iteratec

DEVSECOPS LEITFADEN

10 Tipps zur Implementierung von DevSecOps in Unternehmen

Building Blocks für eine starke DevSecOps Organistion

Organisationen mit einer gelebten DevSecOps-Kultur haben kürzere Go-to-Market Zeiten, geringere Fehlerraten und eine 9 x höhere Wahrscheinlichkeit, schwere Sicherheitsprobleme zu vermeiden[1]. Um die Potenziale von DevSecOps für den gesamten Secure Software Development Cycle (SDLC) zu heben, müssen Organisation bestehende Silos in ihren Entwicklungsprozessen überwinden und neue Formen der Zusammenarbeit etablieren.

Lernen Sie die zentralen Bausteine einer funktionierenden DevSecOps Organisation kennen und 10 Praxistipps, wie Sie dort hinkommen:

Mindset: Applikationssicherheit als gemeinsame Verantwortung

In der alten Welt war die Aufteilung klar: Entwickler waren für die schnelle Bereitstellung, Security für die Anwendungssicherheit und Operations für die den stabilen Betrieb verantwortlich. DevSecOps bricht diese Silos auf, beseitigt Schuldzuweisungen und vereint alle drei Rollen in einem gemeinsamen Ziel: der schnellen Bereitstellung von Software, die sowohl sicher als auch stabil ist.

Dafür müssen DevSecOps Teams erst einmal ein Bewusstsein entwickeln, dass Security nicht eine Aufgabe einzelner, sondern die gemeinsame Verantwortung aller im Entwicklungsprozess ist. Und die Organisationsstrukturen personell und strukturell daran anpassen.

→ Zu den Tipps 1–4

Prozesse: Empowerment statt Delegation

Herkömmliche Entwicklungs-Workflows sind darauf ausgelegt, die Verantwortung für Anwendungssicherheit auszulagern und fördern damit Silo-Bildung statt einer übergreifenden Zusammenarbeit zwischen Dev, Sec und Ops. Die Folge: Längere Freigabeschleifen, mehr Abstimmungsaufwand und längere Release-Zyklen.

Damit die am Entwicklungsprozess beteiligten Personen in der Lage sind, Verantwortung zu übernehmen, müssen Prozesse etabliert werden, die den Einzelnen empowern anstatt Verantwortung auszulagern.

→ Zu den Tipps 5–7

Rollen: Brückenbauer gesucht

DevSecOps verändert Aufgaben und Tätigkeitsbereiche von Mitarbeiter*innen in Entwickler-, Operations- und Security-Teams. Um dem Rechnung zu tragen, müssen auch Rollen und Funktionen innerhalb dieser Teams teilweise neu interpretiert und an die Zusammenarbeitsmodelle angepasst werden.

Zudem kann die Einführung zusätzlicher Rollen – insbesondere in Transitionsphasen – dabei helfen, DevSecOps-Prozesse in diesen Teams zu etablieren, Mitarbeiter*innen zu befähigen und den Team-übergreifenden Wissensaustausch zu fördern.

→ Zu den Tipps 8–10

[1]Palo Alto Networks, 2022: The State of Cloud native Security 2022

10 Tipps für hochperformante DevSecOps Teams

Der Weg zur gelebten DevSecOps Praxis kann lang und steinig sein. Nutzen Sie unsere Praxistipps zum Auf und Ausbau funktionierender DevSecOps Organisationen:

1.

Ein Bewusstsein für die Folgen von Sicherheitslücken schaffen.

Um Applikationssicherheit als gemeinsame Aufgabe wahrnehmen zu können, müssen zunächst alle am SDLC beteiligten Personen ein gemeinsames Verständnis von Security entwickeln. Oft fehlen etwa in Entwicklungsteams das notwendige Wissen und das Bewusstsein darüber, welche Auswirkungen bestimmte Fehler im Code auf die Sicherheit von Komponenten und Services haben können. Awareness-Trainings und Workshops zu den OWASP Top 10 oder Capture-the-Flag Simulationen können dabei helfen, zentrales Know-how zu vermitteln und eine gemeinsame Verständigungsbasis zu schaffen. Das umfasst Entwickler ebenso wie die Management- und Entscheider-Ebene.

2.

Gemeinsame Ziele und Kennzahlen etablieren.

Entwickler und Security-Verantwortliche verfolgen im Alltag oft vermeintlich unterschiedliche Interessen. Gemeinsame Zielkennzahlen können deshalb dabei helfen, gegenseitige Vorbehalte und Widerstände zu überwinden. Damit sie zum gemeinsamen Projekterfolg beitragen, sollten sie Outcome vor Output stellen und Effekte, die sich aus der Kombination von technischem Fachwissen und der Minimierung von Reibungsverlusten über den gesamten Softwareentwicklungszyklus hinweg ergeben, messbar machen.

Dazu gehören beispielsweise:

→ Häufigere Deployments
→ Kürzere Lead time to Change (LTTC)
→ Geringere Ausfallraten bei Änderungen
→ Kürzere Mean Time to Recovery (MTTR)

3.

