STATE OF MODERN APPLICATION SECURITY

Insights From 400+ AppSec Practitioners

TROMZO | www.tromzo.com

A word from our CEO

Harshil Parikh
CEO, Tromzo

DevOps culture and rapid cloud adoption has developers shipping code faster than ever and security is struggling to keep up.

This is especially true in modern organizations where developers are using CI/CD systems, which make up the majority of companies today. Outnumbered 100 to 1 and stuck spending all their time on low-value manual work, many application security professionals are frustrated and overwhelmed.

At Tromzo, we've been spending a lot of time thinking about what's holding application security back. We wanted to know – what are the biggest challenges in appsec today? What's the relationship like between developers and security? What would make appsec programs more effective? To answer these questions and more, we commissioned a survey of over 400 AppSec professionals for our first annual State of Modern Application Security Report.

Our goal is to help CISOs and security leaders better understand what's preventing their application security programs from scaling so they can keep up with the fast pace of modern software development.

We hope you find these findings helpful as you develop your 2022 Application Security Strategy.

Application security posture confidence remains high, yet 67% have experienced an incident in the past year. The juxtaposition of these two facts highlights the need for better visibility into what's happening between AppSec and Development teams. If security teams are confident that security precautions are being implemented in the development environment when, in fact, they are not, there exists an increased risk of a severe security incident.

40% have 5,000 or more security vulnerabilities that need to be addressed, and that rate has quickly increased over the past 12 months. This explosion in vulnerabilities is a universal problem that AppSec teams must address sooner rather than later.

42% are seeing more false positives and noise than ever before. False positives and alert noise are by-products of security tools that lack context and are deployed during the quality assurance stage rather than end-to-end approach.

Reducing friction between developers and security would have the most significant impact on improving the application security program. Any attempt to minimize conflict between security and a development team that doesn't include shifting security left is not likely to succeed. As long as security is a gate implemented at the end of the development cycle, it will cause developer friction and cause delays in deploying secure applications.

Developers ignoring security is the greatest challenge. This problem will only be solved by a platform that enables AppSec teams to keep pace with modern development and scale their application security program.

Integrating security checks throughout the SDLC would dramatically improve the relationship with developers. We must have from security gates to security guardrail that empower developers to develop secure code.

Beginning on September 23, 2021, we surveyed 403 US-based security leaders who work at organizations where the developer team uses CI/CD systems. The survey was conducted online via PollFish using organic sampling. Learn more about the Pollfish methodology here.

In order to provide greater context around these findings, here are more details on who we surveyed. In total, 406 qualified individuals completed the survey. Participants included a mix of job levels, company sizes, and industries.

Industries

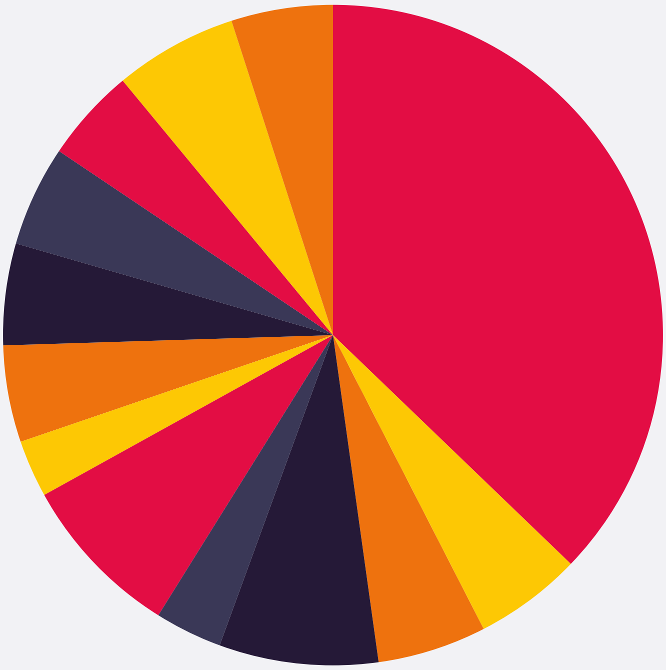
By far, the largest group of security practitioners we surveyed (37.2%) work for technology companies. This group was more than four times larger than the next largest group, utilities/energy companies (8.1%). Other significant industries represented in our survey include healthcare (7.6%), retail (6.2%), insurance (5.4%), and finance (5.2%).

Job Titles

While security job titles are not always indicative of precise job responsibilities, the responses to our job title question confirm that we hit the mark for gathering the information we were after. The perceptions of these security practitioners accurately reflect what it's like to shoulder the responsibility of providing security in a modern development environment.

