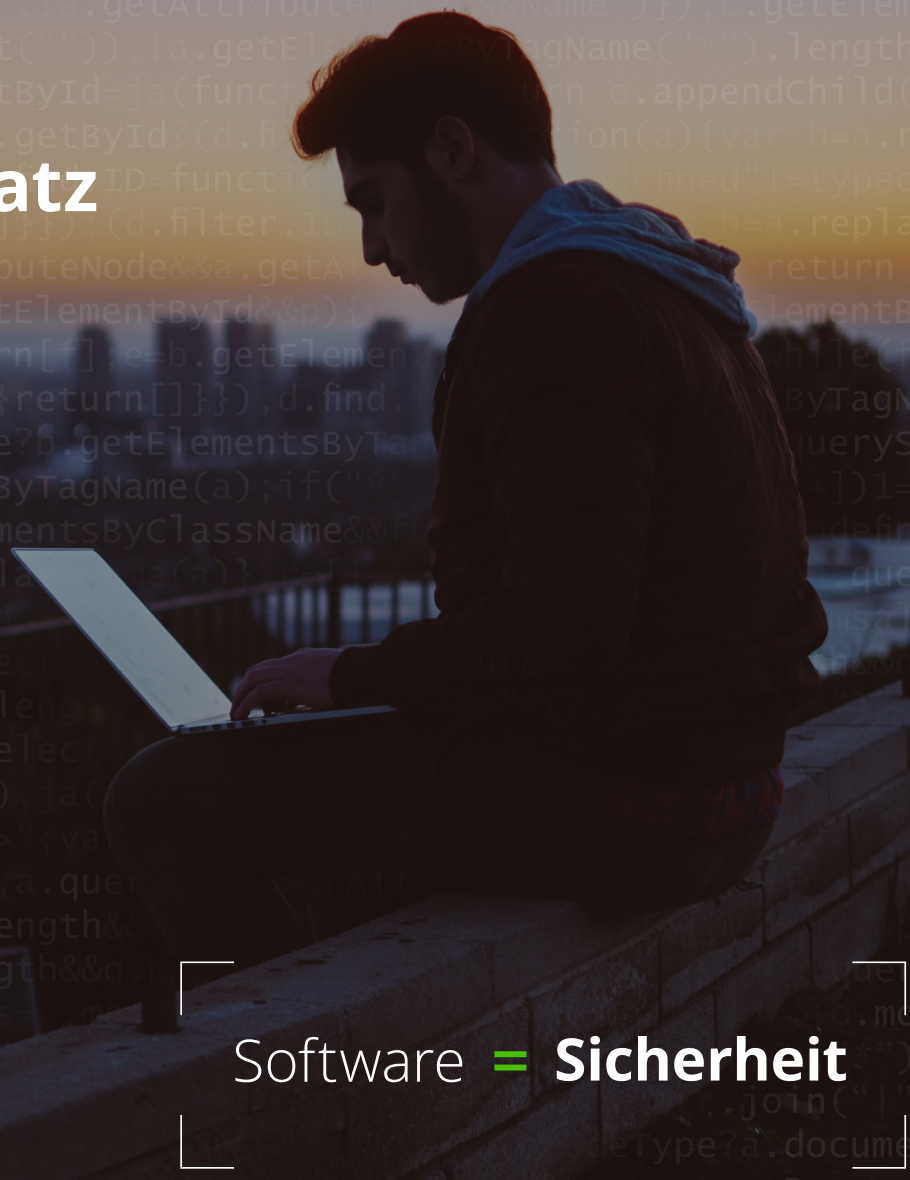
Checkmarx

Ein ganzheitlicher Ansatz für die Einbettung von Security in DevOps

Ein Best-Practices-Leitfaden



Software = **Sicherheit**

Inhalt

Einführung

Unternehmen nutzen DevOps als Entwicklungs- und Betriebsmodell, um die Softwarebereitstellung und -implementierung zu automatisieren. Führende Security- und Entwicklungsexperten stellen dabei fest, dass herkömmliche Software-Security-Konzepte nicht an dieses neue Modell angepasst werden können. Die Sicherheit wird daher oft als Hürde für DevOps angesehen. Doch das muss nicht so sein. Mit den richtigen Tools, Services und Prozessen lässt sich Sicherheit nahtlos in eine DevOps-Umgebung einfügen. In diesem Leitfaden stellen wir die beste Strategie zur Einbettung automatisierter Security Scans in DevOps vor.

DevOps-Grundlagen

Die DevOps-Bewegung schuf eine Kultur und Atmosphäre, in der die Entwicklung, das Testen und die Bereitstellung von Software schneller, regelmäßiger und zuverlässiger erfolgen sollte. Dieser Kulturwandel führte zur Einführung von Continuous Integration (CI) und Continuous Delivery (CD), die heute die Eckpfeiler von DevOps darstellen (siehe Abb. 1).

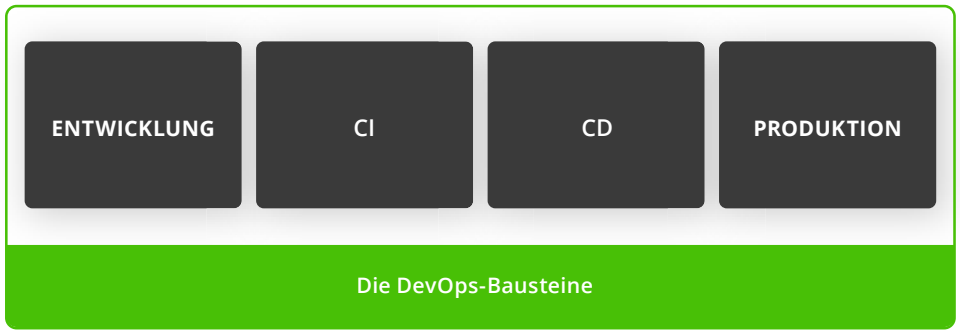


Abb. 1

Im Grunde genommen geht es bei DevOps um Prozesse, Verbindungen, Automatisierung und Tools während der gesamten Entwicklungs-, Test- und Bereitstellungsphase. Noch wichtiger ist jedoch, dass es bei DevOps um die „Automatisierung der Tools“ sowie um die verschiedenen Tools geht, die bei der Softwareentwicklung zum Einsatz kommen. Was bei den Grundlagen von DevOps bislang jedoch noch nicht berücksichtigt wurde, ist die Frage, an welcher Stelle der Softwareentwicklung die Software-Sicherheit ihren Platz haben soll. Für Unternehmen, die sicherere Software entwickeln möchten, ist der Einsatz mehrerer Application-Security-Testing (AST)-Lösungen innerhalb von DevOps unerlässlich, um die Schwachstellen in unkompiliertem Code, Code in laufenden Anwendungen und Open-Source-Komponenten zu beheben. Lassen Sie uns nun der Frage nachgehen, warum das so ist, und einen Blick auf die derzeit vorhandenen AST-Lösungen werfen.

Unterschiedliche AST-Lösungen finden unterschiedliche Software-Schwachstellen

Lösungen für Static Application Security Testing (SAST) werden verwendet, um unkompilierten Code während der Softwareentwicklung in DevOps-Prozessen inkrementell zu scannen (testen). Der Code befindet sich noch in unkompiliertem Zustand, und mit statischen Tests sollen Fehler wie SQL-Injektion schneller aufgespürt werden. SAST-Lösungen eignen sich hervorragend, um auf Code-Ebene aufzuzeigen, wo und wie Schwachstellen im Quellcode behoben werden können. SAST passt hervorragend zu integrierten Entwicklungsumgebungen (IDEs), Issue-Trackern und Build-Tools zur Unterstützung von CI/CD-Workflows. SAST fügt sich nahtlos in DevOps ein, da es keine nennenswerten Verzögerungen verursacht.

Lösungen für das **Interactive Application Security Testing (IAST)** erkennen Konfigurationsfehler im Zuge der etablierten Funktionstest in der Laufzeitumgebung – also vor dem Roll-out der Anwendung. Es wäre unklug, davon auszugehen, dass Anwendungen nach der Entwicklungsphase frei von Schwachstellen sind oder dass Code in laufenden Anwendungen nicht mehr getestet werden muss. IAST erkennt, wie alle Teile einer Anwendung zusammenwirken und zur Laufzeit funktionieren. Dadurch kann es Schwachstellen in laufenden Anwendungen aufdecken, die von Angreifern ausgenutzt werden könnten.

IAST passt optimal zu DevOps, da es keine Verzögerungen verursacht, die über den veranschlagten Zeitraum hinausgehen, der für die Durchführung von Funktionstests erforderlich ist.

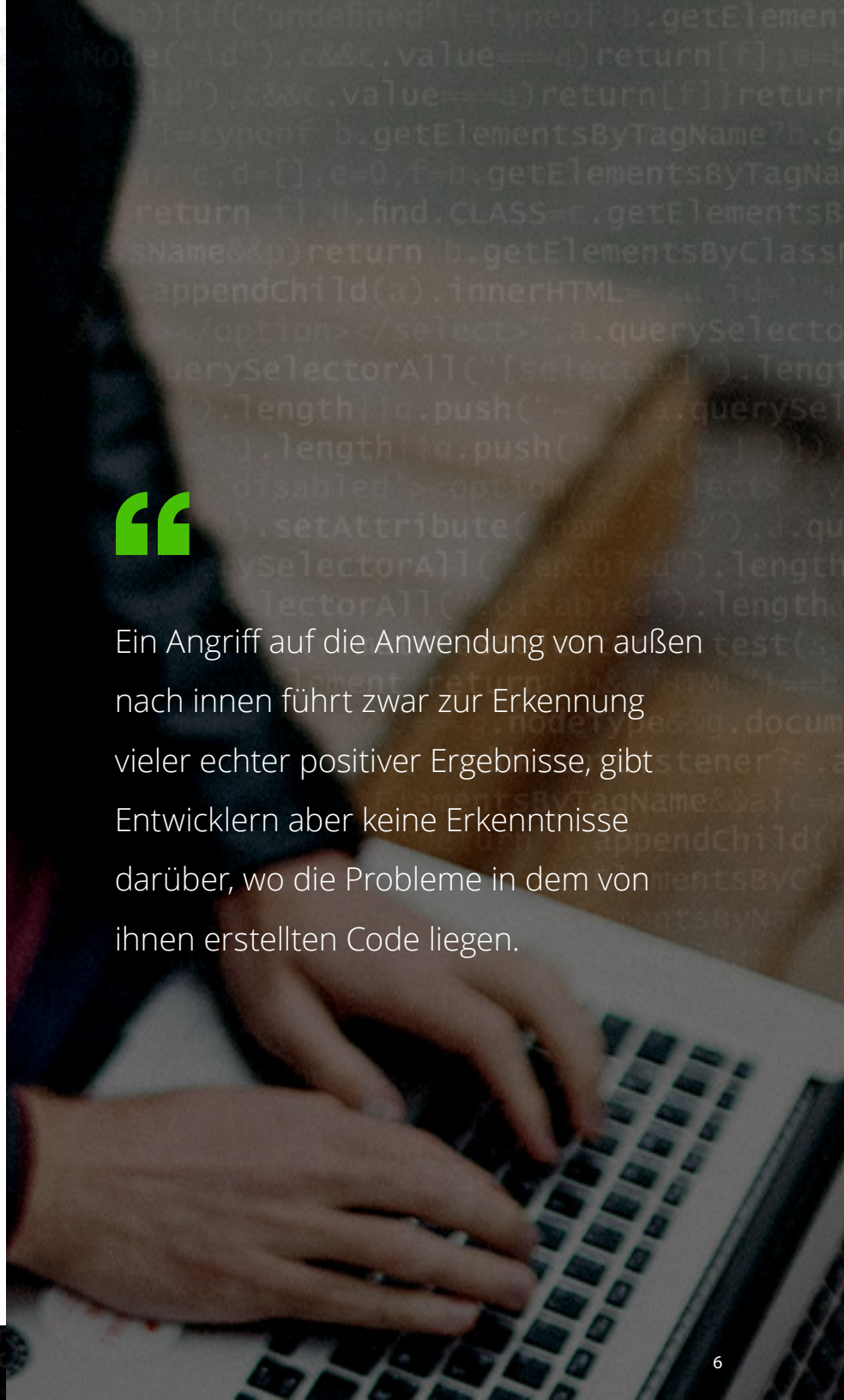
SCA-Lösungen (**Software Composition Analysis**) verschaffen Entwickler-, Security- und Operations-Teams wichtige Einblicke, mit denen sie den Risiken von Open Source Software in den von ihnen erstellten, eingesetzten und gewarteten Anwendungen effizient begegnen. Die Analyse und Administration der verwendeten Open-Source-Komponenten stellt sicher, dass anfällige Komponenten entfernt oder ersetzt werden, bevor sie zum Problem werden. SAST fügt sich nahtlos in DevOps ein, da es keine nennenswerten Verzögerungen verursacht.

DAST-Tools (**Dynamic Application Security Testing**) erkennen Schwachstellen in laufenden Anwendungen, indem sie die Anwendung von außen angreifen. DAST beschränkt sich auf reflektierende Arten von Schwachstellen, da DAST-Lösungen im Wesentlichen nicht erkennen, was innerhalb einer Anwendung geschieht. DAST-Testergebnisse liefern auf Codeebene keine Hinweise darauf, wo sich Software-Schwachstellen befinden, was Entwicklern das Beheben erkannter Schwachstellen erschwert. DAST-Tools können die von DevOps geforderten schnellen Bearbeitungszeiten nicht leisten.

Auch **Pentesting-Lösungen** sollten hier angesprochen werden, auch wenn diese Testverfahren im Allgemeinen nicht zu den AST-Lösungen zählen. Viele Unternehmen nutzen solche Lösungen, um sicherzustellen, dass ihre Anwendungen frei von Schwachstellen sind, aber dies geschieht sehr spät im SDLC, häufig erst nach dem Einsatz von DAST-Lösungen. Sie werden häufig einfach als Routineübung gesehen und die Ergebnisse dieser Tests können Entwicklern nur sehr wenig Anhaltspunkte dafür geben, welche Probleme zu beheben sind.

Im Rahmen der Security-Gate-Mentalität ist der zeitliche Abstand zwischen den Pentests oft sehr lang, und die Tests werden während der Entwicklungsphase nicht wiederholt, sondern erst durchgeführt, nachdem die Anwendungen oft schon live oder kurz davor sind. Ein Angriff auf die Anwendung von außen nach innen führt zwar zur Erkennung vieler echter positiver Ergebnisse, gibt Entwicklern aber keine Erkenntnisse darüber, wo die Probleme in dem von ihnen erstellten Code liegen.

Nach diesem Überblick über die verschiedenen AST-Lösungen möchten wir als nächstes untersuchen, wo AST-Lösungen in DevOps eingebettet werden können.



“Ein Angriff auf die Anwendung von außen nach innen führt zwar zur Erkennung vieler echter positiver Ergebnisse, gibt Entwicklern aber keine Erkenntnisse darüber, wo die Probleme in dem von ihnen erstellten Code liegen.

Einbettung von AST-Lösungen in DevOps

In Abbildung 2 sehen Sie die gerade vorgestellten AST-Lösungen mit dem Dev-Bereich auf der linken und dem Ops-Bereich auf der rechten Seite. Diese Abbildung veranschaulicht, in welchen Phasen des DevOps-Modells die verschiedenen AST-Lösungen am besten eingesetzt werden.

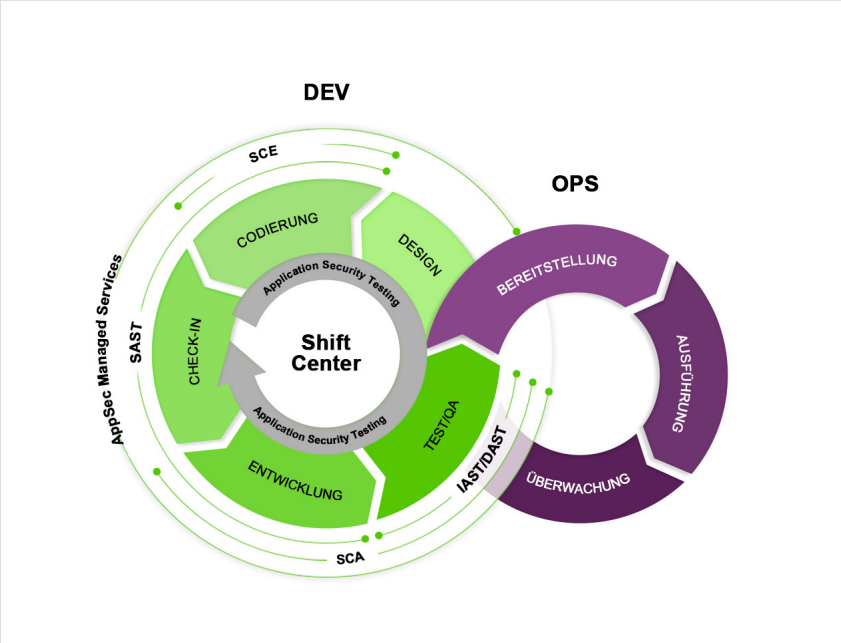


Abb. 2

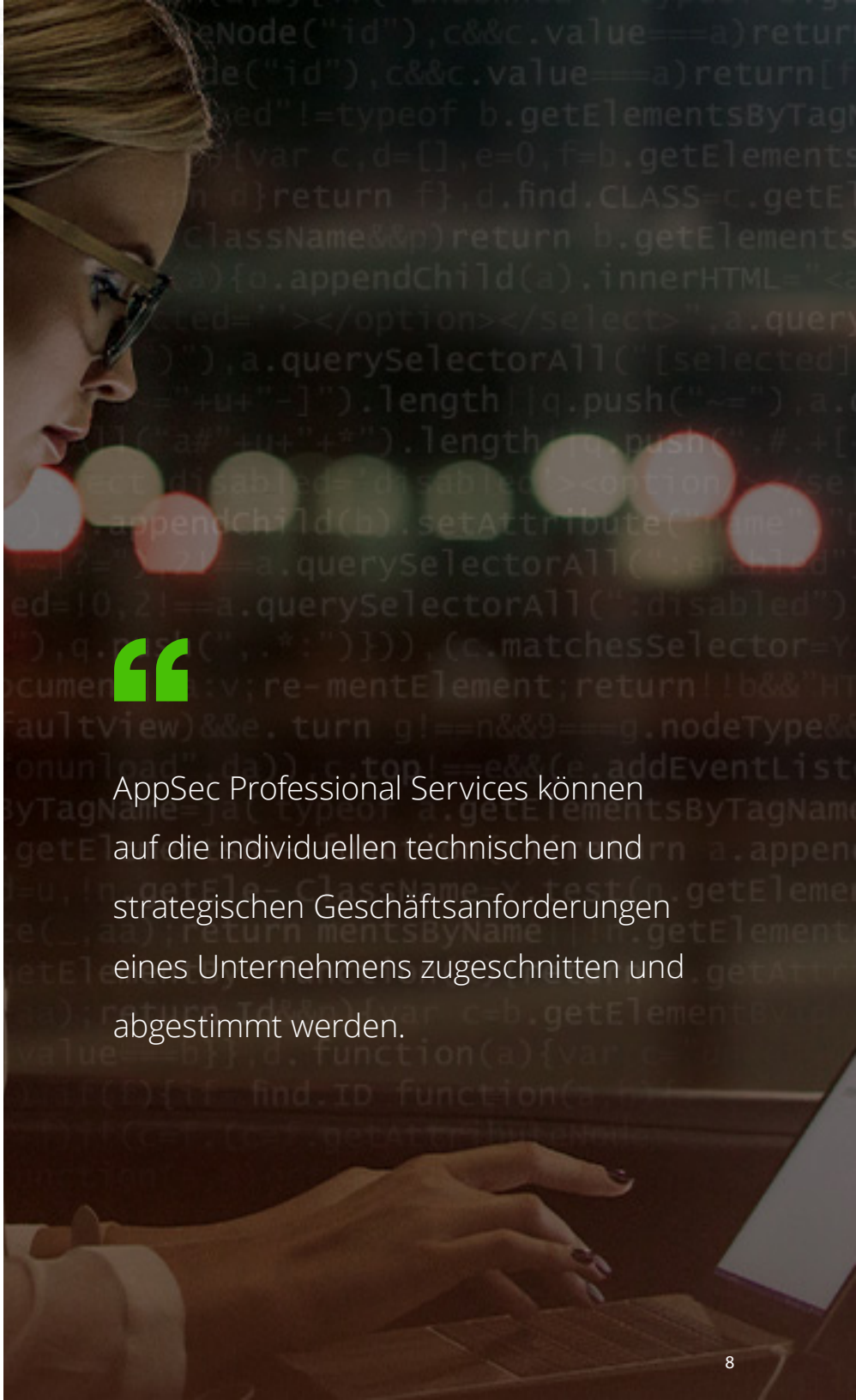
Wie wir in Abbildung 2 sehen können, müssen AST-Lösungen in die Dev-Phasen eingebettet werden und zeigen dann auch in Ops Wirkung. Die passenden AST-Lösungen für jede Dev-Phase werden durch die dünnen grünen Linien um die verschiedenen Phasen herum hervorgehoben. Die folgenden Punkte beschreiben, in welche Phase(n) AST-Lösungen am besten in Dev passen:

- SAST wirkt in den Phasen CODIERUNG, CHECK-IN, ENTWICKLUNG und TEST/QA
- IAST wirkt in der Phase TEST/QA
- SCA wirkt in den Phasen ENTWICKLUNG und TEST/QA
- DAST wirkt in der Phase TEST/QA (jedoch wie bereits erwähnt mit Einschränkungen)
- Pentesting wirkt in erster Linie außerhalb der Softwareentwicklung

AppSec-Services haben wir zwar noch nicht angesprochen, aber in Abbildung 2 sind sie als die äußere grüne Linie zu sehen. Sie erlauben es Unternehmen, ihre Application Security in Teilen an Dritte auszulagern, die bei der Einführung und Implementierung von Anwendungssicherheitsprozessen, sicheren Codierungspraktiken, Sicherheitstests und der Behebung von Schwachstellen unterstützen.
Dazu können auch Kick-Start-Programme gehören, die bei der Implementierung und Integration von Sicherheitslösungen in den SDLC, bei der Analyse von Sicherheitsergebnissen, bei der Reduzierung von Fehlalarmen, bei Schulung und Threat-Modellierung und vielem mehr helfen sollen. Unternehmen, denen in der Anfangsphase von DevSecOps interne Ressourcen und Fachkenntnisse fehlen, können von diesen Services profitieren.

