SANS

Analyst Program

Sponsor Perspective

DevSecOps Success: What's Team Metrics Got to Do with It?

Written by **Kenneth G. Hartman**

October 2022

VERACODE

©2022 SANS™ Institute

Introduction

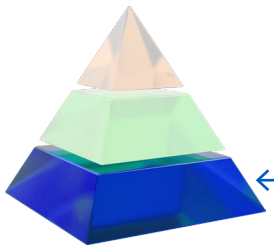
We all seek success, but how do we measure it? This question is on the mind of any DevSecOps practitioner interested in continuous improvement amid the volatile forces at play in the cloud, such as continuous change and a globally exposed, internet-facing attack surface. A central premise of modern management is *what is measured is controlled*, and its corollary, *what is not measured, is not controlled*.

The SANS 2022 DevSecOps Survey asked participants to rank the major KPIs they use to measure the success of their DevSecOps activities.[1] This paper builds on those findings and identifies metrics that leadership can use to measure DevSecOps maturity. Rather than providing an exhaustive list of metrics, we propose a leadership approach to using metrics to mature a DevSecOps program using selected metrics as examples.

Selecting Metrics to Promote Maturity

DevSecOps teams that are serious about improvement should select metrics appropriate to their process maturity. Not every task a team works on should be reflected as a metric—only the most critical initiatives that have the biggest impact. A group serious about improving software security might set an objective such as: "Implement application security testing in the CI/CD (continuous integration/continuous delivery) pipeline to prevent vulnerabilities in the code pushed to production." Aligned with this objective, different metrics might be selected at different stages of maturity to show meaningful incremental progress toward achievement of this objective.

For example, an organization may decide to focus on static application security testing (SAST) first and then add in dynamic application security testing (DAST) next. In this example, an initial metric would be something like: "The percentage of applications in the codebase with an initial SAST scan." This metric focuses on getting the various teams to integrate the SAST scanner with their CI/CD pipeline and using that integration to scan their application at least once. This is an essential first hurdle that demands considerable leadership and change management.



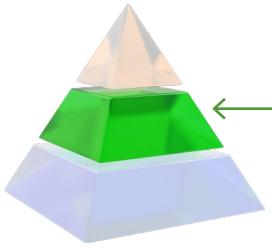
Initial-Stage SAST Metric

Percentage of applications in the codebase with an initial SAST scan

Continuing with this example, the next metric might be: "Percentage of applications in the codebase with a passing SAST scan in the past seven days." After each team has integrated the SAST scanner with their code pipelines, they need to use it regularly. All code has a lifecycle, regardless of how well that lifecycle is defined. Some applications are being actively developed whereas others are just being maintained. Other code might need to be deprecated.

[1] "SANS 2022 DevSecOps Survey: Creating a Culture to Significantly Improve Your Organization's Security Posture," September 20, 2022, www.sans.org/white-papers/sans-2022-devsecops-survey-creating-culture-improve-organization-security/ (Registration required for download.)

Unfortunately, due to the unrelenting nature of security research, new vulnerabilities are found on an ongoing basis, even in mature code. Therefore, all code currently running in production must be scanned recently, regardless of when it was deployed. This second metric gauges the DevSecOps team's adoption of SAST scanning company-wide and might result in some pruning of the codebase to reduce the maintenance burden.

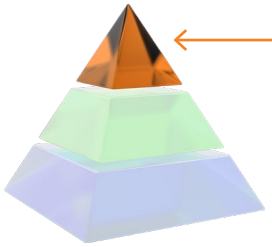


Adoption-Stage SAST Metric

Percentage of applications in the codebase with a passing SAST scan in the past seven days

At this stage, it may be appropriate to have a lower threshold for what is considered a passing scan. This threshold should be configurable in your SAST scanning tool. Keeping the goal within sight is critical to developers so they can feel like they are having success. Once the initial threshold has been met, leadership can raise the bar until all vulnerabilities are being managed at the desired, steady, stated level of compliance.

Although the scan results include a thorough explanation of the findings, complete with remediation advice, a developer may have questions about a particular finding. In such a case, the developer can schedule a consultation call with a Veracode Application Security Consultant to discuss the result and learn how to remediate it. This consultation service fosters faith in the trustworthiness of the tool while increasing the developer's level of understanding.



