Sicherheitsaspekte im Zusammenhang mit Container-Virtualisierung

Oliver Dieke, 11.07.2022

Container heutzutage sehr verbreitet
- Umfrage 2019: 87% befragter Personen,
- 2017: 55%
- (501 IT-Mitarbeiter befragt)

ggü. anderer Techniken gewisse neue Sicherheitsrisiken

[2], [9]
Bild: Container, 15.06.2022, 01:00, https://www.mtcontainer.de/container/hardtop-container/
Bild: Security, 15.06.2022, 01:00, https://southpark.fandom.com/wiki/Security_Guard

Agenda

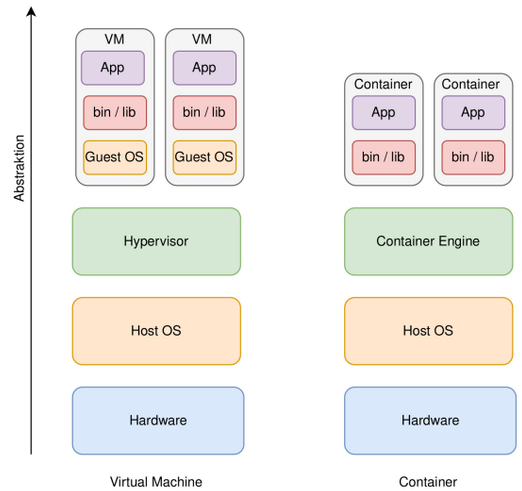
Agenda kurz durchgehen
Hinweis auf Verlinkung der Quellen und Aushändigung der Präsentation und Moderatornotizen

„Container sind eine **Virtualisierungstechnik** im Computerumfeld, die Anwendungen inklusive ihrer Laufzeitumgebungen voneinander trennt.“[1]

Definition Container
vorlesen,
kurz drauf eingehen,
Überleitung zur Grafik auf nächster Folie

[1]

1 Basiswissen



- OS-Level-Virtualisierung (vs. Hardware-Level-Virtualisierung bei VM)
- Isolierung von Prozessen (vs. Isolierung von Maschinen bei VM)

- normalerweise teilen sich Prozesse Ressourcen, hier nicht
- Features die diese Illusion (Glauben getrennt zu sein) ermöglichen:
    - namespaces: zur Individualisierung: Host-OS muss logische Umgebungen schaffen und verwalten, in denen Prozesse, Dateien und Netzwerk voneinander getrennt sind -> große Verantwortung für Host-OS
    - cgroups: kontrollieren der Ressourcen -> Monitoring und Metering

- Vorteil 1: quasi unendlich Portability (kann überall deployed werden, da der Container Informationen über alles enthält, was er braucht wie Bibliotheken usw.)
- Vorteil 2: minimiert Configuration Drifts, da Container zerstört und reproduziert werden

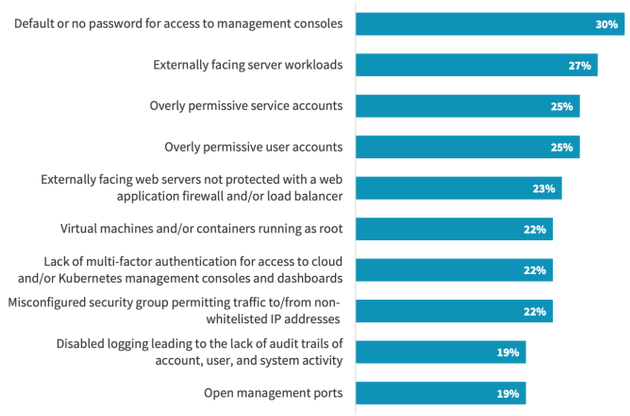
(vs. Vorteil: quasi unendlich Hardware Flexibility bei VM)

[1], [3], [5], [10]

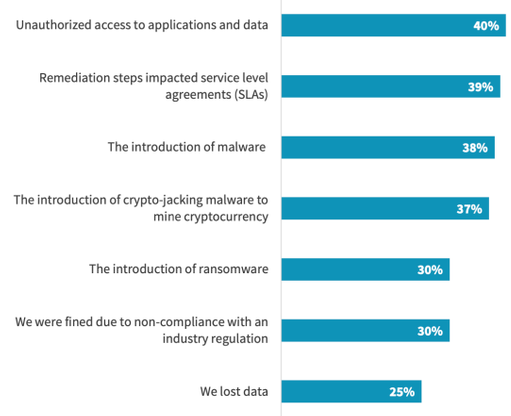
2 Häufige Fehler und Beispiele

Häufigste Fehlkonfigurationen und Folgen [2]

Ten most common cloud misconfigurations in the past 12 months.



