Checkliste für die Suche nach vertraulichen Informationen

6 wichtige Kriterien für entwicklerfreundliche Secrets-Scanning-Lösungen

Entwickler schreiben vertrauliche oder geheim zu haltende Informationen (Secrets) mitunter direkt in den Quellcode, um Services, die sie für die Erstellung und Implementierung von Anwendungen benötigen, nahtlos aufzurufen oder sich bei ihnen zu authentifizieren. Doch dieses sogenannte Hardcoding stellt ein enormes Risiko dar, da die sensiblen Informationen dabei nicht auf sichere Weise gespeichert werden. Hartcodierte Secrets – wie Passwörter, Anmeldedaten, API-Schlüssel und wichtige Tokens – können in Quellcode, Build-Protokollen, Infrastructure-as-Code (IaC), Repositorys etc. offengelegt werden. Fallen wichtige Anmeldedaten Angreifern in die Hände, könnten diese sich damit privilegierten Zugriff verschaffen und dann Daten ausschleusen, Code manipulieren, sensible Informationen stehlen, Services deaktivieren oder mutwillig exorbitante Kosten verursachen.

Da Secrets so oft hartcodiert werden – besonders in der matrixbasierten Entwicklung und in cloudnativen Anwendungen – suchen Hacker zunehmend gezielt danach.

Mit Lösungen zur Sicherung vertraulicher Informationen können Sie diesen Risiken vorbeugen und ein abgerundetes Programm für die Codesicherheit einführen. Doch nicht alle Tools zum Scannen auf Secrets bieten die nötige Breite und Tiefe, um solche vertraulichen Informationen zu identifizieren und zu schützen.

In dieser Checkliste beschreiben wir die sechs wichtigsten Kriterien, auf die Sie bei der Beurteilung einer Secrets-Scanning-Lösung achten sollten.

1. Prüfung von Anwendungscode und Infrastructure-as-Code

Unternehmensgeheimnisse können in den unterschiedlichsten Codetypen enthalten sein. Ihre Secrets-Scanning-Lösung sollte daher in der Lage sein, alle wichtigen Dateitypen zu prüfen, z. B. Container-Images, IaC-Vorlagen und Quellcode. Ohne eine solche vollständige Prüfung können Sie sich kein Gesamtbild aller potenziell gefährdeten vertraulichen Daten in Ihrem Unternehmen machen.

Angenommen, Ihre Secrets-Scanning-Lösung prüft die IaC auf sensible Daten, nicht aber den Anwendungscode. Dann werden Ihre Entwickler auf potenzielle Probleme in IaC-Vorlagen und -Dateien hingewiesen und können diese beheben. Das Risiko eines Datenlecks bleibt aber weiter bestehen, solange Sie Anwendungscode an anderen Stellen in Ihrer Codebasis haben, der nicht gescannt wird.

Prüft Ihr Secrets-Scanner hingegen sowohl den Anwendungscode als auch die IaC auf diese vertraulichen Informationen, profitieren Sie von einer umfassenden Abdeckung und erhalten eine Übersicht über alle Orte, an denen möglicherweise Secrets gestohlen werden könnten. Ihre Tools für die Ermittlung von Secrets verursachen außerdem etwas weniger Aufwand durch False Positives, weil auch der Kontext einer IaC-Datei rund um die vertraulichen Informationen berücksichtigt wird.

Empfohlene Kriterien für Secrets-Scanning-Lösungen:

- Prüfung von IaC und Anwendungscode auf vertrauliche Informationen
- Berücksichtigung des Kontexts einer IaC-Datei rund um die vertraulichen Informationen zur Verbesserung der Scan-Ergebnisse

2. Entwicklerfreundliche Integrationen

Die Einbeziehung der Entwickler in die Sicherung vertraulicher Informationen ist der effizienteste Weg, um potenzielle Gefahren für Anmeldedaten zu identifizieren, zu beseitigen und zukünftig idealerweise von vornherein zu vermeiden. Damit die Entwickler dies möglichst reibungslos tun können, sollte Ihre Secrets-Scanning-Lösung nativ in Entwicklertools integrierbar sein. Damit versetzen Sie Ihre Entwickler in die Lage, Sicherheitsprobleme rund um vertrauliche Informationen wirksam zu verhindern, und reduzieren gleichzeitig die Anzahl der Kontextwechsel, die der Produktivität und Arbeitsmoral der Entwickler abträglich sind.