Mature-Stage SAST Metric

Percentage of applications in the codebase that get a passing SAST scan on the first pass

An organization that has come up to speed with SAST scanning might decide to raise the bar further by transitioning to the following SAST metric: "Percentage of applications in the codebase that get a passing score on the first pass." This metric is intended to shift the focus of secure software development earlier in the workflow. Of course, developers could game the system by waiting to commit code to the main branch until they were sure it did not contain known security flaws, but then again, isn't this the exact behavior we want to encourage? Veracode enables a developer to create a sandbox that permits code still in development to be submitted for static analysis without counting toward the metric. This metric encourages developers to pretest their code before it is committed to the main branch and know with certainty that it will pass.

In this section, we covered how leadership can leverage different metrics to focus efforts on the DevSecOps practices they are attempting to promote. Next, let's discuss how to apply this process to additional metrics.

Building on Success

Certain types of vulnerabilities can only be found by scanning the application while it is running; therefore, DAST should be used to complement SAST. Based on the SANS 2022 DevSecOps Survey, 70% of respondents believe that DAST is a "useful" or "very useful" security practice. Yet only 42% indicated that security testing (including DAST or SAST) is performed as part of the build process.[2] Benchmark data like this is a strong indicator that a metric addressing DAST would be appropriate for many DevSecOps teams. Veracode also supports DAST integration into the CI/CD pipeline.

Therefore, leadership might decide to phase in DAST metrics in a similar manner as the previously discussed SAST metrics. Figure 1 summarizes these metrics.

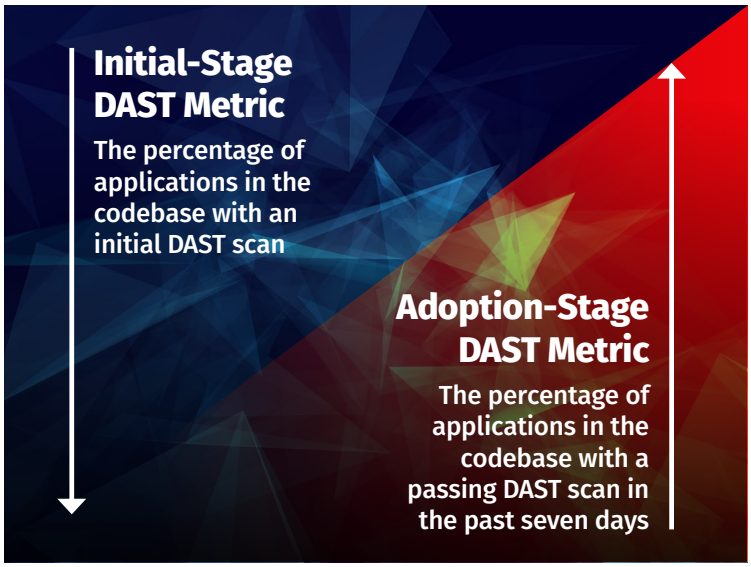


Figure 1. DAST Metrics

As another example, consider the topic of supply-chain attacks. Veracode's 2022 *State of Software Security* report says that 97% of the typical Java applications consist of open-source code libraries and that 77% of flaws in third-party libraries remain unfixed after three months, with 50% remaining unfixed after 12 months.[3] (Refer to Figure 2 for details.)

The best practice is to use an artifact repository as the source of truth for all approved third-party code. Downloading third-party code directly from the internet (from sources such as Docker Hub) is inherently dangerous because even if a package is not malicious at the time of download, there are no guarantees it will not be poisoned in the future. Diligent organizations have a formalized process for vetting the packages admitted to the artifact repository and scanning them to detect any new vulnerabilities.

