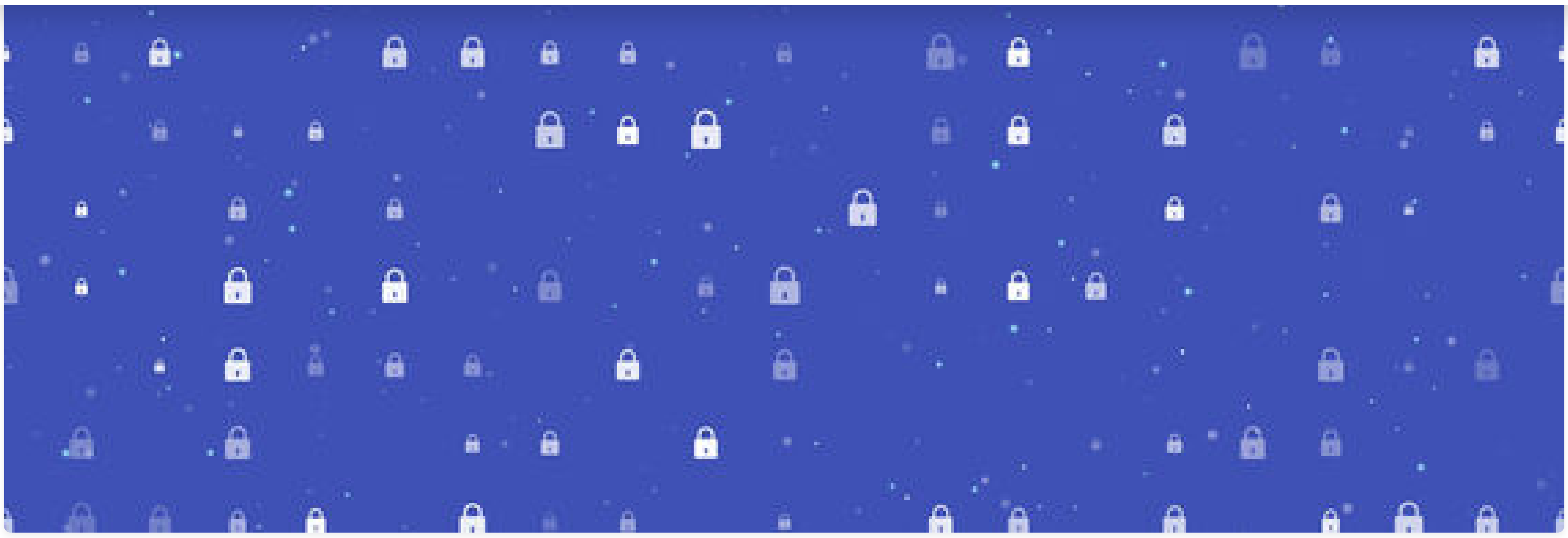
Steve Howard



Interview with Brian Fox, CTO and co-founder of Sonatype

"96% of open-source Java downloads with known-vulnerabilities could have been avoided"

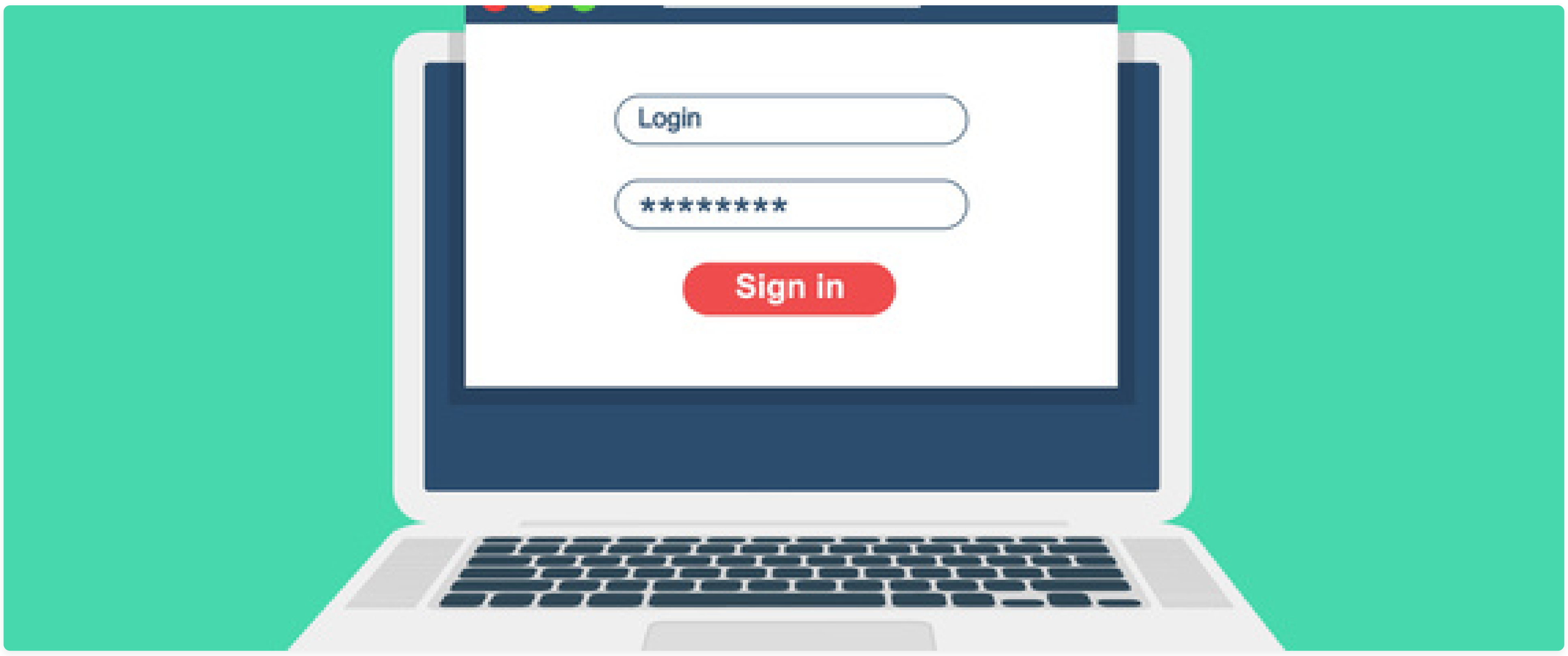
Brian Fox



Keep your system fully protected

Robotic Process Automation: Establishing Best Practices

Irina Lunin



Best practices for optimum security

Putting an End to Weak Passwords This World Password Day

devmio Editorial Team

devmio Basic Access

- ✓ Thousands of articles, series, ebooks and columns
- ✓ Intelligent AI search engine AskFrank
- ✓ Read wherever you want - on desktop, mobile or in the app
- ✓ Cancellable on a monthly basis

REGISTER

Social

Follow us on social media for more news, content and background stories from our authors, editors and events. Share your personal experience with us.

Learn more

App

Access your digital knowledge base everywhere with our mobile apps.

Download on the App Store

GET IT ON Google Play

Expand your knowledge

As a subscriber to our newsletter, you will be the first to know about new content.

Email * SUBMIT

Start your trial month now

Not convinced yet? Check out our training portfolio for free!

TEST NOW