WhiteSource

WHITESOURCE DEVSECOPS INSIGHTS

Security vs. Developers: The DevSecOps Showdown

IS DEVSECOPS MORE THAN A BUZZWORD FOR ORGANIZATIONS?

DevSecOps requires processes and tools that enable weaving security throughout the DevOps pipeline.

Developers share ownership of security, and the traditional silos between development and security teams are broken down.

Most organizations believe they are in the process of adopting DevSecOps tools and practices. Are they?

We surveyed over 560 application security professionals and software developers to better understand the state of DevSecOps implementation.

KEY INSIGHTS

01 Most security professionals and developers feel forced to compromise on security in order to meet deadlines.

02 AppSec tools are purchased to 'check the box', disregarding developers' needs and processes.

03 Huge gaps in AppSec knowledge and skills among developers are neglected by organizations.

04 Security professionals' top challenge is vulnerability prioritization, but the lack of standardized processes leads to friction with developers.

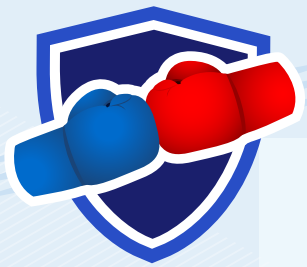
01

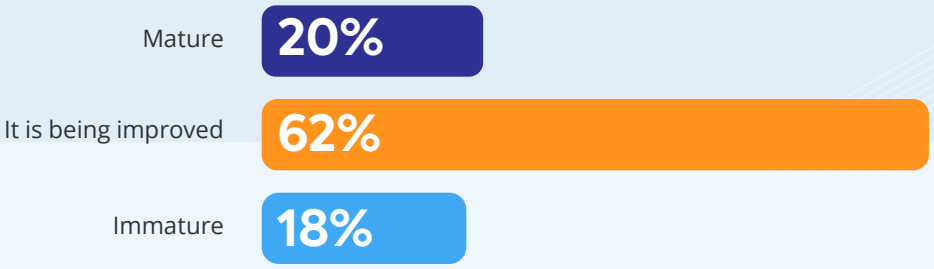
73% OF SECURITY PROFESSIONALS AND DEVELOPERS FEEL FORCED TO COMPROMISE ON SECURITY

Most respondents think that they are in the process of DevSecOps maturity

Most security professionals and developers believe their organizations are in the process of adopting DevSecOps tools and practices.

How would you describe the maturity of your organization's DevSecOps practices?

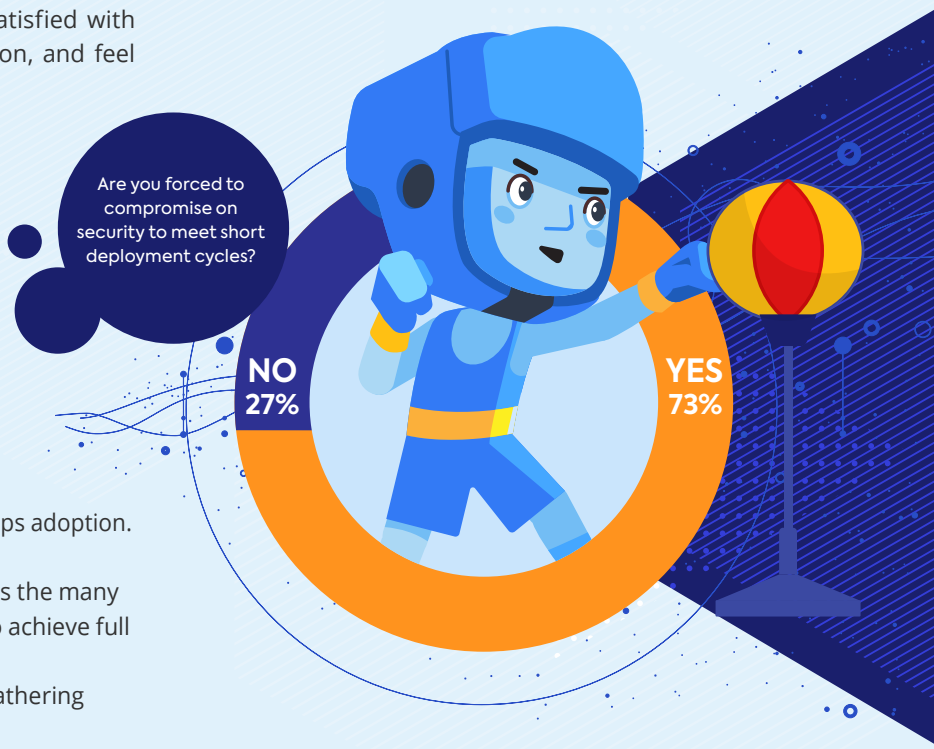




But — both security professionals and developers sacrifice security for speed

Many security professionals and developers are not satisfied with the AppSec processes implemented in their organization, and feel that security is sacrificed to achieve speed.

