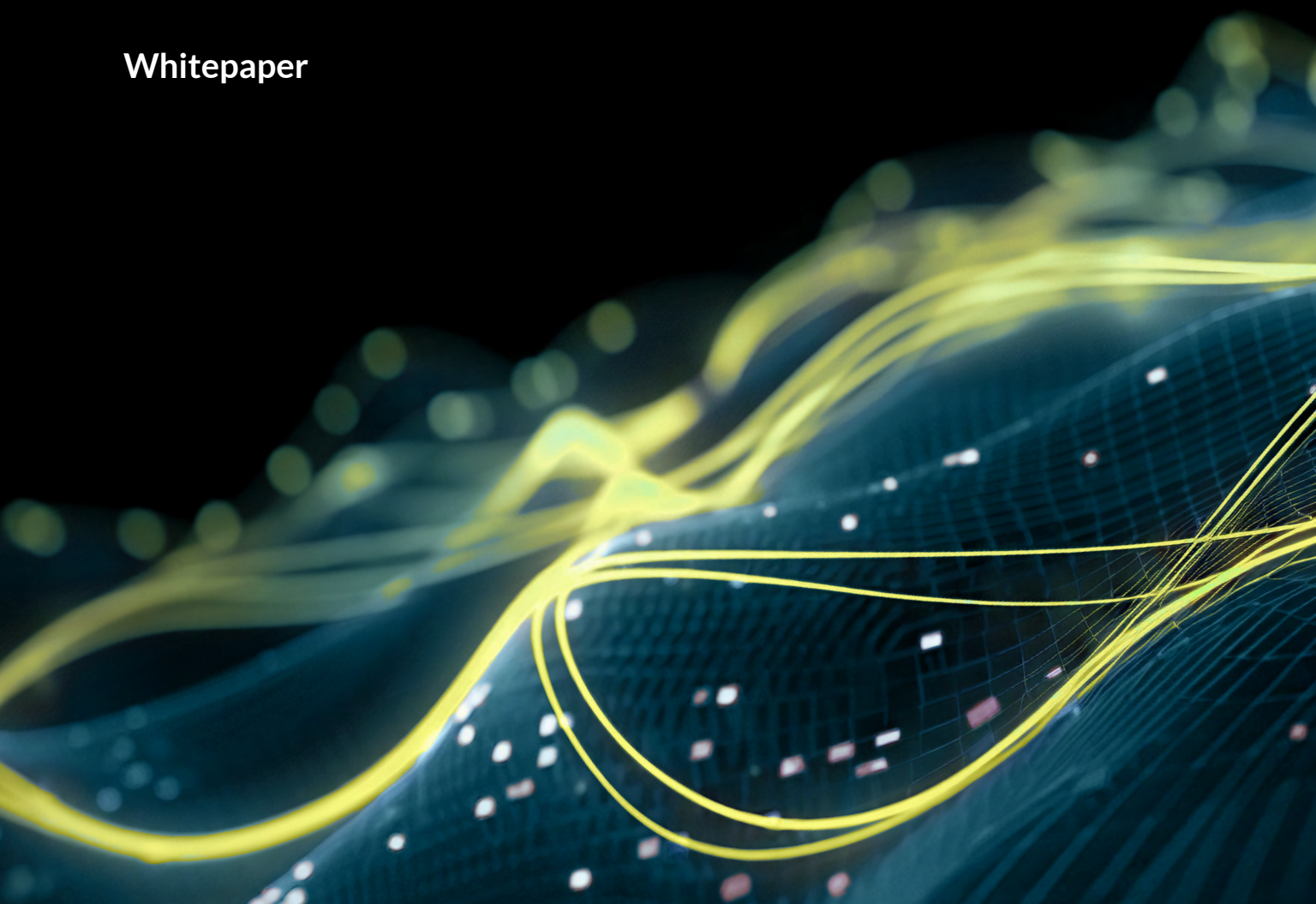
VERACODE

A Smarter Way to Secure Apps: **The Power of Veracode Fix**

Whitepaper



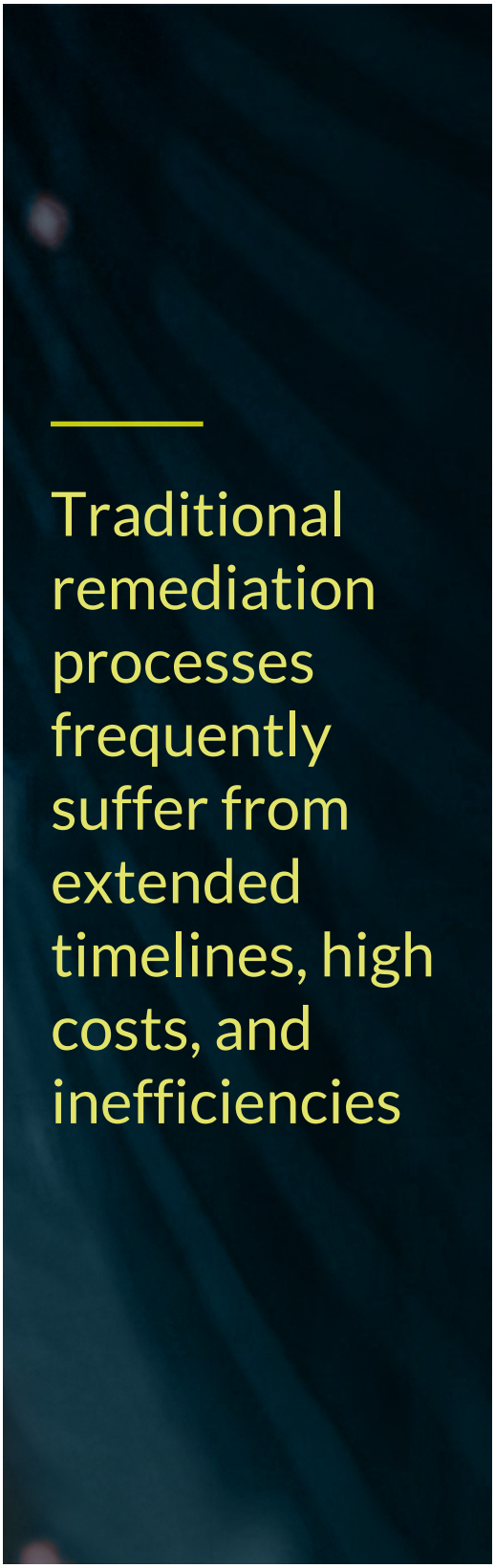
In today's digital landscape, managing software vulnerabilities is a challenge. Veracode Fix, powered by AI, streamlines remediation, reduces MTTR, and enhances security. This whitepaper explores its value amid growing security risk and complexities.

Introduction

In today's fast-paced digital landscape, application development teams face the daunting challenge of managing software vulnerabilities effectively. The growing backlog of security flaws, commonly referred to as "security debt," poses significant risks to organizations, hindering agility and impeding software velocity. To address this critical issue, Veracode Fix emerges as a game-changing solution, leveraging cutting-edge AI technology to streamline the remediation process and enhance the developer experience. This whitepaper explores the value of Veracode Fix for application development teams and application security managers, highlighting its ability to reduce Mean Time to Remediate (MTTR), mitigate breach risks, and optimize software security outcomes.

Streamline Vulnerability Remediation

The growing complexity of modern software development practices, coupled with the evolving landscape of cybersecurity threats, presents a significant challenge for security teams. Understanding and effectively addressing security risks have become increasingly intricate tasks, and staying ahead of compliance expectations has never been more demanding. As traditional remediation processes frequently suffer from extended timelines, high costs, and inefficiencies, security teams find themselves dealing with a growing backlog of vulnerabilities and frustrated developers.



Exploring the Multitude of Methods for Producing Insecure Code

In the modern development landscape, the proliferation of tools and methods for generating code has greatly expanded the avenues through which developers can produce applications.

While these tools offer remarkable efficiency and flexibility, they also introduce new vectors for creating potentially insecure code. Developers now have access to a variety of methods ranging from automated code generators to sophisticated machine learning models. Each of these methods, while innovative and powerful, carries its own set of risks and considerations, especially in the context of security. It's crucial for developers to be aware of these potential vulnerabilities as they leverage these advanced tools to create and implement software solutions.

Some examples include:

Code Generators

Tools that automatically generate code based on specific inputs or templates. Examples include Swagger Codegen, Yeoman, and SQLalchemy's Alembic.

Library and Framework Usage

Implementing libraries and frameworks which provide pre-written code for common tasks. Examples are React.js for UI, TensorFlow for machine learning, and Bootstrap for front-end development.

