DevSecOps – Integration of App Security in DevOps

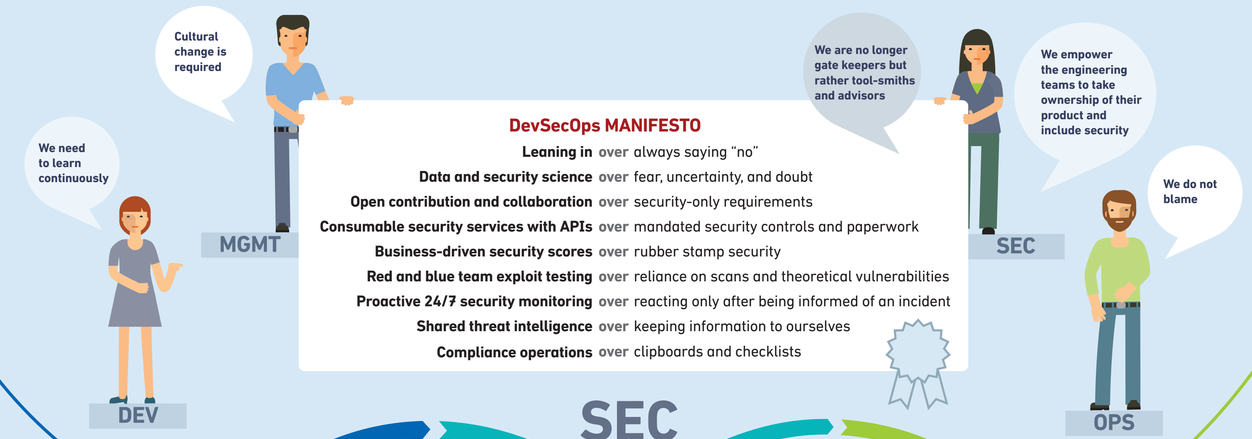
IT Spektrum vormals OBJEKTspektrum
Digitaler Wandel & Software-Architektur für Profis

STATIC APPLICATION SECURITY TESTING (SAST)

A SAST solution helps developers to detect vulnerabilities in their code before it is pushed into production. It analyzes applications from the inside out to check for flaws in the code before it is compiled.

Best practices

- SAST helps increase application security awareness among software developers, but it can only identify a limited set of vulnerabilities.
- SAST can often raise false alarms. Schedule an initial cleaning phase for a skilled developer to analyze all of the issues and flag any false positives.



DevSecOps MANIFESTO

Leaning in	over	always saying "no"
Data and security science	over	fear, uncertainty, and doubt
Open contribution and collaboration	over	security-only requirements
Consumable security services with APIs	over	mandated security controls and paperwork
Business-driven security scores	over	rubber stamp security
Red and blue team exploit testing	over	reliance on scans and theoretical vulnerabilities
Proactive 24/7 security monitoring	over	reacting only after being informed of an incident
Shared threat intelligence	over	keeping information to ourselves
Compliance operations	over	clipboards and checklists