Cross-funktionale Teams einrichten.

Die zentrale Herausforderung beim Aufbau einer funktionierenden DevSecOps Organisation besteht darin, vorhandene Silos aufzubrechen. Einer der effektivsten Wege, um das zu erreichen, besteht in cross-funktionale Teams, die sowohl aus Entwicklern und Operations-Verantwortlichen sowie aus Security-Experten bestehen. Auf diese Weise können bei der Entwicklung neuer Features Sicherheitsaspekte von Beginn an mitberücksichtigt werden.

4.

Von Gating zu Pairing.

Klassische Gating-Prozesse mit definierten Projektabschnitten bzw. Meilensteinen fördern die Silo-Bildung und erzeugen langwierige Feedbackschleifen. In DevSecOps Organisationen mit dem Ziel einer geteilten und gemeinsamen Verantwortung sollte Gating daher von Konzepten der gegenseitigen Verantwortung abgelöst werden. Neben cross-funktionalen Teams, in denen Entwickler und Security-Experten unmittelbar zusammenarbeiten, können auch Pair-Programming Modelle effektiv dazu beitragen, Fehler und Sicherheitslücken bereits im Entwicklungsprozess zu minimieren.

5.

Die Zeit vom Issue zur Behebung minimieren.

Je höher der Feature-Druck und je kürzer die Release-zyklen, desto schwieriger ist es für Entwicklungsteams Security-relevante Issues nachzuhalten. Deshalb ist es umso wichtiger, die Zeit vom Deployment bis zur Rück-meldung an das Entwicklungsteam möglichst kurz zu halten und zusätzliche Schleifen über das Security-Team zu vermeiden. Security-Tools mit integrierter Rückmel-dung in Echtzeit, wie beispielsweise fördern die unmit-telbare Behebung von Security-relevanten Issues dort, wo sie entstehen.

6.

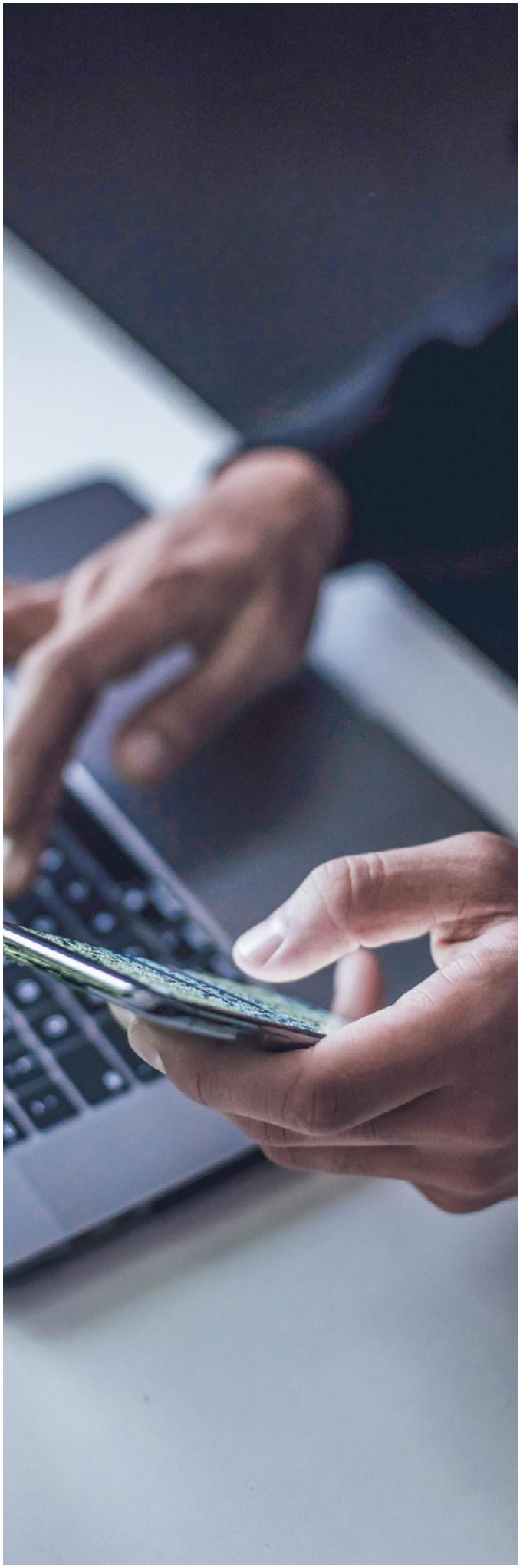
Einen Single-Point-of-Truth schaffen.

Neben unmittelbarem Feedback ist team-übergreifende Transparenz über die vorhandenen Security-Isses eine weitere wichtige Voraussetzung für erfolgreiche DevSe-cOps Organisationen. Über einen Single Point of Truth, etwa in Form eines Vulnerability Management Systems, sollten dabei alle am SDLC beteiligten Personen direkten Zugriff auf die identifizierten Security Issues und den Stand der Bearbeitung haben.