Achten Sie bei der Beurteilung von Secrets-Scanning-Lösungen unbedingt darauf, dass Feedback direkt in Entwicklertools und DevOps-Workflows einfließt.

Empfohlene Kriterien für Secrets-Scanning-Lösungen:

- Native Integration in vorhandene Entwicklertools wie VCS und IDEs
- Integration in DevOps-Workflows wie CI/CD-Pipelines
- Anzeige der ungeschützten Secrets in ihrem Kontext für eine rationelle Risikopriorisierung und -beseitigung
- Blockierung von Push-Übertragungen vertraulicher Informationen in ein Repository, bevor eine Pull-Anforderung über einen Pre-Commit-Hook geöffnet wird; Erkennung und Meldung ungeschützter Secrets beim Prüfen der Pull-Anforderung

3. Mehrdimensionaler Ansatz beim Scannen auf Secrets

Am häufigsten werden ungeschützte vertrauliche Informationen gefunden, die reguläre Ausdrücke – z. B. Zugriffstokens, API-Schlüssel, Verschlüsselungsschlüssel, OAuth-Tokens, Zertifikate etc. – enthalten. Durch die Prüfung auf reguläre Ausdrücke stellt die Secrets-Security-Lösung fest, ob eine bestimmte Zeichenfolge dem Muster anderer Secrets dieser Art entspricht, z. B. einem

AWS-Zugriffsschlüssel. Vertrauliche Informationen können aber auch in anderen, weniger gut vorhersehbaren Formaten vorliegen, deren systematische Erkennung schwieriger ist.

Neben der Prüfung auf reguläre Ausdrücke sollte eine vollständige Lösung auch nach Schlüsselbegriffen suchen, die häufig gemeinsam mit Passwörtern oder anderen vertraulichen Informationen verwendet werden. Anspruchsvollere Methoden zur Identifizierung vertraulicher Informationen suchen zudem nach Mustern mit hoher Entropie und beurteilen, ob Zeichenketten wie `EuN21!HHvaS%JPTQU&cx` sich so stark von echter Sprache unterscheiden, dass es sich wahrscheinlich um Secrets handelt. Dies ist die schwierigste Art der Prüfung, da sie zu zahlreichen False Positives und damit viel Arbeitsaufwand führen kann. Darauf gehen wir im nächsten Abschnitt noch näher ein.

Wenn Sie wirklich alle potenziell ungeschützten Secrets identifizieren wollen, sollten Sie sich für eine Lösung entscheiden, die nach regulären Ausdrücken, Schlüsselbegriffen und Zeichenketten mit hoher Entropie sucht. Viele Secrets-Scanning-Tools suchen jedoch einzeln nach bestimmten Arten von Secrets oder beschränken sich bei der Prüfung auf eine Phase des Entwicklungszyklus. Dadurch generieren sie potenziell viele False Positives oder übersehen einen Teil der Secrets komplett.

Die Wirksamkeit einer Lösung für die Sicherheit vertraulicher Informationen hängt auch davon ab, wie umfänglich die genutzten Erkennungsfunktionen sind und wie tiefgehend die Prüfung ist. Nutzt Ihre Lösung eine umfangreiche Signaturbibliothek, um ein breites Spektrum von Secrets mit bekannten, vorhersehbaren Ausdrücken zu erkennen und zu melden, können Sie davon ausgehen, dass die Prüfung umfassend genug ist.

