ShiftLeft

2022 AppSec Shift Left Progress Report

Introduction

In the shadow of a global pandemic, 2022 began with much of the security and development communities still reeling from the widespread Apache Log4Shell vulnerability. Its disclosure in December 2021 highlighted how dependent today's software world is on open source projects, and how difficult it is to find everywhere an organization is impacted by a zero-day vulnerability.

From an enterprise security perspective, application development, security, and DevSecOps teams have worked to implement processes that make their release cycles faster and applications safer.

Despite the macro environment and its associated uncertainty, we did find some positive trends in the application security space that we'll highlight in the following pages.

> **There is a pressing need to implement more rigorous and predictable mechanisms for ensuring that products function securely, and as intended.**

U.S. President Joe Biden

Executive Order on Improving the Nation's Cybersecurity

In this report, we'll evaluate how the changes and advancements in static application security testing (SAST) and intelligent software composition analysis (SCA) have helped development and DevSecOps teams work better together to fix security issues faster.

A note about the statistics and data in this report

Our 2021 Progress Report featured data from May 1, 2020 through April 20, 2021. For comparison purposes, we have based this report's data on the same time range, May 1, 2021 through April 20, 2022. For generalized data, we have used lifetime data based on aggregated customer usage of ShiftLeft CORE.

Key Findings

76%

new vulnerabilities fixed within 2 sprints

The longer that issues persist in-the-wild, the more risk they often pose to security and the business. That's why AppSec and Security are measured by their mean time to remediate (MTTR). We see that scan frequency is increasing (daily scans are up 68% over 2021) which provides results of what needs to be fixed faster and more often to dev teams.

It's also the driver behind *shifting left*. The philosophy is to move testing and fixing into the development process so that devs and AppSec can secure code faster. In the past year, we've seen our customers really embrace and operationalize *shifting left*, which has allowed them to fix new security issues within a sprint or two of discovery.

97%

OSS vulnerabilities to be addressed

Security has often been a bottleneck to software development, creating more work for engineers and adding more days to software release timelines. But not every vulnerability that security tools put on a list needs to be fixed.

With ShiftLeft, customers can see a 97% reduction in the number of open source vulnerabilities that must be addressed, with only 3% being attackable.

37%

Mean Time To Remediate (MTTR) new vulnerabilities

The measures most often used by security teams when dealing with vulnerabilities are the mean time to detect (MTTD) and more importantly, mean time to remediate (MTTR)

ShiftLeft CORE customers saw a 37% improvement in MTTR from 2021 to 2022, down from 19 days to 12 days. This reduction can be attributed to more frequent, earlier testing that allows devs and security teams to resolve issues more rapidly.

90sec

Median Scan Time

Legacy SAST solutions that took hours or days to scan an application stood in the way of developers getting their work done. And that meant wasted time and issues left in the dust as technical debt.

ShiftLeft CORE has been built to quickly analyze code and dependencies and match the speed at which developers code, with a median scan time of 1 minute 30 seconds.

4%

Log4j found in customer code that was vulnerable

While any zero-day often results in a barrage of executives and board members calling CISOs asking, "Are we affected?" not every zero-day needs to add gray hairs to the security team.

ShiftLeft made the conversation about Log4j and Log4Shell easy, and overall, we found that 96% of Log4j in use across our customer base was not vulnerable to the Log4Shell zero-day. Instead of causing panic, we helped teams find and quickly and easily prioritize the 4% where versions *were* vulnerable.

The Measure of AppSec:

How quickly things get fixed

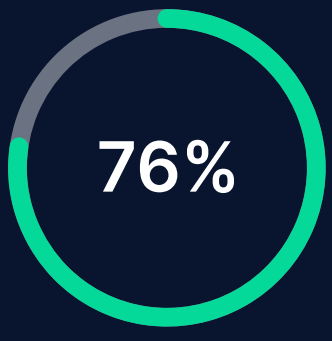
Development, AppSec, and DevSecOps teams continue to struggle to work together cohesively and efficiently to reduce enterprise risk. In order to do that, there needs to be a shared understanding of the goals of the programs and an efficient way to prioritize the shared work to resolve security issues.

The reality of development is that the earlier and faster that issues are found, the easier it is to fix them. This is the essence of *shifting left*.

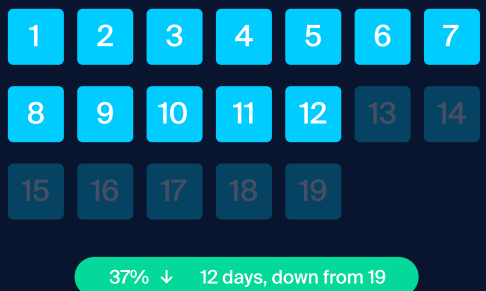
Code is fresh in the minds of developers. A different method can be used to achieve the same result in the application. And issues fixed mean less technical debt to deal with later.