If developers feel they are neglecting security to stay on schedule, something in the DevSecOps process is broken.



Are they covering all of their DevSecOps bases?

There are a lot of moving parts to comprehensive DevSecOps adoption.

The OWASP DevSecOps Maturity Model (DSOMM) shows the many parameters that organizations should address in order to achieve full DevSecOps implementation, including:
• Build and deployment • Culture and org • Information gathering
• Infrastructure • Test and verification

This report will show that while some dimensions of DevSecOps are getting a lot of attention, others are being neglected.

02

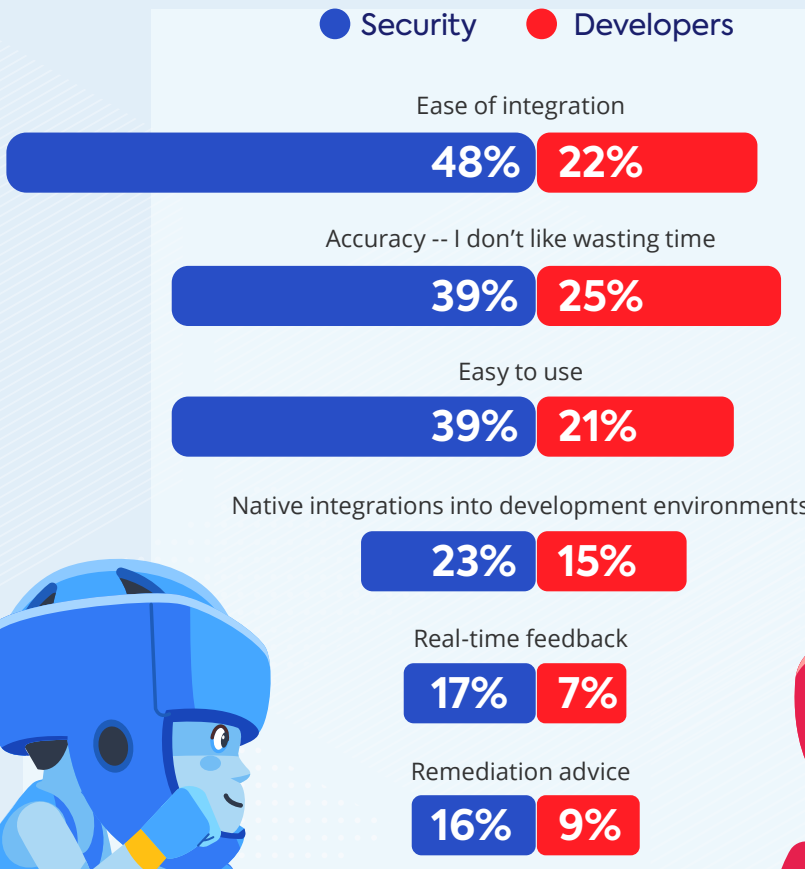
APPSEC TOOLS ARE PURCHASED TO ‘CHECK THE BOX’, DISREGARDING DEVELOPERS’ NEEDS AND PROCESSES

Security and development teams agree which AppSec features support developer adoption

Shifting security left is an important component in DevSecOps. Automated tools enable this process — if developers are willing to adopt them.

Results show that developers and security professionals are aligned when it comes to the features that are important for developer adoption.

