Frachtkontrolle

Container-Images mit Trivy auf Sicherheitslücken durchleuchten

Docker verführt wegen der einfachen Handhabung dazu, Container-Images als Blackbox zu behandeln. Welche Abhängigkeiten in einem Image stecken und ob diese Sicherheitslücken mitbringen, ist nicht immer klar ersichtlich. Mit dem Open-Source-Scanner Trivy klopfen Sie Ihre Images auf Schwachstellen ab.

Von Niklas Dierking

Container erfreuen sich bei Entwicklern, Administratoren und Anwendern großer Beliebtheit, weil sie viel Arbeit abnehmen. Alle Abhängigkeiten, die ein laufender Prozess im Container benötigt, stecken bereits im Container-Image. Einen einsteigerfreundlichen Überblick über Docker verschafft Ihnen der Artikel ab Seite 146.

Der Docker Hub ist die größte Sammlung von Container-Images (eine sogenannte Container-Registry). Dort kann jeder Images für alle möglichen Anwendungsbereiche hochladen. Ob ein Image vertrauenswürdig ist oder Sicherheitslücken enthält, ist jedoch nicht leicht auf Anhieb zu erkennen.

Ein Forscherteam der Technisch-Naturwissenschaftlichen Universität Norwegen untersuchte im vergangenen Jahr 2500 zufällig ausgewählte Docker-Images aus dem Docker Hub auf bekannte Sicherheitslücken. Von den zertifizierten Images wiesen 82 Prozent mindestens eine Sicherheitslücke auf, die laut Bewertungssystem CVSS (Common Vulnerability Scoring System) als „schwerwiegend“ gelten. Bei den Community-Images waren es 68 Prozent der Abbilder. Am besten haben die offiziellen Images abgeschnitten, die das Unternehmen Docker selbst auf Schwachstellen kontrolliert. Aber auch diese enthielten noch zu 46 Prozent eine schwerwiegende Lücke. Die Studie haben wir unter ct.de/y2pe verlinkt.

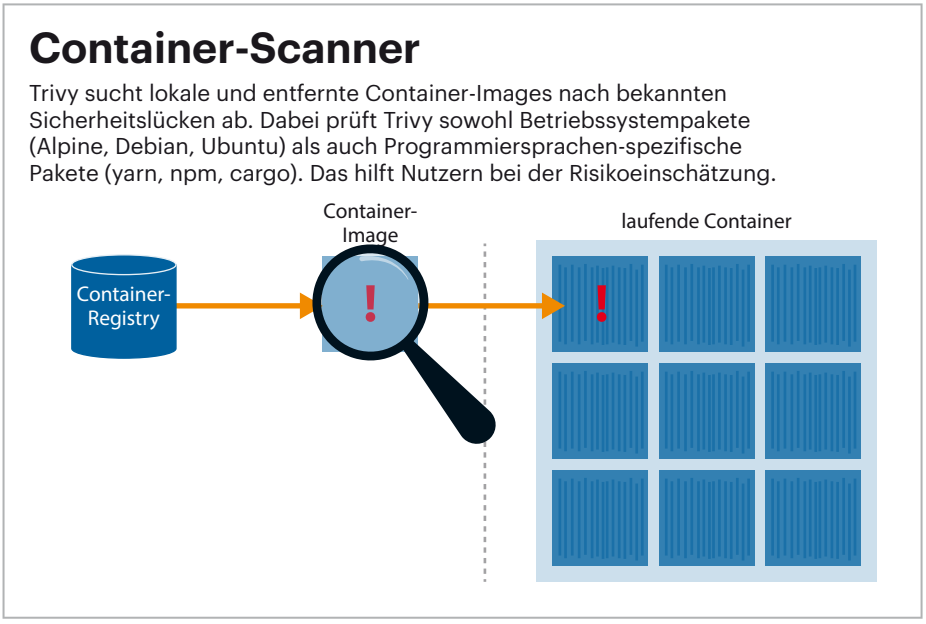
Mit dem Open-Source-Tool „Trivy“ durchleuchten Sie Images nach bereits katalogisierten Sicherheitslücken. Trivy gleicht dazu die Betriebssystempakete des Basis-Image (Alpine, Debian, Ubuntu), sprachenspezifische Pakete (npm, yarn, cargo), Bibliotheken und weitere Abhängigkeiten mit einem eigenen Schwachstellenregister ab. Das befüllt Trivy aus der etablierten Schwachstellendatenbank NVD (National Vulnerability Database), die sich aus dem CVE-Verzeichnis (Common Vulnerabilities and Exposures System) speist und Sicherheitslücken mit dem Bewertungssystem CVSS (Common Vulnerability Scoring System) einen Schweregrad zuweist. Darüber hinaus nutzt Trivy die CVE-Tracker und Security-Ratgeber diverser Open-Source-Projekte wie den Ubuntu-CVE-Tracker, die Rustsec-Advisories oder die Go Vulnerability Database.