This “scan early, scan often, fix faster” mentality has become the mantra of ShiftLeft customers, and it has produced excellent outcomes for them. Three-quarters of all new vulnerabilities found by ShiftLeft are fixed within 2 sprints.

A metric used by most AppSec teams is how quickly they’re able to identify and fix new vulnerabilities. It’s often illustrated by the mean time to remediate (MTTR). From 2021 to 2022, we saw a 37% reduction in MTTR across all of our customers, from 19 days to just shy of 12.



76% new vulnerabilities are fixed within 2 sprints



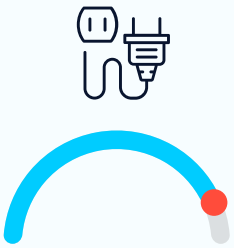
Reduction in Mean Time To Remediate (MTTR) new vulnerabilities

The *Shift Left* requires Culture, Commitment, and Collaboration

Of course, fixing 76% of all new vulnerabilities in 12 days is the *average that we saw across our customers.*

What's remarkable is that we continue to see organizations where dev and AppSec teams are tightly aligned fixing *more vulnerabilities, faster than the average.*

But, that shift can only happen when you have the people, processes, *and* technology to make it work. Here are a few examples of how the culture, relationships, and shared commitment to security and shifting left affected the outcomes in a few of our customers:



90%

new vulns fixed in 15 days

Customer

A large, multinational utility company

Environment

Microservices and smaller applications with an average of 4,000 lines of code per app (JavaScript-heavy)

DevSecOps Relationship

Very strong, with security testing built into dev workflows

This customer's AppSec team is scanning applications daily, returning those results to their development teams in regular team syncs, and are fixing 90% of them within 15 days. The remediation of new vulnerabilities is also built into their sprint planning, allowing them to fix nearly all vulnerabilities within the sprint that they're found.

95%

new vulns fixed in 5 days

Customer

A SaaS-based cybersecurity company

Environment

Cloud-native microservices

DevSecOps Relationship

Strong, with product security tied to AppSec.

Unsurprisingly, our cybersecurity customer has a laser focus on releasing secure software. This alignment and strong drive on the part of developers and their AppSec and product security teams has led them to resolve 95% of new vulnerabilities in only 5 days, on average. This is a testament to the speed and accuracy of their scans and efficiency of their teams.

63%

new vulns fixed in 21 days

Customer

A multinational logistics and transportation company

Environment

Monolith and microservices

DevSecOps Relationship

Weakening with turnover, but rebuilding under new leadership

This customer experienced some of the negative business outcomes of the pandemic, and we saw the development and AppSec teams go through turnover that led to their efficiency changing. The leadership of both teams changed, as did the dynamic between their teams. Last year, we saw that they were fixing 91% of their vulnerabilities in 19 days (on average). This year, we've seen their fix rates dropping to 63% fixed within 21 days (on average).

Most common application vulnerabilities for 2022

As a guideline and best practice, the Open Web Application Security Project (OWASP) publishes a list of what they believe are the most impactful vulnerabilities. Updated mid-2021, we saw some changes. As much as things change, they also stay the same.

From our report in 2021 to this year, we saw many of the same vulnerabilities in the top 10, but there were a few changes:

Most Common Vulnerabilities 2021		Most Common Vulnerabilities 2022
1	Cross-site Scripting (XSS)	1 NEW Log forging
2	Header injection	2 ↓ Cross-site Scripting (XSS)
3	SQL Injection (SQLi)	3 — SQL Injection (SQLi)
4	Remote code execution	4 ↑ Deserialization
5	Deserialization	5 ↑ Prototype pollution
6	Directory traversal	6 — Directory traversal
7	Prototype pollution	7 NEW Server-side Request Forgery (SSRF)
8	Insecure cookie	8 NEW Weak Hash
9	Mail injection	9 NEW Header injection
10	Sensitive environment variable	10 NEW NoSQLi

Additional Resources: BLOG: A Guide to the OWASP top 10

Attackability is the path to better prioritization

Fixing all vulnerabilities should not be the goal of a security team–there will always be more vulns than time and resources to resolve them. Instead, teams must fix those which contribute real risk to the application and its underlying data and processes.