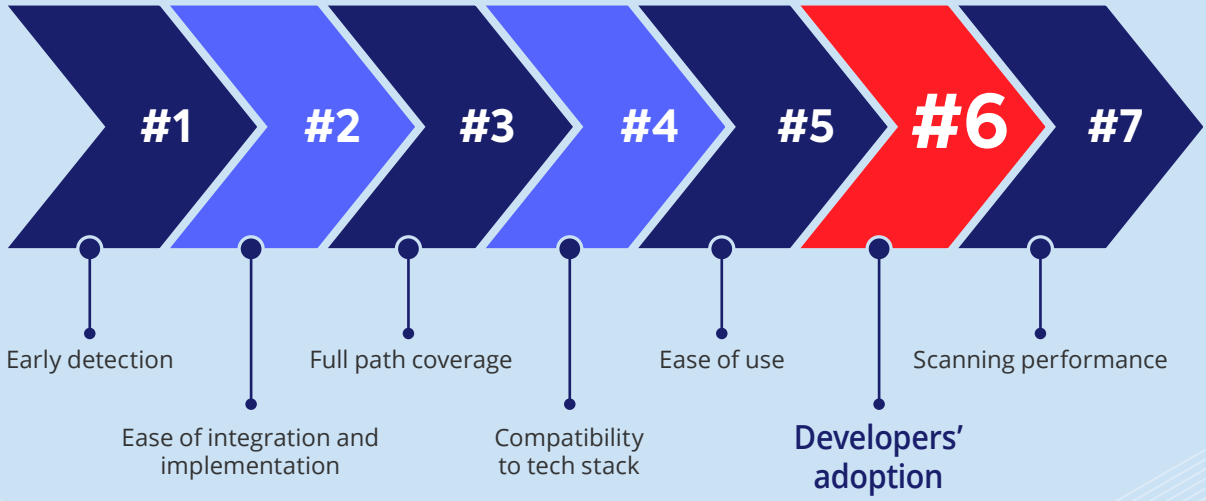
Which feature is the most important when it comes to developers adopting certain AppSec tools?



But — security professionals have their own considerations

When choosing an AppSec tool, developers' adoption gets very low priority from security professionals. Security needs such as detection, and ease of implementation, take priority in security professionals' considerations.

When considering an AppSec tool, which of the following are most important to you?



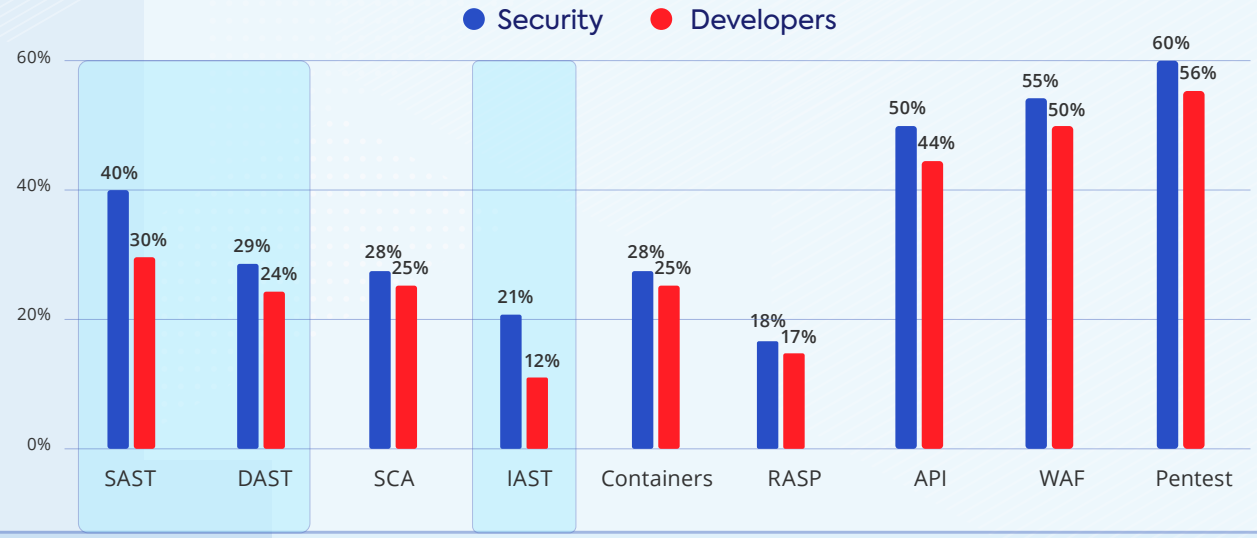
The result: Some AppSec tools are purchased, but not used by developers

When AppSec tools are purchased disregarding developers' adoption, the result is that the tools are left to gather dust.

