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
Download am Ende über tinyURL

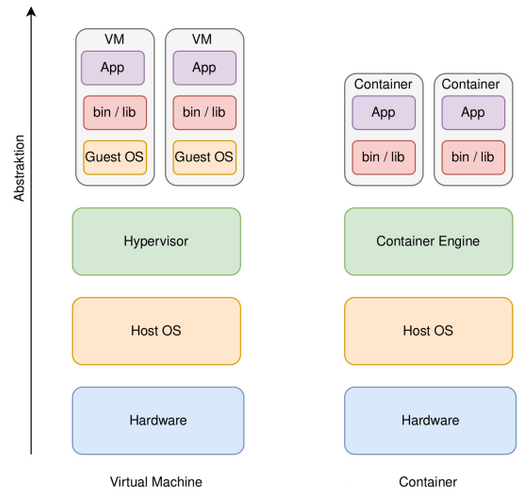
„Container sind eine **Virtualisierungstechnik** im Computerumfeld, die Anwendungen inklusive ihrer Laufzeitumgebungen voneinander trennt.“[1]

Definition Container
vorlesen
kurz drauf eingehen
Überleitung zur Grafik auf nächster Folie (Virtualisierungstechniken)

[1]

1 Basiswissen

Virtual Machine vs. Container



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

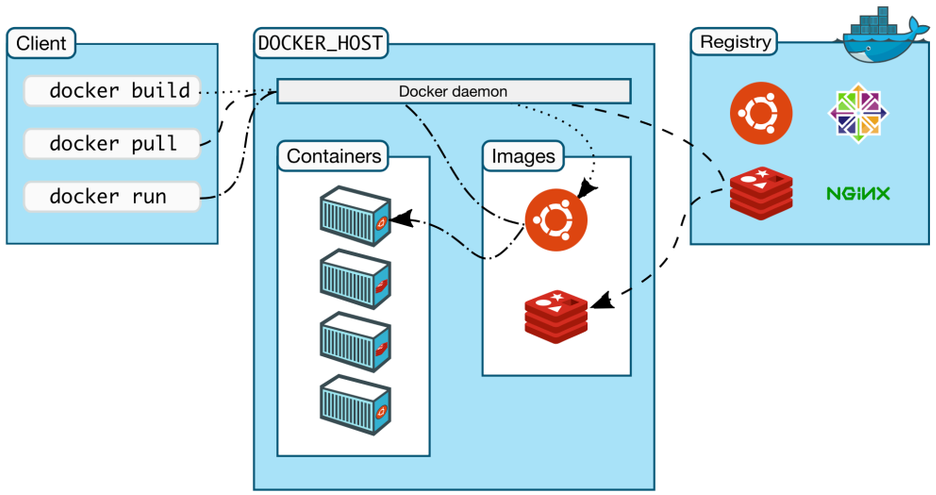
- Vorteil 1: quasi unendlich Portability (kann überall deployed werden, da der Container Informationen über alles enthält, was er braucht wie Bibliotheken usw.)
- Vorteil 2: minimiert Configuration Drifts, da Container zerstört und reproduziert werden

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

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

1 Basiswissen

Container-Architektur am Beispiel von Docker



[16]