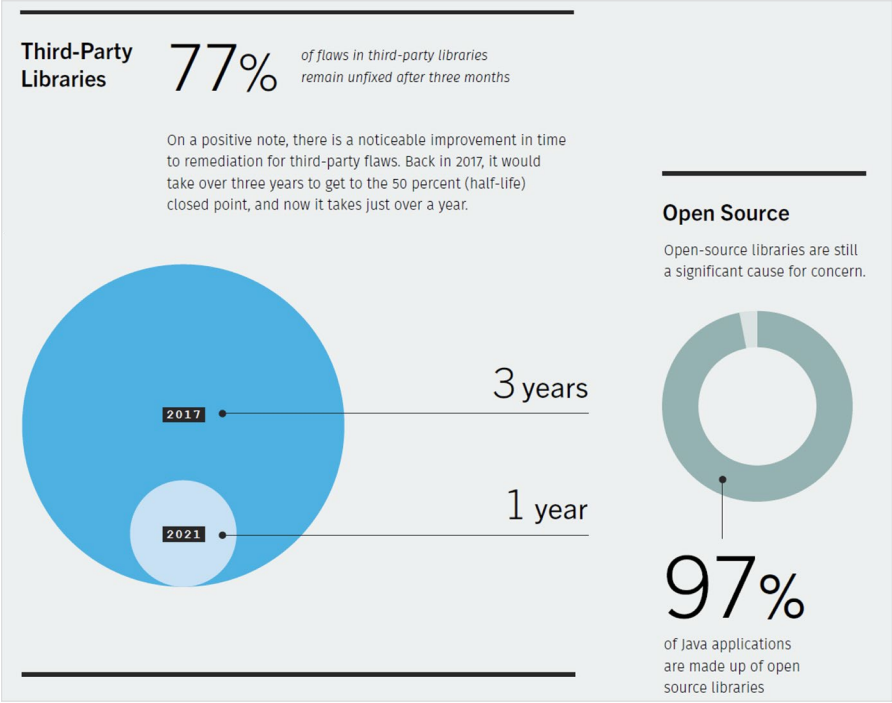
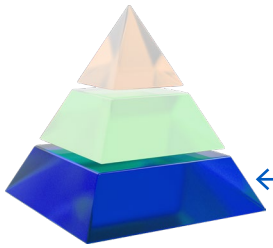


Figure 2. Data on Third-Party Libraries from the Veracode 2022 State of Software Security Report (Used with Permission)

[2] "SANS 2022 DevSecOps Survey: Creating a Culture to Significantly Improve Your Organization's Security Posture," September 20, 2022, www.sans.org/white-papers/sans-2022-devsecops-survey-creating-culture-improve-organization-security, p. 13. (Registration required for download.)

[3] "State of Software Security v12," www.veracode.com/state-of-software-security-report

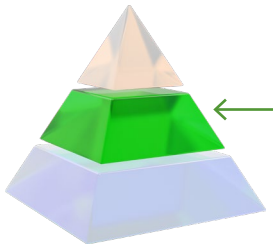
The first challenge is to generate an inventory of all third-party software used by the organization's codebase. In fact, this would make a great initial metric because "Inventory and Control of Software Assets" is one of the 18 CIS Critical Security Controls.[4] Tools such as Veracode's Software Composition Analysis (SCA) can be used to analyze a codebase, identify vulnerabilities, detect license violations, and generate a software bill of materials (SBOM) or, in other words, the software inventory.[5]



Initial-Stage Code Provenance Metric

Percentage of applications in the codebase with a SBOM

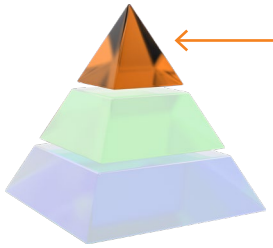
The next step is to define the formalized process for vetting the packages admitted to the artifact repository. Admittedly, it can take some effort for the DevSecOps team to reach a consensus on this process. An alternative approach for the quick win can be grandfathering any packages that pass the SCA scan as of a specific date and incrementally raising the bar as the vetting process is hammered out. After the list of authorized packages is determined, administrators can configure the scanner to detect when an unapproved package is introduced. The adoption stage metric becomes: "Percentage of third-party code packages that have been vetted."



Adoption-Stage Code Provenance Metric

Percentage of third-party code packages that have been vetted

While working on the adoption stage code provenance metric, the approved packages should be preserved in the artifact repository, and the CI/CD pipeline should be refactored to pull the packages from the artifact repository. DevSecOps resources should be dedicated to maintaining the artifact repository, including the responsibility to ensure frequent and regular SCA scans of the repository. When this transition is complete, the team will have attained a 100% score on the mature-stage code provenance metric: "Percentage of third-party code pulled from a controlled artifact repository with no known vulnerabilities."



Mature-Stage Code Provenance Metric

Percentage of third-party code pulled from a controlled artifact repository with no known vulnerabilities

The final task is to prevent regressions using a preventive control, such as firewall rules, so the CI/CD pipeline cannot pull down code from the internet.