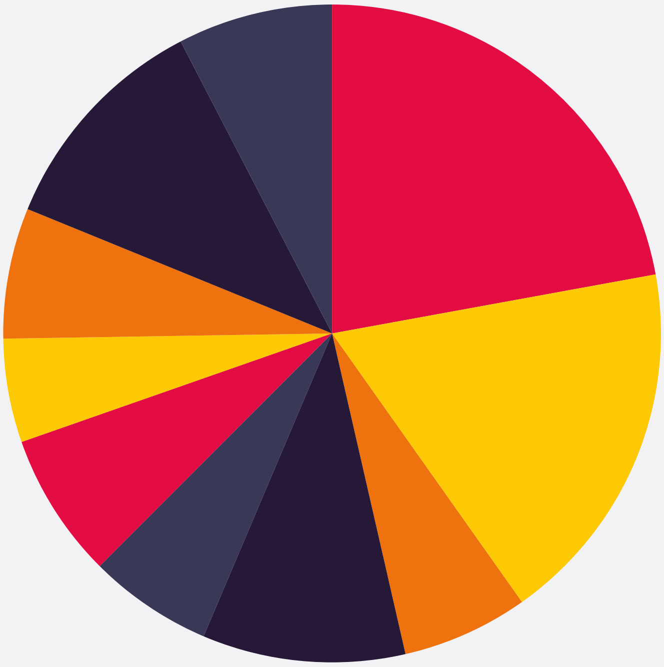
The largest share of respondents (22.2%) are security analysts, followed by security engineers (18%). The only other double-digit group being technical security program managers (11.3%), and there is also a smattering of consultants and pentesters.

What Industry Does Your Company Operate In?



INDUSTRY	%
● Technology	37.19%
● Finance	5.17%
● Insurance	5.42%
● Healthcare	7.64%
● Hospitality	3.45%
● Utilities/Energy	8.13%
● Federal	2.71%
● State/Local Gov.	4.93%
● Education	4.93%
● Manufacturing	4.93%
● Services	4.68%
● Retail	6.16%
● Other	4.68%

What best describes your title?



JOB TITLES	%
● Security Analyst	22.17%
● Security Engineer	17.98%
● Security Architect	6.40%
● Security Administrator	9.85%
● Penetration Tester	6.16%
● Security Specialist	7.14%
● Security Consultant	5.17%
● Security Manager	6.16%
● Technical Security Program Manager	11.33%
● Other	7.64%

**Now, with context around who our respondents were,
Let's take a closer look at what we uncovered.**

Program Confidence And Effectiveness

Application security is racing to keep up with the pace of software development. In this first section, we explore how AppSec professionals feel about the effectiveness of their application security programs.

PART #1

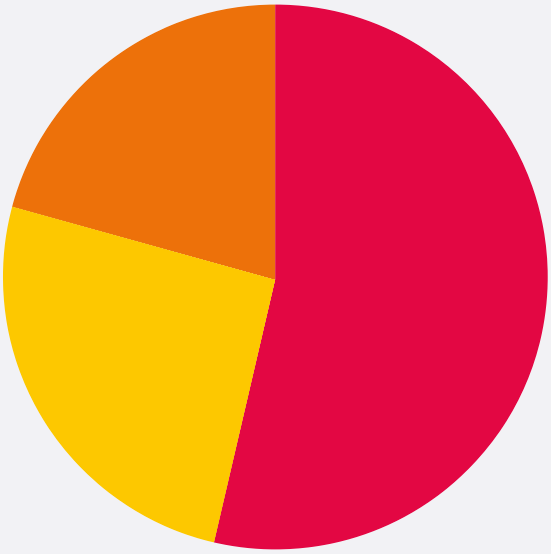
52% are highly confident in their organization’s application security posture.

A slight majority (52.2%) of AppSec practitioners are highly confident in their security posture, and only a fifth (20.4%) are not very confident at all. The results of this survey question may seem surprising, especially in light of the fact that over two-thirds (67.7%) of the respondents have experienced a security incident in the last 12 months, as we’ll see below.

High confidence in their organization's application security posture while at the same time experiencing an incident in the last 12 months makes sense when viewed through the lens of a security analyst that is detached from the development process.

This could be happening because, from their perspective, they have the latest tools and have fed the developers tons of alerts about the code being generated. The organization's application security posture, from that vantage point, looks pretty good, but it lacks context. Too many of the security findings the analysts delivered to the developers in bulk were never remediated, and the vulnerabilities continued to exist long after the security team noted them.