Results of cloud misconfigurations.



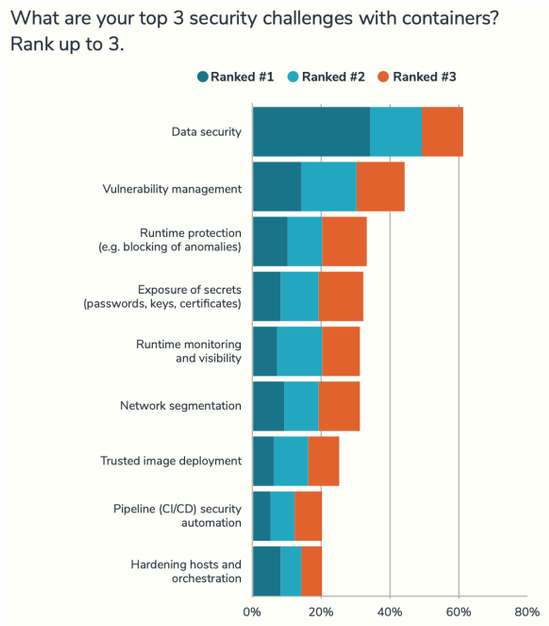
Misskonfigurationen
Default-Passwort oder gar keins
Server Workloads
Zugriffsrechte falsch gesetzt

Folgen
Unauthorisierter Zugriff zu Anwendungen und Daten
Malware
Kryptomining
Datenverlust

[2]

2 Häufige Fehler und Beispiele

Top 3 Security Challenges [9]



Datensicherheit
Schwachstellenmanagement
Runtime protection
Vertrauliche Daten geheim halten
Runtime monitoring
Netzwerksegmentierung

[9]

2 Häufige Fehler und Beispiele

Tesla

https://redlock.io › blog › cryptojacking-tesla

Lessons from the Cryptojacking Attack at Tesla

20 Feb 2018 — The **hackers** had infiltrated **Tesla's Kubernetes** console which was not password protected. Within one **Kubernetes** pod, access credentials were ...

https://arstechnica.com › 2018/02 › tesla-cloud-resourc...

Tesla cloud resources are hacked to run cryptocurrency ...

20 Feb 2018 — "The **hackers** had infiltrated **Tesla's Kubernetes** console which was not password protected," RedLock researchers wrote. "Within one **Kubernetes** ...

https://www.wired.com › Security › Tesla

Hackers Hijacked Tesla's Cloud to Mine Cryptocurrency

20 Feb 2018 — **Hack** Brief: **Hackers** Enlisted **Tesla's** Public Cloud to Mine Cryptocurrency. The recent rash of cryptojacking attacks has hit a **Tesla** database ...

https://blog.neuvector.com › article › cryptojacking-cry...

Cryptojacking and Crypto Mining – Tesla, Kubernetes ... - Blog

Tesla and Jenkins have become the latest victims of data infiltration and cryptojacking. In the **Tesla** case, the exploits started when a **Tesla Kubernetes** cluster ...

https://techbeacon.com › security › tesla-drives-cryptoja...

Tesla drives cryptojack gang's AWS cloud down Kubernetes ...

A **Tesla**-owned AWS account was **hacked** to mine Monero. · The **hackers** drove straight in using an “unsecured” **Kubernetes** admin console (i.e., it had no password).

