WhiteSource

SECURITY
BOULEVARD

Open Source Security Management in the Age of DevOps

INTRODUCTION

WhiteSource and MediaOps (DevOps.com and Security Boulevard) conducted a survey of more than 400 organizations to understand their policies, processes and tools in managing the risk associated with the use of open source components in their applications.

These days, there's an app for virtually anything. Enabling rapid application development and deployment of these apps is a near-endless body of components, most of which are open source: code, scripts, artifacts and more. But while these components are driving faster development and deployment, they also can be a security nightmare for companies that fail to manage and secure them effectively—a scenario that has proven catastrophic in several high-profile incidents over the last few years.

It is estimated that 66 percent to 80 percent of the code comprising most applications today are actually pre-written components that are assembled by the developer team, who then write custom code to add specific functionalities. These components are used and reused as needs arise. Who is responsible for managing the security of these components? And, with so many apps and so many components, is it even possible to manage their security manually?

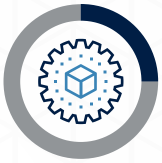
The survey findings reinforce that while some organizations (just under 40 percent) are now trying to actively manage their open source components security, a smaller number are doing so in an automated manner to keep pace with the frequency of deployment. Far too many organizations are still ignoring the risks by not actively monitoring open source component usage at all. Case in point: More than one-third of respondents had no idea that the 2017 Equifax data breach was caused by a vulnerable open source component that had not been updated.

In our analysis of the survey results, we broke out responses into two buckets: those organizations that have adopted a DevOps-DevSecOps methodology and those organizations that have not (IT). The results were striking: In non-DevOps-enabled organizations, any open source security policy and process is left in the hands of InfoSec teams; however, in DevOps-enabled organizations, nearly half have DevSecOps teams working with InfoSec teams to manage open source security. DevOps-enabled groups are far more likely to automate their open source security management with far greater awareness.

KEY FINDINGS

AUTOMATED VERSUS MANUAL

More than 40 percent either did not know whether their manual process detected vulnerable open source components or had no process at all.



25 percent of organizations have an automated process to detect vulnerable open source components.

REMEDIATION OF VULNERABLE OPEN SOURCE COMPONENTS

Almost half of respondents said that severity of vulnerability was the critical factor in determining remediation prioritization.

Only 15 percent remediate more than 75 percent of vulnerabilities found in open source components.

Almost two-thirds of these remediations take place within a week.

WHO IS RESPONSIBLE FOR OPEN SOURCE COMPONENT SECURITY MANAGEMENT?

In DevOps-enabled groups, DevSecOps groups led the management of open source security. Overall, either appsec, information security or DevSecOps were responsible **in about two thirds of organizations.**

About one-third of respondents did not have a specific group tasked with open source component security management in their organization.

RECOGNITION OF OPEN SOURCE COMPONENT MANAGEMENT BEING A SERIOUS SECURITY RISK

EQUIFAX
■ Didn't know
■ Didn't change

More than half of respondents either did not know that the Equifax data breach (during which the PII of millions was stolen) occurred because of an open source vulnerability or said the event did not impact the way their organization monitors open source vulnerabilities.

Overall, concern for vulnerable open source security components was important to **fewer than half of respondents.**