Overall, how confident are you in your organization's application security posture?

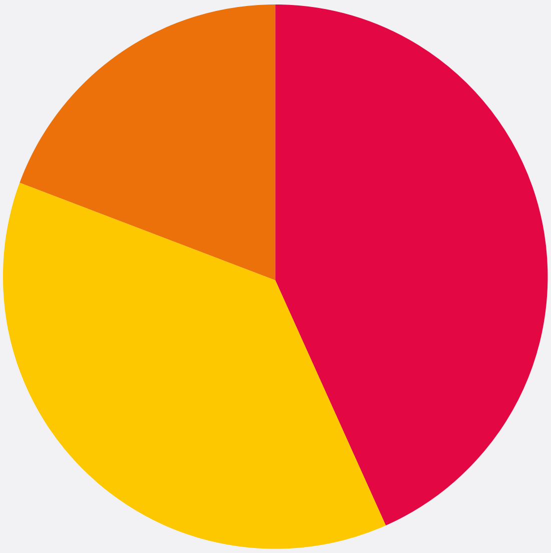


ANSWERS	%
● Highly Confident	52.22%
● Somewhat Confident	27.34%
● Not Very Confident	20.44%

Eight out of ten would be surprised to see their company in the news for an incident related to an application vulnerability.

The answers to this survey question confirm the high confidence security teams have in their tools and work. That only 19.5% of security professionals would not be shocked if their company made the news headlines because of an application security vulnerability may reflect an emotional survival mechanism for the remaining 80.5%. How demoralizing would it be to go to work every day, just waiting for the news vans to show up and put a spotlight on your inability to protect your company from a breach?

How surprised would you be if you saw your company in the news for a breach related to a vulnerability in your applications?

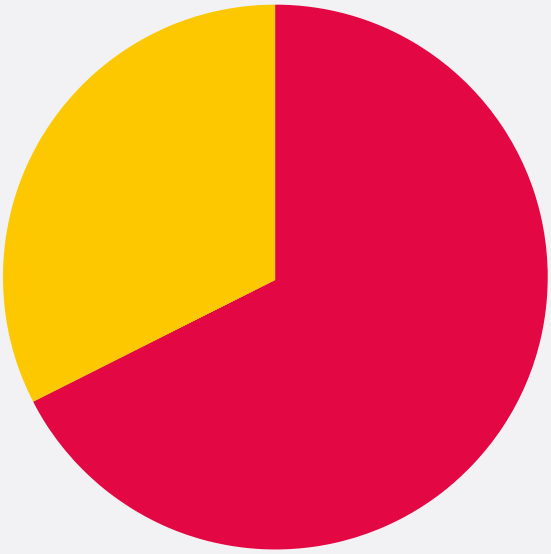


ANSWERS	%
● Very Surprised	44.83%
● Somewhat Surprised	35.71%
● Not Very Surprised	19.46%

Even with high confidence in their security posture, 67% have experienced a security incident in the past 12 months.

As the idiom goes, the proof is in the pudding. Regardless of modern tools wielded by highly educated and competent security teams, security incidents happen every day. It's not always the tools nor the practitioners; it is often the process that fails application security. Security in a silo and as an afterthought is better than having no application security. Still, it invariably causes developer friction and delays delivering secure applications.

Have you experienced a security breach in the past 12 months due to a vulnerability in one of your applications?



ANSWERS	%
● Yes	67.73%
● No	32.27%

Drowning in Vulnerabilities (and Only Getting Worse)

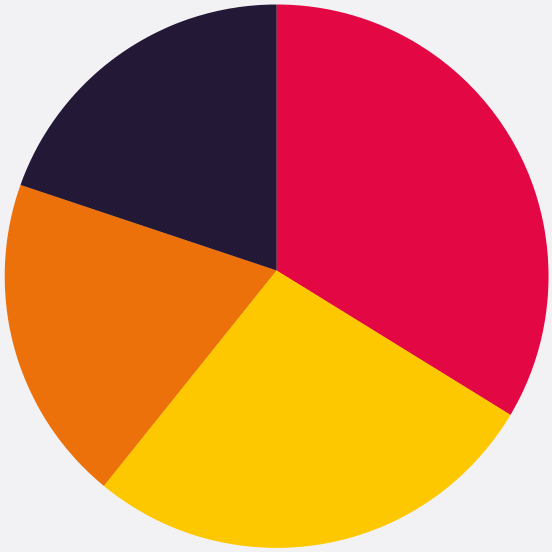
Ask any AppSec professionals and they will tell you what their days are like: they are drowning in a sea of false positives and noisy results as they work to try and prioritize the issues that actually matter. In this section, we'll quantify how big of a problem this really is for AppSec professionals.