AppSec Professional Services können auf die individuellen technischen und strategischen Geschäftsanforderungen eines Unternehmens abgestimmt werden. Außerdem sind sie darauf ausgelegt, sichere DevOps-Initiativen zu optimieren und die Sicherheit von Anwendungen zu stärken, die Einführung und Implementierung zu beschleunigen, um Vorteile früher zu realisieren, und Testverfahren und -konfigurationen zu verfeinern.

In Abbildung 2 finden wir zudem den neuen Begriff SCE, der bisher noch nicht angesprochen wurde. SCE steht für Secure Coding Education (Schulung für sicheres Codieren) und fällt in die Codierungsphase von DevOps. Der nächste Abschnitt beleuchtet kurz SCE im Zusammenhang mit einem AppSec Awareness-Programm im Allgemeinen.



> AppSec Professional Services können auf die individuellen technischen und strategischen Geschäftsanforderungen eines Unternehmens zugeschnitten und abgestimmt werden.

Ein AppSec-Awareness-Programm als sinnvolle Investition

Auf dem Weg zum sicherheitsbewussten Entwickler ist nicht nur die Ausbildung oder Schulung in sicherer Codierung entscheidend. Es geht mehr um betriebliches Bewusstsein, bei der jeder im Unternehmen genau versteht, was Security bedeutet. Wenn Entwickler mit dem Codieren beginnen, sollten sie sich bereits umfassende Gedanken um die Sicherheit machen. AppSec-Teams sollten nicht länger die einzige Gruppe sein, die für die Sicherheit verantwortlich ist. Stattdessen müssen auch Entwickler für die Security verantwortlich sein. Erst dann werden sie in der Lage sein, sichereren Code zu schreiben. Wenn heutzutage alles auf Code umgestellt wird (z. B. Infrastruktur als Code), muss jeder die Sicherheit als oberstes Gebot haben – auch das Management, Entwickler, AppSec-Teams und sogar IT-Teams.

In der Realität sieht es jedoch so aus, dass ein beträchtlicher Prozentsatz der Entwickler kein Vertrauen in die Sicherheit ihrer eigenen Anwendungen hat oder nur wenig oder gar kein tieferes Wissen um Schwachstellen und wie sie entstehen. Diese Lücke besteht, weil Entwickler an der Geschwindigkeit und der Anzahl der funktionalen Fehler in ihrem Code gemessen werden, nicht an der Menge der von ihnen verursachten Sicherheitslücken.

Um diese Lücke zu schließen, verstehen Unternehmen zunehmend, dass sie ihren Entwicklern SCE (im Rahmen eines AppSec Awareness-Programms) zur Verfügung stellen müssen, das direkt in ihre IDEs integriert ist. Mithilfe von Just-in-time-Schulungslösungen, kontinuierlicher Kommunikation und spielerischen Elementen pflegen Sicherheitsmanager eine Kultur der Softwaresicherheit, die Entwickler dabei unterstützt, in ihrer täglichen Arbeit sicher zu denken und zu handeln. Entwickler, die sicher denken und handeln, können die Sicherheit ihrer Software messbar erhöhen, sich wiederholende Codierungsfehler reduzieren und die Anzahl der Software-Schwachstellen, die getestet und behoben werden müssen, deutlich verringern.

Im Vergleich dazu scheitern herkömmliche Schulungsmethoden für die sichere Codierung (z. B. Video-Tutorials, regelmäßige Schulungen im Klassenzimmer und obligatorische Online-Kurse) häufig, da sie zu einfach, kontextfremd und nicht interaktiv sind. Der Grundgedanke hinter AppSec Awareness ist es, das Sicherheitsbewusstsein von Entwicklern zu erhöhen. Das muss zu einem Unternehmensziel werden.

Da DevSecOps mehr bedeutet als nur die Einbettung von AST-Lösungen in DevOps-Prozesse, werden wir uns nun einige Schwerpunktbereiche bei der Automatisierung in DevOps betrachten.

Warum einer dedizierten Integrations- und Orchestrierungsebene eine Schlüsselrolle zukommt

Das Hinzufügen einer SDLC-Integrations- und Orchestrierungsebene zu den vorgenannten AST-Lösungen hilft, die Lösungen zu einer integrierten und benutzerfreundlichen Plattform zu machen, die Unternehmen eine ganzheitliche Sicht auf ihre Softwareschwachstellen bietet und die AST-Automatisierung vereinfacht. Dadurch können Unternehmen Sicherheitsrisiken in großem Maßstab leicht verfolgen, verwalten und beheben. Diese Ebene wird benötigt, um die End-to-End-Automatisierung und den Orchestrierungsfluss, vom Scan bis zum Ticketing, zu vereinfachen und zentral zu verwalten. Wenn Sie AST-Lösungen implementieren möchten, stellen Sie sicher, dass diese mit einer SDLC-Integrations- und Orchestrierungsebene als Teil der Lösung geliefert werden.

Die Automatisierung von AST-Lösungen innerhalb des DevOps-Toolings ist von höchster Bedeutung

Um DevSecOps zu erreichen, müssen Unternehmen AST-Lösungen automatisiert in alle DevOps integrieren, um manuelle Testverfahren zu eliminieren, die in der Vergangenheit möglicherweise zu Verzögerungen geführt haben. Diese AST-Lösungen müssen für Entwickler und Security-Teams so transparent wie möglich sein, damit die DevOps-Flexibilität nicht behindert wird. Automatisierung ist der Schlüssel zur Erfüllung regulatorischer Anforderungen und zum Management des Gesamtrisikos. Um dieses Ziel zu erreichen, müssen AST-Lösungen in der Lage sein, vollständig automatisiert zu werden, und zwar innerhalb der Tools, die oft bereits bei DevOps im Einsatz sind. Über die Automatisierung und die vorhandenen Tools hinaus geht der nächste Abschnitt auf die Aktivitäten ein, die erforderlich sind, um Sicherheitsrisiken in großem Maßstab zu managen und zu reduzieren. Dieses Thema werden wir uns als nächstes ansehen.

+ Wenn Sie AST-Lösungen implementieren möchten, stellen Sie sicher, dass diese mit einer SDLC-Integrations- und Orchestrierungsebene ausgestattet sind.

Umfassendes Management und Reduzierung von Sicherheitsrisiken

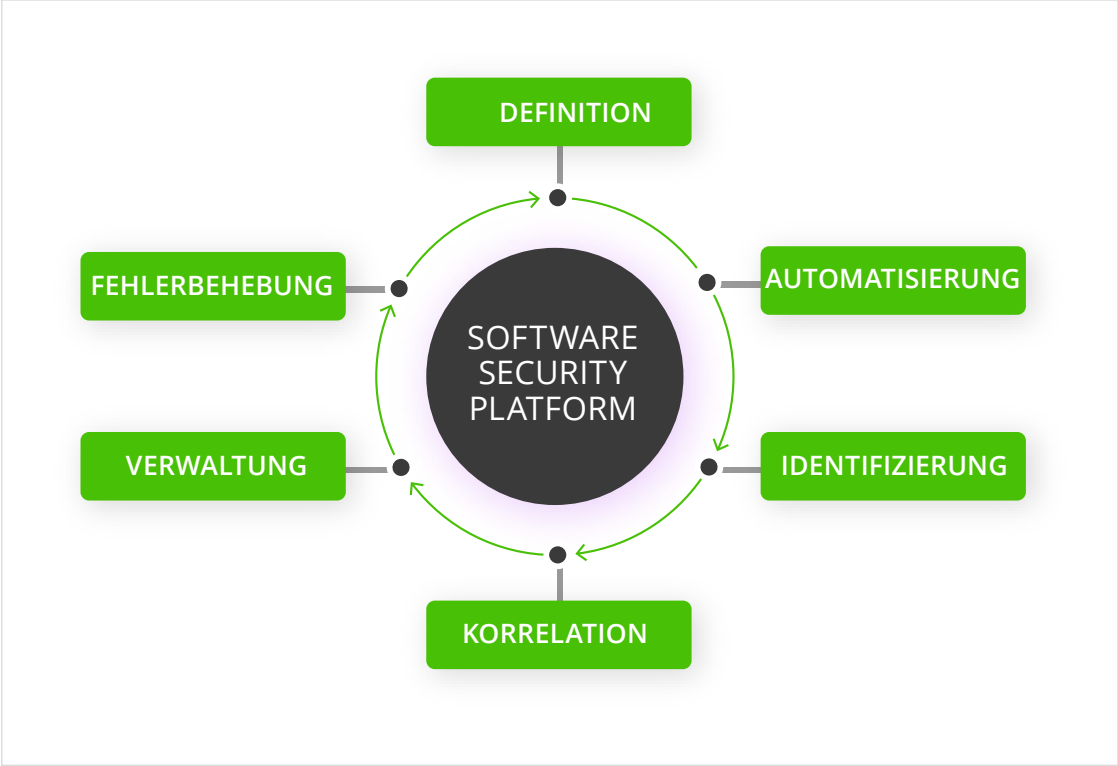


Abb. 3

Bei der Diskussion darüber, wo Sicherheit in DevOps eingebettet werden muss, um DevSecOps zu erreichen, gibt es mehrere Aspekte, die über das allgemeine Testen der Anwendungssicherheit hinausgehen, während andere direkt damit zusammenhängen. Abbildung 3 hebt die Aktivitäten hervor, die durchgeführt werden müssen, um Sicherheitsrisiken in großem Maßstab vollständig zu managen und zu reduzieren. Beginnen wir ganz oben und gehen im Uhrzeigersinn weiter und betrachten dann jedes einzelne dieser Konzepte.

Definition von Sicherheitsrichtlinien

Hier definieren Unternehmen ihre Richtlinien zur Anwendungssicherheit (AppSec) hinsichtlich der akzeptablen und nicht akzeptablen Risiken, die sie bereit sind, einzugehen. Anwendungen werden immer Schwachstellen aufweisen, und kein Unternehmen wird Schwachstellen und funktionale Fehler jemals komplett ausschalten können. Die Definition, welche Risiken akzeptabel und welche nicht akzeptabel sind, ist in dieser Phase zwingend erforderlich. Und es geht darum, zu bestimmen, welche Richtlinienverletzungen Unternehmen dazu veranlassen würde, einen Build rückgängig zu machen.

Die definierte Sicherheitsrichtlinie dient sozusagen als Vereinbarung zwischen AppSec-Teams und Entwicklern, damit beide Seiten verstehen, was von ihnen in Bezug auf die Sicherheit erwartet wird. Diese Richtlinie dient auch als Anleitung dafür, welche Schwachstellen als Ergebnis des Application-Security-Testings zuerst behoben werden sollten. Die Definition von Sicherheitsrichtlinien ist eng mit DevSecOps verbunden, und diese Richtlinien sind für die Messung des Gesamterfolgs Ihrer DevSecOps-Initiativen von entscheidender Bedeutung.

Automatisierung und Integration

Hier integrieren Unternehmen ihre SAST-, IAST- und SCA-Testlösungen in ihre Entwicklungsumgebungen, um die AST-Scans vollständig zu automatisieren. Ohne Automatisierung ist eine Skalierung nicht möglich. Jedes Unternehmen kann wählen, bis zu welchem Grad es seine Prozesse automatisieren möchte, da dies auf viele verschiedene Arten und Weisen geschehen kann. Aber letztendlich möchten Sie sicherstellen, dass Ihre Anwendungen auf einheitliche Weise gescannt werden. Am besten lässt sich das durch die Automatisierung der Scans innerhalb der Build-Umgebung, der Entwicklungsumgebung oder beidem erreichen.

Sie sollten zum Beispiel sicherstellen, dass Sie Ihre AST-Lösungen automatisieren, wenn die Builds in der Build-Umgebung laufen oder wenn Entwickler einen Code-Commit oder eine Pull-Anforderung ausführen. Im letzteren Fall findet diese Automatisierung früher in der Entwicklungsumgebung statt.

Wenn AST-Lösungen in der Codierungsphase automatisiert werden, nutzen Entwicklungsteams Self-Service-Lösungen, um Scans über auf Zusammenarbeit ausgerichtete Codierungsplattformen wie GitHub, Azure DevOps usw. zu automatisieren. Wenn AST-Lösungen während der Build/CI-Phase automatisiert werden, werden CI-Plugins verwendet, um die Scans zu automatisieren. Schließlich schließt die Integration von Ticketingsystemen den Kreis, indem den Entwicklern die relevanten Erkenntnisse aus ihren Scans in Echtzeit zur Verfügung gestellt werden.

Schwachstellen erkennen

Wenn Sie die AST-Lösung auf diese Weise integriert und automatisiert haben, werden in diesem Schritt die AST-Scans durchgeführt. Durch den automatisierten Einsatz von SAST, IAST und SCA sind diese Lösungen vollständig in der Lage, alle Arten von Schwachstellen in Ihren Softwareanwendungen zu erkennen. Das kann Schwachstellen in folgenden Bereichen umfassen:

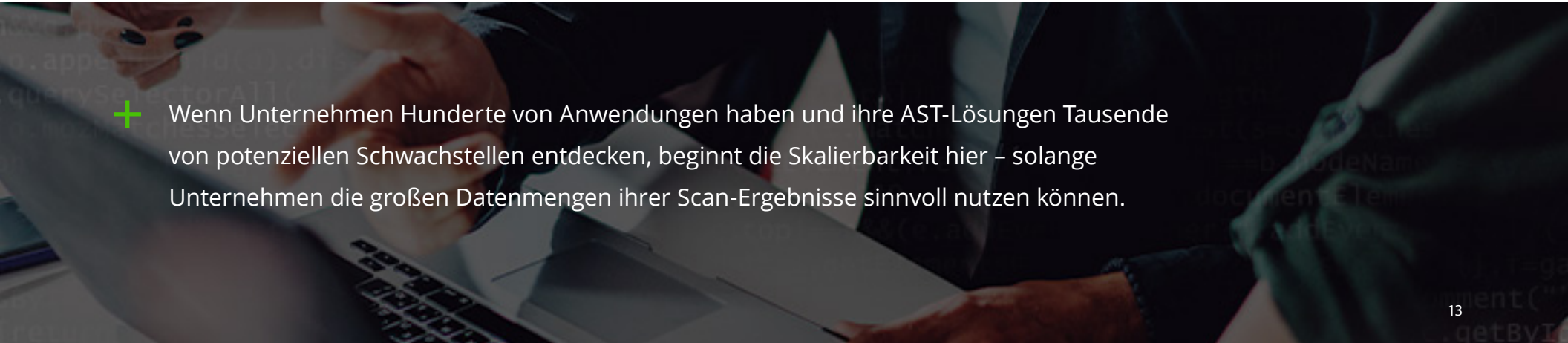
- Nicht kompilierter Code
- Code in der Laufzeitumgebung
- Open-Source-Komponenten

Es geht darum, Codierungsfehler (die Schwachstellen verursachen können) so früh wie möglich zu erkennen, ohne Entwicklung, Bereitstellung und Einsatz von Software-Anwendungen zu verlangsamen, sodass die Agilität von DevOps effektiv aufrechterhalten wird.

Ergebnisse korrelieren

Die Idee hinter der Korrelation besteht darin, das Vertrauensniveau und die Priorität der risikoreichen Ergebnisse (erkannte Schwachstellen) von AST-Lösungen zu erhöhen. Das ist insbesondere dann der Fall, wenn Sie die gleichen Ergebnisse von verschiedenen Scan-Lösungen korrelieren können. Wenn Sie bei den statischen Tests feststellen, dass eine Anwendung für SQL-Injections anfällig ist und IAST denselben Befund während interaktiver Tests bestätigt, können Sie, wenn Sie beide Befunde zusammen korrelieren, eher darauf vertrauen, dass der Befund wirklich positiv ist.

In diesem Fall ist die Wahrscheinlichkeit, dass ein Befund reproduzierbar ist, extrem hoch. Dann muss die Schwachstelle eher früher als später behoben werden. Wenn Unternehmen Hunderte von Anwendungen haben und ihre AST-Lösungen Tausende von potenziellen Schwachstellen aufdecken, beginnt die Skalierbarkeit hier – solange Unternehmen die großen Datenmengen ihrer Scan-Ergebnisse sinnvoll nutzen können.