The application vulnerabilities we listed in the previous section can be remedied by applying security best practices. But they need to be tested, found, and shown to the developer as early as they can be in order to help resolve them quickly and educate on how to avoid making those mistakes in the future. The answer, here, is to scan code more frequently to provide just-in-time feedback to the developer.

68%

Increase in applications scanned daily in 2022

It is crucial to find and fix these vulnerabilities in code because threat actors use them as the first step in attacks. Hackers can then gain access to underlying open source software and widely known CVEs.

There are hundreds of thousands of vulnerabilities that exist in the CVE database already, and it has been growing by the tens of thousands every year. So what should an organization do to keep up? Many begin by analyzing where they are impacted to answer the key questions: What applications now in use have this vulnerability? What do those applications do? What data do they touch? How "severe" is this vulnerability?

A question that's not asked as often as it should be, perhaps because many security teams cannot answer it, is:

" Is the vulnerability actually reachable by an attacker? "

If the answer to this question is no, then there is often nothing that should be done. The vulnerability isn't attackable, so remediation will have zero impact on risk. Deprioritize it, and focus on others.

Whether or not a vulnerability is attackable should inform your entire prioritization model. In fact, it will significantly reduce the workload of your AppSec and development teams by as much as 97%.

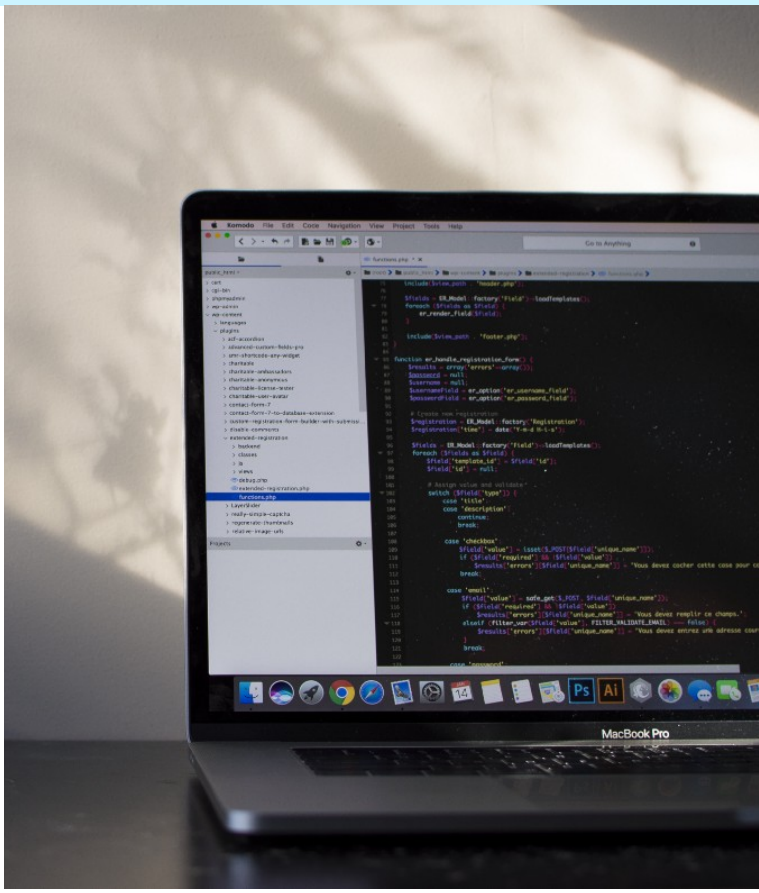
97%

ShiftLeft customers saw a 97% reduction in false positive library upgrade tickets when you consider attackability.

We saw that amongst all of the vulnerable packages in use by applications across our customer base, only 3% of “critical” vulnerabilities were actually attackable.

This chain of custom code tied to open source software represents the full data flow through the application. Without this, siloed testing and results are often misleading and full of false positives.

ShiftLeft exists to help organizations improve the security of their applications by testing applications and their dependencies holistically and without slowing down the development process. The best outcome of using ShiftLeft is increased scan frequency of all or most of an organization’s app portfolio, and more issues being fixed, faster.



