chainguard

The True Cost of CVE Management in Containers:

How and Why CVE Management Is Painful

John Speed Meyers • Chainguard Labs • February 2024

Executive Summary

Do you like vulnerability? If you're an acolyte of emotional guru Brené Brown and thus believe in the power of emotional vulnerability, there's a good chance you do. If you're a software company that builds or deploys containers, not so much.

Common Vulnerabilities and Exposures (aka CVEs) in containers, at least according to the interviews conducted for this study, are a pain (in the vuln). But how big of a pain? More scientifically, how much time do software professionals at organizations that build or deploy containers spend on CVE management? And why?

To answer these questions, Chainguard conducted ten interviews with software professionals at a range of companies that build or operate containers. The interview questions dealt with the processes and workflows that these professionals use to identify, triage, and remediate CVEs in containers. Many of the questions either involved a request for a time estimate of each step of the process or probed the "why" behind the process or workflow

How much time is spent on CVE management?

Based on these interviews, it's likely that many companies spend thousands of hours or more on CVE management each year. This translates into the work of at least a couple full time staff equivalents per year.

Importantly, it appears that some organizations devote an outrageous amount of time to vulnerability management: two interviewed organizations devoted roughly ten or more full-time equivalent staff members to this activity every year. Additionally, the interviews suggested that the time costs of CVE management can be spread across many teams, silently wasting scarce staff resources.

Why is CVE Management with Containers So Painful?

Key reasons include:

1. CVE remediation time depends on the ease of upgrading and testing software, and upgrading and testing software can itself be painful.

2. Developers frequently pick whatever image they want for convenience without regard to the number of CVEs.

3. CVE management is sometimes a pain experienced by not only one organization, but also by its customers and suppliers.

What Does All This Time Wasted on CVE Management Mean for Your Company?

Philosophically, there are two approaches to reducing the time burden. One approach is to create efficient CVE triage and remediation processes, treating the number of CVEs as given. This is a logical and worthwhile approach, but it's not the philosophy that underpins Chainguard Images.

Chainguard's approach, which emphasizes attack surface reduction, is to produce containers that have low-to-zero known CVEs. This reduces the overall count of CVEs that organizations need to manage, freeing software professionals to pursue more valuable (and more innovative or lucrative) work.

On Being Vulnerable

In your personal life (if you believe in the teachings of wellness guru Brené Brown), emotional vulnerability is the source of joy, creativity, belonging and love. In the world of software (if you believe in the interviews and analysis presented later), CVEs are most assuredly not the source of joy, creativity, belonging, or love. Instead, CVEs and their identification, triage, and remediation ("management") for organizations that build or operate containers are a giant time suck and a source of security risk.

Unfortunately, systematic answers to basic questions about CVE management at organizations that build and deploy containers are mostly missing. Basic outstanding questions include:

- [x] How much staff time do companies that build or deploy containers spend on CVE management?
- [x] And why do these organizations spend so much time on CVE management?

The analysis, based on interviews with software professionals throughout 2023, presents tentative answers to these questions.

Methodology

Before jumping into the results, the reader should know a little about the interview methodology. There are several components of an interview-based study: interview questions, finding interviewees, the interviews, and the analysis. Each component bears upon the findings. The questions for these interviews dealt with the time costs of vulnerability triage and remediation and focused on estimates of the typical time spent on sub-components of those processes by the interviewer and their colleagues. The interviewees were all software professionals that handled CVE management as part of their daily work responsibilities and worked at companies that built or operated containers. About half of the interviewees were current or prospective customers of Chainguard and half responded to a web-based survey form distributed through social media. The interviews were approximately 90 minutes. All interviewees were compensated for their time. Finally, the analysis was straightforward about the annual time costs of vulnerability management given interviewee responses and the identification of common themes across interviews.

How much staff time do companies that build and ship container-based software spend on vulnerability management?

TL;DR Based on these interviews, it's likely that many companies spend thousands of hours (or more) on vulnerability management each year.

The interview research revealed that top-level estimates of the total amount of time spent by staff directly triaging and remediating vulnerabilities are possible, at least approximately. Staff were often able to estimate the time spent on these activities by themselves and closely related colleagues and then extrapolate.

See Table 1 for approximate estimates of the total time spent directly by staff on vulnerability management. For this study, vulnerability management refers to the identification, triage, and remediation of known vulnerabilities (often called CVEs). These interviews focused on estimates of day-to-day vulnerability management and not vulnerability management during periods of crisis like log4shell. The table is arranged from organizations estimated to be spending the most hours on vulnerability management to the least.

