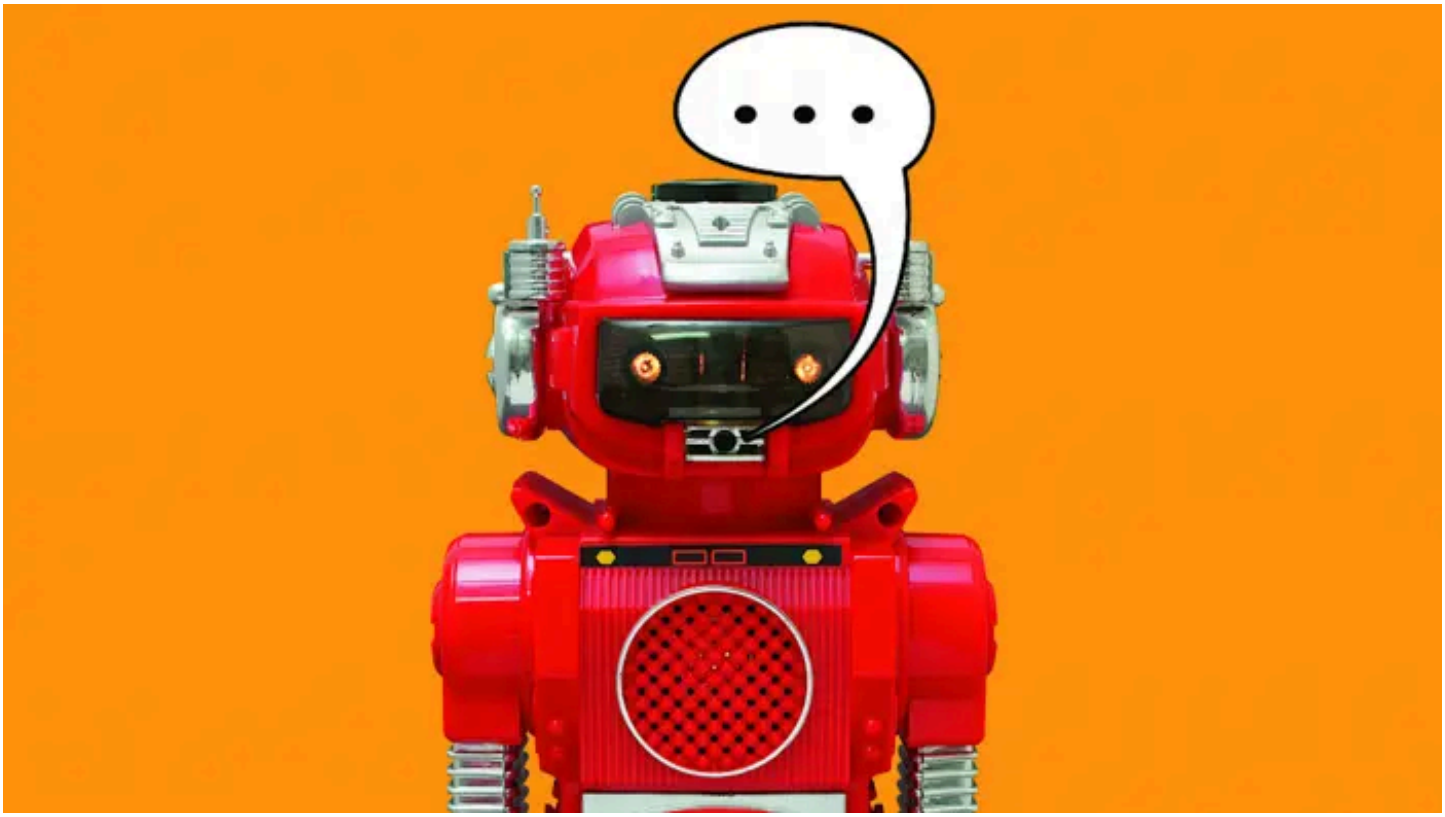
MAGAZIN FÜR PROFESSIONELLE
INFORMATIONSTECHNIK

Drei Fragen und Antworten: Was kann man gegen unsichere Kubernetes-Cluster tun?

24.11.2022 07:30 Uhr Jonas Volkert



Per Kubernetes verwaltete Container-Cluster sind nicht von Haus aus sicher. Dafür ist die richtige Konfiguration wichtig, erklärt iX-Titelautor Christoph Puppe.

Frisch implementiert, sind Kubernetes-Cluster ein Sicherheitsrisiko. Das lässt sich aber ändern: Mit den richtigen Maßnahmen ist eine mit K8s (Kubernetes) verwaltete Container-Infrastruktur der traditionellen IT auch in puncto Security überlegen. Wie, das erklärt iX-Titelautor Christoph Puppe im Interview.

DREI FRAGEN UND ANTWORTEN MIT CHRISTOPH PUPPE

Christoph Puppe ist Managing IT Security Architect und Auditteamleiter für ISO 27001 nach Grundschutz bei SVA System Vertrieb Alexander GmbH, Mitautor des Grundschutz-Kompendiums und ehemaliger Penetrationstester.

? Christoph, warum kann – bei korrekter Konfiguration – der IT-Betrieb über K8s sicherer sein als bei traditioneller IT?

! Container sind zwar eigentlich erstmal ein praktisches Werkzeug zur Paketierung von Anwendungen, aber mit den Namespaces im Kernel und den anderen Limits, die die cGroups bieten, auch eine Sicherheitsbarriere. So wie es früher ChangeRoot oder Jails bei BSD waren, sind es heute Container, die zusätzlichen Schutz bieten. Kubernetes erlaubt es, diese Funktionen schnell und einfach einzusetzen. Auch schützt die mit Kubernetes mögliche Automatisierung des Betriebs vor Fehlern der Betriebsmannschaft. Everything as Code, von Infrastruktur bis Sicherheitseinstellungen, ermöglicht es Fehler zu reduzieren. Oder, mit etwas Pech, Fehler gleich global auszurollen.