When asked which AppSec tools they are using, respondents' answers varied significantly between the security and development teams. Security professionals estimate higher usage for all AppSec tools. It's worth mentioning that the biggest ratio gaps are for SAST, DAST, and IAST. This is another example of the disconnect between the two teams.

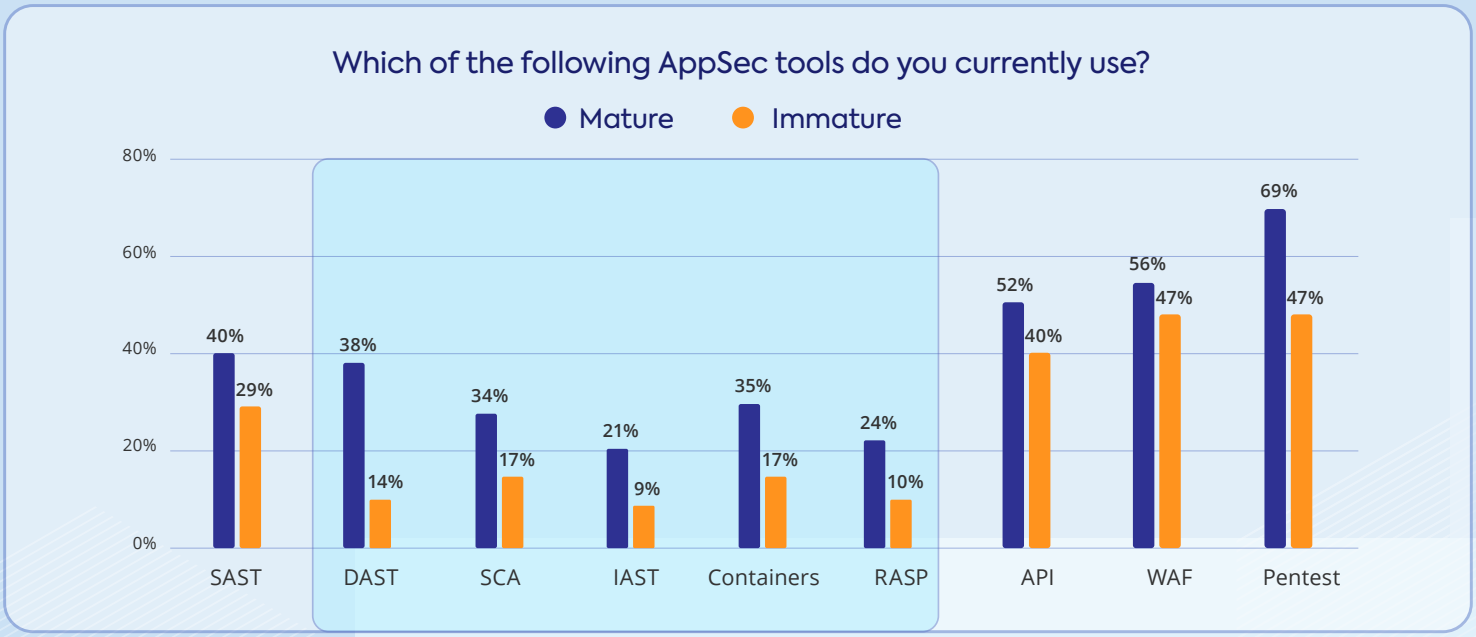
On top of the cost of neglecting security in early stages of development when issues are less expensive and time-consuming to fix, organizations are wasting money on tools that developers don't use.

Which of the following AppSec tools do you currently use?



Mature organizations use significantly more tools

We also analyzed results based on perceived maturity level. There is a clear correlation between perceived maturity level and higher usage of AppSec tools. DAST, SCA, IAST, Containers, and RASP were found to be used at least twice as much in mature organizations compared to immature ones.