Hacker hatten Zugriff auf Kubernetes Konsole (nicht passwortgeschützt)
Resultat: Zugriff auf vertrauliche Daten und Kryptomining über einen Kubernetes Pod

[2], [7], [8], [9]

2 Häufige Fehler und Beispiele

Angriffsmöglichkeiten

- **Container Escape Attack**

Container Escape Attack

- durch Ausführen von Code mit Kernelfunktionen auf dem Container kann es passieren, dass dieser möglicherweise abstürzt oder die User-Rechte zu root-Rechten setzt und komplette Kontrolle über den Host übernimmt

[2], [7], [8], [9]

2 Häufige Fehler und Beispiele

Angriffsmöglichkeiten

- Container Escape Attack

- **Dangling Volume**

Dangling Volume

- Container speichern Daten in bestimmten Bereichen, welche mit dem Container zusammen zerstört werden
- durch bestimmte Befehle ist es möglich, dass Daten außerhalb dieses Bereiches dauerhaft gespeichert werden
- Angreifer kann diese Daten dann über den Kernel auslesen

[2], [7], [8], [9]

2 Häufige Fehler und Beispiele

Angriffsmöglichkeiten

- Container Escape Attack
- Dangling Volume
- **Backdooring Images**

Backdooring Images

- Container werden anhand von Images, welche in Image-Registries gespeichert werden, erstellt
- Angreifer könnten dieses Image modifizieren oder austauschen

[2], [7], [8], [9]

3 Sicherheitsrisiken und Schutzmöglichkeiten

Unterteilung

- Allgemein

- Containerspezifisch

Bsp. lassen sich in zwei Bereiche unterteilen: allgemein und containerspezifisch

3 Sicherheitsrisiken und Schutzmöglichkeiten

3.1 Allgemein

- SQL Injection
- DoS Attacke
- Social Engineering
- Passwörter
- …

betreffen auch VMs und viele andere IT-Komponenten
Beispiele:
- SQL Injection -> prepared statements
- DoS Attacke -> DoS protection
- Social Engineering -> Prävention- und Aufklärungsarbeit
- Passwörter -> Standardpasswörter ändern, sichere Passwörter (siehe Tesla)

3 Sicherheitsrisiken und Schutzmöglichkeiten

3.2 Containerspezifisch

1. Container Images
2. Image Registries
3. Runtime
4. Orchestrierungsplattformen
5. Host-OS

-> kurz erklären was was ist

1. Container Images
2. Image Registries (Speicher der Container Images)
3. Runtime
4. Orchestrierungsplattformen (wie Kubernetes)
5. Host-OS (OS der Hardware auf dem Container laufen)

[5], [6], [11], [13], [14]

3 Sicherheitsrisiken und Schutzmöglichkeiten

3.2 Containerspezifisch

1. **Container Images**
2. Image Registries
3. Runtime
4. Orchestrierungsplattformen
5. Host-OS

- up to date halten, Sicherheitspatches und -updates installieren
- Images regelmäßig scannen um Sicherheitsrisikos und Veränderungen zu erkennen
- Signing („Fingerabdruck“) um kryptografisch die Unverfälschtheit sicherzustellen

[5], [6], [11], [13], [14], [15]

3 Sicherheitsrisiken und Schutzmöglichkeiten

3.2 Containerspezifisch

1. Container Images
2. **Image Registries**
3. Runtime
4. Orchestrierungsplattformen
5. Host-OS

- privat halten und absolute Kontrolle über Typ, Anzahl, Nutzerzugriffe
- Monitoring
- sicherer Registry Host Server zur Vermeidung von Kompromittierungen

[5], [6], [11], [13], [14], [15]

3 Sicherheitsrisiken und Schutzmöglichkeiten

3.2 Containerspezifisch

1. Container Images
2. Image Registries
3. **Runtime**
4. Orchestrierungsplattformen
5. Host-OS

etwas schwierig, weil Container Security Tools eher die Kommunikation anstatt das Geschehen im Container überwachen
- App-Sicherheit up to par halten („at an expected or usual quality“)
- Netzwerkverkehr und Daten monitoren

[5], [6], [11], [13], [14], [15]

3 Sicherheitsrisiken und Schutzmöglichkeiten

3.2 Containerspezifisch

1. Container Images
2. Image Registries
3. Runtime
4. **Orchestrierungsplattformen**
5. Host-OS

haben bereits viel access control capabilities
- Limits setzen wie Anzahl privilegierter User und deren Rechte
- Monitoring der Plattform und der pod Kommunikation innerhalb

[5], [6], [11], [13], [14], [15]

