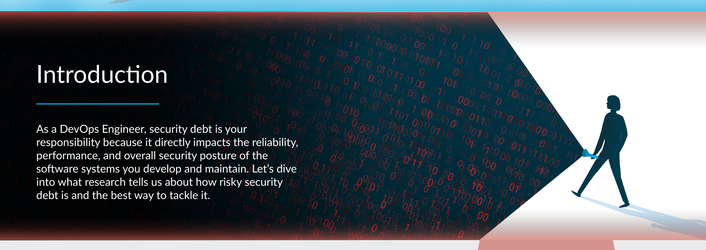
DevOps Engineers: Why Security Debt is Your Responsibility

& What You Can Do About It Based on New Data

Introduction

As a DevOps Engineer, security debt is your responsibility because it directly impacts the reliability, performance, and overall security posture of the software systems you develop and maintain. Let's dive into what research tells us about how risky security debt is and the best way to tackle it.



Step-by-Step Data Exploration

Here's the data behind managing security debt, identifying where the most risk is, and navigating it as a DevOps Engineer (based on the analysis of over one million applications).

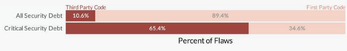
1 Organizations are drowning in security debt

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

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

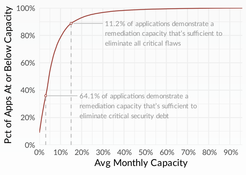
While first-party code makes up the vast majority of overall security debt, most critical security debt comes from third-party code in open-source software. As a DevOps Engineer, it's crucial to consider how testing and remediation efforts will be handled for both first-party and third-party code continuously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

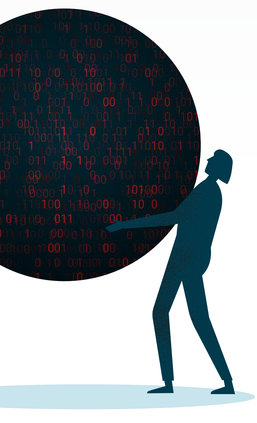
Pro tip: your development and security teams will love you if you help these teams effectively communicate with one another and make it clear which findings must be met with action against compliance or policy requirements.



3 Remediation capacity is constrained

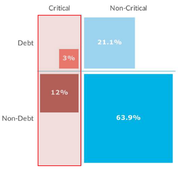
Only 64% of applications demonstrate a sustained capacity to eliminate all critical security debt. This means few teams bail fast enough to reverse the tide of debt once it starts rising. As a DevOps Engineer, it's imperative development teams have what they need to ensure teams prioritize and promptly address critical security debt first, preventing its accumulation and mitigating potential risks.





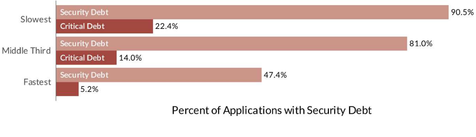
4 Prioritizing which flaws to remediate is essential

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 to sustain a program that eliminates critical security debt. Prioritize 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. Find tooling that works in the IDE where developers work, as well as responsible-by-design AI-assisted remediation tools, like Veracode Fix. This will help teams fix flaws faster – improving release time and reducing unplanned work.



Conclusion

Take data-driven action to tackle security debt effectively, including continuous scanning, prioritization, and AI-assisted remediation.

VERACODE

Download the full report or get a demo of Veracode Fix today