Product Security by Design

4 Steps to Building AppSec Earlier in the SDLC

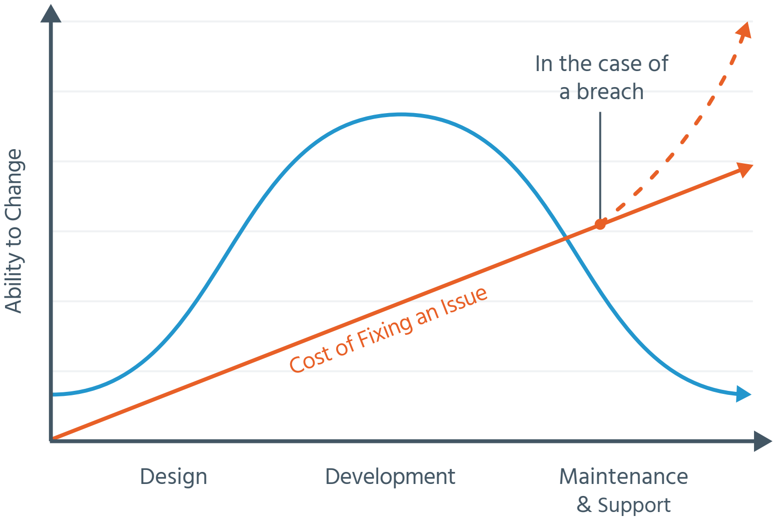
19%

Web applications were identified as the most common attack pattern in the 2018 Verizon Data Breach Investigations Report, comprising **19% of all data breaches**.

It's no secret that agile development teams are pushing code and releasing features directly into production continuously—almost as fast as ideas are conceived. But with web applications representing the most common vector for data breaches[1], the question of how security testing can keep pace with these software builds remains unanswered for many teams. This is where dynamic application security testing, or DAST, comes in.

DAST can enable security and development teams to work together to balance speed with acceptable risk. In fact, leveraging a DAST tool can speed up development time by spotting and addressing vulnerabilities (or bugs) while the product is being built—a concept known as shifting left. By prioritizing security earlier in the software development lifecycle, you can proactively address many of the potential downstream consequences of exposing application vulnerabilities to the wild. This approach helps prevent you and your organization from rolling back features or incurring immeasurable losses in the wake of a breach.

Software Development Lifecycle



[1] 2018 Verizon Data Breach Investigations Report

4 STEPS TO SHIFTING LEFT IN YOUR ORGANIZATION

Build Cross-Functional Partnerships

To successfully shift your application security efforts left (or earlier) in the development process, you first need buy-in from all of the stakeholders involved. Those who lead security, development, and operations teams must agree on both a shared goal to better secure what is being built, and the new tools or processes that will be introduced for that purpose. Addressing these potential sources of friction early on allows all teams to feel invested in process improvements and lend cross-functional support.

Embed Security with Existing Tools

Next, introduce tools that will allow you to automate more tedious aspects of security, embed into development's existing workflows, and play nicely with development's current tools. This way, security is no longer a blocker at the end of the process, but rather folds in as a natural part of the SDLC process.

Streamline Processes Through Orchestration

Further pinpoint tools that are purpose-built to integrate with common continuous integration/continuous delivery (CI/CD) and ticketing solutions, so that vulnerabilities are detected, reported, and prioritized all in the same workflow (one that development knows well). When done right, security can run parallel to other dev work in the background—with no impact on performance or speed.

Get Feedback in Real Time

Consider tools that can put in place quality gates, or policies that halt builds or raise alerts when an agreed upon threshold of vulnerabilities has been detected by a DAST scan. Why? Development not only gets an early warning of the bug, but also the motivation to fix the issue so that integration and testing processes proceed error-free.

It's simple: Shifting left and catching security issues earlier in the development cycle makes them far cheaper to fix and downtime a thing of the past. When it comes time, you can be confident that you're deploying apps and not the risks we've come to associate with them.

Take the next step with your AppSec program.

Learn more about how we can help:
www.rapid7.com/shifting-left