Sonatype

OPEN SOURCE-RICHTLINIEN SICHER DURCHSETZEN

SAGEN SIE „FALSE POSITIVES“ UND „FALSE NEGATIVES“ LEBEWOHL

WHITE PAPER

Applikationen in DevOps-Geschwindigkeit liefern

Die meisten Organisationen implementieren DevOps-Praktiken, um sichere, qualitativ hochwertige Anwendungen schneller als ihre Konkurrenz zu liefern. Aber wie können Sie sichere Anwendungen bedarfsgerecht liefern, wenn Ihre Open-Source-Richtlinien durch Wasserfall-native Prozesse erzwungen werden?

Es gibt einen Grund, warum so viele Organisationen die Sicherheitspraktiken verlagern. Traditionelle SCA-Tools (Software Composition Analysis) verwenden Black- und Whitelists sowie manuelle Prüfprozesse, um die Qualität der Anwendungen nach der Entwicklung zu analysieren. Da diese Methoden jedoch nicht skalieren, erfüllen sie nicht die Voraussetzungen, um DevOps-Geschwindigkeit zu erreichen. Durch die Auswahl sicherer, qualitativ hochwertiger Open-Source- und Drittanbieter-Komponenten zum frühestmöglichen Zeitpunkt innerhalb der DevOps-Pipeline lässt sich die spätere manuelle Nacharbeit eliminieren.

Viele Security-Fachleute bestätigen, dass die Durchsetzung von Open-Source-Richtlinien entscheidend ist. Das beste Vorgehen, um Richtlinien durchzusetzen, ist, sie zu automatisieren und gleichzeitig Flexibilität in verschiedenen Phasen des SDLC (Software Delivery Life Cycle) zu ermöglichen. Berichte über bekannte Mängel und Abhilfemaßnahmen skalieren ebenfalls nicht. Mit präzisen Informationen über die Qualität und Sicherheit von Open-Source-Komponenten können Organisationen Richtlinien bereits am Anfang der DevOps-Toolchain sicher durchsetzen. Aber nicht alle Tools sind gleichermaßen geschaffen. Zusätzlich machen es sich viele Anbieter leicht und daher ist Präzision in einigen Tools schwer zu erreichen.

Bei der Identifizierung ist Präzision entscheidend

Es gibt verschiedene Ansätze, um Komponenten zu analysieren. Dabei ist es wichtig zu verstehen, warum nur ein präziser Ansatz dazu in der Lage ist, Automatisierung und Skalierbarkeit in der Software-Lieferkette zu erzeugen.

Identifikationsansatz	Warum dieser Ansatz nicht genau genug ist
Dateiname - Komponenten werden durch eine alphanumerische Auswertung der Dateinamen identifiziert.	**False Positives -** Dateinamen wie util.jar oder core.js kommen sehr häufig vor (vergleichbar mit dem Versuch im Telefonbuch eine bestimmte Person mit dem Nachnamen Schmidt zu identifizieren). **False Negatives –** Dateien werden ständig umbenannt, vor allem bei JavaScript, daher entziehen sich die Komponenten den Vergleichsergebnissen. **Unvollständig** - Obwohl struts.jar oder jquery.js (zwei beliebte Komponenten) anhand ihres Namens identifiziert werden, bleibt die spezifische Version der Komponente verborgen.
File Hash - Komponenten werden durch eine eindeutige Kennung namens Hash oder Checksum (Prüfsumme) identifiziert	**False Negatives (Unbekannte) -** Entwickler kompilieren Komponenten oftmals neu und Dateien werden regelmäßig nachbearbeitet, verkleinert, verdichtet. Jede noch so kleine Änderung führt zu einem anderen Hash, was die Identifizierung durch diese Methode ausschließt.
NAMESPACE / METADATA - Komponenten werden anhand der Verpackungsanweisungen in Dateien oder Manifesten (z. B. pom.xml - Apache Maven, Gradle - gradle, package.json - npm) identifiziert.	**False Positives -** Die Vervielfältigung von Koordinaten ist ein bekanntes Problem. Nur weil der Paket-Manager denkt, er habe eine Datei mit dem richtigen Namespace, bedeutet das nicht, dass es nicht eine veränderte Datei aus einem anderen Repository ist. **Unbekannte** - Verpackungssysteme führen häufig Komponenten ein, die nicht identifiziert werden können. **Multiples** - oftmals werden mehrere Verpackungssysteme verwendet, wodurch die oben genannten Fehler miteinander verbunden werden.
Quellcode - Komponenten, die durch Scannen des Quellcodes identifiziert werden, wobei Fingerabdrücke gesucht werden, die mit anderen bekannten (inventarisierten) Quellcodes (Code-Snippets) übereinstimmen.	**Verfügbarkeit** - wenn es nicht möglich ist, den Quellcode zu scannen, ist es (offensichtlich) unmöglich, den Code und die entsprechenden Verbindungen zu identifizieren. **False Positives -** Quellcode-Matching basiert auf Snippet-Analyse, die zahlreiche False Positives nach sich zieht. Dies ist deshalb der Fall, weil gemeinsame Code-Muster existieren, die unterschiedliche Abschnitte des Quellcodes gleich erscheinen lassen. **Redundant Positives –** ein beliebiges Stück Quellcode (häufig Open Source) ist oft in mehreren Projekten und innerhalb eines Projektes, möglicherweise in vielen Versionen einer Komponente vorhanden. **Speed -** Das Scannen von Quellcode ist ein langsamer Prozess, der Feedback nicht schnell genug zurückspielt, um den Anforderungen moderner Software Supply Chains gerecht zu werden.