Images durchleuchten

Sie können Trivy einsetzen, um lokale Container-Images zu scannen, ebenso wie entfernte Abbilder, die in einer Container-Registry wie dem Docker Hub oder der GitHub-Container-Registry abgelegt sind. Das funktioniert auch mit Images, die Sie mit dem Befehl `docker save` als Tar-Archiv exportiert haben. Die Software unterstützt das von Docker verwendete OCI- und das Podman-Format. Ebenso durchsucht Trivy den Code öffentlicher GitHub-Repositories nach bekannten Schwachstellen. Trivy gibt es als Paket für die Linux-Distributionen RHEL, CentOS, Debian, Ubuntu und Arch-Linux sowie für macOS via Homebrew-Paketmanager. Eine Anleitung, wie Sie Trivy auf Ihrem System installieren, finden Sie in der Dokumentation des Projekts, die wir unter ct.de/y2pe verlinkt haben.

Auch Trivy selbst läuft im Container. Damit können Sie sich die Installation sparen und den Scanner auch unter Windows und macOS mit installiertem Docker einsetzen. Ersetzen Sie dazu in den beiden folgenden Befehlen einfach trivy durch diesen Befehl:

```
docker run -it aquasec/trivy
```



Sie bedienen Trivy mittels Kommandozeile. Wenn Sie Trivy auf dem Hostsystem installiert haben, dann scannen Sie ein Image, beispielsweise Python 3.4 mit Alpine-Linux als Basis, mit dem folgenden Befehl:

```
trivy image python:3.4-alpine
```

Außer Container-Images können Sie auch öffentliche GitHub-Repositories nach Schwachstellen durchsuchen:

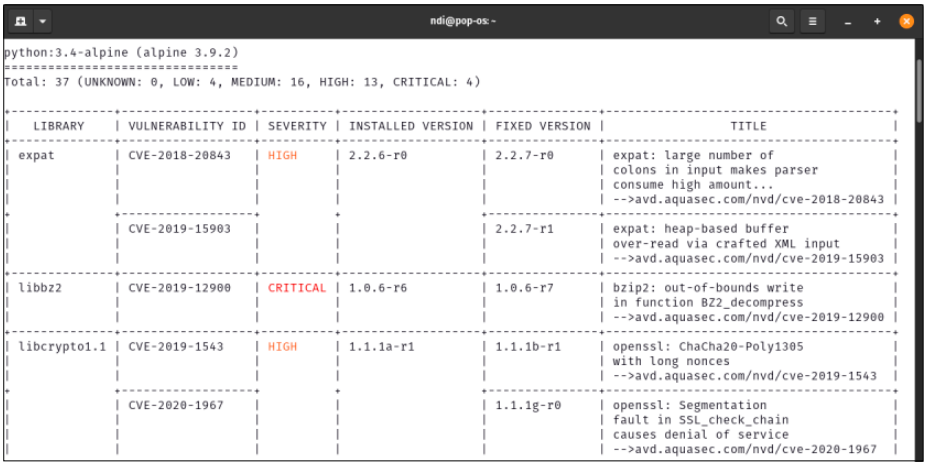
```
trivy repo https://github.com/↵
↳knqyf263/trivy-ci-test
```

Wenn Sie Trivy installiert haben, dann lädt das Tool beim ersten Scan die Schwachstellendatenbank herunter, was einige Sekunden in Anspruch nehmen kann. Die nachfolgenden Scans erfolgen deutlich schneller, weil Trivy die Datenbank lokal als Cache ablegt und künftig nur noch Änderungen herunterlädt. Nutzen Sie Trivy selbst als Docker-Container, dann wird die Datenbank nicht automatisch zwischengespeichert, es sei denn, Sie reichen einen Ordner für den Cache in den Container hinein. Das erledigen Sie, indem Sie den `docker run`-Befehl wie folgt anpassen:

```
docker run --rm -v ${PWD}/cache:↵
↳/root/.cache/ aquasec/trivy ↵
↳python:3.4-alpine
```

Nach dem Abgleich mit der Schwachstellendatenbank präsentiert Trivy Ihnen die Ergebnisse des Scans standardmäßig in einer Tabelle auf der Kommandozeile.