IDENTITY AND ACCESS MANAGEMENT (IAM) BENEFITS

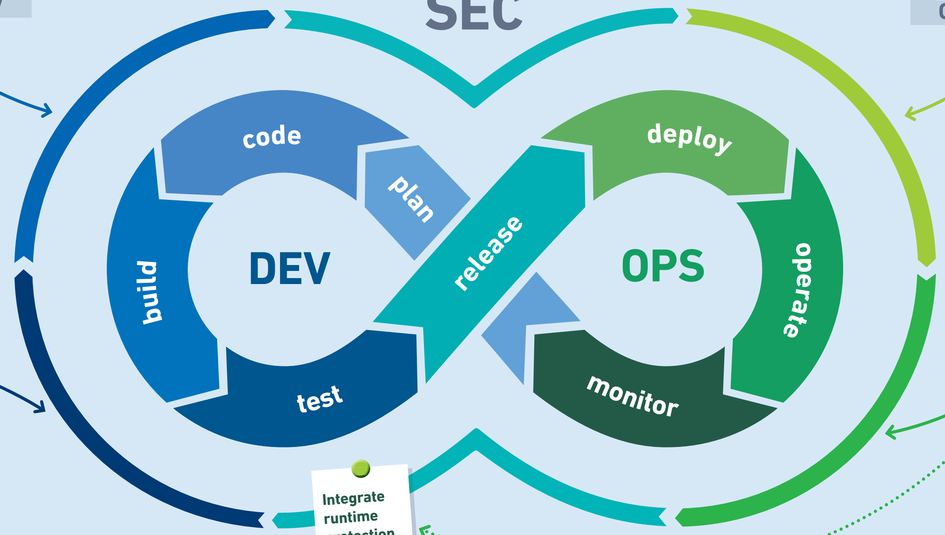
- Prevent account takeover by adding a user-friendly second authentication factor.
- Increase delivery speed and unify the user experience with a single identity for every customer, partner, and employee.
- Add secure login and single sign-on to both legacy and new applications.

Best practices

- Use security services via APIs rather than building them within the application yourself.
- With a modular and expandable IAM platform, you can easily switch to new authentication methods or change your registration flows in the future.
- Practice zero trust and prevent lateral movement.

CONTINUOUS ASSESSMENT

Fixing vulnerabilities is exponentially cheaper if they are caught before production.



CONTINUOUS PROTECTION

Protect against bots and malicious users at runtime and mitigate even unknown attacks before they reach the application.

SHIFT LEFT

SHIELD RIGHT

SOFTWARE COMPOSITION ANALYSIS (SCA)

The vast majority of security vulnerabilities are actually found in transitive dependencies that the developers often do not even know about. SCA tools help you create an inventory of all the third-party components in your products, including both direct and transitive dependencies. You quickly receive warnings for known vulnerabilities as well as support in improving license compliance and code quality.

Best practices

- Run automated checks regularly: Trigger the analysis both for every code change and on a regular basis (such as every night).
- Early checks start in IDE: If an open-source component is vulnerable, you should already receive an actionable warning in your IDE.
- If you can't update the vulnerable component immediately, use "virtual patching" to keep the time window for an attack as short as possible (see WAAP).

"More than 70% of applications contain flaws stemming from the use of open source." (Gartner)

"On average, it takes 200 days for a critical vulnerability to be fully patched." (Source: NTT Application Security Security)

Integrate runtime protection

DYNAMIC / INTERACTIVE APPLICATION SECURITY TESTING (DAST/IAST)

DAST tools simulate external attacks on running applications to reveal runtime vulnerabilities. As black box tests, they work without knowing anything about the application.

Best practices

- Integrate DAST tools with the bug tracking system your DevOps colleagues use.
- Give your developers exactly the right information they need to fix security vulnerabilities immediately. That helps them make security concerns a priority and move your organization closer to a DevSecOps mindset.

Mitigate exploits with virtual patches and buy time (until all components are fully patched).

WEB APPLICATION AND API PROTECTION (WAAP) BENEFITS

- Deploy virtual patches to prevent exploitation before a vulnerability can be fixed
- Protect from unknown attacks and 0-day-exploits
- Enforce user authentication
- Proactive 24/7 security monitoring
- Shared threat intelligence

Best practices

- Use a positive security model that forbids everything the developer has not explicitly allowed (like OpenAPI schema validation).
- Protection should be integrated and activated already upon development, not only in production. This makes it possible to use security features effectively and resolve false positives at an early stage.
- Automate security configurations, testing and validation in the CI/CD pipeline.
- Microgateways are cloud-native, lightweight WAAP solutions that are easy to automate.
- Implement a zero-trust model by protecting each application, microservice or API with a dedicated Microgateway.
- Use ML to detect and block automated threats.

What is WAAP? Nowadays, web application firewalls (WAFs) and API security gateways are often combined into a single product. This product category is called WAAP.

Inhaltliche Entwicklung:

Ergon Informatik AG:
Daniel Estermann und Stefan Dietiker

VSHN AG:
Adrian Kosmaczewski