SPECIFIC FINDINGS

OPEN SOURCE SECURITY RISK AWARENESS

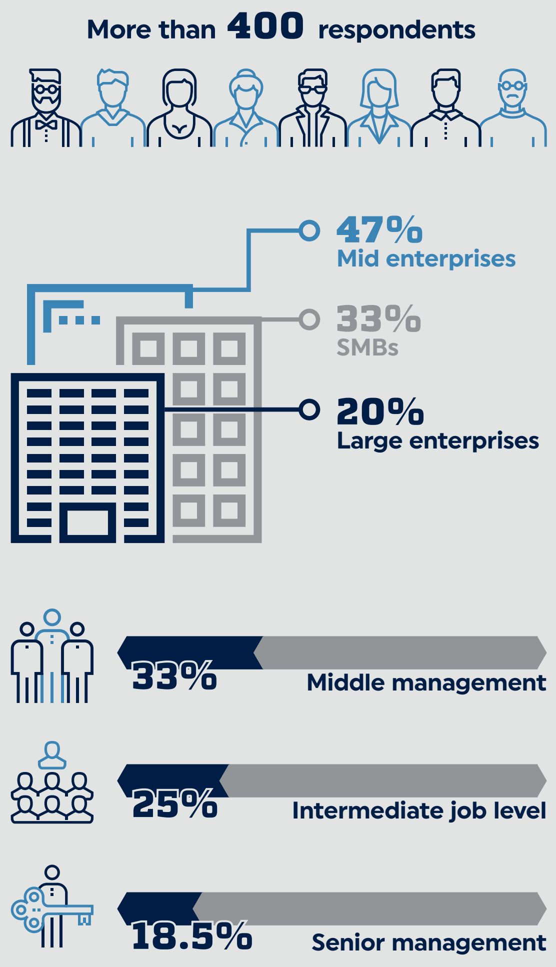
Concern about open source components having known vulnerabilities was overwhelmingly supported by the respondents. **Over 55% of respondents said they were concerned, with another 33% saying they were concerned but were confident they had a process in place that was working to combat this.** Less than 10% said they were not concerned.

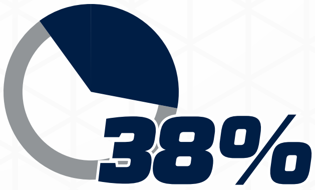
However, this concern and confidence did not translate to actual results. While about half of the respondents said they had an either an automated or manual process in place to detect vulnerable open source components, the other half of respondents said they had either a manual process that they were not sure actually detected vulnerable open source components or they had no process in place at all.

The no process in place for all groups represented fully 20% of respondents and seems to be at odds with the number who are concerned about open source vulnerabilities.

The Equifax data breach, which has become the poster-child for open source component vulnerability due to the sheer number of records affected did not raise awareness as much as one would think.

DEMOGRAPHICS





were unaware that the Equifax data breach was an example of reported open source vulnerability

About one third of respondents were not even aware that the Equifax breach was caused by an open source vulnerability. Another third were aware but said it had not impacted how their organization monitored open source vulnerabilities.

Even taking into account the percentage of those who were aware of (but not impacted by) the breach that were already monitoring open source vulnerabilities, this failure to alter their behavior still shows a surprising, if not staggering percentage of organizations that have not yet learned the lessons of the Equifax breach.

DEALING WITH VULNERABILITIES WHEN FOUND

There was a wide disparity in how organizations deal with vulnerabilities in open source components.

First there was real doubt on the ability of organizations to detect and remediate vulnerabilities in open source components. While 15% said that over 75% of vulnerabilities are indeed found and remediated, slightly over half said that 30% or less of vulnerabilities are remediated.

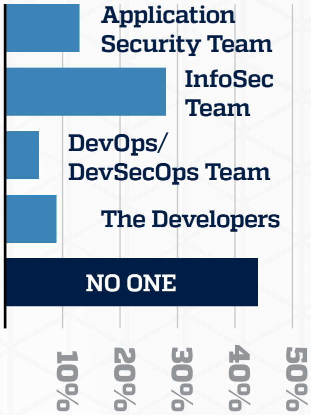
The process to remediate vulnerabilities was pretty evenly split (about 25% each) between (i) open and assigning a trouble ticket to R&D with details and (ii) open and assigning a ticket to R&D with a proposed remediation. Another quarter of the respondents would do research to better understand the impact and severity to decide what (if anything) should be done. About 10% did not have a process in place at all.

The good news is that one half of respondent's organizations did open a ticket for every vulnerability. Also notable is that about 12% reported the vulnerability to DevOps or DevSecOps teams.

