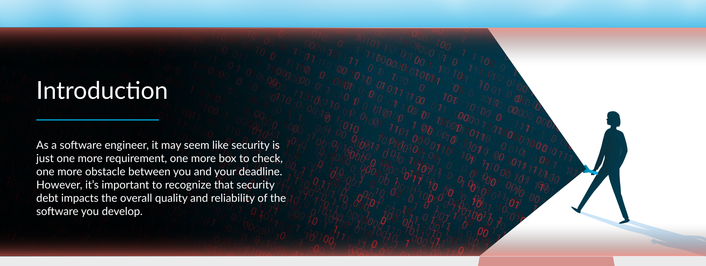
Developers: Why Security Debt is Your Responsibility

& What You Can Do About It Based on New Data

Introduction

As a software engineer, it may seem like security is just one more requirement, one more box to check, one more obstacle between you and your deadline. However, it's important to recognize that security debt impacts the overall quality and reliability of the software you develop.

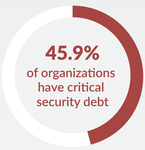


Step-by-Step Data Exploration

Here's the data behind what you can do to release high-quality code and advance your career (based on the analysis of over one million applications).

1 Organizations are drowning in security debt

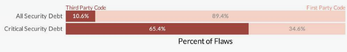
Over 70% of organizations have security debt and nearly half have critical debt (high-severity flaws). These flaws pose a significant risk to businesses, as we define severity as the potential impact on confidentiality, integrity, and availability.



2 Security debt exists in both first-party and third-party code

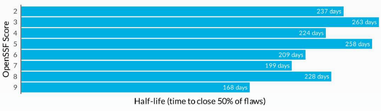
While first-party code makes up the vast majority of overall security debt, most critical security debt comes from third-party code in open-source software. As a developer, it's crucial to consider testing and remediation efforts for both first-party and third-party code continuously throughout development.

Pro tip: use scanners that work in the IDE where you work already.



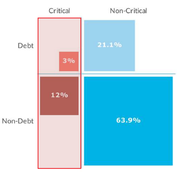
3 Open Source: don't import it and ignore it

You may know the saying "set it and forget it," but when it comes to software, what we see happen with most libraries is "import it and ignore it." The primary advice is to select libraries that are open source and actively developed by a diverse community of contributors. These libraries are more likely to have security controls in place in their repository to make them more secure, and first-party developers tend to be able to address flaws in these libraries more quickly.



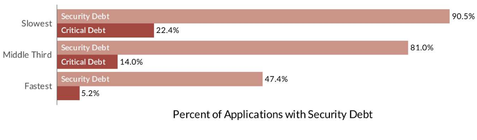
4 Prioritizing which flaws to remediate is essential

Only 15% of all flaws are critical flaws. This subset of flaws represents pound-for-pound the greatest risk exposure to your applications. Be intentional in prioritizing these issues so the work you're doing eliminates critical security debt. Prioritization will likely be handled by the policy set by your organization. Be sure your policy has you prioritizing that 15% and you'll achieve a goal of maximum risk reduction with focused effort.



5 Fixing flaws faster is the path forward

Development teams that fix flaws fastest are 4x less likely to let critical security debt materialize in their applications. Secure coding education, like Security Labs, and AI-assisted remediation tools trained on proprietary data, like Veracode Fix, will help you fix flaws faster – improving release time, reducing unplanned work, and preventing security debt.



Conclusion

Learning secure coding practices, conducting thorough testing, and addressing security vulnerabilities in a timely manner can help protect sensitive data and ensure the integrity of your creations.

[Download the full report](#) or [get a demo of Veracode Fix today](#)

VERACODE