TABLE 1.

Estimate of Total Annual Direct Staff Hours Spent on CVE Management by Company

Company Industry	Estimated # Employees	Estimate of Total Annual Direct Staff Hours Spent on CVE Management
Transport & Logistics	10,000s	20,000
U.S. Federal Organization	100s	15,000
Data Products & Services	1,000s	1,250
IT Consulting	100s	1,000
Application Development Platform	100s	1,000
Application Development Platform	1,000s	200
Developer Tools	100s	150
Observability	10s	100

See appendix 1 for detailed descriptions of these estimates. Estimates were only possible for eight organizations and interviewees.

Several trends stand out.

First, there are some companies and organizations dealing with an immense amount of vulnerability management. One European transport and logistics company spends the hourly equivalent of 10 full-time employees on vulnerability management each year. One U.S. federal organization also spends an extraordinary amount of staff time on vulnerability management. It's worth noting that these organizations are different from the others. The transport and logistics company has many employees (~100K) and appears to be particularly security-conscious (more on the exact form of their security posture later). The U.S. federal organization is a government organization and is subject to particularly onerous security and compliance processes.

Second, there's a second tier of companies that annually expends approximately one thousand hours of staff time on vulnerability management.

Third, there are several companies that report only spending a few hundreds of hours on vulnerability management per year. For instance, one early stage startup simply didn't prioritize security given that its most pressing concern was cash flow.

But there is a key caveat. These interviews revealed that staff often don't have a complete picture of how much time is spent on vulnerability management. There are too many staff across too many teams for there to be a simple tally. For instance, for two interviewees, despite their involvement in day-to-day CVE triage and remediation, estimating the number of hours that their organization devoted to these activities was simply impossible; they had so little confidence that they wouldn't even hazard a rough guess. Other interviewees were able to offer an approximate tally by cobbling together their knowledge of CVE-related activities across different teams.

The teams most commonly performing CVE-related activities were security engineering, platform engineering, and software development.

An organization that is worried about death-by a-thousand-CVEs and the "silent" costs of CVE management should start by investigating the activities of these teams. But there are other teams that also sometimes bear the burden of vulnerability management, including testing or quality assurance teams and compliance teams.

It's worth mentioning one methodological caveat. Measuring how much time professionals, including software engineers, self-report on a task via interviews or surveys is tricky methodological territory. Diary studies or ethnographic methods would be a worthwhile complement to this research and could enrich the software industry's understanding of the time burden of vulnerability management.

List of Specific Vulnerability Management Pain Points

After conducting the interviews and listening to each one for a second time, we created a list of specific vulnerability management pain points. This list is in the subjective order of "more" pain to "less" pain and attempts to explain, at least partially, why modern organizations that build or operate containers spend so much time on vulnerability management.

1

CVE remediation time depends on the ease of upgrading and testing software. For organizations that have low test coverage, require extensive manual testing, slow builds, or very outdated dependencies, the burden of fixing vulnerabilities, even via minor version bumps of a dependency, is high. Think of this as the "DevOps" view of vulnerability management. If you're bad at DevOps (and the majority are), vulnerability management is a particularly big pain.

2

Developers pick whatever base image they want. In the language of economics, this creates negative externalities for other roles in a software company. Developers sometimes (often, in fact, according to platform engineers) pay little heed to the number of known vulnerabilities in base images. The only criteria employed is whether the container enables the application to perform its functional role.

3

CVEs create a burden for customers and container providers, too. Two interviewees reported that vulnerabilities in the images they ship created time burdens not only for their own organization but for other organizations who use their products or services. In one interviewee's case, their customers often scan their images and then write to that company complaining about detected vulnerabilities. In the case of another company, their container provider now spends significant time on vulnerability management trying to provide this company CVE-free containers.

- ***It is hard to estimate the time needed for triaging a particular CVE.*** There are so many variables. Estimates ranged from 20 minutes for simple patch version upgrades to a couple weeks of engineering work when needing to migrate a codebase to a new major version of a dependency. This is an engineering management nightmare. In short, it's lumpy and unpredictable work.

- ***Shifting left doesn't fully make sense for vulnerability management.*** Since new vulnerabilities emerge daily, it's not a one and done process like avoiding writing a SQL injection into your first party code. Therefore, there is a need for vulnerability management "in real time."

- ***False positives were sometimes extremely annoying to the interviewees, so much so that there was notable interest in the vulnerability exploitability exchange (VEX) from several interviewees.*** VEX is a standard for creating documents that communicate whether one or more products is affected by one or more known vulnerabilities.