Large Language Models (LLMs)

Trained on extensive web content, assist developers in code generation. However, this practice entails potential legal risks. The generated code might inadvertently mirror code from a licensed library, leading to unintentional license violations.

Software Composition

Building applications by assembling pre-existing components. Examples include Docker containers, npm packages in Node.js, and Maven dependencies in Java.

Automated Refactoring Tools

Tools that automatically refactor existing code to improve its structure without changing its behavior.

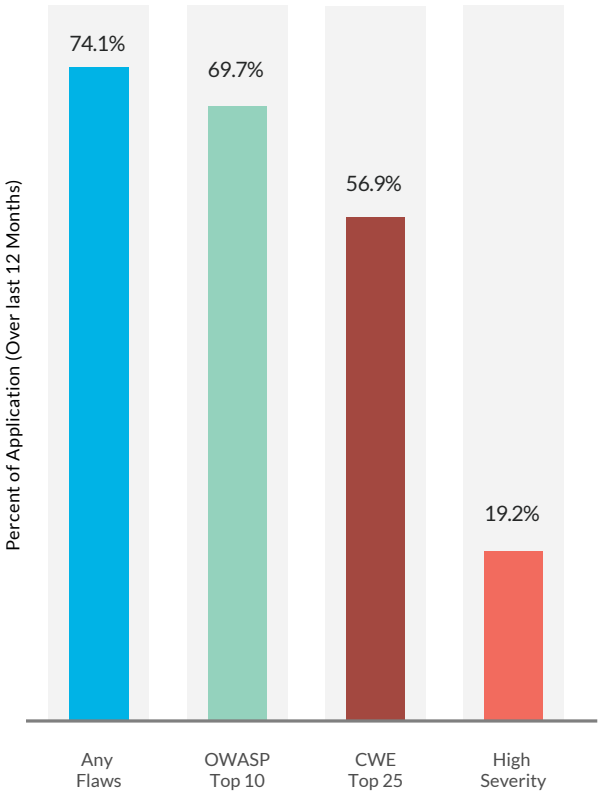
Navigating the Challenges of Open-Source Adoption and Secure Development

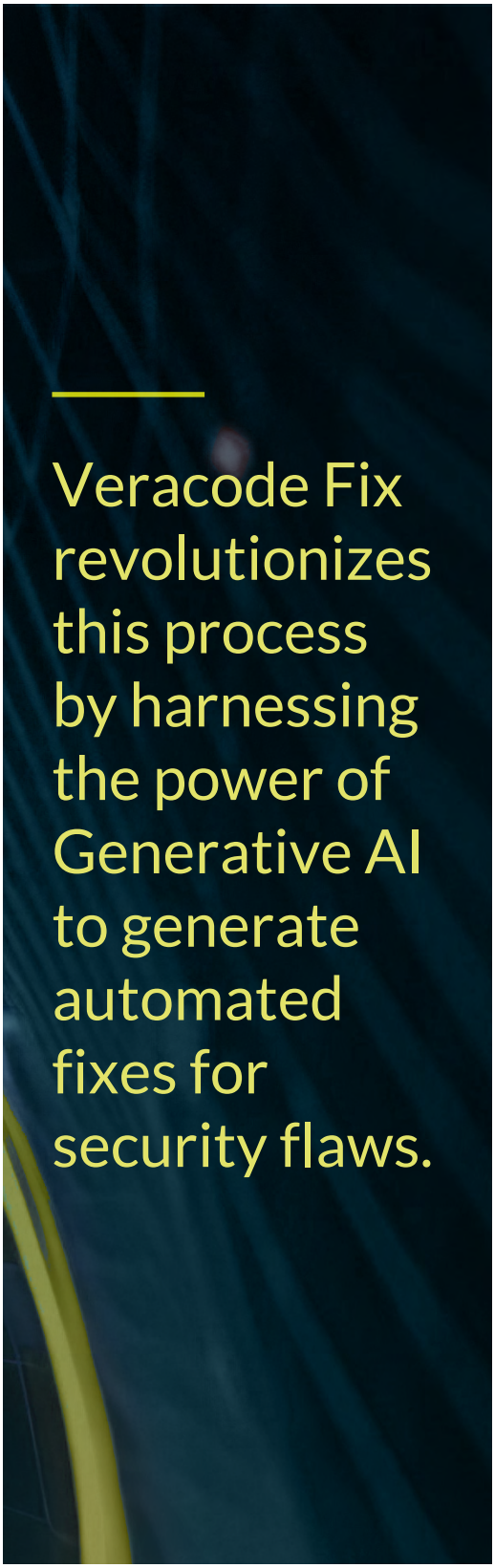
Embracing open-source libraries and freely utilizing code from source code repositories, even in environments where malicious actors or weak coding practices are prevalent, presents a dual challenge. These common developer behaviors not only introduce security risks but also give rise to concerns related to license infringements. In this intricate landscape, security teams must implement strategies encompassing automation, collaboration, continuous learning, and risk prioritization