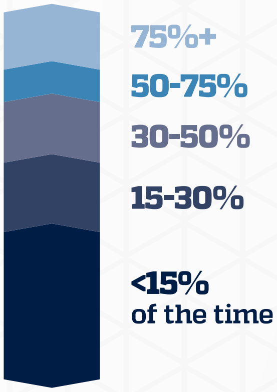
Remediation was prioritized based on several factors. The most common was severity based on CVSS score with about one third of respondents. Next with slightly over a quarter of respondents was prioritizing based on criticality of the project involved. Also notable was that over 15% said the perceived impact of the vulnerability to projects was the factor to determine priority.

In terms of time to remediate there was some good news. Once a vulnerability is discovered, over 50% of respondents indicate time to remediate is would be less than a week.

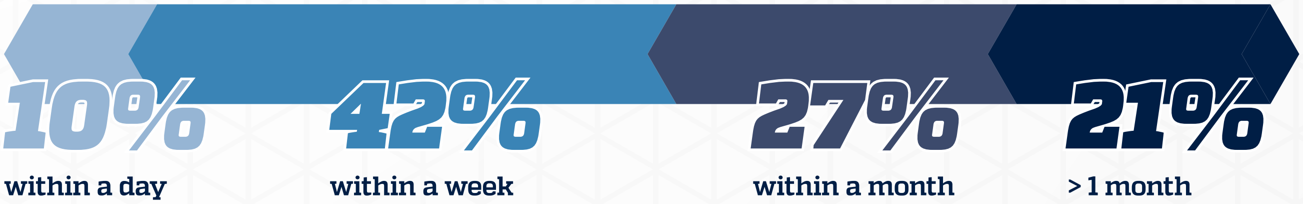
WHO IS RESPONSIBLE FOR DETECTING VULNERABLE OPEN SOURCE COMPONENTS?



WHEN TESTING REVEALS APPLICATION VULNERABILITIES, HOW OFTEN DO THEY GET FIXED?



WHAT IS THE AVERAGE TIME IT TAKES YOUR ORGANIZATION TO REMEDIATE AN OPEN SOURCE COMPONENT THAT WAS FOUND TO HAVE A VULNERABILITY?



CONCLUSION

The findings of this survey clearly show that DevOps- and DevSecOps-empowered organizations generally are much more proactive in managing their open source component vulnerabilities. They are getting their DevOp and DevSecOps teams involved in scanning vulnerabilities. They are automating their open source vulnerability testing. Overall, in non-DevOps and DevSecOps-enabled organizations, awareness of open source vulnerabilities and processes to scan and remediate still lag.

Confidence that the measures adopted to manage components are working and the lag time to remediation still represent critical risks. However, considering the number of organizations that don't acknowledge and recognize the risk, any type of program is better than none at all.

As we have seen in other surveys, including the latest "State of DevOps" reports, the gap continues to widen between high-performing IT organizations that have adopted DevOps/DevSecOps and those that haven't, creating a clear delineation of companies that are managing their open source components for risk and vulnerability and those that are not. And those that are not, unfortunately, are destined to become tomorrow's victims.

"[Open source security management] requires a multi-pronged approach — one cannot simply bring up a single suggestion that will serve as a panacea" said Rami Elron, Senior Director of Product Management at WhiteSource. "It must involve a combination of multiple approaches."

In a DevOps context, all stakeholders are responsible for open source security, while different team members will approach security from different perspectives — all of which offer a synergistic approach. Shared responsibility and awareness should ultimately trump unawareness about open source security risks, which runs rampant today.

"All teams must be willing to share some of their responsibilities, instead of opting for an approach involving delegation of responsibility for open source code, and for security in general, Elron said. "That's a key challenge, really," Elron added.

> ***"Open source security management requires a multi-pronged approach – one cannot simply bring up a single suggestion that will serve as a panacea. It must involve a combination of multiple approaches."***
>
> ***–Rami Elron, Sr. Director of Product Management, WhiteSource***