Buying for compliance

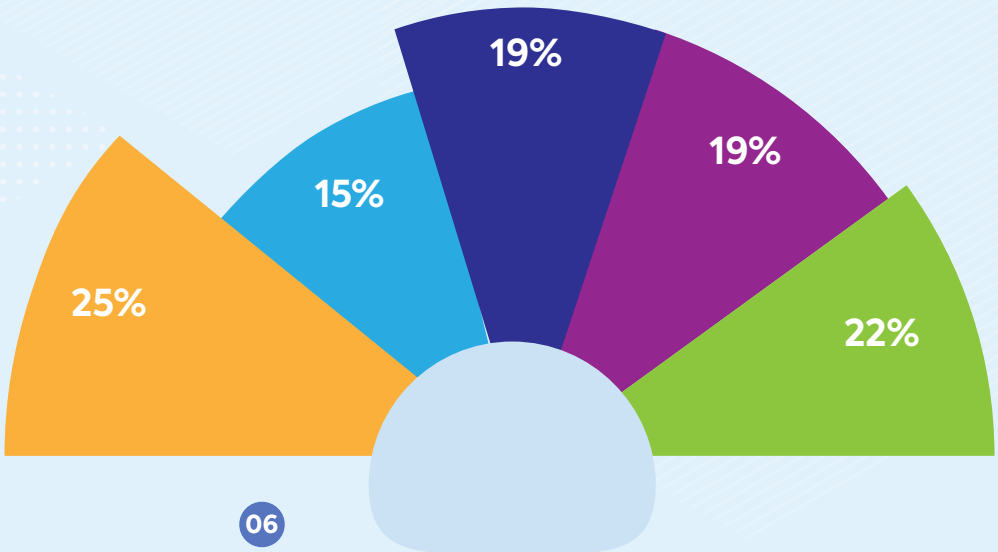
Security professionals rated meeting regulations and compliance with industry standards as the top two reasons for purchasing new AppSec tools.

Regulation and compliance are increasingly pushing companies to up their AppSec game, thereby driving investment in purchasing new tools.

Meanwhile, other security needs like addressing potential threats, are pushed aside. This reflects on organizational culture, making AppSec decisions to 'check the box' rather than improve shift left processes.

How do you justify purchasing new application security tools?

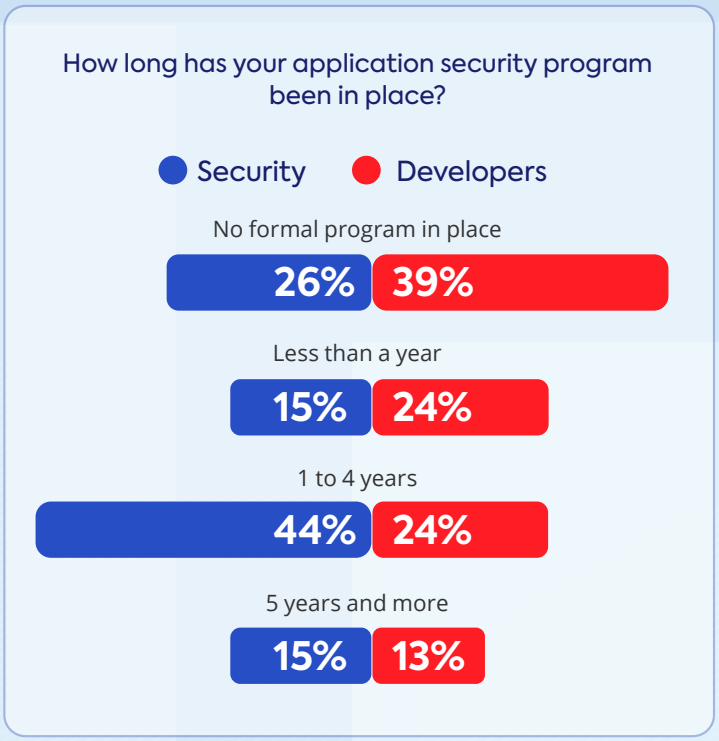
- Meeting industry-specific regulations (HIPAA, PCI etc.)
- Direct response to security audit findings
- Using well-known public incidents to demonstrate benefit (or risk)
- Including AppSec costs in general IT security spending
- Compliance with industry standards such as ISO/IEC 27034