Additional Resources:

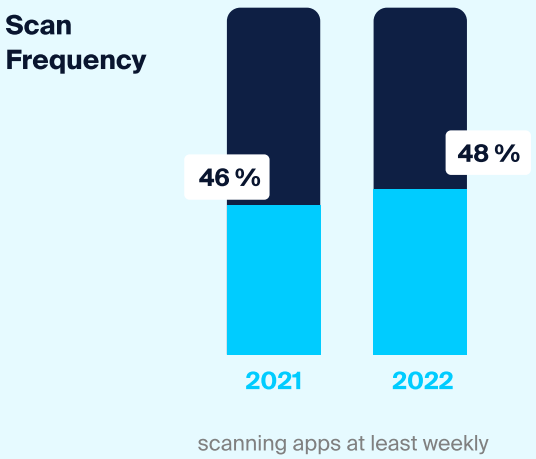
[BLOG: Modern AppSec Tools Must Focus on Attackability, Not Chasing Bugs]

Things Are Getting Faster, Not Slowing Down

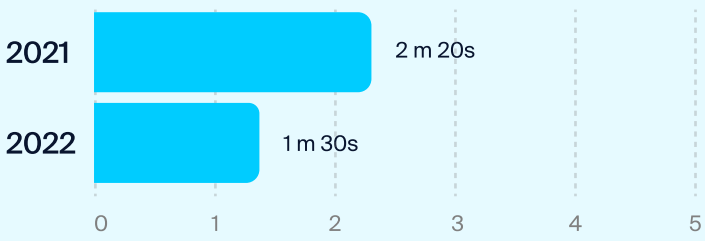
Application teams have embraced new technologies like microservices and cloud, and new processes like DevOps and DevSecOps as a way to innovate faster. And testing needs to keep up the pace.

We saw an increase in applications being scanned in 2022 across all of our customers, and nearly all of those were being scanned at least weekly.

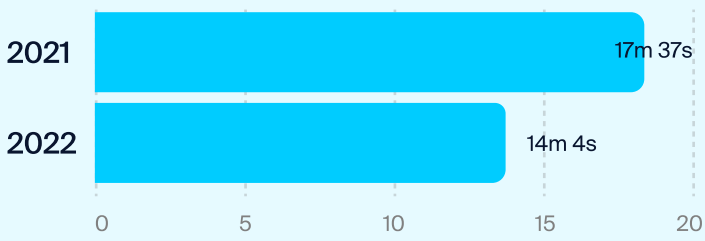
This goes to support teams scanning early and often, and multiple times throughout a typical sprint.



Time to Scan: **Average across all apps**



Time to Scan: **Compiled Languages (95th %ile)**

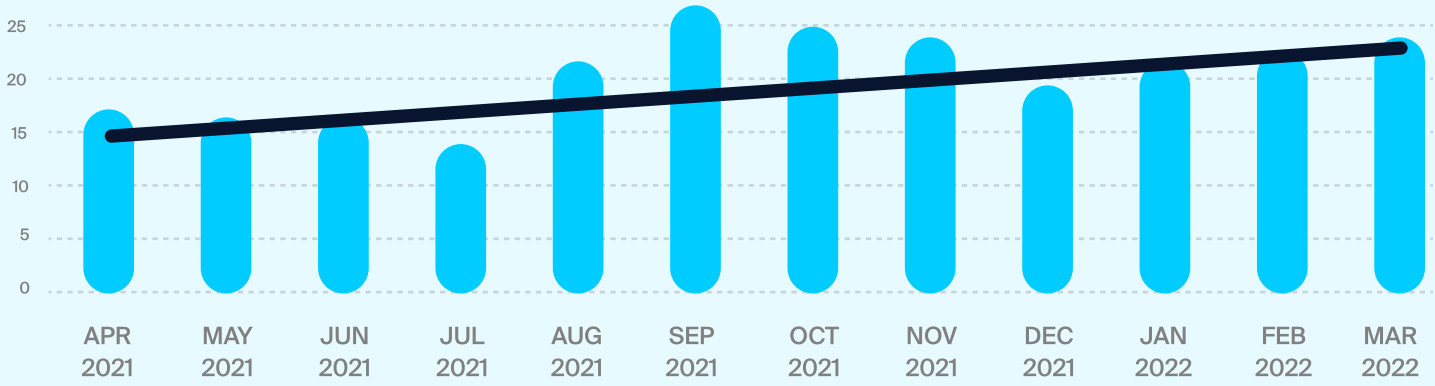


> **“We used to have an application that we literally couldn’t scan with another SAST tool, it was too big. But not for ShiftLeft.”**
>
> **CISO**
> Global 500 Airline

While we’d like to attribute the decrease in scan time to our technology just getting 43% better than it was a year ago, there’s actually an additional reason. We saw the average size of applications in terms of lines of code go down. This aligns with more organizations moving to microservices and smaller, more modular applications.

