

DevSecOps

Wie bringe ich den Security Gedanken in meinen DevOps-Prozess?

Yannick Tresch

DMoove Solutions GmbH

yannick.tresch@dmoove.com

Die DMoove

- Cloud Consulting
- Managed Services
 - DevOps Tools
 - Monitoring
- Kostenoptimierung on AWS
 - Up to 55% savings on EC2
- Partner: AWS, GitLab, Centreon, Microsoft

Introduction to DevSecOps

What is DevOps?



Cultural
Philosophy

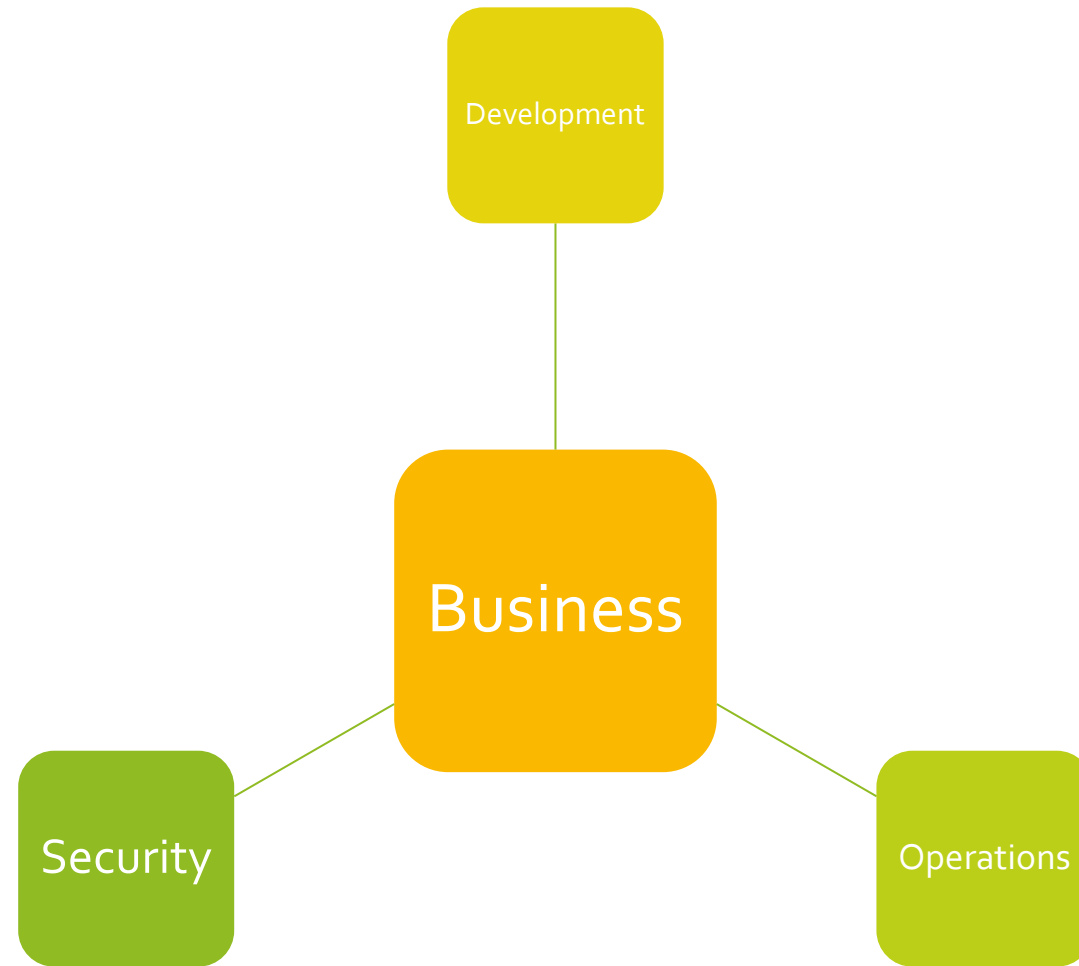


Practices



Tools

**Whats
important?**



What is DevSecOps?

DevSecOps is

- Team effort, not a single security unit
- Automated and continuous security

DevSecOps role

- Not here for code audits
- Implement processes to automate code validation and audit of code and artifacts during CI/CD process

So how should we perform DevSecOps?