PART #2

40% have 5,000 or more security vulnerabilities that need to be addressed and that rate has quickly increased over the past 12 months.

Although the most often-picked answer to our question about how many application security vulnerabilities they currently have that need to be addressed was "less than a thousand" (31%), it is of little comfort. Our results show that 28.8% of teams have between 1,000 and 5,000 vulnerabilities, 20.2% of teams say they have at least 5,000, and another 20% have more than 10,000 unresolved vulnerabilities.

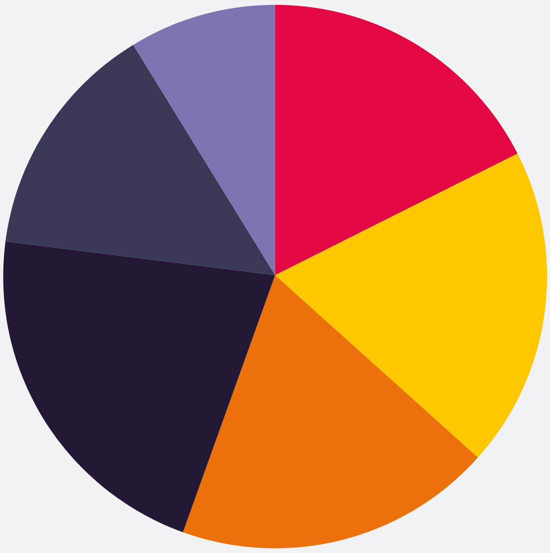
How many application security vulnerabilities do you currently have that need to be addressed?



QUANTITY	%
🔴 Less than 1,000	31.03%
🟡 1,000 - 5,000	28.82%
🟠 >5,000 - 10,000	20.20%
⚫ 10,000+	19.95%

Although the most often-picked answer to our question about how many application security vulnerabilities they currently have that need to be addressed was "less than a thousand" (31%), it is of little comfort. Our results show that 28.8% of teams have between 1,000 and 5,000 vulnerabilities, 20.2% of teams say they have at least 5,000, and another 20% have more than 10,000 unresolved vulnerabilities.

How has the volume of vulnerabilities/issues changed over the past 12 months?



ANSWERS	%
● Increased at a rate of 5x or more	20.44%
● Increases at a rate of 4x	15.52%
● Increases at a rate of 3x	22.17%
● Increases at a rate of 2x	18.72%
● No increase	14.29%
● Decreased	8.87%

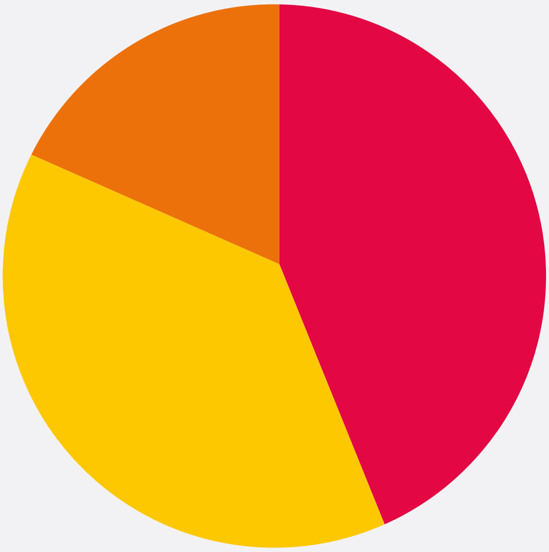
The number of vulnerabilities that are not yet mitigated is alarming, but the rate at which this number is growing is just as concerning. With 22.2% of respondents indicating that the volume of vulnerabilities has increased threefold in the last year, solving this growing problem is imperative.

42% are seeing more false positives and noise than ever before.

To the chagrin of developers and security teams alike, 42.4% of our respondents maintain that they receive more false positives and noise than ever before.

Security testing results are often noisy and require teams to scrub false positives and minor issues before prioritizing the vulnerabilities. Development teams then revise and resubmit the code for another round of testing. Since security tools are often used late in the development process, remediating the issues requires more time. Release dates often slip, and friction between development and security increases.

How is the number of false positives/noise changing?