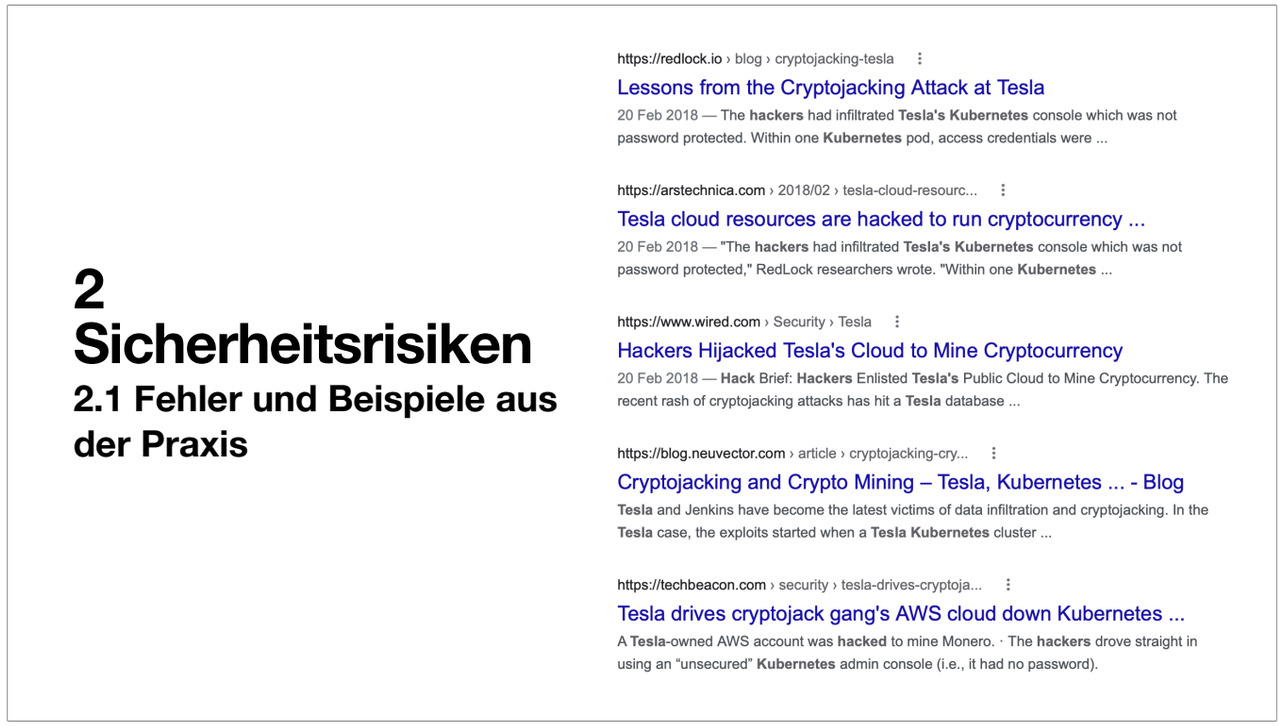
Am Beispiel von Docker, weil es am weitesten verbreitet ist.

Registry - lagert Images
Docker daemon - nimmt Client commands entgegen
Docker Host - baut und kostet Container

[16]

2 Sicherheitsrisiken

2.1 Fehler und Beispiele aus der Praxis



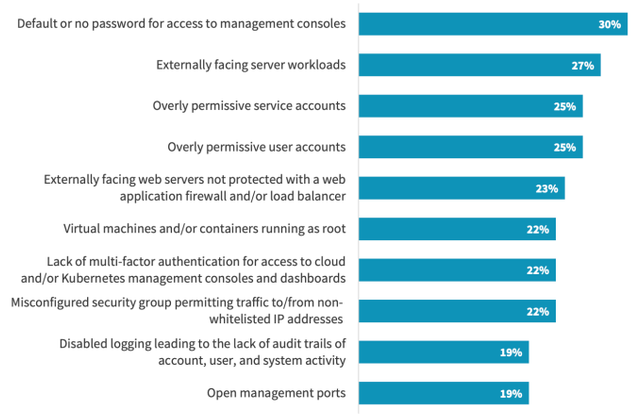
Tesla
Hacker hatten Zugriff auf Kubernetes Konsole (nicht passwortgeschützt)
Resultat: Zugriff auf vertrauliche Daten und Kryptomining über einen Kubernetes Pod

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

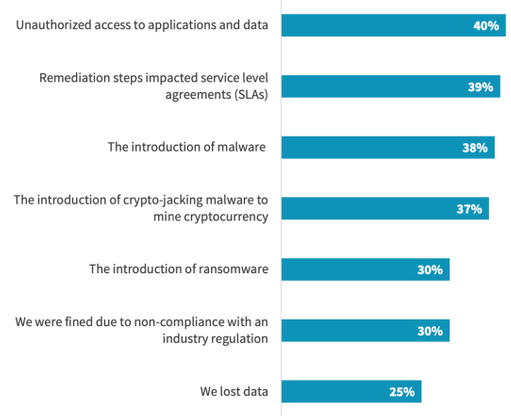
2 Sicherheitsrisiken

2.1 Fehler und Beispiele aus der Praxis

Ten most common cloud misconfigurations in the past 12 months.