7.

Bedrohungsanalysen als Standardschritt einführen.

Eine weitere wirksame Maßnahme, um die Zusammen-arbeit an Security-relevanten Fragen zu fördern und die Applikationssicherheit insgesamt zu erhöhen, besteht darin, bei allen neuen Anwendungen oder größeren Fea-tures standardmäßig eine Bedrohungsanalyse zu Beginn des Entwicklungsprozesses durchzuführen. Durch Ein-bindung aller DevSecOps-Funktionen in dieser initialen Phase ist sichergestellt, dass Security-Aspekte von Beginn an angemessen berücksichtigt werden.



8.

Security Champions in den Entwicklungsteams installieren.

Gewohnte Prozesse zu überwinden und neue DevSecOps Prinzipien im Alltag zu adaptieren, erfordert Zeit und intensive Arbeit. Damit Entwicklungsteams in dieser schwierigen Transitionsphase nicht auf sich allein gestellt sind, empfiehlt sich der Einsatz von sogenannten Security Champions.

Dabei handelt es sich um Mitarbeiter*innen aus dem Entwicklungsteam, die ein spezielles Security-Trainingsprogramm durchlaufen haben. Als Ansprechpartner und Multiplikatoren für Security-bezogene Fragen im Team sorgen sie dafür, dass Applikationssicherheit trotz Zeitknappheit und Ergebnisdruck eine Top-Priorität im Entwicklungsprozess bleibt. Daneben sind sie für die Koordinierung, Verfolgung und Meldung von Sicherheitsproblemen für das Projekt verantwortlich und helfen dabei, Entscheidungen darüber zu treffen, wann das Sicherheitsteam eingeschaltet werden muss. Außerdem fördern sie den Team-übergreifenden Erfahrungs- und Wissensaustausch, indem sie regelmäßig mit anderen Security-Champions in Kontakt stehen.

9.

Security-Experten befähigen.

Neben dem Security-Champion im Entwicklungsteam hat es sich in erfolgreichen DevSecOps Organisationen bewährt, entsprechende Counterparts im Security Team zu installieren, die als dedizierte Ansprechpartner für Rückfragen aus den Entwicklungsteams zur Verfügung stehen und – durch entsprechenden Background oder Schulungen – die gleiche Sprache sprechen wie ihre Dev-Kolleg*innen und das nötige Knowhow mitbringen, um im Entwicklungsprozess auftretende Security-Issues umfassend verstehen, bewerten und Lösungswege aufzeigen zu können.

10.

Scrum Master stärken.

Indem sie dabei helfen, die Team-internen Abläufe, Meetings und Zusammenarbeitsmodelle an die DevSecOps-Praxis anzupassen, nimmt die Funktion des Scrum Masters eine wesentliche Rolle bei der Transformation von agilen Teams in Richtung einer agilen DevSecOps Organisation ein. Unternehmen sollten dem Rechnung tragen, indem Sie diese Rolle stärken und mit dem nötigen Wissen ausstatten, um diesen Übergang kompetent begleiten zu können.

Bereit für Ihre DevSecOps Transformation?

Mit unserem praxiserprobten Vorgehensmodell weisen wir Ihnen den Weg zu einer starken DevSecOps Organisation:

Phase 1:
Bestandsaufnahme

Wir ermitteln den Ist-Zustand Ihrer DevOps-Organisation. Durch Interviews mit Security-Verantwortlichen und Vor-Ort-Workshops im Entwicklerteam identifizieren wir Schwachstellen in der Zusammenarbeit und leiten erste Maßnahmen ab.

(voraussichtlicher Zeitaufwand: 1–2 Tage)

↓

Phase 2:
Awareness & Befähigung

In interdisziplinären DevSecOps Trainings und Schulungen schaffen wir ein Bewusstsein für Security als gemeinsame Verantwortung aller Beteiligen am Softwareentwicklungsprozess.

(voraussichtlicher Zeitaufwand: 2–3 Tage)

↓

Phase 3:
DevSecOps-Praktiken

Wir begleiten ein Pilot-Team bei der Einführung von DevSecOps-Praktiken und unterstützen bei der Integration von Security-bezogenen Aufgaben in den Entwicklungsalltag.

(voraussichtlicher Zeitaufwand 5–10 Tage)

↓

Phase 4:
Skalieren

Wir nutzen die Erkenntnisse aus der Pilotphase, um das DevSecOps-Zusammenarbeitsmodell auf weitere Teams innerhalb der Organisation auszurollen und diese zu befähigen.

(voraussichtlicher Zeitaufwand 2–X Tage)

Interesse?

Vereinbaren Sie jetzt einen unverbindlichen Beratungstermin zur Klärung Ihrer Anforderungen.

→ Jetzt Termin vereinbaren

Ihr Ansprechpartner:

Jan Girlich
Lead Security Advisor