to adeptly navigate and manage security and compliance expectations while staying in step with the ever-evolving realm of software development.

The Veracode State of Software Security Report found that the average time to fix a security finding in cloud native languages is 56 days, which is in line with reports from other industry surveys. If that's not concerning enough, our report revealed that over 74% of applications contained at least one security flaw, with nearly 20% of applications contained a high severity flaw.

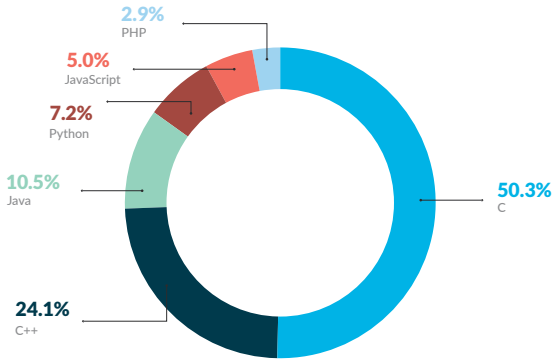
This data shows that C and C++ are the most vulnerable languages, accounting for over 75% of all reported vulnerabilities. Java and Python are also relatively vulnerable, accounting for about 17% of all reported vulnerabilities. JavaScript and PHP are the least vulnerable languages, accounting for about 8% of all reported vulnerabilities.





Veracode Fix revolutionizes this process by harnessing the power of Generative AI to generate automated fixes for security flaws.

These figures speak more to the time and effort required to resolve security flaws than to the priority of doing so. In many cases, developers are confronted with a list of flaws to address in code that they may not have authored and lack the specific security training to be able to easily address the issues. It takes significant time to interpret a finding, research a remediation technique, and then adapt it to the existing code.



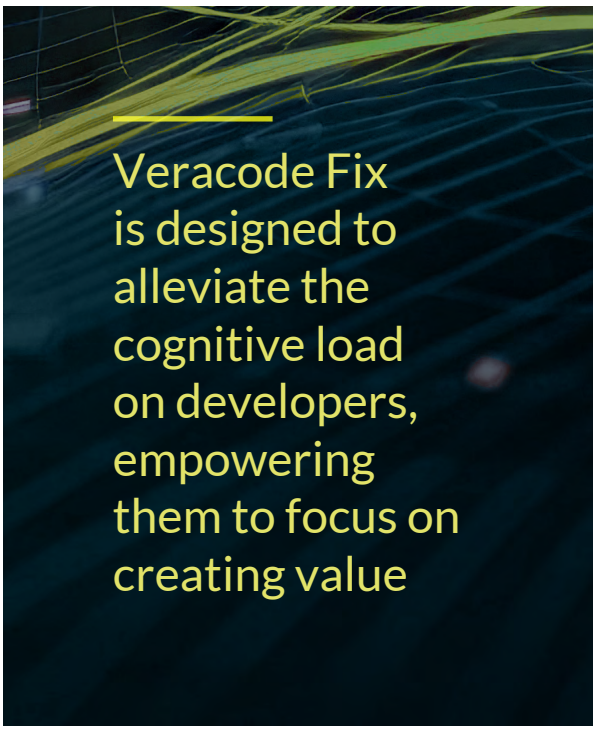
Veracode Fix revolutionizes this process by harnessing the power of Generative AI to generate automated fixes for security flaws. By training the GPT model on carefully curated data and using a comprehensive database of reference patches written by Veracode security researchers, Veracode Fix customizes best practice solutions directly to client codebases.