Results of cloud misconfigurations.



[2]

CLOUD ALLGEMEIN

Häufigste Fehlkonfigurationen und Folgen

Misskonfigurationen

Default-Passwort oder gar keins

Server Workloads

Zugriffsrechte falsch gesetzt

Folgen

Unauthorisierter Zugriff zu Anwendungen und Daten

Malware

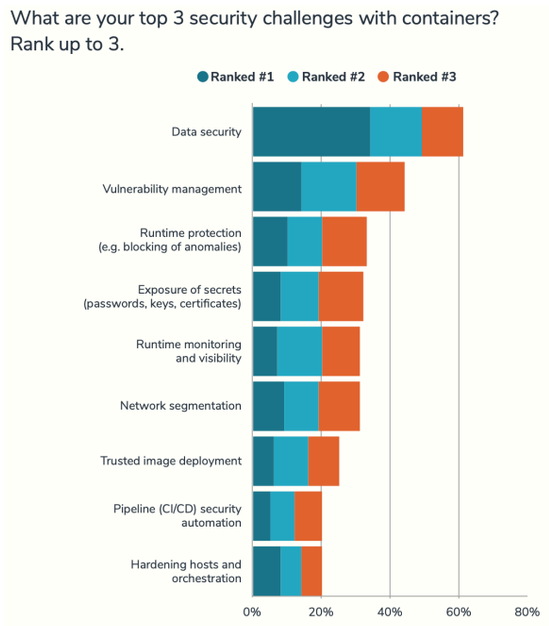
Kryptomining

Datenverlust

[2]

2 Sicherheitsrisiken

2.1 Fehler und Beispiele aus der Praxis



[9]

Top 3 Security Challenges
Datensicherheit
Schwachstellenmanagement
Runtime protection
Vertrauliche Daten geheim halten
Runtime monitoring
Netzwerksegmentierung

[9]

2 Sicherheitsrisiken

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 Sicherheitsrisiken

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 Sicherheitsrisiken

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

next: Brian

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

2 Sicherheitsrisiken

Interview: Brian Fox

- GNU Bash shell (1989)
- Erste interaktive Online-Banking Software (1995)
- Open Source Wahlsystems (2008)

- Chief Architect @ Holaplex
- CEO @ Kano.One
- CEO @ Opus Logica
- …

Brian Fox
- Autor der GNU Bash Shell (1989)
- Erstellte erste interaktive Online-Banking Software (1995)
- Ersteller eines Open Source Wahlsystems (2008)
- **Enkel des Erstellers des Monopoly Maskottchen - Rich Uncle Pennybags**
- Chief Architect @ Holaplex
- CEO @ Kano.One
- CEO @Opus Logica
- und viele weitere Unternehmen…

Probleme verschiedener Unternehmen bei denen er gearbeitet hat
Er darf nicht sagen, um welche es sich dabei handelt.

„An attempted DNS attack caused the pull of the base container image to come from a malicious server. By redirecting the SSL request to a completely different server the attacker was able to deliver a container that already had a back door in it, which would allow the attacker to gain access to the running container“

Brian Fox

„An attempted DNS attack caused the pull of the base container image to come from a malicious server. By redirecting the SSL request to a completely different server the attacker was able to deliver a container that already had a back door in it, which would allow the attacker to gain access to the running container“

DNS Attacke verursachte, dass ein Container Image von einem bösartigem Server geladen wurde, welches bereits eine Backdoor inne hatte, wodurch der Angreifer Kontrolle über den Container bekam.