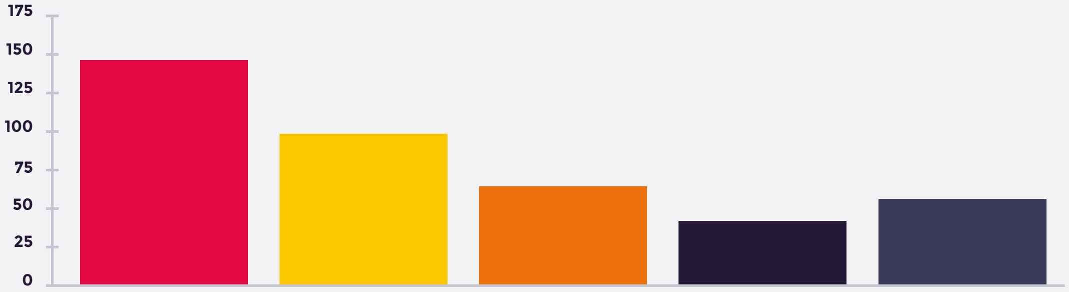
ANSWERS	%
● We're getting more than ever	42.36%
● We're getting about the same as usual	38.42%
● We're getting less than usual	19.21%

How are teams reducing false positives?

For those teams that said they were getting less than the usual number of false positives and noise, we wanted to know what they were doing to accomplish this reduction.

Deploying modern AppSec tools (35.5%), including defining custom detection rules (24.4%), account for nearly 60% of this improvement. Good old-fashioned hard manual work (16%) rounded out the top three answers.

If less than usual, what changes have you made to reduce the number of false positives?



ANSWER	%
● Implement modern AppSec tools like Software Composition Analysis, Runtime Application Self-Protection, IAST	35.47%
● Define custom detection rules in scanning engine to reduce false positives and false negatives	24.38%
● Manually triaged scan results before sending to development teams for remediation	16.01%
● Disabled default scan rules and only enabled rules that make sense to your organization	10.59%
● Use Machine Learning in AppSec stack	13.55%

The Relationship Between AppSec and Developers

Developers and security have very different priorities. Developers are focused on shipping code while security is focused on keeping their organizations safe. Despite having different priorities, it’s never been more important for these two groups to collaborate and work together. This section highlights how security feels about their current relationship with developers.

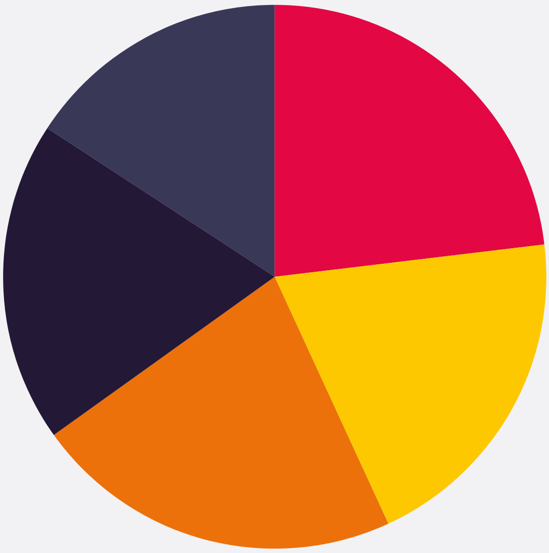
PART #3

Silos continue to hamper communication.

When presented with options to describe their relationship with developers, 22.9% of our security professionals chose "teams are siloed with little communication between them." This situation is unfortunate but reflective of the conditions within many organizations. The upside is that 62% reported that they have at least basic communication and can work with developers.

The success of an organization's application security program depends on a high level of communication and collaboration between developers and security teams. Where friction exists, security concerns can go unchecked.

Which of the following most accurately describes the relationship your security team has with developers?

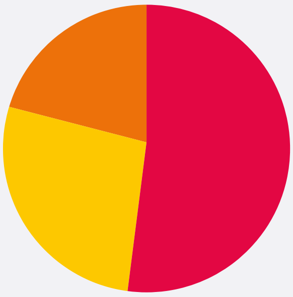


ANSWERS	%
● Teams are siloed with little communication between them.	29.91%
● Security and development teams have basic communications established. Security services are done on request.	21.43%
● Security and development teams work together. Each team has a security champion and there are regular security trainings for developers	22.91%
● Teams collaborate together. Security reviews performed with developers and system admins. Security processes are automated and integrated into SDLC	17.73%
● None of the above	15.02%

51% say the relationship between developers and security is improving and most feel developers are taking security seriously.