In this section, we have covered additional metrics that leadership can introduce as appropriate to build on the momentum of DevSecOps improvement. These are only examples and certainly do not constitute an exhaustive list.

[4] "The 18 CIS Critical Security Controls," www.cisecurity.org/controls/cis-controls-list

[5] "Software Composition Analysis (SCA)," www.veracode.com/products/software-composition-analysis

Benchmarking Success

In the introduction, we mentioned management's penchant for measuring success. The next logical question is: What do we measure against? Comparisons across time can be used to make decisions on allocating DevSecOps resources, and comparisons with industry peers can demonstrate due care. Due care is the ongoing duty of management to manage risks in a manner at least commensurate with industry norms. This is where metric benchmarks come into play.

Although any metric that more than one company measures can serve as a benchmark, the best benchmarks have substantial data supporting them and have been identified by industry consensus. Figure 3 shows the top KPIs (metrics) as identified in the SANS 2022 DevSecOps Survey.

Certainly, any of the top-ranked metrics listed in Figure 3 are good candidates, but leadership should select just a few of them based on the top behaviors they want to encourage. Too many metrics can dilute the team's focus. On the other hand, comparisons against industry benchmarks can be very motivating to teams when framed properly by leadership.

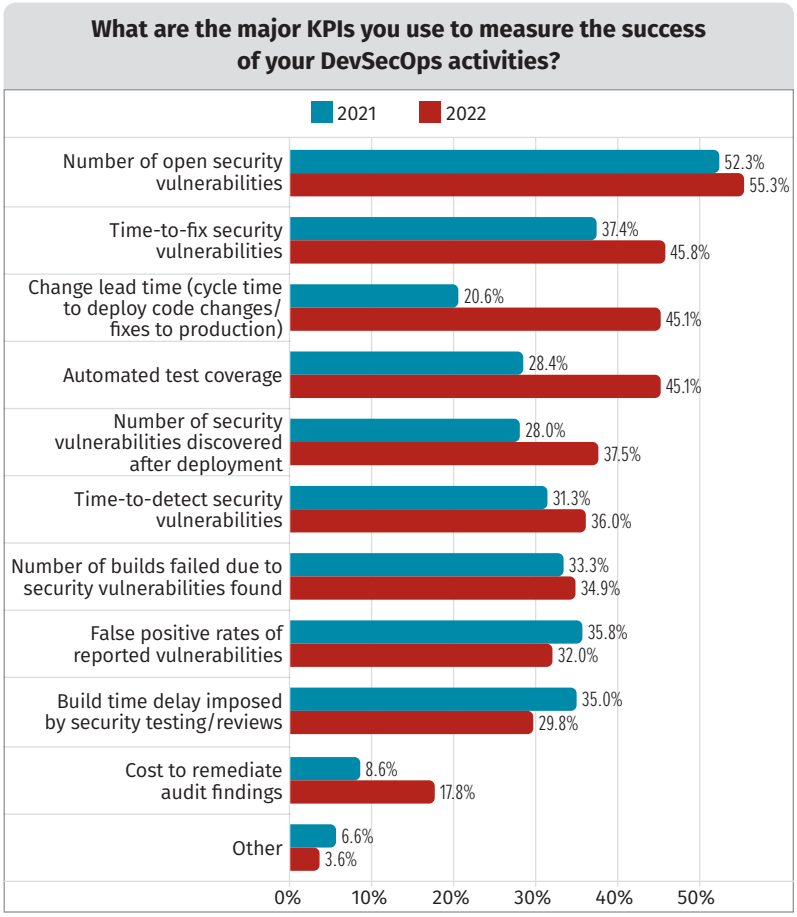


Figure 3. Top DevSecOps KPIs, 2021 and 2022

Final Thoughts

Leadership should use metrics to focus the efforts of DevSecOps teams on the most important "next actions" that are appropriate to the team's level of process maturity. The metrics should track the status of the goals that were set to achieve the continuous improvement objectives.

Similarly, leadership should use industry security benchmarks to identify opportunities and as a yardstick for success. The insights gained from examining the team's metrics in the context of industry benchmarks can help leadership shape and refine the continuous improvement objectives, creating a virtuous cycle.

Sponsor

SANS would like to thank this paper's sponsor:

VERACODE