oxsecurity

Key Performance Indicators (KPIs) for Application Security: Lessons from Leading CISOs

Over the past three months, the OX team had the opportunity to engage with a variety of Chief Information Security Officers (CISOs) at industry events and meetings. Throughout these conversations, a common theme consistently emerged: the crucial role of metrics. Assessing the effectiveness of security initiatives can be complex without well-defined Key Performance Indicators (KPIs). Many teams struggle to identify which metrics to prioritize, especially in application security. This white paper will delve into several KPIs that have repeatedly come up in discussions and are key to evaluating an organization's application security posture.

Security Coverage

A foundational aspect discussed was security coverage. This refers to a basic inventory of the technical capabilities an organization possesses and how well they are deployed across all relevant assets. If static code analysis and software composition analysis tools are operational on 60-70% of your pipelines, the coverage might be calculated as 2x60 or 120-140%.

Then, the obvious question arises: What constitutes "good" coverage?

There are eight essential elements in today's development environments, including:

Git Posture

API Inventory (BOM)

SaaS BOM

Artifact BOM

Code Scanning

Open-source Reputation

Container Security

Infrastructure as Code

As new technologies and threats emerge, coverage maps will need to adapt. For instance, some organizations are embedding CSPM configuration results into this map, and some integrate DAST or vulnerability assessment.

Mean Time to Triage (MTTT)

With over 100 new vulnerabilities disclosed daily, reducing MTTT is essential to prevent the accumulation of security risks. This metric gauges the time from the initial detection of a security weakness to when a fix is scheduled. It serves as an indicator of AppSec capacity requirements and helps prioritize vulnerabilities based on their severity and exploitability. The MTTT varies widely among organizations and depends on several factors including the severity of the issue, the organization's resources, and its specific security protocols.

Kill Chains vs. Vulnerability (Signal to Noise Ratio)

This metric focuses on distinguishing between the common "noise" and significant threats. A good signal-to-noise ratio, such as 1:20 to 1:100, helps prioritize issues that truly need attention. By filtering out noise and prioritizing actionable threats, organizations can allocate resources more effectively, ensuring that security efforts are directed towards mitigating significant risks.

Developer Empowerment

Another significant metric is developer empowerment, which gauges how effectively developers can independently address security issues without direct support from application security teams. Gartner predicts, "by 2026, 70% of platform teams will integrate application security tools as part of internal developer platforms to scale DevSecOps practices, up from 20% in 2023." Therefore, having a metric that assesses a developer's ability to identify, understand, and remediate vulnerabilities on their own is crucial. Some CISOs shared how organizations track the involvement of engineering managers who oversee team-specific security postures, providing valuable snapshots of current security exposure.

Measuring Backlog Bloat

Backlog bloat refers to the accumulation of unresolved security vulnerabilities over time. By systematically measuring, organizations can gain insights into their backlog management efficiency. Key metrics to assess include:

Total Number of Unresolved Vulnerabilities: Regularly track the number of unresolved vulnerabilities to identify trends and spikes that may indicate increasing backlog bloat.

Average Age of Unresolved Vulnerabilities: Measure how long vulnerabilities remain unresolved in the backlog. Older vulnerabilities can indicate prioritization issues or resource constraints.

Resolution Rate: Calculate the rate at which vulnerabilities are resolved over a specific period. A decreasing resolution rate may signal growing backlog bloat.

New Vulnerability Rate: Monitor the rate at which new vulnerabilities are discovered and added to the backlog. A high rate of new vulnerabilities can exacerbate backlog bloat if not matched by an adequate resolution rate.

Backlog Clearance Time: Estimate the time required to clear the backlog at the current resolution rate. This metric helps in planning and resource allocation for effective backlog management.

Service Level Agreements (SLAs)

SLAs define response times and resolution targets for addressing security vulnerabilities, often sparking debate as definitions vary based on external obligations to governments or vendors, or internal legal commitments to standards like PCI 4.0 or FedRamp. Metrics can include:

- **Time to Acknowledge:** Measures the time taken from when a security vulnerability is reported to when it is acknowledged by the security team. An example SLA could require that all security vulnerabilities be acknowledged within 24 hours of being reported.
- **Time to Assess:** Records the time from acknowledgment of the vulnerability to its initial assessment. The SLA might stipulate that the initial assessment should occur within three days to determine the severity and impact of the vulnerability.
- **Time to Remediate:** Tracks the time from when the vulnerability is first reported to when it is fully remediated. SLAs for remediation can vary based on the severity of the vulnerability, e.g., critical vulnerabilities might have a remediation SLA of seven days, while less critical ones might have more extended periods such as 30 days.
- **Patch Deployment Time:** Measures the time taken to develop, test, and deploy patches for identified vulnerabilities. An SLA might require that patches for high-priority issues be deployed within a specific timeframe, such as 14 days from the identification.
- **Regression Testing Time:** Ensures that the patch does not adversely affect existing functionalities. The SLA could specify that regression testing should be completed within five days of patch deployment.
- **Incident Response Time:** Refers to the time taken to respond to and start mitigating the impact of an exploited vulnerability. SLAs might demand that response actions commence within hours of detection, depending on the criticality.
- **Vulnerability Disclosure Time:** Measures the time from when a vulnerability is confirmed to when it is disclosed publicly, often coordinated with the release of a patch. SLAs here ensure that the organization controls the narrative and timing of the disclosure.

Preventative Measures

Advanced organizations often evaluate how many issues were preemptively resolved before affecting the codebase. Addressing vulnerabilities early can drastically reduce costs compared to remedial actions, highlighting the importance of proactive security measures. Two examples include:

COST SAVINGS

The IBM Cost of a Data Breach Report frequently highlights that organizations with fully deployed security automation (which includes preventative measures) incur significantly lower data breach costs compared to those without. For example, the 2020 report indicated that automated security measures could reduce the cost of a data breach by up to $3.58 million.

REDUCTION IN VULNERABILITIES

Studies such as the one from the Ponemon Institute suggest that organizations that invest in preventative measures such as security testing during the software development lifecycle (SDLC) can reduce the number of security defects in released software by up to 50%.

Conclusion

Effective application security requires a comprehensive set of KPIs to measure and enhance security practices. By focusing on security coverage, developer empowerment, adherence to SLAs, reducing Mean Time to Triage, maintaining a good signal-to-noise ratio, implementing preventative measures, and managing backlog bloat, organizations can significantly improve their security posture and resilience against threats. These metrics, drawn from the insights of leading CISOs, provide a roadmap for organizations striving to achieve robust application security.

About OX Security

At OX Security, we're simplifying application security (AppSec) with the first-ever Active ASPM platform offering seamless visibility and traceability from code to cloud and cloud to code. Leveraging our proprietary AppSec Data Fabric, OX delivers comprehensive security coverage, contextualized prioritization, and automated response and remediation throughout the software development lifecycle. Recently recognized as a Gartner Cool Vendor and a SINET 16 Innovator, OX is trusted by dozens of global enterprises and tech-forward companies. Founded by industry leaders Neatsun Ziv, former VP of CheckPoint's Cyber Security business unit, and Lior Arzi from Check Point's Security Division, OX's Active ASPM platform is more than a platform; it empowers organizations to take the first step toward eliminating manual AppSec practices while enabling scalable and secure development.

INTERESTED IN LEARNING MORE VISIT:
WWW.OX.SECURITY/BOOK-A-DEMO/

oxsecurity