03

HUGE APPSEC KNOWLEDGE AND SKILLS GAPS ARE STILL NEGLECTED BY ORGANIZATIONS

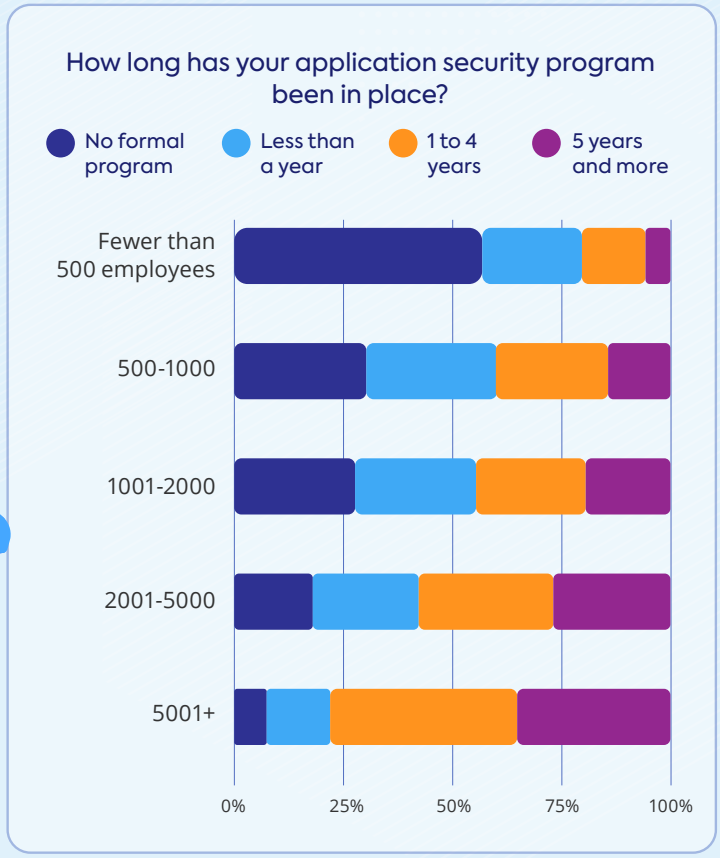
AppSec program? If it exists, most developers aren't aware



These responses are yet another sad demonstration of how out of sync security and development teams are.

Nearly 60% of security professionals say that they have had a security program in place for at least one year, while only 37% of developers are aware of a program running longer than a year. If there is an AppSec program in place, most developers aren't aware of it.

While some organizations still don't have an AppSec program up and running, we can clearly see that the larger the organization, the more likely it is to have a formal program in place, for a longer time.



Developers do not get training (although security believes they do)

Minimal efforts are invested in training developers, even though lack of skilled AppSec personnel is a top challenge.

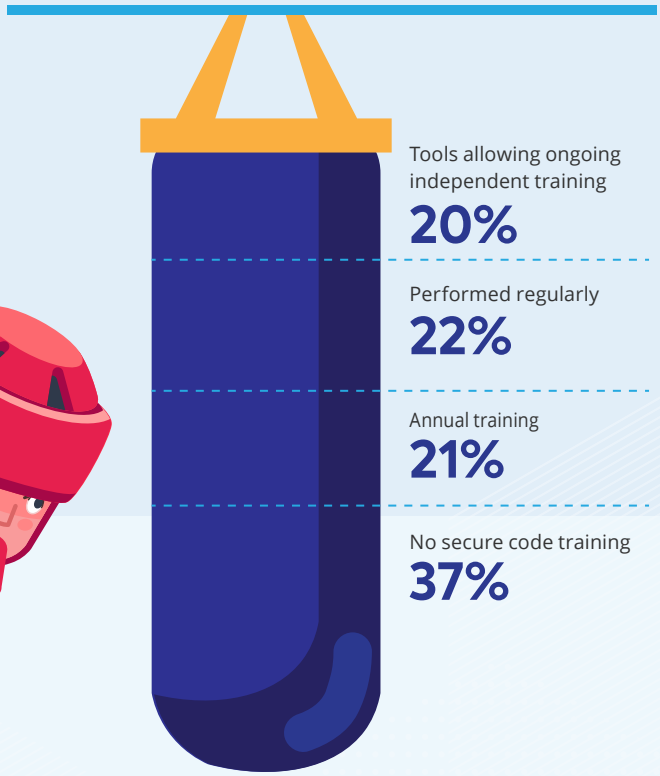
Nearly 60% of developers stated they have either no secure coding training or only an annual event.

This is a big hindrance to DevSecOps maturity. Improving developers' AppSec skills is as wise an investment as purchasing AppSec tools. It boosts shift left practices, and helps bridge the divide between security and development teams.

