Softwareentwicklung mit Security

DevSecOps – auch eine kulturelle Veränderung?

Softwareentwicklung ist ein komplexes Thema. Nicht immer steht dabei die Security im Zentrum. Hier setzt der DevSecOps-Ansatz an.

→ VON NIKLAUS MANSER

DER AUTOR

Niklaus Manser spezialisiert sich bei der Swiss Infosec AG seit 2017 im IT Security Consulting. Seit 2020 leitet er den Fachbereich IT Security, ist seit 2022 Mitglied der Geschäftsleitung und seit 2023 Executive Partner der Swiss Infosec AG.

Die Entwicklung von Software ist vielschichtig. Es gibt zahlreiche Programmiersprachen, Frameworks, Design Patterns, Methoden und Praktiken. Wo es viele Möglichkeiten gibt, gibt es auch viele Möglichkeiten für Fehler und Sicherheitsdefizite. 2023 wurden 30'000 neue Schwachstellen bzw. CVE (Common Vulnerability Enumeration, https://cve.org)-Nummern publiziert, was täglich rund 80 neuen Verwundbarkeiten entspricht. Tendenz steigend: 2024 waren es bis Anfang November ebenfalls bereits 30'000. Der Umgang mit Verwundbarkeiten stellt für Softwareproduzentinnen und -nutzerinnen gleichermassen eine Bedrohung dar. Unautorisierte Zugriffe und Data Breaches, die Kompromittierung von unternehmensinternen Netzwerken oder Denial-of-Service-Angriffe können die Folge sein.

WO BLEIBT DIE SECURITY?

Moderne Softwareentwicklungsprojekte erfolgen zumeist nach dem agilen Modell. In der klassischen Rollenverteilung erzeugen Softwareentwickler Programmcode und implementieren Features, Softwaretester prüfen, ob die Anforderungen korrekt umgesetzt werden und Plattform-spezialisten sorgen dafür, dass die Applikationsinfrastruktur einsatzfähig ist.

Die Sicherheit des entwickelten Softwareprodukts hat in vielen Fällen kurze Beine. Funktionale Erweiterungen stehen im Vordergrund der Entwicklungsvorhaben: Nur selten ist Security ein primäres Verkaufsargument, zu oft wird Security als reiner Kostenpunkt wahrgenommen. Aus Perspektive der Eigentümer der Software ist Security somit zumeist eher notwendiges Übel als eine sinnvolle Investition. Dieses Phänomen wird verstärkt durch den Umstand, dass die Verantwortung für Security gerade bei beauftragter Softwareentwicklung aufseiten der Eigentümer nicht ausreichend wahrgenommen wird: «Wir kennen uns ja nicht aus, wir erwarten, dass sich der beauftragte Entwickler darum kümmert.» Naja. Den beauftragten Softwareentwicklungsunternehmen fehlen handkehrum teilweise noch organisatorische Strukturen sowie geeignete Awareness- und Ausbildungsprogramme.

In der Konsequenz wird Security somit häufig erst zu einem späten Zeitpunkt berücksichtigt, oftmals mit einem Penetration Test kurz vor Go-Live. Die Behebung von Schwachstellen, die durch einen Penetration Test identifiziert werden, kann unter anderem zu unschönen Flickarbeiten führen. Probleme werden zudem tendenziell nicht auf einer grundlegenden Designebene angepackt, auch wenn die zentralen Probleme teilweise auf der Designebene bestehen.

«Konsequenter Einbezug der Security in die Softwareentwicklung bringt einen Mehrwert.»

Niklaus Manser

DEVSECOPS ALS PROAKTIVER ANSATZ

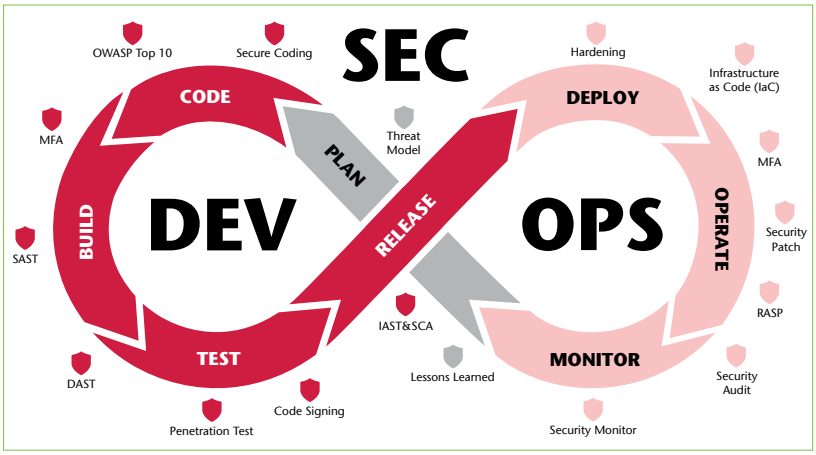
Anstatt Sicherheitsbedenken bei Softwareprodukten reaktiv zu adressieren, setzen mehr und mehr Organisationen auf proaktive Ansätze. Mittels DevSecOps wird die Security in alle Abschnitte des Software-Lebenszyklus eingebracht. Die Etablierung von DevSecOps ist nicht trivial, da oftmals auch kulturelle Veränderungen nötig sind. Umso wichtiger ist es, bei der Realisierung eines DevSecOps-Ansatzes nach be-

Bild: Swiss Infosec AG

stimmten Punkten vorzugehen.