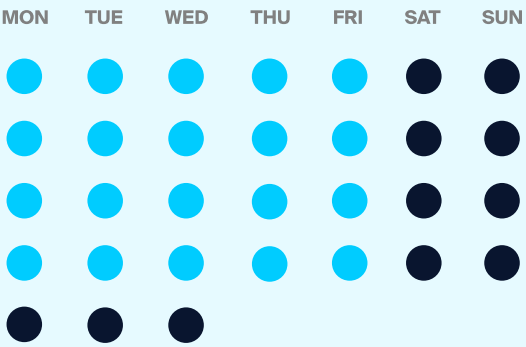
Over the last year, we’ve introduced language support for JavaScript, Typescript and Python, which are typically smaller applications. But also, our analysis technology got faster with regard to Java and .NET, which make up approximately 40% of applications that our customers test.

Average # of Scans Per App



We see scan frequency trending upward in 2022, where the average number of ShiftLeft scans per app, per month, was 20. This aligns roughly with a daily average scan cadence if you consider weekdays as active work days where code is being committed and pull requests created.

20
scans per app, per month



Additional resources:
[BLOG: Best Practices for AppSec in the Cloud]
[BLOG: Key Considerations When Choosing SAST Tools]

What's In Your App?

The concept of understanding the make up of an application–which code is custom, which open source libraries were used–has always been important. With recent supply chain attacks and United States Presidential Executive Orders, it's now absolutely critical. A software bill of materials (SBOM) helps by providing insight into what open source software is used in an application.

On December 9, 2021, Apache disclosed that the Log4j 2 utility contains a critical vulnerability that allows unauthenticated remote code execution (RCE). Log4j is a popular open source logging package included as a dependency in a lot of major frameworks, such as Apache Struts2.

The Log4Shell vulnerability allows attackers to run arbitrary code by sending requests to servers running vulnerable versions of Apache Log4j, and it was given the CVE number CVE-2021-44228.

This vulnerability, due to the widespread use of the framework, set most IT and security teams into crisis mode. The challenge for most organizations was being able to quickly understand where Log4j was being used in their applications and whether they were at risk.

With ShiftLeft, our customers were able to instantly identify whether and where they were using a vulnerable version of Log4j. In our analysis of thousands of applications which used the Log4j framework, we only found that 4% of them were actually vulnerable. This helped our customers focus on that small portion of apps which represented a risk to the business.

Vulnerability management and application teams

Many development teams are organized by the application that the team is responsible for. Across the applications ShiftLeft has analyzed, we found that only 1 in 3 apps actually had an attackable vulnerability. Thinking about it like this, that means that 2 of 3 application teams are actually doing unnecessary work if they're spending time upgrading libraries.

2 of 3 application teams are actually doing unnecessary work

ShiftLeft helps these teams understand whether their applications have attackable vulnerabilities–the ones which present real risk and must be fixed–or if they can simply add fixes and upgrades to their backlog to be resolved as time allows.

Additional Resources

BLOG: Who is responsible for open source security?
BLOG: Secure Software Summit: The State of OSS Security
BLOG: Detecting Log4j in your Apps

Conclusion

The end goal of any application security and DevSecOps program is to enable developers to move faster without compromising on security and compliance. What we've seen this year is that teams are, indeed, scanning more frequently, fixing vulnerabilities faster, and becoming more focused on the efficiency and effectiveness of their work.

Speed and accuracy are requirements of any SAST and SCA tool in use by an organization. False positives and "unattackable" vulnerabilities waste developer time and break down confidence in security tools, impacting morale and security poorly.

Here at ShiftLeft, we continue to hear from teams about the challenges that they face in order to release software at scale. To help them overcome those challenges, our platform's three main goals are to:

Bridge the AppSec Divide

Scan modern applications in minutes instead of hours or days so your developers can keep coding. When critical issues are found early, they're easy to fix while code is still fresh in the developer's mind. With fast, accurate results from ShiftLeft, you can build trust and break down siloes between devs and security.

Fix the Risk that Matters Most

ShiftLeft automatically prioritizes vulnerabilities that are reachable by attackers and pose the greatest risk to the business. Quickly remediate the most serious vulnerabilities to reduce the available attack surface, decrease security debt and prevent attackable code from being deployed.

Release Secure Code at Scale

Achieve faster time-to-market by securing more code in less time. Accelerate digital transformation without slowing down software development. ShiftLeft customers scan their applications more often and fix security issues faster than their peers. No other platform helps organizations realize more continuous DevSecOps improvement.

We hope that you've found this information useful and inspiring. If you're just getting started on your DevSecOps journey, or if you're already knee-deep in the transformation, reach out to us. Let us show you how our code security platform can help you release secure code at scale.

-Team ShiftLeft

www.shiftleft.io
(877) 331-9092 HQ: Santa Clara, CA