„When a developer creates a container to run some scripts or a database, they may assume that the entire container is a secure environment, and they utilize default username and password pairs for the applications running in the container. This is a serious risk on a network that may have many different containers running in it, where the security rules can be managed differently for each devops group. If container B has network access to container A, and container A is running a networked database with default superuser access, something malicious in container B can read or change information in the database on container A.“

Brian Fox

„When a developer creates a container to run some scripts or a database, they may assume that the entire container is a secure environment, and they utilize default username and password pairs for the applications running in the container. This is a serious risk on a network that may have many different containers running in it, where the security rules can be managed differently for each devops group. If container B has network access to container A, and container A is running a networked database with default superuser access, something malicious in container B can read or change information in the database on container A.“

Entwickler nehmen möglicherweise an, dass ein Container eine gesicherte Umgebung ist und verwenden Standard-Anmelde-Daten. Das ist ein sehr großes Sicherheitsrisiko, weil es unter Umständen möglich ist, durch Kommunikation zwischen Containern Informationen zu erlangen

DevOps: Zusammenschluss von Development und Operations, Möglichkeit der Zusammenstellung einer Arbeitsgruppe

„Developers do not keep up with security updates, with Docker itself, and with versions of application software. […]“

„In one instance, an older version of Rails was running in a container that was accepting connections from the internet at large. This earlier version had some significant security flaws that allowed for database injection attacks, literally destroying the database. We survived this attack because we had aggressive backup policies, and the rate of change of information in the database was relatively small. “

Brian Fox

„Developers do not keep up with security updates, with Docker itself, and with versions of application software. […]“
„In one instance, an older version of Rails was running in a container that was accepting connections from the internet at large. This earlier version had some significant security flaws that allowed for database injection attacks, literally destroying the database. We survived this attack because we had aggressive backup policies, and the rate of change of information in the database was relatively small. “

Entwickler bleiben mit Sicherheitsupdates nicht up to date.
Einmal lief eine ältere Version von Rails in einem Container, welcher Verbindungen über das Internet akzeptierte. Durch die veraltete Version waren Datenbank Injection Attacks möglich, welche die Datenbank komplett zerstörten. Das Unternehmen hat den Angriff nur überlebt, da es eine strikte Backup Policy verfolgte und der Informationsverlust dadurch minimal war.

(Rails: server side web application framework written in Ruby)

2 Sicherheitsrisiken
2.2 Unterteilung

- Allgemein

- Containerspezifisch

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

3 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 Schutzmöglichkeiten

3.2 Containerspezifisch

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

1. Container Images
2. Image Registries
3. Runtime
4. Orchestrierungsplattformen (wie Kubernetes)
5. Host-OS

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

3 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 um kryptografisch die Unverfälschtheit sicherzustellen

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

3 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 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 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]

pod: Ansammlung aus einem oder mehreren Containern, kleinste Einheit einer Kubernetes-Anwendung

3 Schutzmöglichkeiten

3.2 Containerspezifisch

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

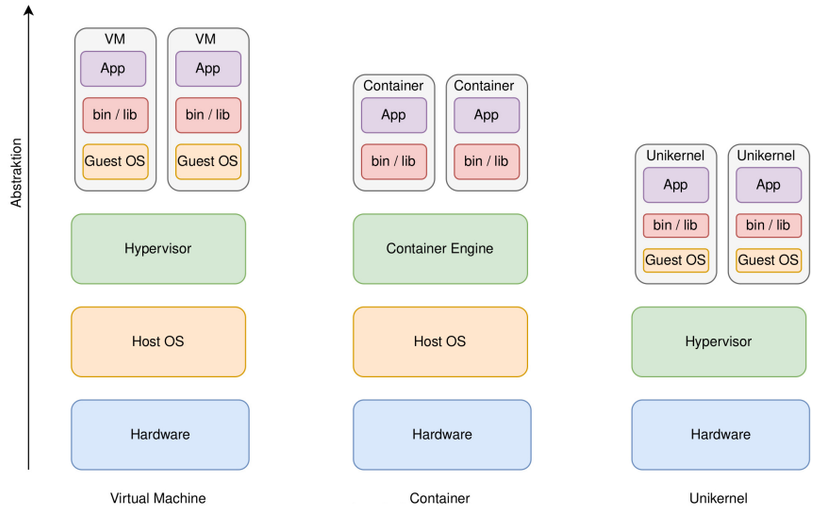
Größtes Risiko und großer Schaden möglich, weil wenn Host-OS kompromittiert = alle Container darauf kompromittiert und möglicherweise hat ein Angreifer Zugriff auf die gesamte Umgebung

- 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

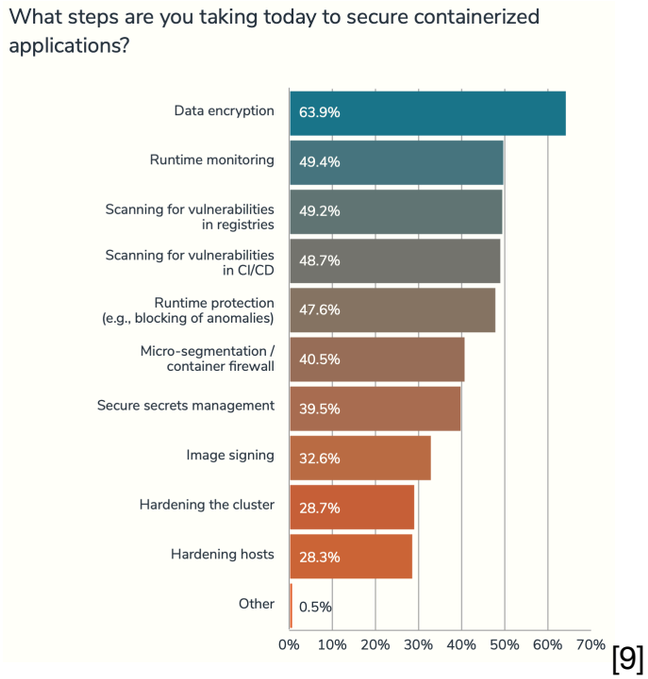


Host-OS - Lösung der Sicherheitsprobleme: Unikernels

- 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

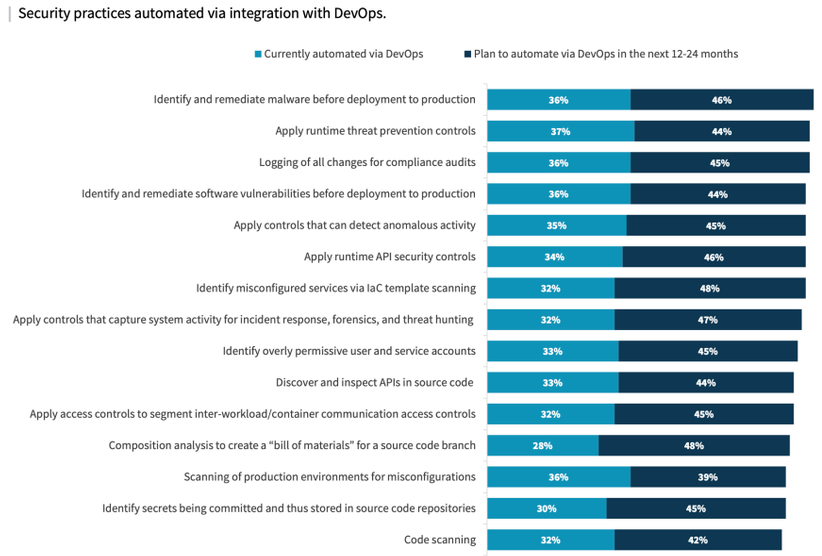


[9]

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
- Aber viel Automatisierung möglich
- Herausforderung: Plattformunabhängigkeit der Sicherheit

->>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
[16]	Docker overview	Docker	10.07.2022, 10:35	Link

https://tinyurl.com/
34e33dbr

HoME
HOCHSCHULE
MERSEBURG
University of
Applied Sciences

www.hs-merseburg.de