3 Sicherheitsrisiken und Schutzmöglichkeiten

3.2 Containerspezifisch

1. Container Images
2. Image Registries
3. Runtime
4. Orchestrierungsplattformen
5. **Host-OS**

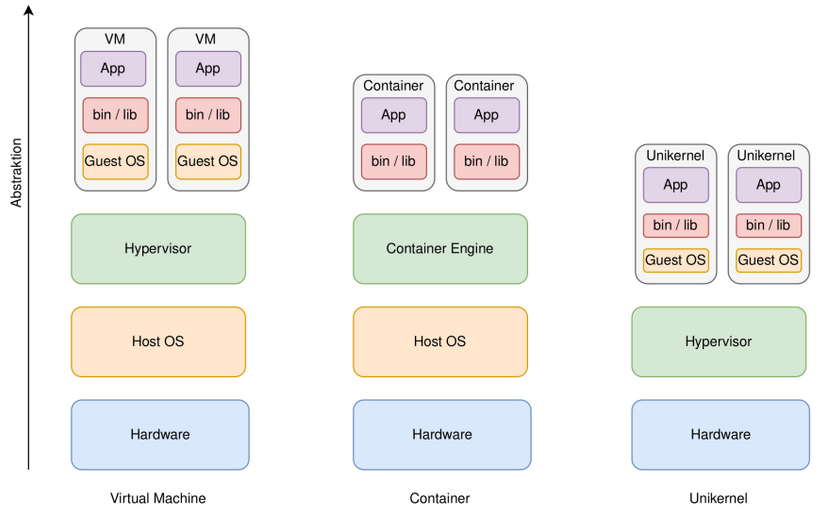
Größtes Risiko und großer Schaden möglich, weil wenn Host-OS kompromittiert = alle Container darauf kompromittiert und möglicherweise hat ein Angreifer Zugriff auf die gesamte Umgebung

- Zugriffsrechte und Kontrolle im OS
- Monitoring
- schmales OS, ohne viel Schnickschnack -> Übergang zu Unikernels

[5], [6], [11], [13], [14], [15]

3 Sicherheitsrisiken und Schutzmöglichkeiten

Unikernels

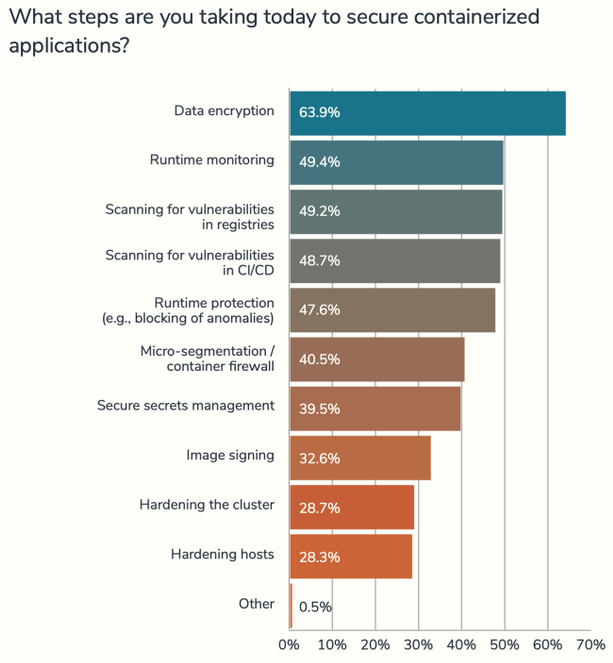


Host-OS - Lösung der Sicherheitsprobleme: Unikernels

- Kompilieren Source-Code in ein individualisiertes OS, welches nur die Funktionen besitzt, die von der Anwendung benötigt werden
- Vorteile: klein, schnell und sicher

[5], [6], [11], [13], [14]