Empfohlene Kriterien für Secrets-Scanning-Lösungen:

- Prüfung auf regelmäßige Ausdrücke, Schlüsselwörter und Entropie
- Einsatz domainspezifischer Erkennungsfunktionen für Secrets
- Umfassende signaturbasierte Richtlinienbibliothek als Grundlage
- Kontinuierliche Prüfung auf exponierte Anmeldedaten im gesamten Entwicklungszyklus – vom Build bis zur Laufzeit
- Prüfung aller Quellcodedateien und der gesamten Versionshistorie, um auch tief in die Codebasis eingebettete Secrets zu finden

4. Genau abgestimmte Erkennung von Mustern mit hoher Entropie

Durch die Prüfung auf reguläre Ausdrücke und Schlüsselwörter lässt sich ein erheblicher Prozentsatz der hartcodierten Anmeldedaten finden. Secrets ohne konsistente oder identifizierbare Muster bleiben dabei aber unerkannt. Wenn Sie nur diese signaturbasierten Prüfmethoden verwenden, werden Secrets wie Benutzernamen und Passwörter mit zufälligen Zeichenfolgen nicht gefunden. Um dies zu vermeiden und eine wirklich mehrdimensionale Prüfung durchzuführen, benötigen Sie einen Secrets-Scanner mit entropiebasierten Prüfungen, die auf einem genau abgestimmten Entropiemodell basieren.

Zur Analyse einer Zeichenkette ermittelt ein entropiebasiertes Secrets-Scanning-Tool, wie sehr die Zeichenkette sich von allen Wörtern der englischen Sprache unterscheidet. Zeichenketten, die sehr stark von tatsächlichen englischen Wörtern abweichen, werden von dem Tool als ungeschützte Secrets markiert. Diese Art der Prüfung kann Secrets erkennen, die anderen Prüfmethoden entgehen. Das entropiebasierte Scannen kann aber auch einige False Positives produzieren. Angenommen, Sie verwenden sehr lange Variablennamen wie zum Beispiel `redirectForOptimization`. Diese Variable ist sehr eindeutig kein Wort. Einfache entropiebasierte Scanner würden sie daher wahrscheinlich fälschlicherweise als Secret einordnen.

Es gibt verschiedene Möglichkeiten, diese False-Positive-Rate zu verringern. Als ersten Schritt sollten Sie eine Lösung wählen, die das Entropiemodell unter Berücksichtigung des Kontexts der Zeichenkette kalibriert, um komplexe Typen vertraulicher Informationen zu erkennen. Achten Sie bei der Lösungsauswahl außerdem darauf, dass schlüsselwort- und entropiebasierte Scans kombiniert werden. Lösungen mit einem mehrdimensionalen Ansatz bieten den Vorteil, dass sie die False-Positive-Rate senken und gleichzeitig eine umfassendere Abdeckung ermöglichen, um alle ungeschützten Secrets zu finden.

Empfohlene Kriterien für Secrets-Scanning-Lösungen:

- Sowohl signaturbasiertes als auch entropiebasiertes Scannen auf Secrets
- Entropiemodell berücksichtigt den Kontext der Zeichenketten, um komplexe Arten von Secrets zu identifizieren und die False-Positive-Rate zu reduzieren

5. Einbindung in eine Komplettlösung für die Codesicherheit

Das Scannen auf Secrets stellt lediglich eine Komponente einer umfassenden Codesicherheitslösung dar. Da die Nutzung separater Sicherheitstools für jede einzelne Komponente einer cloudnativen Codebasis zu erhöhter Komplexität führt, ist es am effizientesten, die Codesicherheit mithilfe einer zentralen Überwachungslösung zu konsolidieren.

Das Aufspalten der Codesicherung in verschiedene Anwendungsbereiche (wie Scannen auf Secrets, IaC-Sicherheit, Software Composition Analysis [SCA] usw.), kann zu Wildwuchs und Lücken in der Abdeckung führen. Zudem kann es damit schwierig sein, der Tatsache Rechnung zu tragen, dass Secrets, Sicherheitslücken und Fehlkonfigurationen in allen Phasen des Entwicklungszyklus auftreten können. Werden Sicherheitsprobleme von unterschiedlichen Tools erkannt, verliert man leicht den Überblick. Das erschwert den Entwicklern die Arbeit zusätzlich. Verwenden Teams unterschiedliche Sicherheitstools für die Infrastruktur- und Anwendungssicherung, wird es schwieriger, Probleme nach Schweregrad zu priorisieren.