Grundsätzlich ist mit Kubernetes und damit den Containern die Kontrolle über die auf unseren Servern laufenden Prozesse granularer geworden. Wo früher das BSI sagte "Ein Dienst pro Server", haben wir jetzt einen Dienst pro Pod. Und dieser wird automatisch überwacht und neu gestartet, wenn er nicht mehr antwortet und – sofern man Sicherheitsprodukte einsetzt – wenn er Dinge tut, die er nicht tun darf. Was hier auch klar zu sagen ist: Ein Kubernetes Cluster ist erstmal sehr unsicher. Nur die gute Parametrisierung von IAM (Identity and Access Management), über Netz-Richtlinien bis Überwachung bringt dann die zusätzliche Sicherheit. Wie alles bei Kubernetes hängt es immer von der zusätzlichen Software ab, wie sicher der Cluster ist oder sein kann.

? Müssen sich Admins einen komplett eigenen Plan zurechtlegen oder gibt es Orientierungspunkte für den sicheren K8s-Betrieb? Und: Gibt es eine Priorisierung der Maßnahmen?

! Jede Behörde oder Organisation mit "Sicherheit" im Titel hat mittlerweile einen Standard oder eine Anleitung für die Absicherung veröffentlicht. Auch das BSI hat seit diesem Jahr zwei Bausteine im Grundschutzkompendium. Der "SYS.1.6 Containerisierung" und "APP.4.4 Kubernetes", die ich die Freude hatte, schreiben zu dürfen, umfassen alle Phasen des Betriebs, inklusive Planung sowie CI/CD-Pipelines und Werkzeuge dafür. Die Anforderungen sind aufgeteilt in "Basis", "Standard" und "Erhöht". In dieser Reihenfolge sind sie auch abzuarbeiten.

Auch sehr hilfreich und mehr auf die technischen Details der Konfiguration fokussiert ist der Kubernetes-Benchmark des Center for Internet Security (CIS). Dies ergänzt die BSI-Bausteine sehr gut, so wie die Bausteine auch die Benchmarks ergänzen, da die Bausteine auch die organisatorischen Themen adressieren und die Benchmarks extrem detaillierte Vorgaben für die Konfiguration sind. Bei den technischen Details hat der CIS-Benchmark eine Aufteilung in zwei Level, wobei das zweite für verstärkte Absicherung gedacht ist, aber auch schon das erste Level sehr umfangreich ist.

? Warum lohnt es sich, auch die komplexen Linux-Kernel-Komponenten Capabilities, SECCOMP und SELinux zu beherrschen?

! Diese Funktionen existieren schon lange und kaum jemand nutzt sie. Was schade ist, weil sie viele Angriffe verhindern könnten. Sie sind leider komplex in der Konfiguration und es besteht immer das Risiko, dass sie auch mal einen erlaubten Prozess beenden oder einen erlaubten Zugriff verhindern. Automatisierung und solide Tests sind hier die Chance, dieses Triumvirat der Sicherheitsfunktionen auch wirklich einzusetzen. Also nicht mit Profilen, die fast alles zulassen, sondern dass jeder Container ein eigenes und sehr restriktives Profil bekommt. Dann können diese Werkzeuge ihre Stärken ausspielen und Containern zu einer Sicherheit verhelfen, die auf der Legacy-IT nur selten zu finden ist.

Wem das zu kompliziert ist, der kann sich auch mit etwas Geld behelfen und eine der kommerziellen Cloud Workload Protection Platforms (CWPP) kaufen. Die administrieren sich zumindest teilweise von alleine, da sie durch die Beobachtung von Containern die Profile erlernen. Diese Profile sind dann maximal restriktiv und sehr effizient beim Verhindern von Angriffen gegen bekannte und unbekannte Schwachstellen. Auch lassen sich diese Werkzeuge gut in die CI/CD einbinden. So wird der zusätzliche Schutz automatisiert zum Teil des Betriebs.

Christoph, vielen Dank für Deine Antworten. Die **Titelstrecke der neuen iX 12/22 [1]** ist gleichzeitig der Startschuss einer neuen Tutorial-Reihe zum Absichern von Kubernetes. Ein weiterer Artikel beleuchtet die im Interview angesprochenen Linux-Komponenten Capabilities, SECCOMP und SELinux. Ab sofort finden iX-Abonnentinnen und -Abonnenten außerdem **Zugriff auf die Artikel über Heise+ [2]**.

*In der Serie „Drei Fragen und Antworten" will die iX die heutigen Herausforderungen der IT auf den Punkt bringen – egal ob es sich um den Blick des Anwenders vorm PC, die Sicht des Managers oder den Alltag eines Administrators handelt. Haben Sie Anregungen aus Ihrer tagtäglichen Praxis oder der Ihrer Nutzer? Wessen Tipps zu welchem Thema würden Sie gerne kurz und knackig lesen? Dann **schreiben Sie uns gerne [3]** oder hinterlassen Sie einen Kommentar im Forum.*

[4]

(jvo [5])

URL dieses Artikels:

https://www.heise.de/-7349827

Links in diesem Artikel:

[1] https://www.heise.de/select/ix/2022/12

[2] https://www.heise.de/plus/

[3] mailto:fo@ix.de

[4] https://www.heise.de/ix/

[5] mailto:jvo@ix.de