4 Was Unternehmen unternehmen
[9]

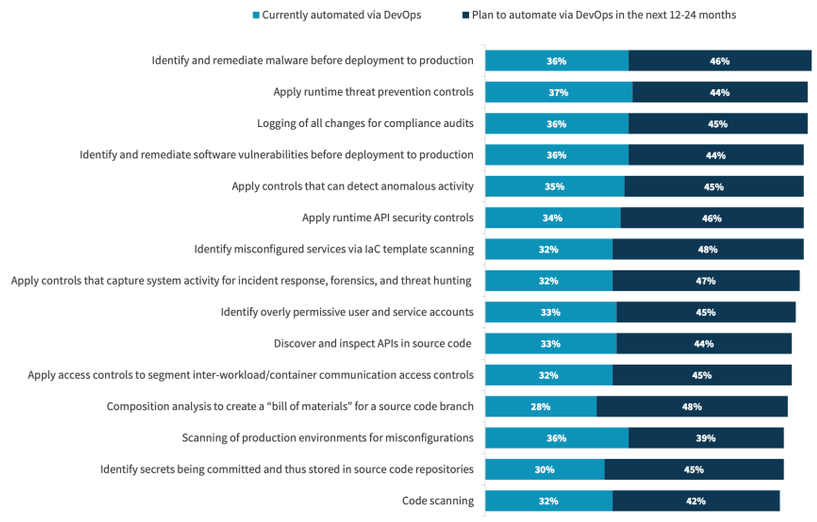


Viele der eben erwähnten möglichen Maßnahmen sichtbar:
Datenverschlüsselung
Monitoring
Scanning
Runtime protection
Image Signing

[9]

4 Was Unternehmen unternehmen

[2]



Automatisierung von Sicherheitsmaßnahmen:
Malware und sonstige Anomalien erkennen und entfernen vor Deployment
Logging
Erkennung von Accounts mit Rechteüberschreitungen

[2]

5 Zusammenfassung und Fazit

Container-basierte Virtualisierung hat viele Vorteile wie
- gute Performance,
- sehr flexibel und portabel - kann fast überall deployed werden (mögliche Einschränkung z.B. durch ARM vs. x86),
- ermöglichen gute Elastizität und Skalierbarkeit von Anwendungen

<u>Sicherheit</u>
- Neue Sicherheitsrisiken, aber oft einfache, bewährte Lösungen -> akzeptabler Aufwand
- Viel Automatisierung notwenig -> möglicherweise fundamentale Änderungen in Betriebsabläufen
- Herausforderung: Portability der Sicherheit (plattformunabhängig)

<u>Fazit</u>
- Container ermöglichen es Unternehmen von manuellen und teuren Sicherheitsmodellen auf besser skalierbare und effizientere zu wechseln
- Insgesamt gute Balance zwischen Sicherheit, Performance und User Experience möglich

[7], [15]

Quellen

[1]	Was sind Container?	Dipl.-Ing.(FH) Stefan Huber, Dr. Jürgen Ehneß	13.06.2022, 21:25	Link
[2]	The Maturation of Cloud-native Security: Securing Modern Applications and Infrastructure	Enterprise Strategy Group	18.05.2022, 23:00	Link
[3]	Container vs. VM security: Which is better?	Ed Moyle	18.05.2022, 23:10	Link
[4]	From virtualization security issues to cloud protection opportunities: An in-depth analysis of system virtualization models	Maxime Compastié, Rémi Badonnel, Olivier Festor, Ruan Heb	18.05.2022, 23:20	Link
[5]	Container Security: Issues, Challenges, and the Road Ahead	Sari Sultan, Imtiaz Ahmad, Thasos Dimitriou	18.05.2022, 23:25	Link
[6]	A Survey on Security Isolation of Virtualization, Containers, and Unikernels	Michael J De Lucia	18.05.2022, 23:30	Link
[7]	Study of Container-Based Virtualisation and Threats in Fog Computing	Poornima Mahadevappa, R.K. Murugesan	10.06.2022, 21:20	Link
[8]	Lessons from the Cryptojacking Attack at Tesla	RedLock CSI Team	11.06.2022, 23:20	Link
[9]	2019 Container Adoption Survey	Portworx and Aqua Security	11.06.2022, 23:25	Link
[10]	Containers vs VMs: What's the difference?	IBM Technology	10.06.2022, 22:00	Link
[11]	Container Security Explained	IBM Technology	11.06.2022, 22:45	Link
[12]	Container Security 101	The Linux Foundation	11.06.2022, 23:00	Link
[13]	Unikernel Systems is now part of Docker	Docker	12.06.2022, 12:00	Link
[14]	Unikernel Technologies	Docker	12.06.2022, 12:10	Link
[15]	NIST Application Container Security Guide	Murugiah Souppaya, John Morello, Karen Scarfone	16.06.2022, 23:50	Link