This AI-augmented approach significantly reduces MTTR for prevalent vulnerabilities in Java, C#, and JavaScript/TypeScript, enabling developers to remediate flaws in minutes rather than months. By building a Responsible-by-Design AI solution, Veracode can ensure the accuracy, safety, and consistency that application security requires, while delivering the flexibility and customization that Generative AI can provide.

Enhancing the Developer Experience

Veracode Fix is designed to alleviate the cognitive load on developers, empowering them to focus on creating value rather than constantly revisiting old code. By covering a substantial portion of open findings in codebases, Veracode Fix reduces the risk of exploitable vulnerabilities.

For example, in controlled tests, the remediation time for a CWE-117 vulnerability in a Java application was reduced from 35 minutes to just 3 minutes using Veracode Fix. This efficiency and ease of use significantly enhance the developer experience, fostering productivity and job satisfaction. Developers can move away from the onerous task of interpreting security findings then researching and testing security fixes to reviewing and approving AI suggested fixes.

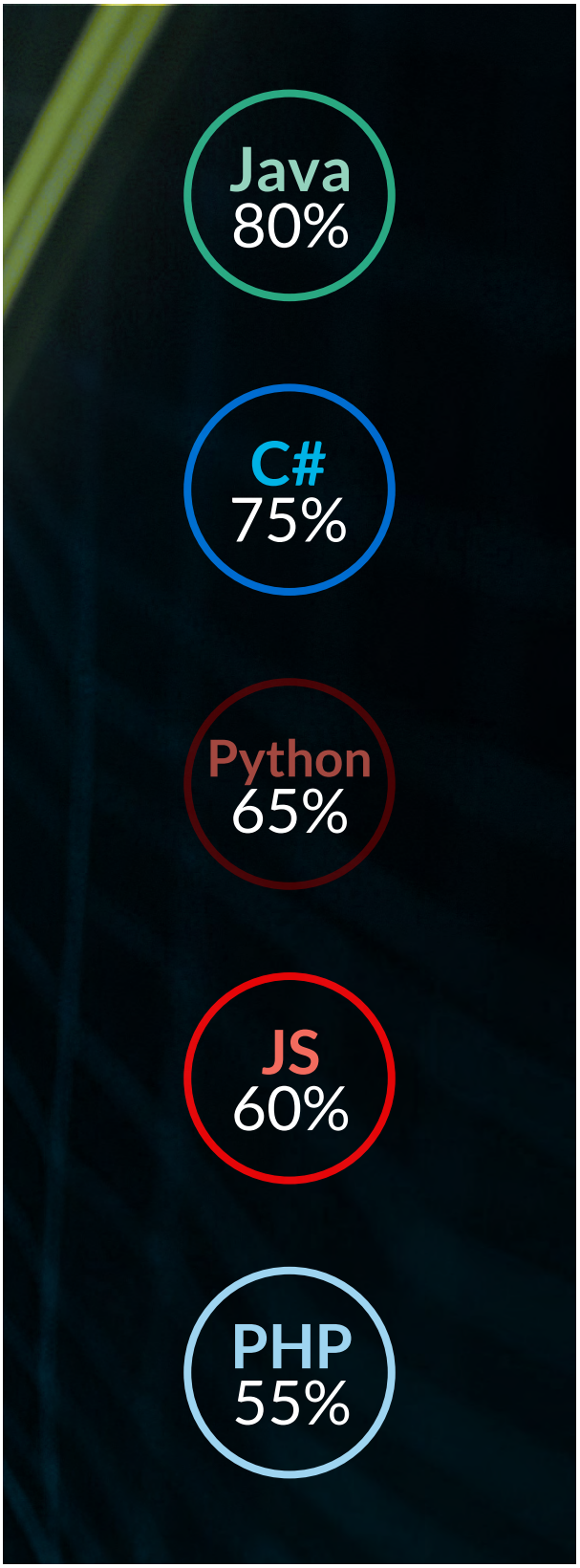


Mitigating Breach Risks

For a Veracode Fix customer with even a small security backlog of 2000 open flaws this represents a potential unlocking of over 6 months of developer effort (see how much time your organization can save with our ROI calculator)

As well as reducing the toil of correcting detected security mistakes, developers can use Veracode to create less of them. Using a combination of Veracode Static Analysis, Veracode Software Composition analysis, Veracode Container security, and Veracode Fix early in the development lifecycle, developers can detect and resolve code security, container and IaC issues before committing code. The result: Cleaner code into the CI/CD pipeline, improved delivery velocity, and a reduced security backlog.

The presence of unaddressed vulnerabilities poses a significant risk to organizations, increasing the likelihood of data breaches and system compromises. Veracode Fix's comprehensive coverage of common vulnerabilities in Java, C#, and JavaScript/TypeScript significantly lowers the risk of data breaches. By addressing 92% of vulnerabilities[1] in Java, 98% in JavaScript/TypeScript, and 86% in C#, using Veracode Fix to reduce the time to remediation can substantially reduce the risk of data breaches. This reduction translates into substantial cost savings, considering the average cost of a data breach is $4.45 Million USD according to IBM.



Efficacy of Veracode Fix Across Programming Languages

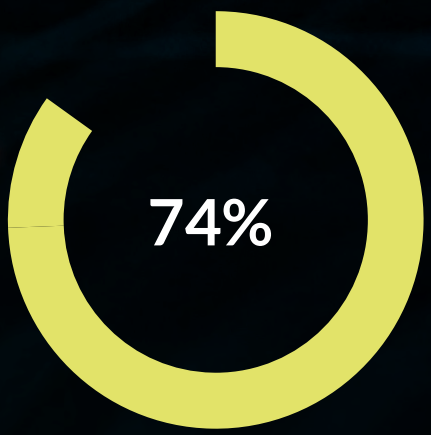
To underscope the unique value proposition of Veracode Fix, consider the following data indicating the percentage of issues it can address by programming language:

Veracode Fix is particularly effective in resolving issues in java and C#, while also significantly addressing challenges in Python, JavaSript, and PHP.

Veracode Fix is equipped to tackle a variety of critical issues,including:

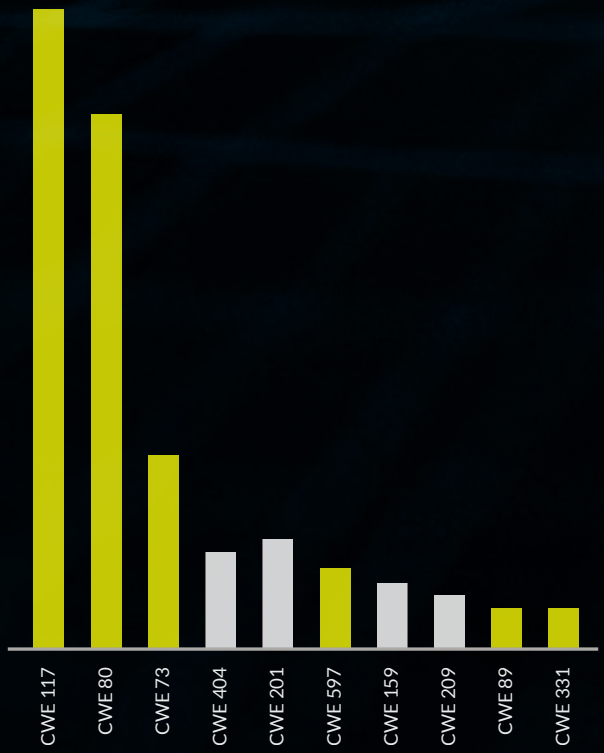
- SQL Injection
- Cross-Site Scripting (XSS)
- Insecure Direct Object References (IDORs)
- Commnand Injection
- Path Traversal
- Broken Authentication and Session Management
- Exposure of Sensitive Data
- Use of Insecure Cryptographic Algorithms
- Injection Flaws
- Cross-Site Request Forgery (CSRF)

Efficacy of Veracode Fix Across Programming Languages



Percent of Veracode Static Analysis Java findings with Veracode Fix CWE coverage.

Top 10 Java CWE Static Analysis Findings



Conclusion

Veracode Fix emerges as a critical tool in the arsenal of application development teams and application security managers. By seamlessly integrating into the development workflow and offering AI-driven remediation suggestions, Veracode Fix enhances developer productivity, reduces security debt, and mitigates breach risks. As organizations strive to balance speed, innovation, and security in an increasingly digital world, Veracode Fix plays a pivotal role in transforming the vulnerability remediation process. Embrace Veracode Fix and unlock the full potential of your application security strategy.

Author
Brian Roche, CPO Veracode

Learn more at **www.veracode.com**, on the **Veracode blog** and on **Twitter**.