It is encouraging to note that 51.7% of security teams feel that their relationship with developers in their organization is improving. While we didn't correlate responses for each individual, one can imagine that the 20.7% of respondents that feel their relationship with developers is getting worse can also be found among the 22.9% that work in a siloed environment.

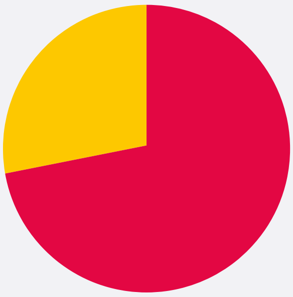
How are you seeing this relationship evolving?



ANSWERS	%
● It's getting better	51.72%
● It's getting worst	27.59%
● It's staying the same	20.69%

To continue with the positive results, it is hopeful to know that, by and large, the respondents feel that developers take security seriously. In today's threat-laden world, it's hard to imagine it being otherwise.

Do you feel developers take security seriously?

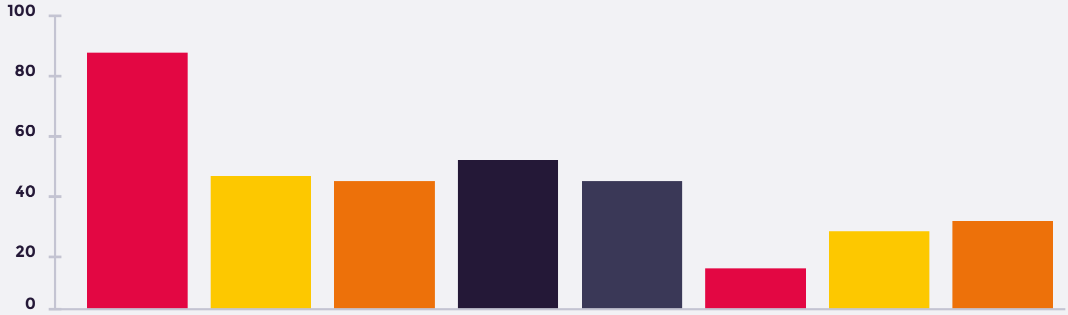


ANSWERS	%
● Yes	70.44%
● No	29.56%

Integrating automated security checks throughout the SDLC would have the most significant impact on the relationship with developers.

Organizations with a more mature application security program employ an end-to-end approach to AppSec. This strategy delivers superior results to the traditional method by providing developers with feedback on their security earlier in the process (shifting security left). It allows them to leverage integration and automation throughout the SDLC. There are still shortcomings, such as data integrations to maintain, information siloes, and false positives, but 21.9% of our respondents feel that integrating security checks throughout the SDLC is the most significant thing they could do to improve the relationship with developers.

What is the #1 thing you could do to improve the relationship with developers?



	ANSWER	%
●	Integrate automated security checks throughout the SDLC to keep up with release cycles	21.92%
●	Integrate security checks earlier in the SDLC to reduce work on developers	13.05%
●	Cross-train security and development teams to increase collaboration through common language and processes	12.56%
●	Have security focused developers working within the dev teams to bridge any communication gaps with security teams	14.53%
●	Communicate triaged AppSec issues directly to developers in their own systems (Jira, Slack, Github etc)	12.56%
●	Ensure only true positives and priority issues are sent to developers	6.65%
●	Gamify AppSec so developers have an additional incentive to work on security bugs	8.87%
●	Other	9.85%

Challenges & Areas For Improvement

Previous sections put hard numbers behind the state of modern application security programs. Next, we'll explore the top challenges teams are facing along with what they feel could be done to make their programs more effective.

PART #4

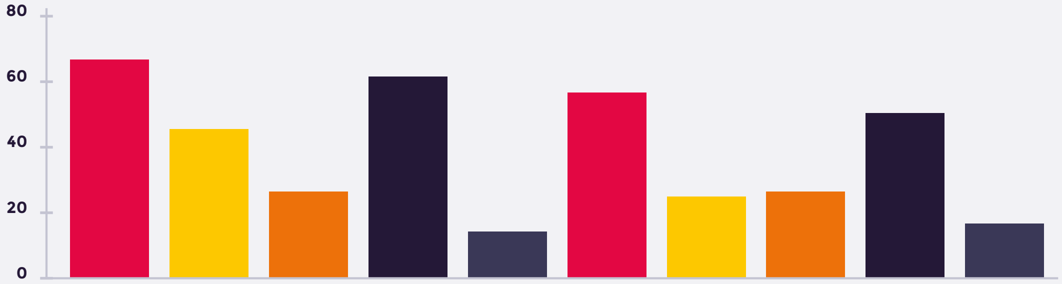
Developers not doing what security asks is the greatest challenge teams face.