10-Punkte-Check zur Realisierung eines DevSecOps-Ansatzes:

1. Agile Sicherheitsorganisation: Security Champions schaffen eine Kultur für Cybersicherheit und sorgen dafür, dass Sicherheit tief im Entwicklungsprozess verankert wird. Die Rolle des Security Champion sollte mit genügenden zeitlichen Ressourcen ausgestattet werden.
2. Automatisierte Security-Tests: Die DevOps-Pipeline ist mit automatisierten Security-Tests und Quality Gateways zu ergänzen, mit denen der Code statisch oder idealerweise dynamisch auf Verwundbarkeiten überprüft wird.
3. Sicherheitsbezogene Ausbildung der Mitarbeitenden: Eine kontinuierliche Ausbildung der Mitarbeitenden sorgt für ein grundlegendes Sicherheitsverständnis. Unterschiedliche Methoden zur Sensibilisierung begünstigen ein Erreichen möglichst aller Mitarbeitenden.
4. Sichere Entwicklungs- und Build-Umgebungen: Damit die Sicherheit des Programmcodes gewährleistet ist, müssen grundlegende Cyber-«Hygieneregeln» eingehalten werden: Alle Systeme sind regelmässig oder automatisiert mit Sicherheitspatches zu aktualisieren.
5. Sichere Applikationsplattformen und Laufzeitumgebungen: Zentraler Bestandteil des sicheren Betriebs einer Applikation sind sichere Applikationsplattformen und Laufzeitumgebungen. Um diese zu erhalten, müssen Herstellerempfehlungen oder Best Practice-Empfehlungen konsequent umgesetzt werden. Egal, ob klassisches Serversystem oder Containerplattform: Diese Umgebungen sind nicht secure-by-default, sondern müssen explizit sicher konfiguriert werden.
6. Secure Coding Principles: Eine verbindliche Vorgabe zu sicheren Entwicklungsprinzipien ist zu dokumentieren und zu schulen. Die Erstellung, Schulung, Operationalisierung und fortlaufende Weiterentwicklung dieser Vorgabe übernimmt idealerweise der eingesetzte Security Champion (s. Punkt 1).
7. Threat Modeling: Ein Bedrohungsmodell ermöglicht einen Blick auf sicherheitsbezogene Fragestellungen rund um das Entwicklungsprojekt. Threat Modeling Workshops helfen, bei Projektinitialisierung und der kontinuierlichen Weiterentwicklung des Produkts, reale Bedrohungen zu erkennen und mit gezielten Massnahmen entgegenzuwirken.



Review der DevSecOps Pipeline und zugehöriger Prozesse

8. Abuse Cases: Zur Entwicklung einer sicheren Applikation ist es wichtig, diejenigen Angriffe, gegen die die Applikation gerüstet sein muss, zu identifizieren. Hier helfen Abuse Cases, die die Diskussion und Implementation elementarer und explizit geforderter Sicherheitsfeatures ermöglichen.
9. Loggen von Security Events: Das Aufzeichnen sicherheitsrelevanter Vorfälle innerhalb der Applikation in Logfiles ermöglicht ein sicherheitsbezogenes Monitoring, zeitnahe Auswertungen oder auch die Einrichtung von Alerts.
10. Penetration Testing: Penetration Tests helfen, die Umsetzung von Sicherheitsanforderungen und die Einhaltung von Best Practices im Bereich der sicheren Softwareentwicklung zu überprüfen. Sinnvollerweise wird hierbei mit unterschiedlichen Anbietern abgewechselt, damit die Software von unterschiedlichen Anbietern mit potenziell unterschiedlichen Fokuspunkten untersucht wird und die Sicherheit des Produkts möglichst breit bestätigt wird.

DEVSECOPS REVIEW

Mittels angemessener Prozesse und der Nutzung zeitgemässer Tools und Settings bei Entwicklung, Deployment und Operations kann die Sicherheit von Softwareprodukten nachhaltig aufrechterhalten werden. Eine Überprüfung der Security über den ganzen Lifecycle ist entscheidend. Die zuständigen Lead-Entwickler werden dabei idealerweise von externen Spezialisten unterstützt.

Ja, die Etablierung von DevSecOps verlangt eine kulturelle Veränderung. Der konsequente Einbezug der Security in allen Abschnitten des Software-Lifecycles bringt jedoch einen entscheidenden Mehrwert, der diese Veränderung mehr als rechtfertigt. ←

IMPRESSUM

Das offizielle Publikationsorgan des VIW

Herausgeber:
VIW – Wirtschaftsinformatik Schweiz
→ www.viw.ch

Podcasts, Videos & mehr:
→ https://my.viw.ch

VIW in den sozialen Netzwerken: