KONTINUIERLICHE PIPELINE-SICHERHEIT

SO STARTEN SIE MIT DEVSECOPS IN IHRER CI/CD-PIPELINE

Inhaltsverzeichnis

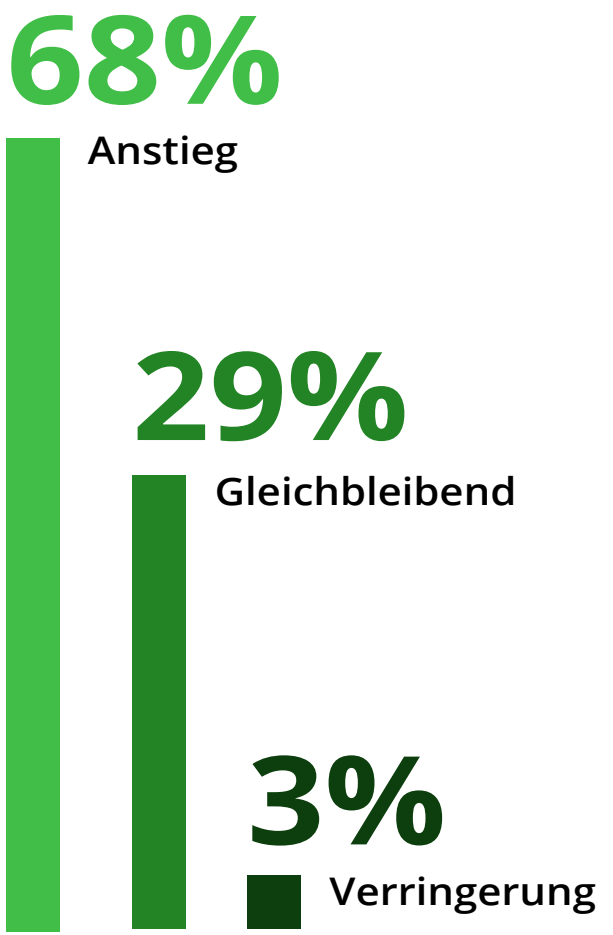
Einführung

Wenn DevSecOps einfach wäre, dann bräuchten wir gar nicht darüber zu reden, denn dann wären wir vor allen Schwachstellen geschützt und es gäbe gar keine Cyber-Angriffe. Eine echte DevOps-Strategie zu implementieren ist schwierig, aber eine erfolgreiche DevSecOps-Strategie ist noch schwieriger. Wir wissen das aus bitterer Erfahrung durch Angriffe und Vorfälle bei Unternehmen wie Equifax, Marriott, Facebook und Google - die deutlich machen, wie wichtig es ist, Software-Schwachstellen frühzeitig zu entdecken.

Die Philosophie von DevSecOps ist die Integration von Sicherheitspraktiken in den DevOps-Prozess. Mit einem methodischen Ansatz können Unternehmen eine sichere Software-Pipeline erstellen und ausliefern, und zugleich sicherstellen, dass sie alle bekannten Schwachstellen frühzeitig im Lebenszyklus der Softwareentwicklung entschärfen. Einer der größten Fehler von Unternehmen, die keinen ganzheitlichen DevSecOps-Ansatz verfolgen, besteht darin, Sicherheit als nachträglichen Gedanken zu behandeln. Die Sicherheitsprüfung ist keine isolierte Aufgabe, die Identifizierung von Schwachstellen ist untrennbar mit dem Lebenszyklus Ihrer Softwareentwicklung verbunden.

Der stetig wachsende Einsatz von Open-Source-Software in Unternehmen setzt die Code-Basis potenziellen Schwachstellen und Verstößen gegen die Lizenzbestimmungen aus, die in Open-Source-Komponenten versteckt sind. Die Frage ist... Wie sichern wir unser Ökosystem für Softwareentwicklung und -bereitstellung kontinuierlich ab, um diese Risiken zu mindern, zumal die Häufigkeit und Intensität der Angriffe zunimmt?

Die Beseitigung von Schwachstellen und die Sicherstellung der Lizenzkonformität muss eng in Ihre CI/CD-Pipeline integriert werden, um so früh wie möglich auf Probleme reagieren zu können. Integrierte und kontinuierliche Pipeline-Sicherheit ist möglich, aber sie erfordert die Zusammenarbeit der IT-, Entwicklungs-, Sicherheits- und der Operativen Teams.



Veränderung in der Nutzung von Enterprise Open Source in den letzten 12 Monaten

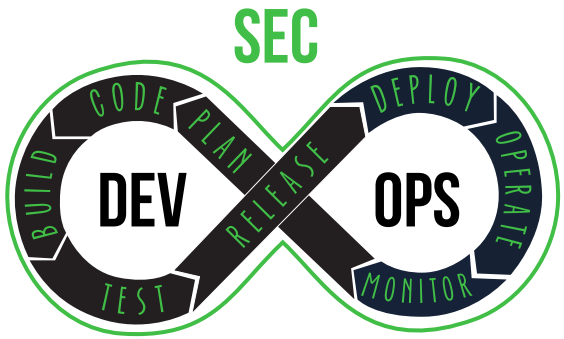
Quelle: Red Hat - The State Of Enterprise Open Source

Warum Sicherheit das gesamte Team angeht

Das Innovationstempo hat die Verwendung von Open-Source-Software vorangetrieben, um Anwendungen schneller zu erstellen und zu entwickeln, anstatt alles von Grund auf neu zu programmieren. Aufgrund des unerbittlichen Drangs zur Innovation nutzen Entwickler mehr und mehr Open-Source-Software als je zuvor. Open-Source-Code hat seine Vorteile, der offensichtlichste ist die Beschleunigung der Software Entwicklung. Aber es hat auch Nachteile. Der Code und seine Exploits werden zu öffentlichen Informationen, die es Hackern ermöglichen, auf diese Informationen zuzugreifen und clevere und gezielte Angriffe zu planen.

Die Notwendigkeit, Ihre Software früher im SDLC zu sichern, ist heute ein Muss für jedes Unternehmen. Einer der Hauptgründe einen DevSecOps-Ansatz zu verfolgen, ist die Tatsache dass es 100:1 mehr Entwickler als Sicherheitsexperten gibt und Sie das Sicherheitswissen verteilen müssen, um die Sicherheitslücke schneller schließen zu können.

DevSecOps ist nicht länger ein Wunschzettel für einen CIO, sondern eine IT-Strategie, die ein integraler Bestandteil jedes Softwareentwicklungslebenszyklus sein muss. In diesem ebook gehen wir auf DevSecOps als Ganzes ein, konzentrieren uns aber hauptsächlich darauf, wie Sie das Risiko der Verwendung von Open-Source-Software (OSS) in Ihrem Produktionscode mit Software Composition Analysis (SCA) am besten reduzieren können



DEV : OPS : SEC

100 : 10 : 1

Verhältnis von Entwicklern zu Betriebs- und Sicherheitsingenieuren

Umgang mit Schwachstellen in Open-Source-Software und Compliance mit Software Composition Analysis

Moderne Anwendungen bestehen heute aus bis zu 90 % OSS-Komponenten. Das bedeutet, dass der Großteil der Software, die wir täglich erstellen, bereitstellen und nutzen, mit größter Wahrscheinlichkeit mehr Schwachstellen enthält als je zuvor. Da OSS offen zugänglich ist, bedeutet dies, dass Hacker leichten Zugang zum Code haben und nach Schwachstellen suchen können. Open-Source-Software stellt nicht nur ein Risiko durch Schwachstellen dar, sondern kann auch komplexe Lizenzierungsprobleme für Unternehmen mit sich bringen. Eine mögliche Komplikation könnte sein, dass eine Lizenz besagt: "Wenn Sie diese Komponente verwenden, müssen Sie Ihren Code als „Open Source“ bereitstellen."

Idealerweise möchten Sie alle Ihre OSS-Komponenten so früh wie möglich im Entwicklungsprozess auf Lizenzkonformität und Schwachstellen überprüfen und identifizieren. Zu wissen, welche Komponenten Sie in Ihrem gesamten Anwendungsportfolio haben und diese im Auge zu behalten, ist ein absolutes Muss und sollte letztlich automatisiert werden. Dies sollte ein integraler Bestandteil Ihrer CI/CD-Pipeline sein, um Ihre Entwicklungs- und Release-Geschwindigkeit auf Kurs zu halten.

In der Vergangenheit war es für die Absicherung des SDLC und der Produktion erforderlich, einen Scan der Komponenten durchzuführen. Heute gibt es verschiedene Lösungen, die ein höheres Maß an Sicherheit und Compliance-Überwachung erreichen können. Sie scannen direkt in Ihre IDE, Ihr Repository Manager, Ihre CI/CD-Pipeline und kann sogar Ihre Container-Images scannen. Für die Open-Source-Sicherheits- und Compliance-Überwachung wäre eine nativ integrierte SCA-Lösung am besten geeignet.

Sicherheitsverletzungen kosten pro Jahr weltweit 600 Milliarden Dollar

Quelle: McAfee und das Center for Strategic and International Studies, 2018

Was ist eine Software-Kompositionsanalyse (Software Composition Analysis (SCA))?

Die Verwendung von Werkzeugen zur statischen Code-Analyse kann Ihnen bei der Identifizierung von Schwachstellen in Ihrem eigenen, selbst entwickelten Code helfen. Allerdings werden damit keine Schwachstellen in den Komponenten identifiziert, von denen Sie abhängig sind (depencies). Der Bedarf an Software Composition Analysis entstand durch den zunehmenden Einsatz von Open-Source-Komponenten, die von Entwicklern verwendet werden, um mit dem Innovationstempo Schritt zu halten. Unternehmen hatten Schwierigkeiten, die Verwendung von Open-Source-Komponenten in verschiedenen Teams und an verschiedenen Standorten zu verwalten und nachzuverfolgen, und benötigten einen automatisierten Weg.

SCA umfasst die Verwaltung und Überwachung der Lizenzkonformität und Sicherheitsschwachstellen in Open-Source-Komponenten. Zu wissen, welche OSS verwendet wird und welche Abhängigkeiten bestehen, ist ihr Hauptziel. Nach der Identifizierung der Open-Source-Komponenten liefern die SCA-Tools Informationen über die Lizenz, die Versionsnummer und darüber, ob es bekannte Sicherheitsschwachstellen gibt die mit ihr verbunden sind.

Fortschrittliche SCA-Tools bieten Funktionen zur Durchsetzung von Richtlinien, die eine automatische Überwachung von Open-Source-Komponenten ermöglichen. Diese können so konfiguriert werden, dass je nach Kontext des zu überprüfenden Objekts unterschiedliche Verhaltensweisen bei identifizierten Sicherheits- oder Compliance-Verstößen aktiviert werden. Ein Beispiel wäre, dass ein Build einer hochsensiblen Anwendung aufgrund einer Schwachstelle fehlschlägt, während der Build einer Testanwendung mit der gleichen verwundbaren Komponente nicht fehlschlägt. Das Tool vergleicht jede Open-Source-Komponente in Ihrem Code mit Ihren Richtlinien und löst je nach Ergebnis verschiedene Arten von automatisierten Aktionen aus.

Es ist wichtig, die folgenden Punkte zu beachten, wenn Sie sich mit Tools zur Analyse der Softwarezusammensetzung beschäftigen:

1. **Technologie-Unterstützung:** Wie universell ist das Tool und unterstützt es alle im Unternehmen verwendeten Programmiersprachen und Softwarepaket Formate

2. **Ökosystem-Integration:** Das SCA-Tool muss mit Repositories, IDEs, Paketmanagern, CI-Servern und mehr über Out-of-the-Box-Integrationen sowie über umfangreiche offene APIs integriert werden

3. **Datenbank:** Eine umfassende Lizenz- und Schwachstellenintelligenz ist ein Muss, um Ihnen die beste Abdeckung zu bieten

Die Analyse der Softwarezusammensetzung ist nur so gut wie die Daten dahinter

Das hört sich alles großartig an, denken Sie vielleicht... alles, was Sie tun müssen, ist ein paar tolle Tools kaufen, sie in Ihr Tool-Ökosystem zu integrieren und voila, Sie machen SCA wie ein Meister. Das mag zwar bis zu einem gewissen Grad stimmen, aber Ihr Erfolg bei der Erreichung des Ziels, Schwachstellen und Lizenzprobleme wirklich zu entschärfen, kann durchaus begrenzt sein. Hier ist der Grund dafür:

Sie können die beste Integration von SCA-Tools die es gibt haben, aber eine Sicherheitsscan-Lösung ist nur so gut wie die Datenbank mit Schwachstellen aus der sie Informationen zieht. Die Verwendung einer Datenbank, die nicht auf dem neuesten Stand inpunkto Schwachstellen ist, ist wie der Versuch, jemanden in einer Menschenmenge zu finden, ohne zu wissen, wie er aussieht. Einige Unternehmen sehen die Verwendung von MITREs Common Vulnerability Enumeration (CVE) in der National Vulnerability Database (NVD) als Standard zur Absicherung. Viele Sicherheitsexperten beharren darauf, dass es nicht mehr ausreicht, sich auf CVE und NVD für Schwachstellendaten zu verlassen.

Risk Based Security zum Beispiel, die dafür bekannt sind, einen der aktuellsten und umfassendsten Schwachstellen-Informationsdienste zu haben, haben jedes Jahr die Gesamtzahl der von CVE und NVD identifizierten und katalogisierten Schwachstellen in den Schatten gestellt. Bis heute haben sie über 97.000 Schwachstellen identifiziert und katalogisiert, die nicht in den CVE/NVD-Datenbanken zu finden sind (Stand: Ende 2020).

VulnDB vs. CVE Schwachstellen insgesamt	VulnDB	CVE
Total Entries	244,477	146,860
Real Time Alerts	Yes	No
Average Daily New Entries	68	30
Exploit Detail	Yes	Limited
Remediation Detail	Yes	Limited
Social Risk Scoring	Yes	No
Vendor & Product Risk Ratings	Yes	No
RESTful API & Integrations	Yes	No
On-demand Vuln Research	Yes	No

Quelle: vulndb.cyberriskanalytics.com und cve.mitre.org

Die größten Herausforderungen und Best Practices von DevSecOps und dem Einsatz von Open-Source-Software

1. Gewinnung von Sicherheitswissen

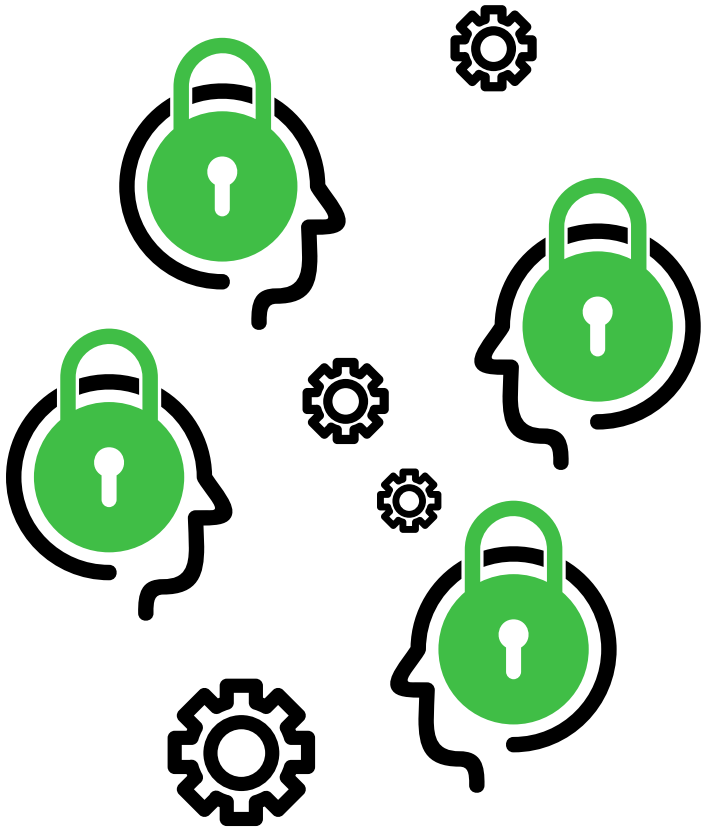
Im Dschungel von Continuous Integration und Continuous Delivery geht es den Entwicklern vor allem um die Geschwindigkeit, wie schnell sie die Anwendung codieren und bereitstellen können. Das bedeutet, dass sie nicht darüber nachdenken, wie "sicher" ihr Code ist oder ob die Open-Source-Softwarekomponenten, die sie einbinden, irgendwelche Schwachstellen aufweisen oder ob sie mit den Lizenzrichtlinien ihres Unternehmens übereinstimmen. Sie stehen unter Lieferdruck. Jedoch können Entwickler und Betriebsingenieure die Anwendungssicherheit nicht länger ignorieren. Wenn Sie keine richtig geschulten oder sachkundigen Entwickler haben, ist es wahrscheinlicher, dass sie anfälligen Code schreiben und unsichere OSS-Komponenten in ihre Anwendungen einbinden. Das ist ein Rezept für eine Katastrophe - vor allem, wenn sich diese in der Produktion befinden! Die Verringerung von Reibungsverlusten in Ihrem Prozess kann ebenfalls eine Herausforderung sein, wenn Sie die Akzeptanz von Sicherheit erhöhen und gleichzeitig die Release-Geschwindigkeit beibehalten wollen.

Wenn Sie damit beginnen, Sicherheit in den Lebenszyklus Ihrer Softwareentwicklung zu integrieren, ist das gesamte Team verantwortlich für Erfolg oder Misserfolg. Damit Einzelpersonen und Teams neue Sicherheitsprozesse, -konzepte und -technologien wirklich annehmen können, müssen sie verstehen, warum. Wenn sie wissen, warum sie etwas Neues tun sollen, können sie diese neuen Ideen leichter annehmen. Ein Hauptaugenmerk sollte auf einem gut vorbereiteten und trainierten Team liegen. Das bedeutet, dass Entwickler und aller weiteren Mitarbeiter in Sicherheitspraktiken geschult werden müssen, was die Einstellung eines erfahrenen DevSecOps- oder Sicherheitsingenieurs erfordern könnte.

Die Ausbildung beginnt bei den Teams, die die Software schreiben. Es ist von unschätzbarem Wert, Entwickler darin zu schulen, bessere Coding-Praktiken zu befolgen und zu wissen, wie sie sicheren Code schreiben und anfälligen Code identifizieren und beheben können. Je mehr Entwickler über die Risiken schlechter Coding-Praktiken wissen, desto besser, was eine solide Grundlage Ihres DevSecOps-Prozesses bildet. Außerdem müssen Entwickler und Betriebsteams unbedingt für alle neuen Technologien oder Lösungen geschult werden, deren Einsatz von ihnen erwartet wird.

Das Ziel von DevSecOps ist es, die Sicherheit in alle Phasen des Softwareentwicklungs-Workflows einzubinden, und muss mit einer "Ich bin für die Sicherheit verantwortlich"-Mentalität für Entwickler, Operations- und Sicherheitsteams gleichermaßen beginnen. Dies kann bedeuten, dass bestimmte Teammitglieder neue Aufgaben und Verantwortlichkeiten haben werden, was eine Herausforderung sein kann. Möglicherweise müssen Sie neue Mitarbeiter mit spezifischen Fachkenntnissen einstellen, die bei der Analyse von Sicherheitsbedrohungen und der Einrichtung von Best Practices für die Sicherheit helfen. Sicherheitsschulung ist in der Tat ein nie endender Prozess und sollte als ständige Anstrengung betrachtet werden.

Die Softwarekompositionsanalyse ermöglicht Ihren Entwicklern einen Einblick in alle OSS-Komponenten und deren Abhängigkeiten, die sie verwenden möchten. Sie liefert Lizenz- und Abhängigkeitsinformationen und zeigt, ob es Schwachstellen oder Lizenzverletzungen gibt. Dies kann bereits während der Entwicklung in der IDE und in Ihrem Repository-Manager erfolgen und sich über Ihre Build-Tools und Deployments erstrecken. Dies gibt dem Entwickler das Wissen, Verstöße früher zu finden und spätere teure Sicherheits- oder Compliance-Kosten zu mindern.

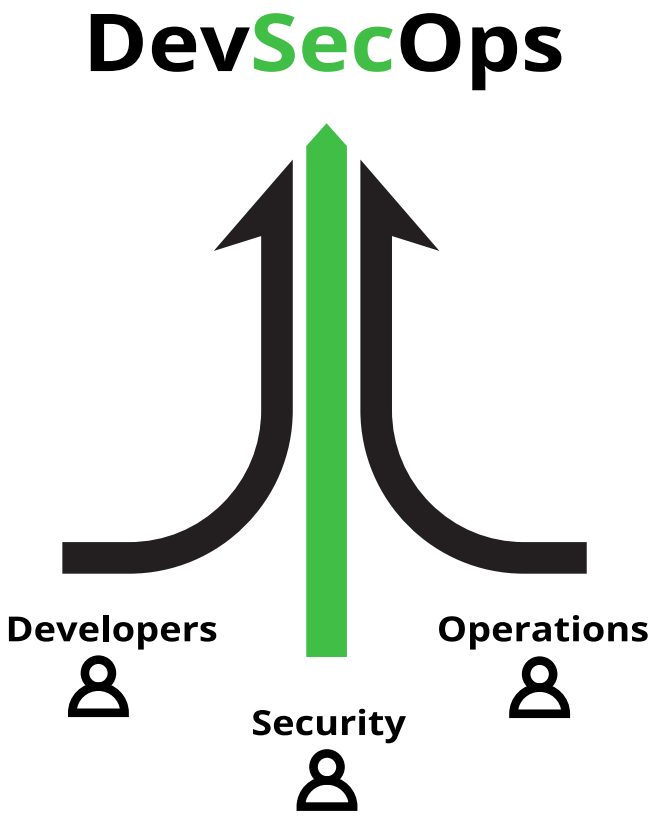


2. Aufbrechen von Silos

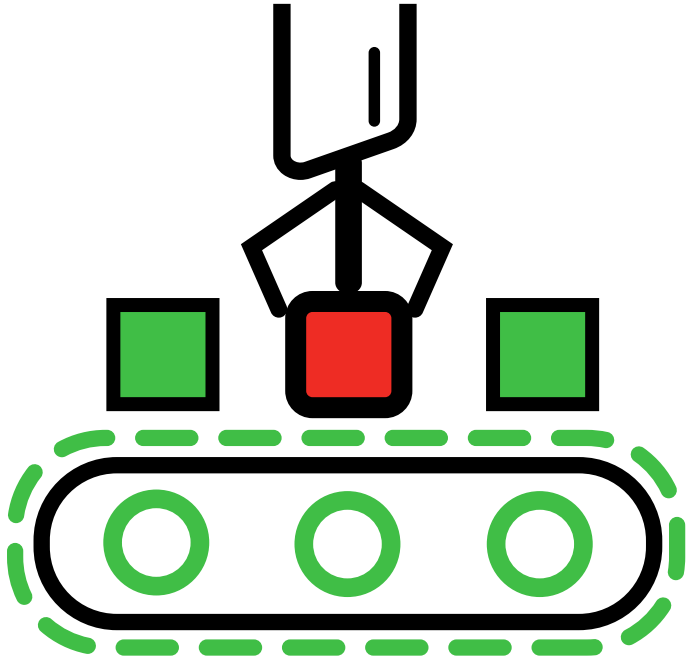
Erfolgreiches DevSecOps erfordert eine grundlegende Abkehr von traditionellen Ansätzen zur Produkt- und Softwarebereitstellung. Damit ein DevSecOps-Prozess erfolgreich ist, müssen nicht nur Prozesse, Organisationen und Ausbildung geändert werden, sondern auch Teams, die normalerweise in ihren eigenen Silos arbeiten (IT, Entwickler, Betrieb und Sicherheit), müssen nun zusammenarbeiten. Sie müssen sich so entwickeln, dass sie kooperativ und unterstützend sind und Menschen, Kultur, Prozesse und Organisationsstrukturen in Frage stellen. Es muss eine konsequente Zusammenarbeit über Abteilungen hinweg geben, und die Teams müssen zusammenarbeiten, um die Lücken zu überbrücken und sich auf ein gemeinsames sicherheitsbewusstes Ziel zu konzentrieren. Jedes Team muss sich neue Denk-, Arbeits- und Kommunikationsweisen aneignen, um sich zu vereinen.