The most popular answer to the question about the number one challenge of their application security program—Developers not doing what security asks—may seem a little self-serving of security teams. It's more telling, however, to look at the top three answers for more context.

It's easy for security professionals to believe that if everyone did what I told them to do (16.5%), the organization would have better application security. But look at the number two and number three answers too.

Lack of visibility across scanners and tools (15.5%) is all but a tacit admission that they could do better if the organization afforded them better visibility. And, a need for more security awareness (14%) reflects the security teams' frustration that others may not be taking risks as seriously as they should.

Which of the following do you consider to be the #1 challenge of your application security program?



ANSWER	%
● Developers not doing what security asks	16.50%
● Too much noise and false positives	11.08%
● Too many tools	6.90%
● Lack of visibility across scanners and tools	15.52%
● Too much time spent on manual work	3.94%
● Changing culture of organization to cultivate a security awareness	14.04%
● Training development staff in secure coding practices	7.14%
● Educating management on the threats and mitigation strategies	7.64%
● Reviewing legacy and third party applications for security risks	12.56%
● Shortage of capable security staff	4.68%

Reducing friction between developers and security would have the number one impact to improve the application security program.

When asked what would have the most impact toward improving their application security program, the number one answer—chosen by as many of the respondents as the number two and three solutions combined—was reducing friction between the developer and security teams (20.9%). The tension between these two groups is a severe problem in many organizations.

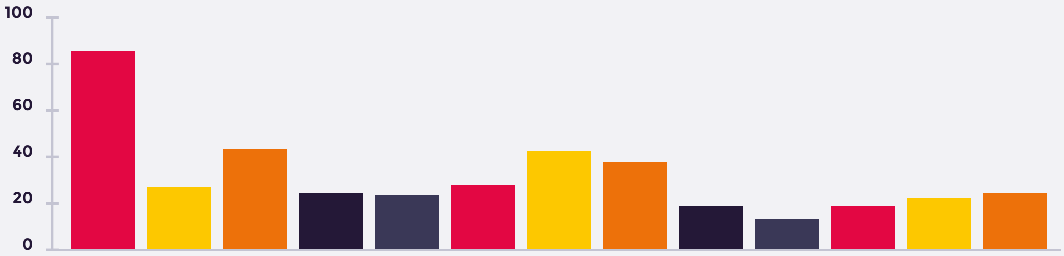
Security feedback lagging the development process.

