SOLUTION BRIEF

LACEWORK

Code security

Visibility and context across the entire cloud-native application lifecycle

OVERVIEW

Software is the lifeblood that powers today's leading businesses. But with millions of lines of code that are constantly being added, forked, and updated, how can teams effectively secure their code and protect systems, data, and their reputations?

Application security testing is essential. However, legacy tools tend to generate excessive noise, lack the context necessary to understand vulnerabilities in terms of risk, and fail to give security teams much needed visibility. Developers waste time on false positives and low-risk vulnerabilities. Security teams struggle to understand who owns the running code, which teams need more help, and how to effectively reduce risk.

THE LACEWORK SOLUTION

Instead of addressing code security and cloud security as separate practices or standalone gates, Lacework addresses them as part of a single security continuum that spans the entire cloud-native application life cycle, from code to run.

Lacework empowers teams to understand third-party and first-party code vulnerabilities in terms of relative risk. Continuous end-to-end visibility and context across the cloud-native application life cycle facilitates a risk-based approach and empowers teams to understand and manage vulnerabilities within the context of their specific applications and environments.

Security teams effectively manage runtime risks with downstream development context such as what code is secure, where compensating controls are required, and who owns the running code. Likewise, development teams can more efficiently deliver secure code by gaining upstream runtime context such as whether code is running, what version or versions are in use, where it is running, and whether it's behaving as expected.

Challenges

- Security teams struggle to understand and communicate vulnerabilities in terms of relative risk
- Developers waste precious time on inaccurate findings and false positives and cannot prioritize what matters most
- Security teams lack visibility into code ownership, remediation progress, and who needs support

Lacework benefits

- Gain visibility into all third-party software, their direct dependencies, and any associated risks
- Uncover first-party code defects that could become zero-day exploits and minimize false positives
- Understand vulnerabilities in terms of relative risk to prioritize which fixes will have the biggest impact on reducing overall risk
- Boost developer productivity and eliminate time wasted on inaccurate findings and those they didn't introduce

Software composition analysis (SCA)

Lacework SCA gives customers continuous visibility into their third-party software, indirect dependencies, and associated vulnerabilities. Lacework SCA goes beyond simply identifying whether you have certain vulnerable third-party packages in your code. Teams gain continuous visibility into who introduced them into the code, when and how these packages are used, whether they are actively called in running workloads, and exactly which line of code needs updating.

This empowers security and development teams to more accurately prioritize what matters most and quickly isolate and remediate risk. Organizations understand vulnerabilities in terms of risk, security teams know who is responsible for fixing the code, and developers gain automated remediation of vulnerable dependences.

Software bill of materials (SBOM)

With Lacework SCA, teams gain an always-up-to date software bill of materials for every application. It includes information about the version and source of each component, and provides teams a clear understanding of the ingredients that make up their code.

Beyond SBOM generation, Lacework empowers teams to manage and share SBOM information with downstream consumers of this data in an easily digestible manner. It also allows them to identify and address any open-source licensing issues that may put them at risk — like identifying overly-restrictive licenses and licenses that may require the licensee to make their code available under the same terms.