Ein DevSecOps-Prozess ist leichter zu erreichen, wenn Sie bereits einen etablierten DevOps-Prozess haben. Das Ziel bei DevSecOps ist die Einführung einer "Shift Left"-Mentalität über die verschiedenen Teams hinweg. Das bedeutet eine Praxis, bei der sich die Entwicklungs- und Betriebsteams früher auf Qualität, Sicherheit und Problemvermeidung (durch Tests auf Sicherheit und Verstöße) konzentrieren. Eine Lösung wäre, Mitglieder des Sicherheitsteams in die Entwicklerteams einzubinden, damit sie in deren Ausbildung einbezogen werden, Code früher im Lebenszyklus sehen und bei Code-Reviews und täglichen Standups oder Scrums anwesend sein können.

Ein umfassendes und integriertes SCA-Tool kann es Entwicklern, Betriebs- und Sicherheitsteams ermöglichen, alle OSS-Metadaten für alle ihre Artefakte, Builds und Container-Images zu sehen. Sie alle sehen und teilen dieselben Daten und können so zusammenarbeiten, um Schwachstellen und nicht konforme Lizenzen frühzeitig zu finden und zu entschärfen. Dies überbrückt die Kluft zwischen den isolierten Teams und sie werden durch die Verantwortung und das Ziel geeint, schwachstellenfreien und konformen Code zu veröffentlichen.



3. Automatisierte Sicherheit und Compliancet



Eine Herausforderung bei der Bereitstellung von sicherem und konformem Code können "menschliche Fehler" sein, durch die sich Schwachstellen in Produktionsversionen einschleichen können. Manuelle Sicherheitsprozesse sind bei der Menge an Code, die heute produziert wird, einfach unmöglich. Jeder Entwickler bräuchte einen eigenen Sicherheitsingenieur, der die Sicherheitsprobleme überprüft und identifiziert. Darüber hinaus müssen Sie auch den bestehenden Code untersuchen, um neue Schwachstellen zu identifizieren.

Alle sicherheitskritischen Funktionen sollten daher automatisiert und in Ihr bestehendes Tool-Ökosystem integriert werden. Dies wird dazu beitragen, einige der "menschlichen Fehler" zu mindern, die die Schwachstellen in die Produktionsversionen einschleichen können. Das bedeutet, dass einige Aufgaben, die bisher manuell erledigt wurden, mit einem speziellen Tool und Prozess automatisiert werden müssen. Ein Beispiel hier wäre ein "Security Gating", ein automatisierter Prozess, der Builds fehlschlagen lässt oder die Promotion verhindert, wenn bestimmte Sicherheitsbedingungen nicht erfüllt sind.

Wenn Sie heute kein Software Composition Analysis-Tool verwenden, haben Sie wahrscheinlich Schwierigkeiten, die Verwendung von Open-Source-Software in Ihren Teams zu verwalten und zu verfolgen. Wenn Sie versuchen, die Geschwindigkeit Ihrer Releases zu erhöhen, wird die Verwendung von Open-Source-Software durch die Entwickler stetig zunehmen. Außerdem ist es für Ihre Entwickler schwierig zu wissen, ob die OSS-Komponenten, die sie verwenden, irgendwelche Sicherheitslücken oder Lizenzverletzungen enthalten, was im Nachhinein problematisch sein kann. Sie haben vielleicht ein Inventar der Open-Source-Komponenten, die in Ihrem Binary Repository Manager verwendet werden, aber was Sie wirklich brauchen, ist ein umfassendes Wissen über die Komponenten, ihre Abhängigkeiten und die damit verbundenen Schwachstellen und Lizenzverletzungen. Durch die Integration von SCA-Tools in Ihre Toolkette können Sie die Identifizierung und Entschärfung von Schwachstellen und Lizenzverletzungen automatisieren.