In der Kopfzeile der Tabelle finden Sie eine Zusammenfassung der Ergebnisse. Für das Beispiel-Image `python:3.4-alpine` hat Trivy insgesamt 37 Schwachstellen gefunden, denen es die Schweregrade „Unknown“, „Low“, „Medium“, „High“ und „Critical“ in der Spalte „Severity“ zuordnet. In der Tabelle finden Sie unter „Library“ auf der linken Seite das Paket oder die Programmbibliothek, die die Schwachstelle enthält. „Vulnerability-ID“ listet die Kennung, mit der die Schwachstelle katalogisiert wurde. Der Spalte „Installed Version“ entnehmen Sie, in welcher Version die betroffene Abhängigkeit vorliegt. Besonders praktisch: Unter „Fixed Version“ informiert Trivy Sie direkt, in welcher Version die Entwickler das Problem behoben haben. In der „Title“-Spalte ganz rechts lesen Sie eine Kurzbeschreibung der Schwachstelle und finden einen Link mit weiterführenden Informationen.



Trivy listet alle relevanten Informationen zu gefundenen Schwachstellen in einer übersichtlichen Kommandozeilenausgabe auf. CVE-Nummer und Kurzbeschreibung dienen der weiteren Recherche.

Blick schärfen

Lassen Sie sich von der Flut der gefundenen Schwachstellen zunächst nicht verunsichern. Nur weil ein Container-Image bekannte Schwachstellen enthält, heißt das nicht automatisch, dass in Ihrem konkreten Anwendungsfall ein Sicherheitsrisiko besteht. So könnte es möglich sein, dass der Prozess im laufenden Container gar nicht mit der betroffenen Abhängigkeit interagiert.

Um die Analyse einzugrenzen, können Sie Trivy auch anweisen, nur besonders schwerwiegende Schwachstellen auszugeben. Mit folgendem Befehl lassen Sie sich beispielsweise nur Lücken anzeigen, die als „High“ oder „Critical“ eingestuft werden:

```
trivy image --severity HIGH,CRITICAL \
python:3.4-alpine
```

Wenn Sie bereits recherchiert haben, dass bestimmte Lücken für Ihren Anwendungsfall nicht relevant sind, dann können Sie in dem Verzeichnis, in dem Sie Trivy ausführen, auch eine Textdatei namens `.trivyignore` anlegen. Dort gelistete Schwachstellen ignoriert Trivy beim nächsten Scan. Beispiel:

```
# Risiko ist akzeptabel
CVE-2019-19242
# Für Anwendungsfall nicht relevant
CVE-2020-11655
```

Seine volle Stärke entfaltet Trivy, wenn Sie Scans automatisieren, beispielsweise mittels GitHub Actions. So können Sie beispielsweise spezifizieren, welchen Exit-Code Trivy ausgeben soll, wenn bestimmte Schwachstellen gefunden werden:

```
trivy image --exit-code 0 --severity \
MEDIUM,HIGH python:3.4-alpine

trivy image --exit-code 1 --severity \
CRITICAL python:3.4-alpine
```

Das ist besonders hilfreich, wenn Sie planen, Trivy in automatisierte Buildprozesse oder eine CI/CD-Lösung einzubinden. In diesem Beispiel schlägt der Buildprozess fehl, wenn Trivy eine kritische Sicherheitslücke aufspürt und den Exit-Code 1 ausgibt. Der Prozess läuft aber durch, falls es sich lediglich um eine mittelschwere Schwachstelle handelt. Die Entwickler stellen in der Dokumentation des Projektes Vorlagen bereit (siehe ct.de/y2pe), um Trivy in GitHub Actions und weitere CI/CD-Lösungen wie GitLab CI oder Travis CI zu integrieren.

Fazit

Mit Trivy verschaffen Sie sich im Handumdrehen einen Überblick über mögliche Schwachstellen in Container-Images und können so eine Risikoeinschätzung vornehmen. Das Scan-Tool hilft Administratoren und Container-Enthusiasten bei der Suche nach Images mit der kleinstmöglichen Angriffsfläche. Fortgeschrittene Anwender unterstützt Trivy als Kontrollmechanismus in automatisierten Buildprozessen. *(ndi@ct.de)* c't

Trivy-Dokumentation und GitHub-Repository: ct.de/y2pe