Erweiterter binärer Fingerabdruck

Die Unfähigkeit anderer Ansätze, Komponenten präzise zu identifizieren, veranlasste Sonatype dazu, die notwendige Zeit und das Kapital zu investieren, um den erweiterten binären Fingerabdruck zu erfinden. Sonatype erkannte, wie wichtig präzise Informationen sind, und nahm sich der Aufgabe an, neue Techniken, Prozesse und Algorithmen innerhalb der Datenwissenschaft zu schaffen.

Das Ergebnis sei hier anhand zweier Beispiele dargestellt:

- **Java** - Eine Anwendung wird als Java Enterprise Archive (EAR)-Artefakt ausgeliefert. Das EAR enthält drei Java-Web-Anwendungen (WAR). Jede dieser WARs enthält zahlreiche JAR-Dateien, darunter eine mit einer Java-JAR-Datei, die durch Patching (Forking) eines Open-Source-Projekts erstellt und neu kompiliert wurde, um eine neue benutzerdefinierte JAR zu erstellen. Mithilfe des “Advanced Binary Matching” von Sonatype können alle verschachtelten Dateien erfolgreich identifiziert werden. Darüber hinaus löst unser patentierter Algorithmus das Ähnlichkeitsergebnis aus und zeigt eine enge partielle Übereinstimmung mit dem abgespaltenen Open-Source-Projekt einschließlich der Projektversion an.
- **Javascript** - Eine Anwendung wird als selbstextrahierende Zip-Datei ausgeliefert. Die Datei enthält eine Anzahl von JavaScript-Dateien, einschließlich einer Datei mit einer Abhängigkeit für jQuery. Diese Datei wurde jedoch umbenannt und hat daher alle Informationen zur Identifizierung der jQuery-Abhängigkeit verloren. Mithilfe des “Proprietary Matching” von Sonatype wird die Komponente identifiziert und auf die ursprüngliche jQuery-Abhängigkeit zurückverfolgt, einschließlich der spezifischen Version.

Bei Data Intelligence ist Präzision entscheidend

Sobald man in der Lage ist, Komponenten präzise zu identifizieren, wird im nächsten Schritt der Software-Supply-Chain-Automatisierung sichergestellt, dass die Metadaten, die die Attribute einer Komponente beschreiben, hinreichend genau sind. Damit können Entscheidungen über die Akzeptanz einer Komponente durch automatisierte Prozesse gefällt werden.

Wie bei der Komponentenidentifikation wird das Durchsuchen verschiedener Datenbanken nach Metadaten, wie beispielsweise Sicherheitsinformationen, anhand des Namens einer Komponente, unweigerlich schlechte Informationen liefern. Angesichts dessen investiert Sonatype massiv in ein Expertenteam, das eigene Forschung betreibt, um tiefer gehende Informationen zu Komponenten zu erschließen. Diese Experten nutzen öffentliche und private Daten-Feeds, beliebte Repositories (wie GitHub) und überwachen Projekt-Webseiten. Jedoch verlassen sie sich niemals auf diese Informationsquellen ohne gründliche Kuration (d. h., Daten werden dahin gehend aufbereitet, dass sie von automatischen Abfragen wiederverwendet werden können).

Für die Lizenzierung bedeutet das, sich nicht nur auf die Lizenzerklärungen des Projekts zu verlassen, sondern auch die Header im Quellcode zu prüfen. Für Sicherheitsmängel bedeutet das, den Fehler im Wurzel-Algorithmus zu finden und Wege zur Behebung oder alternativen Auflösung des Fehlers zu dokumentieren. Für andere Attribute bedeutet dies sowohl quantitative als auch qualitative Einschätzungen der architektonischen und projektbezogenen Integrität.

Richtlinien sicher durchsetzen

Präzision ist die einzige Möglichkeit, Teams in die Lage zu versetzen, bessere Entscheidungen zu treffen, damit sie schneller skalieren können, mit Kontrollen, die flexibel genug sind, die Unternehmensrichtlinien im Zusammenhang mit den entwickelten Anwendungen zu reflektieren.