"Bei Organisationen, die keine Sicherheitsautomatisierung eingesetzt hatten, waren die Kosten für Sicherheitsverletzungen um 95 Prozent höher als bei Organisationen mit vollständig eingesetzter Automatisierung (durchschnittliche Gesamtkosten einer Sicherheitsverletzung ohne Automatisierung: 5,16 Millionen US-Dollar gegenüber 2,65 Millionen US-Dollar bei vollständig eingesetzter Automatisierung)."

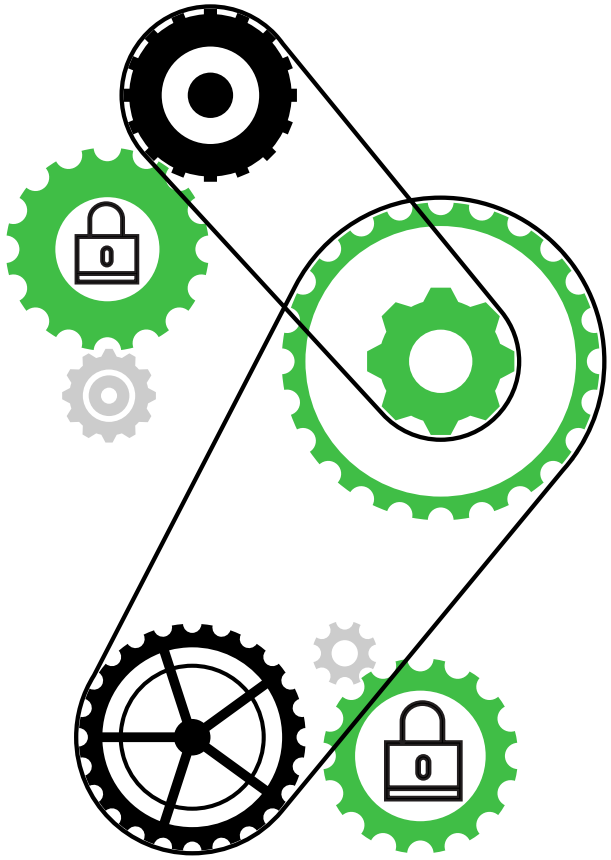
Quelle: Cost of a Data Breach Report 2019, IBM Security

4. Übernahme von Sicherheitsprozessen

Unternehmen die DevOps praktizieren haben eine bestehende Toolkette, mit der sie ihre tägliche Arbeit erledigen. Von ihnen zu verlangen, noch ein weiteres System oder Tool zu übernehmen, um Sicherheits Probleme in ihrem Code aufzuspüren, würde sie möglicherweise aus dem Kontext bringen und proaktiv nach Verstößen zu suchen ist lästig. Das Ergebnis ist weniger Produktivität und potenziell weniger Akzeptanz.

Damit Entwickler neue Tools annehmen, muss die neue Pipeline ihrer bisherigen DevOps-Pipeline so wenig gestört wie möglich werden. Es gibt viele neuere Sicherheitstools, darunter viele, die sich direkt in bestehende Ökosysteme integrieren oder in diese "einbinden" lassen. Entwickler möchten nicht damit belästigt werden Software oder Komponenten zu scannen. Mit integrierten Sicherheitstools wird der Code automatisch nach gescannt. Der Entwickler muss lediglich entscheiden, wie er mit den Ergebnissen umgeht, so wie er es heute mit anderen Arten von Fehlern tut.

Idealerweise sollten Sie ein Software Composition Analysis-Tool wählen, das sich in Ihre IDE, Ihren Repository Manager und Ihre Build-Tools integrieren lässt. Es würde dazu dienen, etwaige Schwachstellen oder Lizenzverletzungen während des gesamten Lebenszyklus der Softwareentwicklung zu identifizieren. Ein wichtiger Aspekt, den Sie berücksichtigen sollten, ist, welche Programmiersprachen und Pakettypen Ihre Entwickler verwenden, und daher müssen diese von der SCA-Lösung unterstützt werden.



5. Sicherheitslösungen für containerisierte Umgebungen

Container werden zu einer gängigen Praxis für Cloud-basierte Systeme. Sie helfen bei der Bereitstellung skalierbarer Systeme und erhöhen die Geschwindigkeit, mit der Unternehmen neue Dienste und letztlich Software viel schneller als bisher bereitstellen. Aber wie immer bei neuen Technologien, kommen sie mit neuen Sicherheits- und Compliance-Risiken.

Es gibt mehrere Sicherheitstechnologien, um diese neuen Risiken zu mindern, z. B. SCA-Tools für die CI/CD und Container-Runtime-Tools für den Schutz in der Runtime.