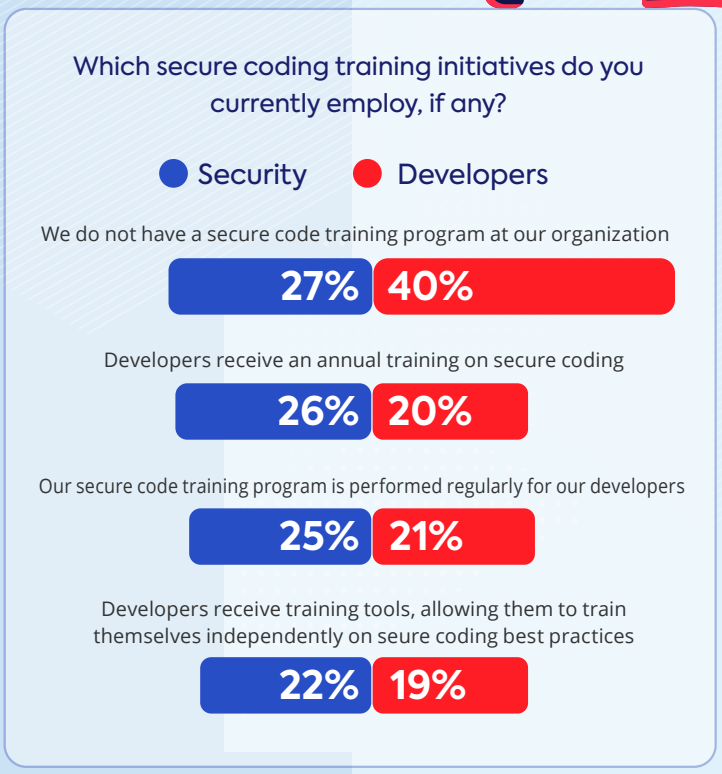
In addition, it's an investment in prevention of security issues by teaching developers how to avoid them from the start.

The siloed culture is again reflected by the differences in training perception between security professionals and developers.

Which secure coding training initiatives do you currently employ, if any?



Which secure coding training initiatives do you currently employ, if any?



04

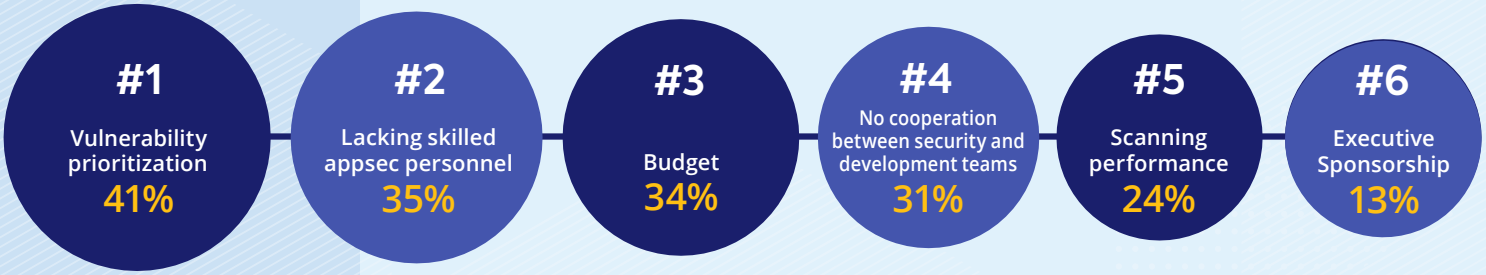
SECURITY PROFESSIONALS' TOP CHALLENGE IS PRIORITIZATION, BUT THEY LACK STANDARDIZED PROCESSES

Security professionals view prioritization as their top challenge

Addressing security debt is a huge issue for security professionals today. The more security testing tools they use, the more alerts they are required to address.

Since fixing every single vulnerability and staying on schedule is unrealistic, it's crucial to reduce the load of new vulnerabilities early in the SDLC, by helping developers to prioritize and remediate the issues as soon as they are detected.

What are the biggest challenges in implementing and running your AppSec program?