Jede Organisation, jedes Team und jede Anwendung ist grundsätzlich einzigartig. Deshalb sollte geklärt werden, zu welchem Zeitpunkt im Software-Lebenszyklus die Attribute einer Komponente untersucht werden sollen.

- Jedes Mal, wenn ein Entwickler eine neue Komponente herunterlädt?
- Wann immer ein Entwicklungsteam ein Build produziert?
- Während der Vorfreigabeprüfung?
- Wenn Anwendungen in die Produktion übergeben werden?

Welcher Zeitpunkt ist der richtige?

Die Antwort lautet natürlich, dass jeder dieser Zeitpunkte wichtig ist, weil die Welt moderner Software-Entwicklung keine Einheitsgröße ist. In der Tat ist jede Situation anders und deshalb ist es entscheidend, Intelligence-Tools für Komponenten zu haben, die flexibel genug sind, um in jeder Phase der DevOps-Tool-Chain Mehrwerte zu bieten. Dazu gehören die Entwickler-IDE, Build-Systeme, Repository-Manager, CI/CD-Tools, Image-Konstruktoren, Delivery-Orchestrierung und Produktionslaufzeitumgebungen.

Nach der Bereitstellung ist das Sichern einer Stückliste und die Überwachung des Ökosystems bezüglich neuer Informationen, wie beispielsweise das Feststellen einer neuen Sicherheitslücke, ebenfalls zwingend erforderlich.

Resümée

Wenn es darum geht, Open Source-Komponenten zur Herstellung moderner Software zu verwenden, lautet das Fazit: Präzise Informationen sind unerlässlich. Werkzeuge, die mangelnde Präzision aufweisen, lassen sich nicht auf die Bedürfnisse moderner Software-Entwicklung abstimmen. Bei ungenauen und/oder unvollständigen Daten müssen Unternehmen

Schwachstellen, Lizenz- und andere Qualitätsprobleme bewältigen, die direkt zu höheren Kosten führen und Innovationen reduzieren.

Sonatypes „Data Services“ und „Advanced Binary Fingerprinting“ bieten eine einzigartige Lösung, die es Unternehmen ermöglicht:

- **Innovationen zu fördern**, indem Teams in die Lage versetzt werden, die qualitativ hochwertigsten Open Source-Komponenten präzise zu identifizieren.
- **Schnell zu skalieren** mit Komponenten-Informationen, die präzise genug sind, um Automatisierung in jeder Phase des Software-Lebenszyklus zuzulassen.
- **Die Verwendung von Komponenten mit flexiblen Richtlinien zu kontrollieren**, die eine granulare Entscheidungsunterstützung über verschiedene Teams, Sprachen und Anwendungsprofile hinweg fördern.

Mit genauen Identifikationsmöglichkeiten sind Sie in der Lage, die Software Supply Chain fehlersicher zu machen. Dies bedeutet, dass Risiken und Ineffizienzen, die Innovationen reduzieren, sicher beseitigt werden. Dies bedeutet auch, das volle Potenzial talentierter Entwickler auszuschöpfen, um schneller zu innovieren und auf einem globalen Spielfeld effektiver konkurrieren zu können.

Wir freuen uns über alle Ihre Fragen und möchten Sie ermutigen, den unglaublichen Wert der Sonatype „Data Services“ und des „Advanced Binary Fingerprinting“ zu testen. **Einer der nächsten Schritte wäre, unser kostenloses Tool “Application Health Check” auszuprobieren oder heute noch einen Demo-Termin zu vereinbaren.**

Sonatype

Im vergangenen Jahr haben Entwickler 52 Milliarden Komponenten aus dem Central Repository angefordert, um Software-Anwendungen herzustellen, die die Welt steuern. Darüber hinaus nutzen Unternehmen mit mehr als 120.000 Installationen weltweit die Sonatype Nexus Lösungen, um wiederverwendbare Komponenten zu verwalten und die Qualität, die Geschwindigkeit und die Sicherheit ihrer Software Supply Chains zu verbessern. Sonatype ist im Privatbesitz mit Kapitalbeteiligung von New Enterprise Associates (NEA), Accel Partners, Hummer Winblad Venture Partners, Morgenthaler Ventures, Bay Partners und Goldman Sachs.
Weitere Informationen finden Sie auf : www.sonatype.com

Hauptsitz
8161 Maple Lawn Blvd, Suite 250
Fulton, MD 20759
Vereinigte Staaten – 1.877.866.2836

Niederlassung für Europa
1 Primrose Straße
London EC2A 2EX
Großbritannien

Niederlassung APAC
5 Martin Place, Level 14
Sydney 2000, NSW
Australien

Sonatype Inc.
www.sonatype.com

ASERVO Software

ASERVO Software GmbH
Konrad-Zuse-Platz 8
81829 München

Tel.: +49 (0) 89 7167182-40
Fax: +49 (0) 89 7167182-55

Email: kontakt@aservo.com
www.ASERVO.com