- ***There are too many steps in CVE triage.*** There are lots of disparate sources of information and it's easy to end up with fifty browser windows open. This is because each individual source of information about a given CVE is often of relatively low-quality, failing to explain much. The software professional doing triage is therefore forced to look at many different sources of information, putting together the puzzle pieces of a CVE.

- ***Adjusting the severity of a vulnerability's score for your application's particular environment is not always easy with current tooling.*** Additionally, whether an exploit already exists for a vulnerability and whether that vulnerability is known to have been exploited in the wild are desired information.

- ***Software teams often roll their own database and user interface solutions to store and aggregate vulnerabilities.***

Minimizing CVEs Isn't Just About Time Savings

The interviews on the time burden of CVEs also revealed that reducing CVEs isn't only about avoiding toil. In addition to the general desire to avoid compromise, there are at least three other reasons that interviewees identified as key reasons to reduce CVEs.

> **First, reducing the overall count of CVEs, and especially reducing false positives, ensure that security teams and developers can focus on true positives, vulnerabilities that are legitimate and of immediate importance. Reducing CVEs, in other words, is a way to increase the signal-to-noise ratio.**

Second, reducing CVEs, even low or medium severity CVEs, reduces the dangers of "exploit chaining," combining exploits, even exploits of low severity vulnerabilities, to achieve compromise. The organization that spent the most time on vulnerability management did so largely due to the belief that fixing low and medium vulnerabilities was important because of the danger of exploit chaining.

Third, managing vulnerabilities is not simply a time cost, but it is a distraction from other core tasks that software teams need to focus on. In other words, it breaks the flow for software professionals, reducing soft developer productivity.

What Does All This Time Wasted on CVE Management Mean for Your Company?

When confronting the problem of CVE management, companies can choose one, both, or none of the two options.

The "efficiency" approach emphasizes efficient CVE triage and remediation processes, treating the number of CVEs as given. This is a logical and worthwhile approach. This approach prioritizes technical solutions that automatically detect false positives or do other forms of reachability analysis.

There's an alternative approach, one that does not treat the number of CVEs as a given, and instead seeks to radically reduce the CVE burden. This might be called the "CVE zero" approach. This approach, which emphasizes minimalism, attack surface reduction and rapid patches, underpins Chainguard Images. Chainguard's approach is to produce containers that have low-to-zero known CVEs. This reduces the

overall count of CVEs that organizations need to manage, freeing software professionals to pursue more valuable (and more lucrative) work.

Of course, organizations can pursue both approaches. And, sadly, organizations can also pursue neither approach, letting the CVE time burden problem fester, sapping developer productivity and diverting scarce human talent to an often tedious task.

In short, CVEs are a pain. This research can help organizations know how big that pain is and why it is so big. The next and unanswered question is: what is your organization going to do about it?

Visit images.chainguard.dev to try a virtually 0 CVE Chainguard Image today.

Appendix 1

Back-of-the-Envelope Math for Time Organizations Spend Annually on Vulnerability Management

Transport & Logistics Company
Interviewee estimated 36 software teams. 12 hours per week for each team. 36 * 12 = ~430 hours per week.
430 hours/week * 50 weeks = ~20,000 hours.

U.S. Federal Organization
15 security engineers performing 20 hours a week of CVE triage. 300 hours/week * 50 weeks = ~15,000 hours.

Application Development Platform
Interviewee estimated 2-4 per week for each of two security engineers and 15 hours from developers doing remediation per week. 20 hours/week * 50 weeks = 1,000 hours.

Data Products & Services
Interviewee estimated ~25+ hours a week for whole company. Two hours per day from security team + fix times for devs. 25 hours/week * 50 weeks = 1,250 hours.

Application Development Platform
Interviewee estimated that triage and remediation takes a couple days a month. 16 hours/month * 12 months = ~200 hours.

Observability Company
Interviewee estimated only a couple hours a week. 2 hours/week * 50 weeks = ~100 hours.

Developer Tools Company
Interviewee estimated 2-3 hours per week. 3 hours/week * 50 weeks = ~150 hours.

IT Consulting Company
Interviewee estimated 5-6 hours a week per platform engineer. 3-4 platform engineers doing this activity each week. 20 hours/week * 50 weeks = ~1000 hours.

Thank you to Lewis Denham-Parry, Patrick Flynn, Garry Ing, Narayan Iyengar, Kirby Koo, Dan Luhring, Tracy Miranda, John Osborne, James Rawlings, Jed Salazar, Kaylin Trychon and many others.