Too many unactionable alerts

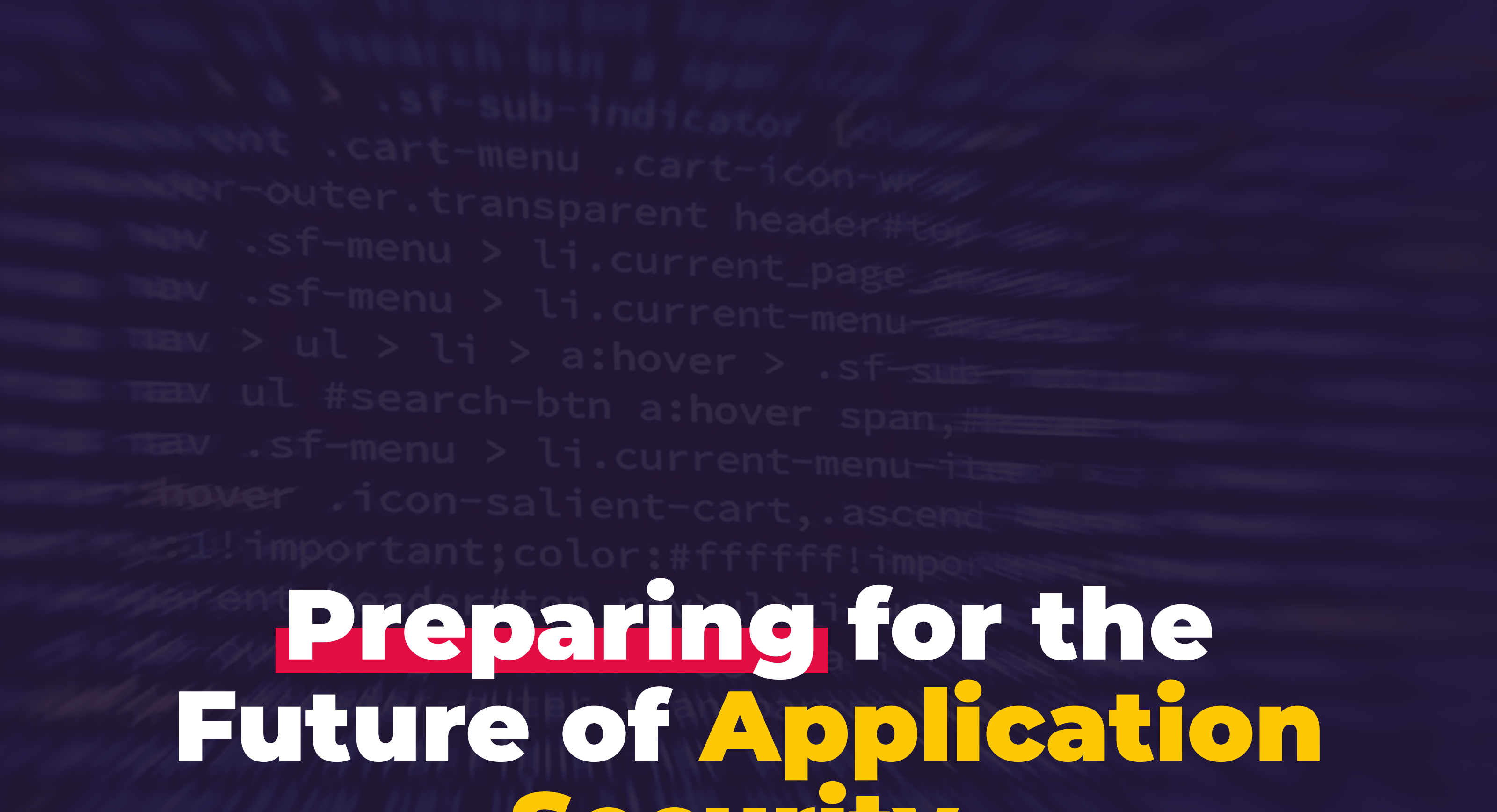
The bottleneck caused by the need for manual review of scan results

The number two and three survey results will also go a long way toward mitigating developer friction. By improving visibility across scanners and tools (10.6%) and integrating testing and notification into the CI/CD pipeline (10.3%), there will be fewer unactionable alerts, and automated testing and notification can happen in near real-time

To make your application security program more effective, what would have the most impact? [Choose the #1]



ANSWER	%
● Reduce friction between developers and security	20.94%
● Reduce noise and false positives	6.40%
● Improve visibility across all our scanners and tools	40.59%
● Automate more workflows and processes	5.91%
● Centralize appsec management	5.67%
● Integrate security into the SDLC (DevSecOps)	6.65%
● Integrate automated testing and notification in the CI/CD pipeline	10.34%
● Add security tooling for integration and automation including DAST/SAST/IAST applications	9.36%
● Add security related issues to ticketing/issue management system	4.68%
● Adopt and follow secure coding standards such as OWASP	3.45%
● Implement Web Application Firewall or RASP	4.68%
● Have developers run security scanning tools themselves	5.42%
● Hold development teams accountable for fixing vulnerabilities on time	5.91%



Preparing for the Future of Application Security

PART #5

As development teams continue to own more aspects of security testing and remediation, AppSec teams will need to transform themselves to provide security expertise for solving complex challenges and maintain oversight of the developer teams' performance on security. While developer teams might own tactical security tasks, the AppSec team will continue to be the experts in making risk-based decisions and driving security accountability across the development teams.

For too many companies, developers continue to get bombarded with noise and false positives from security scanners that lack the context needed to make security effective. Application security is still considered the sole responsibility of the security team.

At Tromzo, we imagine a world where security becomes self-service and developers can effortlessly determine security measures appropriate for their work and tune out the noise. A world where security becomes a first-class citizen in developer workflows and security teams are empowered to do meaningful work.

As an industry, ***application security*** is taking its first steps towards becoming integrated into developer workflows. Both opportunities and challenges lie ahead as we strive to make ***application security*** an enabler that assists developers in building secure software faster, empowering businesses to make the digital transformation journey safer and faster.

CISOs must eliminate the friction between developers and security so AppSec teams can scale their application security programs. Achieving this at scale requires a developer-first approach to security, and security must be made easy for developers so they can focus on shipping great software. Only then can AppSec teams focus on higher-value strategic work.

Developer-first is the future of application security.

Ready to eliminate friction between developers and security so you can scale your application security program?

VISIT www.tromzo.com

TROMZO

Make Application Security
Easy for Developers.