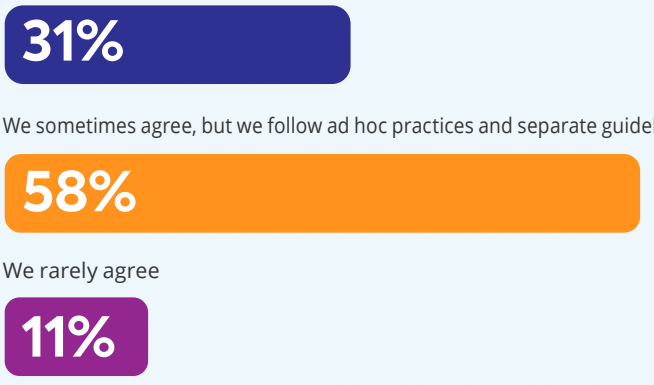


However, most organizations lack a standardized prioritization process

Only 31% of organizations have a defined and agreed-upon prioritization process. Lacking a standardized practice, most teams rely on ad-hoc practices, or follow separate guidelines for development and security teams.

The results of continuously having to renegotiate a prioritization strategy are expensive. Valuable time is wasted, delaying remediation and critical security issues, which are left open until teams come to a decision.

To what extent do the security team and development team in my organization agree on which application vulnerabilities need to be fixed?



Lack of standard practices also leads to friction between teams

An important component of DevSecOps is company culture, and cooperation between security and development teams, which was also listed by security as a top challenge.

Standards and processes form the shared language of teams, without them communication challenges arise. Friction between teams slows down both development and security, and enforces the security silos that DevSecOps is meant to break down.

An AppSec champion helps skills, prioritization, and communication

Lack of skilled personnel was the second biggest challenge listed by security.

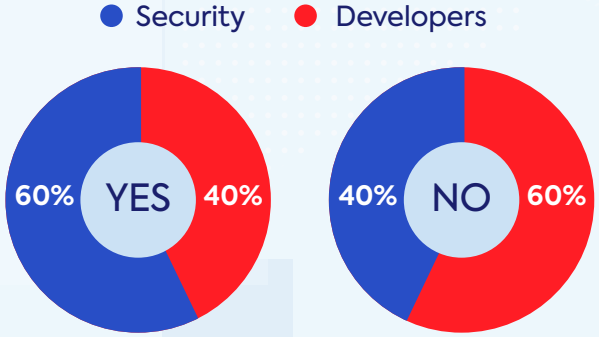
Appointing an AppSec champion for development teams is an important first step towards bridging the skills gap.

Unfortunately, only 40%-60% of organizations have an AppSec champion in organizations. More evidence of the divide between teams is that even when security professionals say there is one, developers don't always agree.

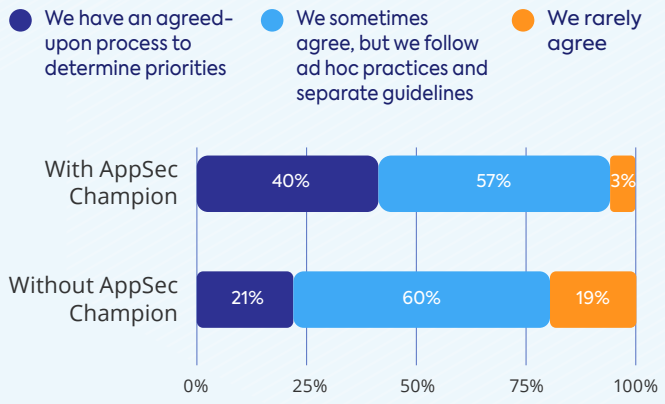
When cooperation between teams is encouraged, standardized processes are more common, and agreement is more easily achieved.

Teams with an AppSec champion have nearly twice the chance to easily reach an agreement by relying on a standardized process.

Do you have an assigned champion in your development team who takes the lead on AppSec objectives?



To what extent do the security team and development team in my organization agree on which application vulnerabilities need to be fixed?



How to Break the Silos and Advance Towards DevSecOps Maturity

While organizations are making an effort to achieve DevSecOps maturity, our research shows most still have a way to go.

Most developers and security professionals are still struggling to make security more agile, and feel that security is left behind in favor of achieving deadlines. Prioritization is a major challenge for teams, and as security debt grows, sharing ownership over security is crucial.

When choosing automated tools, organizations need to invest in solutions that can be easily integrated into development processes. Developers need training to get up to speed when it comes to AppSec and secure coding. At the base of it all is organizational culture.

Organizations must continue working to break the silos between security and developers. If all of the moving parts of DevSecOps aren't embraced throughout the organizations, from stakeholders across all teams, DevSecOps will remain a buzzword.