- Implement security as code
- Keep security in your mind while developing
- Implement iterative solutions and not „the best answer“
- Not only scanners but real attacks on services
 - Learn how to defend
 - Learn how to attack
- Automate defense procedures



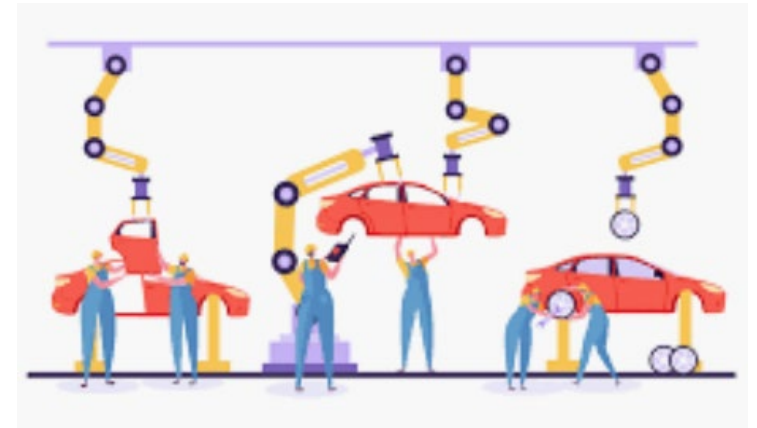
Target:
Ensure data
security

DevSecOps

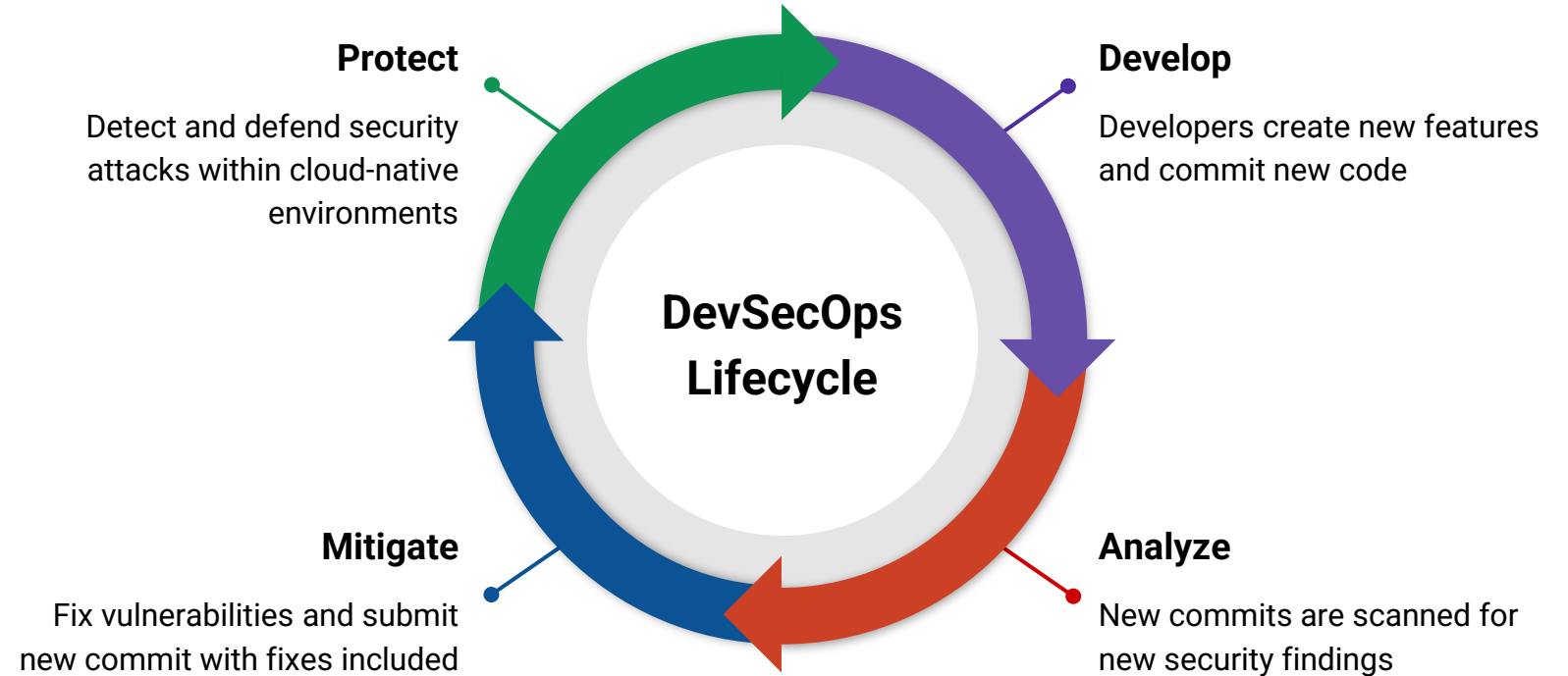
New problems need new solutions

The CI pipeline is your software assembly line. It must be:

- Standardized
- Protected
- Measured
- Inspected



DevSecOps lifecycle



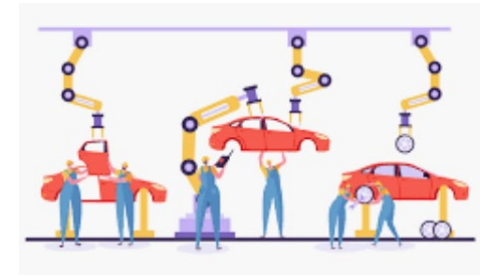
Application Security Testing and Remediation

Shift-left

Managing vulnerabilities



What happens when you find 10k vulnerabilities at the end of the SDLC?



What if you empower Dev to deal with each one at the point where it is introduced?

What means shift-left?

Implementing a successful “shift left” is straightforward; however, steps are often overlooked. Security scans and associated results are:

- Run as early and as often as possible into the development lifecycle
- Built into CI with results showing up within the developer workflow
- Run on the delta code change making results available within minutes
- Easy to understand and include actionable next steps to resolve

Advantages for the developer

Contextual

- Within CI/CD dev workflow - accountable person
- Security dashboard for Security

Congruent with DevOps processes

- Iterative within dev, tests every code change
- Immediate cause/effect of code changes

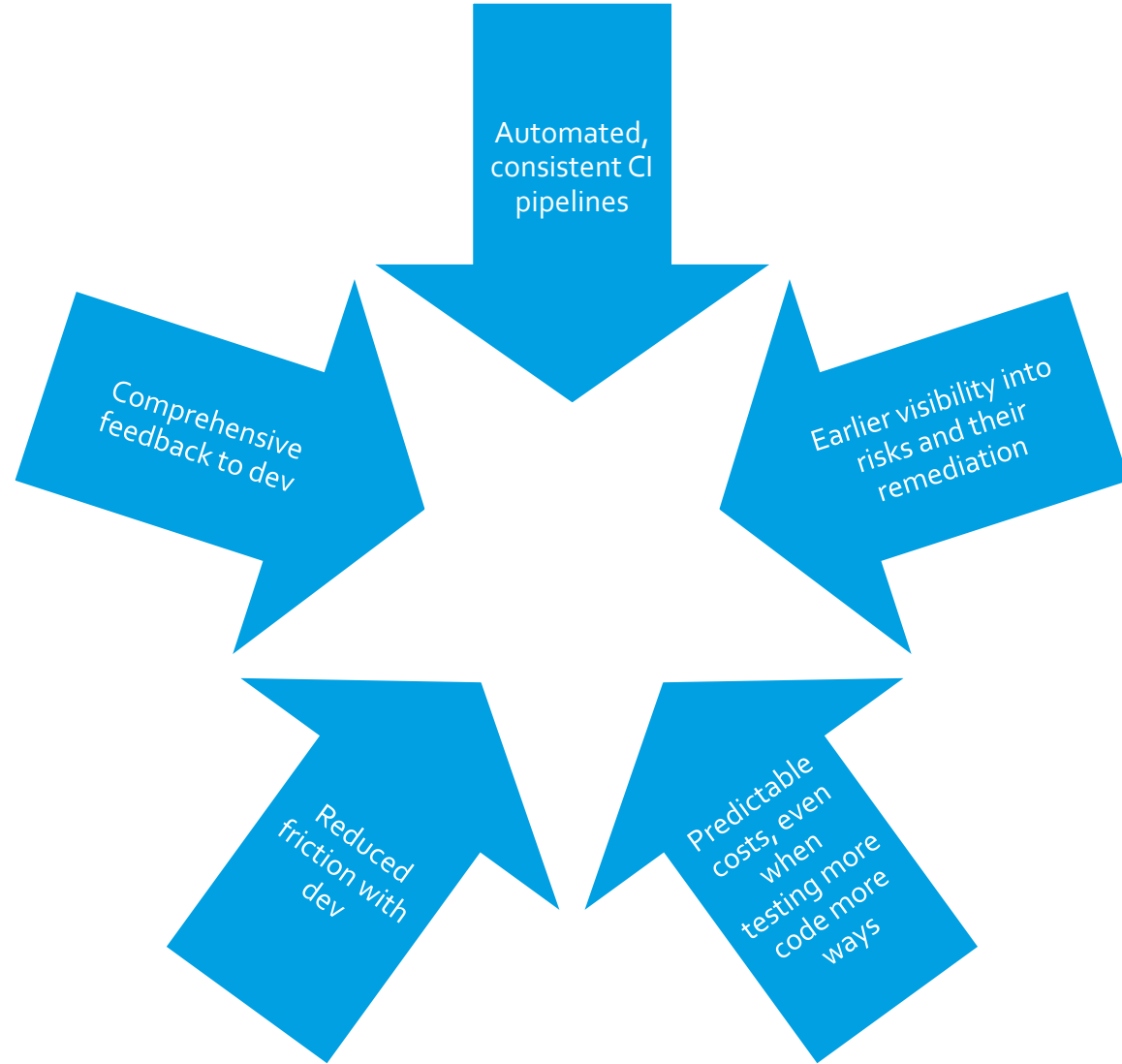
Integrated with DevOps tools

- Create issues
- Auto remediation
- Production feedback

Efficient and automated

- Eliminate work wherever possible
- No context-switching
- Less tracking/triaging and more value-added security

Advantages for the security engineer



Security Testing

Tooling for security testing

Static Application Security Testing (SAST)

- In SAST, application is tested from inside out.
- This type testing is a developer's approach of testing.

Dynamic Application Security Testing (DAST)

- In DAST, application is tested from outside in.
- This type testing is a hacker's approach of testing.

Dependency scanners

- Scans source code for dependencies.

The screenshot displays the GitLab Security Dashboard for a project named 'simply-simple-notes'. At the top, a 'Project's current vulnerability state' callout points to a summary bar showing counts for different severity levels: Critical (0), High (3), Medium (13), Low (30), Info (3), and Unknown (0). Below this, a table lists individual vulnerabilities. Callouts on the right side of the image point to specific parts of the