+ Wenn Unternehmen Hunderte von Anwendungen haben und ihre AST-Lösungen Tausende von potenziellen Schwachstellen entdecken, beginnt die Skalierbarkeit hier – solange Unternehmen die großen Datenmengen ihrer Scan-Ergebnisse sinnvoll nutzen können.

Schwachstellen beheben

Die Behebung von Schwachstellen hat zwei Aspekte. Zum einem geht es darum, was zu beheben ist, und zum anderen darum, wie es sich beheben lässt. In Bezug auf die Frage, was behoben werden sollte, ist kein Entwickler in der Lage mit Tausenden erkannten Schwachstellen umzugehen. Sie müssen sicherstellen, dass Sie all diese Funde so priorisieren können, dass ein Entwickler sie bearbeiten kann.

Wenn alle erkannten Schwachstellen an die Entwickler gehen (ähnlich wie sie alle Defekte aus ihrem aktuellen Defektmanagement-Tool erhalten), trägt dies dazu bei, die Behebungszeit und die Einführung von AST zu beschleunigen. Wenn die Entwickler darüber hinaus Best Practices zur Behebung einer bestimmten Schwachstelle erhalten, können sie schnell und effizient die größte Anzahl von Schwachstellen beheben. Einfach ausgedrückt, müssen sich die Entwickler auf das Wesentliche konzentrieren können und diejenigen Schwachstellen beheben, die das größte Risiko zuerst exponentiell reduzieren würden.

Wenn ein Team einmal entschieden hat, was zu beheben ist, oft auf der Grundlage der eingangs unter „Definition von Sicherheitsrichtlinien“ dargelegten Richtlinie, ist die nächste Entscheidung, wie die Schwachstellen behoben werden sollen. Hier kann Secure Coding Education (SCE) eine große Hilfe sein. SCE kann Entwicklern mit einem speziell auf diese Art von Schwachstelle zugeschnittenen Tutorial beibringen, wie man eine bestimmte Schwachstelle behebt. Das funktioniert besonders gut, wenn SCE direkt in die IDEs der Entwickler integriert ist.

Management und Monitoring

Management und Monitoring sind die Bereiche, in denen Unternehmen die Leistungskennzahlen oder KPIs ihres Application-Security-Programms verfolgen. Auf diese Weise können Unternehmen erkennen, ob im Laufe der Zeit die Anzahl der Schwachstellen abnimmt, die Rate der Einführung neuer Schwachstellen abnimmt und die Rate schwerer Schwachstellen ebenfalls abnimmt. Es gibt alle Arten von KPIs, die Unternehmen verwenden, um festzustellen, ob ihr Security-Programm wirksam ist.

Ein Teil dieses KPI-Zyklus besteht darin zu wissen, welche Bereiche verbesserungsbedürftig sind und welche nicht. So können die Teams feststellen, ob die Sicherheitsrichtlinien eingehalten werden oder nicht, oder ob die Entwickler mehr Schulung, Tools oder Anreize benötigen. Die Teams können zum Beispiel auch feststellen, ob die vorhandene Richtlinie weiter verfeinert werden muss. All diese Aktivitäten ermöglichen es Unternehmen, den aktuellen Status ihres Programms und den Grad der Verbesserungen zu messen. Sie schafft auch eine nicht endende Feedback-Schleife zurück zu Ihren Entwicklern. Es geht darum, eine kontinuierliche Verbesserung im gesamten DevSecOps-Ökosystem zu ermöglichen.

Fazit

Die Ziele dieses Leitfadens bestehen darin, ein gewisses Maß an Verständnis dafür zu fördern, wo Security in die DevOps-Kultur eines Unternehmens eingebettet werden sollte. Was durch die Einbettung von Sec in DevOps auf möglichst automatisierte Art und Weise wirklich erreicht wird, ist eine sicherere Software, die das Gesamtergebnis des Unternehmens verbessert und gleichzeitig das Risiko reduziert.

Checkmarx

Über Checkmarx

Software-Sicherheit für DevOps und mehr.

Checkmarx stellt die für Software-Security erforderlichen Infrastrukturen her: einheitlich mit DevOps und nahtlos in jede Phase Ihres SDLC eingebettet, vom unkomplilierten Code bis hin zu Laufzeitüberprüfungen. Unsere ganzheitliche Plattform setzt den neuen Standard für die Einbettung von Security in die moderne Entwicklung.

Checkmarx bietet Ihnen:

Sicherheit von Anfang an

Checkmarx bietet die umfassendste und einheitlichste Software Security Platform der Branche, die SAST, SCA, IAST und AppSec Awareness eng miteinander verknüpft, um Sicherheit in jeder Phase der CI/CD-Pipeline zu gewährleisten und Sicherheitsrisiken einzudämmen.

Vereinfachte AST-Automatisierung

Checkmarx lässt sich nahtlos in gängige Software-Release-Orchestrierungs- und agile Planungstools wie IDEs, Build-Management-Server, Bug Tracking Tools und Quellcode-Repositorys integrieren, um Sicherheitsrichtlinien automatisch durchzusetzen.

DevOps-Geschwindigkeit

Nur Checkmarx ermöglicht es Ihnen, Softwarerisiken mit der Geschwindigkeit von DevOps im Griff zu halten und Anwendungen schnell und sicher herzustellen, ohne die Arbeitsabläufe der Entwickler zu unterbrechen.

Umfassende DevSecOps-Kompetenz

Wir kennen Software in- und auswendig. Wir sind mit Sicherheitsfragen vertraut, wie sonst kaum jemand. Entwickler vertrauen auf Checkmarx.

Software = **Sicherheit**