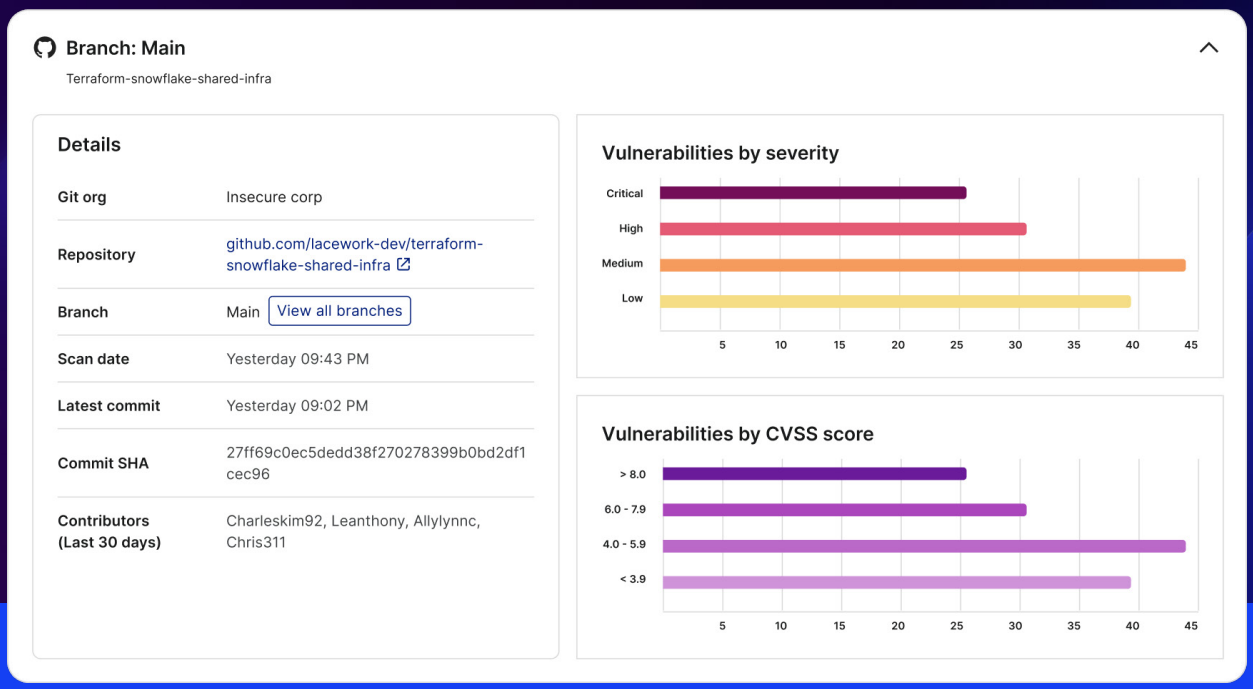


Figure 2: Lacework analyzes and monitors the flow of contaminated data throughout the application to identify risks like SQL injection.

Static application security testing (SAST)

Organizations must also understand how their first-party code could be exploited by an attacker. Lacework SAST identifies source-code weaknesses and defects in code that you write in-house, essentially automating an expert-level secure code review. Unlike SCA, SAST does not find known common vulnerabilities and exposures (CVEs). Instead, it analyzes source code patterns and data flows to identify possible weaknesses that an attacker could exploit to bypass security controls, run malicious commands, and exfiltrate sensitive data.

A key benefit of SAST is that it analyzes 100% of the codebase. However, historically it's been plagued by noisy results with a large number of false positives. To provide highly-accurate findings that enable high fix rates, Lacework provides unique analysis of application control and data flows to pinpoint hard-to-find application exploits. For developers, Lacework also provides a quicker version of SAST that can scan code for security defects as it is written.

Because the platform intimately understands an application's constructs, Lacework SAST automatically identifies whether a developer has already built compensating controls within the code to safely mitigate a weakness. Our deep analysis and complete understanding of language constructs and frameworks creates higher rates of true positives and lower rates of false negative and false positive findings.

Deep SAST for application security engineers

Lacework SAST allows application security teams to find and fix the most exploitable parts of their codebases. Our platform can help security teams scale, with an engine that can analyze millions of lines of code per minute.

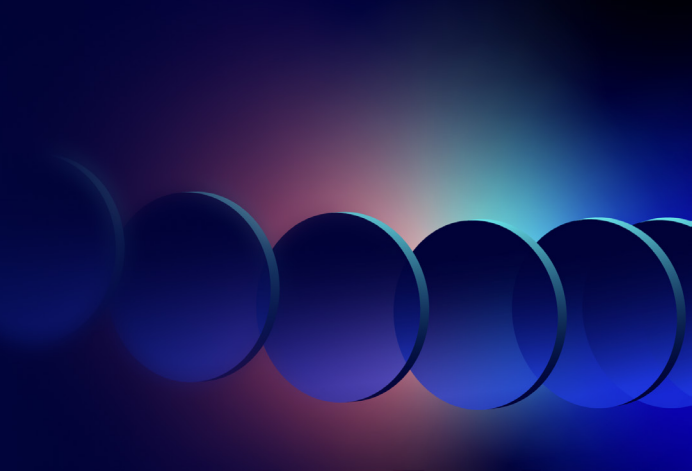
- **Trace data flows throughout the application.** An unprecedented understanding of how data flows through an application allows teams to find previously unknown security defects and determine whether compensating controls exist.
- **Code attribution.** The platform captures git history to establish ownership over code and ensure developers are fixing the vulnerabilities that they introduced.
- **Built by and for security engineers.** Security engineers can easily tune the SAST engine with their own safe functions/types to meet the needs of their unique codebases and minimize false positives.

SAST for developers and dev teams

Lacework SAST allows development teams to find common weaknesses or vulnerabilities as they write code. The platform produces fast results along with actionable guidance on how to fix security flaws.

- **Easy setup and run.** Lacework SAST can integrate with source code management (SCM) tools like GitHub, GitLab, and Bitbucket and can automatically detect any new code repositories. The platform can also integrate with continuous integration (CI) tools like GitHub Actions and GitLab Pipelines or with integrated development environments (IDEs) like Visual Studio (VS) Code.
- **Clear security guidance.** Developers receive automated guidance on how to fix security issues as they write code. This guidance is clear and straightforward, in language familiar to developers.
- **Speed and focus.** By capturing git history, the platform allows developers to see only the vulnerabilities that they've introduced.

> **"I've been in the industry for many years. When we sat down with our infrastructure and DevOps teams to review Lacework, that was the only time I've ever seen all the teams agree on a solution."**
>
> JOHN TURNER, SENIOR SECURITY ARCHITECT, LENDINGTREE



> "With Lacework's behavioral ML-led approach, we'll dramatically reduce security alerting while automating key audit data delivery so that we can focus on the secure development of new products for our customers. The Polygraph Data Platform will completely transform the way our security team works."
>
> LUKE CLINCH, GLOBAL DIRECTOR OF INFRASTRUCTURE, VERIFONE SYSTEMS, INC.

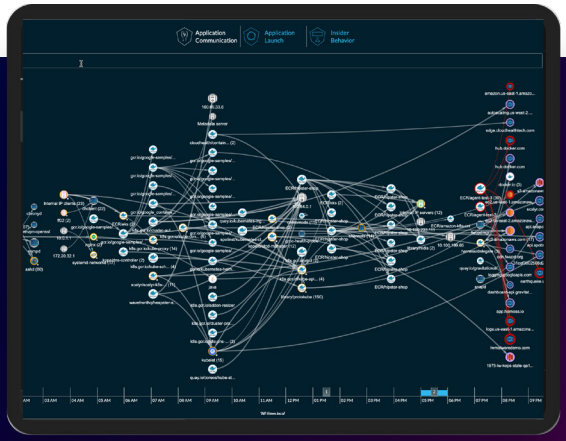
Infrastructure as code (IaC) security

The Lacework platform extends automated security and compliance checks of IaC early in the development process to prevent misconfigured cloud services from being deployed. Within minutes, Lacework can be connected to code repositories or CI/CD pipelines. We support templates such as Terraform, AWS CloudFormation, and Kubernetes/Helm and can scan Dockerfiles. Teams can use Lacework to automatically discover IaC templates and code repos used across the organization.

Lacework adds security guardrails that automatically and continuously scan IaC for policy violations and immediately alert developers as they are writing code. When IaC is submitted (pull request or commit) to a code repository, Lacework will automatically test it against any assigned security and compliance policies.

Organizations can also use Lacework to test IaC once it has been packaged for delivery to ensure production parameters, like us-west-1, are within policy.

All identified misconfigurations are recorded within the code repository, as well as the Lacework UI, to ensure both development and security teams have shared visibility into the security of the code. Developers receive immediate and automated remediation guidance for fixing the code, as well as the ability to auto-remediate with just one click.



Want to learn more?

Get in touch

Watch an on-demand demo >>