Suchen Sie ein SCA-Tool aus, das rekursiv alle Schichten eines Containers durchsucht und Sicherheitsschwachstellen und Probleme mit der Lizenzkonformität identifiziert. Das Tool sollte in der Lage sein, Verstöße in einem bestehenden Container sowie während der Erstellung eines Containers zu identifizieren und sollte auch in der Lage sein, einen Build fehlschlagen zu lassen oder die Verwendung eines Containers zu verhindern. Es ist wichtig, eine Lösung zu wählen, die sich direkt in den Repository-Manager, die Build-Plattform und die Container-Infrastruktur integrieren lässt, um die Akzeptanz sicherzustellen und teure Nacharbeiten an der Software zu vermeiden, oder schlimmer noch, eine unsichere Produktionsfreigabe.

„Sicherheit ist die größte Herausforderung für Kubernetes-Anwender: 46 % der Befragten nannten Sicherheit als ihre größte Herausforderung."

Quelle: Die neue Stack-Analyse von Cloud Native Umfrage der Stiftung Computing (Herbst 2017)

Fazit

Die Einführung einer DevSecOps "Security Everywhere"-Mentalität ist heute für jedes Unternehmen unerlässlich. Das Streben nach schneller Innovation geht Hand in Hand mit einer zunehmenden Nutzung von Open-Source-Software. Das bedeutet, dass die Notwendigkeit eines soliden OSS-Sicherheits- und Lizenz-Compliance-Prozesses in Zukunft noch dringender sein wird. Und warum? Weil die zunehmende Verwendung von OSS in Anwendungen diese zu einem großen und attraktiven Ziel für Cyberangriffe und Hacker macht.

Was nehmen wir also daraus mit?

- Etablieren Sie DevSecOps als einen Eckpfeiler Ihres Softwareentwicklungslebenszyklus
- Verankerung von Sicherheitswissen und -verantwortung in Ihren Entwickler- und Betriebsteams
- Verwendung von Best Practices für Sicherheit und Compliance und Anwendung von Taktiken zur kontinuierlichen Verbesserung
- Verwenden Sie eine integrierte Suite von DevSecOps-Tools, die Sicherheit und Governance automatisieren können
- Stellen Sie sicher, dass Ihre Tool-Suite eine universelle Softwarekompositionsanalyse-Lösung enthält
- Nutzen Sie die umfassendste und aktuellste Datenbank für Schwachstelleninformationen

DevSecOps ist keine exakte Wissenschaft. Es gibt mehr als einen Weg, um sicherzustellen, dass Ihre Softwareimplementierungen sicher und konform sind. Um bei DevSecOps erfolgreich zu sein und Open-Source-Software sicher zu nutzen, ist ein starkes Fundament aus Wissen, Zusammenarbeit, Automatisierung und Eigenverantwortung erforderlich. Ohne die Einführung von DevSecOps setzen Sie Ihr Unternehmen dem Risiko einer Sicherheitsverletzung aus, was sich direkt in Hunderten von Millionen Dollar an potenziellen Entschädigungszahlungen, verlorenen Kunden, entgangenen Einnahmen und Kosten für Software-Nachbesserungen niederschlagen kann.

Über JFrog

JFrog hat es sich zur Aufgabe gemacht, kontinuierliche Updates durch“Liquid Software“ zu ermöglichen und Entwickler in die Lage zu versetzen, qualitativ hochwertige Anwendungen zu liefern, die sicher und ohne Ausfallzeiten an die Endbenutzer fließen. Unsere Lösungen entsprechen den Anforderungen Ihres Geschäftsmodells und unterstützen On-Prem-, Cloud-, Hybrid- und Multi-Cloud-Konfigurationen. Mehr als 6.000 Kunden verlassen sich auf JFrog, um ihre Binaries für ihre geschäftskritischen Anwendungen zu verwalten, darunter mehr als 70% der Fortune 100 - Unternehmen wie Amazon, Facebook, Google, Netflix, Uber, VMware und Spotify vertrauen JFrog.

Um mehr über JFrog-Lösungen zu erfahren, schauen Sie sich diese Webinare an:

Einführung in JFrog Artifactory

DevSecOps mit JFrog Xray - Universelle Komponenten- & Auswirkungsanalyse

Probieren Sie es selbst aus - JFrog Xray FREE Trial

Referenzen:

http://info.nowsecure.com/rs/201-XEW-873/images/NowSecure-WhiteHat-2018-Application-Security-Report.pdf
Logz.io, "Der DevOps-Puls 2018", 2018
https://www2.deloitte.com/content/dam/Deloitte/uk/Documents/technology/deloitte-uk-tech-trends-2019-chapter7-devsecops.pdf
https://techbeacon.com/security/6-devsecops-best-practices-automate-early-often