table: 'File and/or line where vulnerability resides' points to the 'Description' column; 'Scanner that identified the vulnerability' points to the 'Scanner' column; 'Issues related to vulnerability' points to the 'Activity' column; and 'Remediated vulnerability awaiting review' points to a vulnerability with a status of 'Detected' and a severity of 'High'.

Status	Severity	Scanner	Description	Identifier	Scanner	Activity	
Detected +1 more	All severities	All scanners					
<input type="checkbox"/>	Detected	Confirmed	High	Uncontrolled Memory Consumption in Django requirements.txt	CVE-2019-6975 + 1 more	Dependency Scanning	<input type="checkbox"/> 0
<input type="checkbox"/>	Detected	Confirmed	High	CVE-2019-14697 in mustl registry.gitlab.com/gitlab-examples/security/simply-simple-notes/master:1c4ac8d1e03c2d84da b86b47fe22b69d8ab918f6	CVE-2019-14697	Container Scanning GitLab	<input type="checkbox"/>
<input type="checkbox"/>	Detected	Detected	High	Denial of service in Flask requirements.txt	CVE-2019-1010083 + 1 more	Dependency Scanning	<input type="checkbox"/> 1
<input type="checkbox"/>	Detected	Detected	Medium	Insecure HTTP Method - DELETE	CWE-200 + 2 more	DAST	<input type="checkbox"/> 0
<input type="checkbox"/>	Detected	Detected	Medium	HTTP Only Site	HTTP Only Site + 2 more	DAST	<input type="checkbox"/> 0

Today: Providing increased visibility into security risk

Vulnerable projects

ytredmoove.com dmooveteam/dtag-training/dtag-demo-container:container/Dockerfile	1 C 1 H 0 M 88 L	updated a few seconds ago
ytredmoove.com dmooveteam/tsi-digitize:container/Dockerfile	1 C 1 H 0 M 88 L	updated a minute ago
ytredmoove.com dmooveteam/gitlabrunnerroles/ecs-gitlab-runner:template.yaml	0 C 4 H 2 M 3 L	updated a few seconds ago
ytredmoove.com dmooveteam/tsi-digitize:app.yaml	0 C 4 H 2 M 2 L	updated a minute ago
ytredmoove.com dmooveteam/aws-accounts/dmoove-aws-lz:cfn/dmoove_ci_infra.yaml	0 C 4 H 1 M 2 L	updated a few seconds ago
ytredmoove.com dmooveteam/gitlabrunnerroles/aws-builder:awsers/package.json	0 C 2 H 2 M 1 L	updated a few seconds ago Fix vulnerabilities
ytredmoove.com		

Whitesource Alternative:
Snyk



Fragen?

LinkedIn: Yannick Tresch
E-Mail: yannick.tresch@dmoove.com

