6 Best Practices für DevSecOps für Entwicklungsteams

1 Reduzieren von Risiken aufgrund von Anwendungsabhängigkeit

Möglicherweise gibt es zu behebende Schwachstellen in den Softwarekomponenten, die für Ihre Anwendungsentwicklung verwendet werden. Sie können Tools zur Analyse der Softwarezusammensetzung (Software Composition Analysis, SCA) einsetzen, um die Risiken in der Entwicklungskette zu mindern, besonders wenn Sie Open-Source-Softwarekomponenten verwenden. Suchen Sie nach SCA-Tools, die folgende Fähigkeiten haben:

- Scannen der Anwendungsabhängigkeiten, um sicherzustellen, dass sie frei von bekannten Schwachstellen sind.
- Unterstützung bei der automatischen Einhaltung von Softwarelizenzen durch Identifizieren der Komponenten und ihrer Lizenzen sowie Kennzeichnung möglicherweise inkompatibler Lizenzen.
- Sicherstellen, dass Anwendungsabhängigkeiten aktuell sind und aus einer aktiven Community stammen, die Updates erstellt.
- Sind Teil der automatisierten Anwendungsentwicklung und auch der Entwicklungsumgebung. Dies gibt Entwicklungsteams die Möglichkeit, Probleme zu beheben, bevor Anwendungen integriert werden, was die Anzahl der Fehler bei der Anwendungsentwicklung reduziert.

2 Vereinheitlichung von Code- und Konfigurationsmanagement

Das GitOps-Paradigma, das sich in Kubernetes- und Container-Umgebungen durchgesetzt hat, beinhaltet Verfahren, die Ihre Sicherheitslage erheblich verbessern können. Dies beginnt bereits bei der Entwicklung:

- Anwenden bewährter Entwicklungspraktiken für die Quellcodeverwaltung (Source Code Management, SCM) auf die Konfiguration. Durch Verwenden derselben Steuerelemente für Check-In, Zusammenführung und Genehmigung können Änderungen an der Konfiguration der Infrastruktur bis zu einer bestimmten Person und einem bestimmten Zeitpunkt nachverfolgt werden.
- Anstatt sich auf Ops zu verlassen, sollten Entwicklungsteams die Konfiguration schon früh in den Prozess einbeziehen und eine Vision für die geplante Produktionsumgebung der Anwendung entwickeln. Das Verwenden der gleichen Art von Umgebung und Sicherheitskontrollen für Entwicklung, Test und Produktion erleichtert das Konfigurationsmanagement im gesamten Lebenszyklus.
- Verwenden Sie eine automatisierte Entwicklungspipeline, um Container-Images und binäre Artefakte für die kontinuierliche Integration/Bereitstellung (CI/CD) zu entwickeln. Beim Bereitstellen dieser Images in der Produktion sollten keine Ad-hoc-Änderungen erforderlich werden.
- In einem SCM-System dürfen keine sensiblen Daten gespeichert werden. Verwenden Sie Tools zum Scannen von Konfigurations- und Container-Images, um sicherzustellen, dass sie keine eingebetteten, vertraulichen Daten enthalten.

3 Schutz von Anwendungsgeheimnissen

Es ist wichtig, Identitäten und vertrauliche Daten wie Passwörter, Token und Schlüssel während des gesamten Nutzungszeitraums der Anwendung zu verwalten. Der Zugriff auf SCM-Systeme, Container-Registrys und Binär-Repositories muss reguliert werden. Anmeldeinformationen, die von Anwendungen für den Zugriff auf Datenbanken und Dienste verwendet werden, sowie die für automatisierte Builds und Testprozesse benötigten Daten müssen ebenfalls geschützt werden. Vertrauliche Daten können versehentlich offengelegt werden, wenn sie in SCM-Systemen oder Konfigurationsdateien gespeichert sind. Zum Schutz von Anwendungsgeheimnissen:

- Richten Sie schon früh im Lifecycle eine Infrastruktur zur Identitätsverwaltung und Zugangskontrolle ein.
- Erwägen Sie den Einsatz eines Datentresors oder eines Hardware-Sicherheitsmoduls (HSM) zum Verwalten und Sichern vertraulicher Daten im Ruhezustand und bei der Übertragung. Datentresore sind in der Regel Softwarelösungen, während HSM spezielle Hardware verwenden, um ein höheres Maß an Schutz zu bieten. Beide sollten in die Infrastruktur des Identitätsmanagements integriert werden.