Empfohlene Kriterien für Secrets-Scanning- und Codesicherheitslösungen:

- Einbindung in eine zentrale Codesicherheitslösung, um Toolwildwuchs einzudämmen und Lücken bei der Abdeckung zu minimieren
- Kontinuierliche Prüfung auf ungeschützte Secrets, Fehlkonfigurationen in IaC-Dateien und Schwachstellen in Open-Source-Code und Container-Images
- Vollständige Übersicht über alle Sicherheitsprobleme und Abhängigkeiten durch ein Lieferkettendiagramm
- Überwachung und Verhinderung von Fehlkonfigurationen und Schwachstellen im gesamten Entwicklungszyklus, um alles vom Code bis zur Cloud abzudecken

6. Laufzeitanbindung mit Lösungen zum Management der Infrastrukturzugriffsrechte in der Cloud

Wenn Sie sowohl Identitäten mit zu weit gefassten Zugriffsrechten als auch ungeschützte Secrets in der Cloud identifizieren und korrigieren möchten, sollten Ihre Secrets-Scanning-Lösung und Ihre Lösung für das Management der Infrastrukturzugriffsrechte in der Cloud (Cloud Infrastructure Entitlement Management, CIEM) aufeinander abgestimmt sein. Mit CIEM-Lösungen lassen sich Identitäten und Privilegien in Cloud-Umgebungen verwalten. Diese Lösungen vermitteln Ihnen eine Übersicht über die Zugriffsrechte in Ihren Cloud- und Multi-Cloud-Umgebungen. Sobald Sie dieses Gesamtbild haben, können Sie Risiken, die durch ungenutzte und zu großzügig bemessene Zugriffsrechte entstehen, identifizieren und beheben. Eine umfassende CIEM-Lösung beinhaltet unter anderem Funktionen für die Ermittlung der effektiven Zugriffsrechte, die anzeigen, wer in Ihrem Unternehmen was tun darf. CIEM-Komplettlösungen verringern außerdem den Rechtewildwuchs, nutzen innovative ML-Funktionen sowie verhaltensbasierte Richtlinien für die Anomalie-Erkennung und kombinieren Laufzeit- mit IaC-Daten.

Mit einer zentralen Lösung für die Sicherung vertraulicher Informationen und für CIEM profitiert Ihr Unternehmen von umfassender Cloud-Identitätssicherheit. Sie erhalten eine Übersicht darüber, wer auf sensible Ressourcen wie Schlüsselspeicher und Secrets zugreifen kann und wie diese Secrets verwendet werden. Nicht zuletzt können Sie mithilfe einer solchen Lösung auch die Anzahl der ungenutzten und in anderer Hinsicht riskanten Berechtigungen reduzieren, die zu nachgelagerten Sicherheitsproblemen führen könnten.

Empfohlene Kriterien für Secrets-Scanning- und CIEM-Lösungen:

- Abstimmung von Feedback aus der Suche nach vertraulichen Informationen mit CIEM-Funktionalität, z. B. Transparenz, Verfolgung und Umfangsanpassung
- Bereitstellung von Kontext, welche Secrets von welchen Identitäten verwendet werden
- Reduzierung von ungenutzten und riskanten Berechtigungen

Wenn Sie diese Kriterien im Hinterkopf behalten, können Sie eine umfassende, mehrdimensionale Secrets-Scanning-Lösung auswählen, mit der sich die Risiken durch exponierte Anmeldedaten minimieren lassen.

paloalto NETWORKS

Oval Tower, De Entrée 99–197
1101 HE Amsterdam, Niederlande

Telefon: +31 20 888 1883
Vertrieb: +800 7239771
Support: +31 20 808 4600

www.paloaltonetworks.de