4 Verwenden sicherer Basisimages

Container-Basisimages sind stark minimierte Linux®-Distributionen. Hunderte Pakete können bereits mit potenziellen Sicherheitslücken vorinstalliert sein. Mindern der mit Container-Images verbundenen Risiken:

- Wählen Sie sichere Images mit zuverlässigen, regelmäßigen und sorgfältig getesteten Updates. Untersuchen Sie die Image-Quellen und die verfügbaren Support-Optionen.
- Verwenden Sie Image-Tools, um nach bekannten Schwachstellen zu suchen. Images sollten auch gescannt werden, um sicherzustellen, dass die Konfigurationen sicher sind und keine vertraulichen Daten eingebettet sind.
- Reduzieren Sie Angriffspunkte und entfernen Sie unnötige Binärdateien, einschließlich Betriebssystem-Tools, die bei einem Angriff verwendet werden könnten.

5 Frühzeitig Fragen zu Compliance und Audits ansprechen

Um Verzögerungen bei der Überführung in die Produktion zu vermeiden, ist es wichtig, bereits in der Entwicklungsphase die Bedingungen für die Einhaltung der Vorschriften und die erforderlichen technischen Kontrollen zu kennen. Automatisierte Prüfungen, um Compliance- und Sicherheitsanforderungen durchzusetzen, können in die Entwicklungspipeline eingefügt werden.

Da die Dokumentation von Verfahren und Richtlinien mindestens 50 % eines Audits ausmachen kann, sollten Sie proaktiv mit der Dokumentation beginnen. Zur Richtliniendokumentation sollten Zugangskontrollen, Änderungskontrollen, Backups und Datenaufbewahrung zählen. Sicherheitsprüfungen wie Anwendungssicherheitstests und SCA sollten in die Verfahrensdokumentation einbezogen werden.

6 Solide Plattform und starkes IT-Ökosystem

Zunehmende Sicherheitsbedrohungen erfordern den Einsatz einer Plattform mit einem umfassenden Sicherheits-Ökosystem, das integrierte und unterstützte Lösungen bietet. Red Hat® OpenShift® ist eine Kubernetes-Plattform für Unternehmen und bietet umfangreiche Funktionen für Entwicklung und Betrieb. Die leistungsstarken Entwicklungs- und Deployment-Pipelines in Red Hat OpenShift sind ideal für die Implementierung automatisierter Sicherheitsprüfungen und -kontrollen. An praktisch jedem Punkt des Prozesses können Sicherheitsprüfungen eingefügt werden, vom Entwickeln des Quellcodes in Images bis hin zur Produktion.

Red Hat arbeitet mit einem Netzwerk von Partnern, die die Sicherheitsfunktionen von Red Hat OpenShift kontinuierlich verbessern und erweitern. In Zusammenarbeit mit Red Hat bieten diese Partner unterstützte Lösungen an, die in Red Hat OpenShift integriert sind. Sie können aus einer Reihe von Lösungen wählen, die Ihren spezifischen Sicherheits- und Organisationsanforderungen entsprechen.

Red Hat CodeReady Workspaces ist eine Kubernetes-native Entwicklungsumgebung zur Beschleunigung der Entwicklung containerbasierter Anwendungen. Sie wird auf Red Hat OpenShift eingesetzt. Red Hat Universal Base Images und Red Hat Runtimes bieten Ihren Anwendungen eine solide Basis aus zuverlässiger Quelle.

Framework für Red Hat DevSecOps

Verschaffen Sie sich einen vollständigen Überblick über den Sicherheits-Lifecycle und erfahren Sie, wie die Sicherheitsfunktionen für die Entwicklung mit dem Red Hat DevSecOps Framework kombiniert werden können. Besuchen Sie red.ht/DevSecOps.

Sicherheitslösungen für die Entwicklung finden

In Webcasts von Red Hat und von Sicherheitspartnern von Red Hat erfahren Sie, wie Sie Sicherheit in Ihren Anwendungs-Lifecycle einbinden können.

Über Red Hat

Red Hat unterstützt Kunden dabei, ihre Umgebungen zu standardisieren, cloudnative Anwendungen zu entwickeln und komplexe Umgebungen mit vielfach ausgezeichnetem Support, Training und Consulting Services zu integrieren, zu automatisieren, zu sichern und zu verwalten.

EUROPA, NAHOST UND AFRIKA (EMEA)
00800 7334 2835
de.redhat.com
europe@redhat.com

TÜRKEI
00800 448820640

ISRAEL
1 809 449548

VAE
8000-4449549

facebook.com/redhatinc
@RedHatDACH
linkedin.com/company/red-hat

de.redhat.com
#F29682_0921