IT-Grundschutz für die Container-Virtualisierung mit dem neuen BSI-Baustein SYS. 1.6

Christoph Haar[1], Erik Buchmann[2]

Abstract: Die Container-Virtualisierung baut auf eine komplexe IT-Landschaft auf, in der Hardware, Betriebssystem und Anwendungen von verschiedenen Parteien bereitgestellt und genutzt werden. Der IT-Sicherheit kommt daher eine große Bedeutung zu. Es gibt jedoch wenig Erfahrung mit der Absicherung der Container-Virtualisierung: Das Grundschutz-Kompendium und die Standards zur Risikoanalyse des Bundesamts für Sicherheit in der Informationstechnik (BSI) wurden erst im November 2017 in überarbeiteter Form neu eingeführt, und der BSI-Baustein SYS. 1.6 zur Container-Virtualisierung wurde erst im Mai 2018 als Community Draft veröffentlicht. Ziel dieser Arbeit ist die Erprobung des neuen Baustein SYS. 1.6 an einem konkreten Fallbeispiel. Dazu wenden wir den neuen Baustein auf ein typisches Docker Szenarion „Shop" an und gehen die Gefährdungsanalyse, Docker-spezifische Gefährdungen sowie entsprechende Maßnahmen zur Abwendung dieser Gefährdungen ein. Wir haben festgestellt, dass der Baustein SYS. 1.6 des Grundschutz-Kompendiums eine umfassende Hilfestellung zur Absicherung der Container-Virtualisierung bietet, und in der Praxis gut anwendbar ist. Wir haben jedoch zwei zusätzliche Gefährdungen identifiziert, die der Baustein nicht ausreichend berücksichtigt.

Keywords: IT-Grundschutz; IT-Sicherheit; Container-Virtualisierung; Docker Container

1 Einleitung

Die Container-Virtualisierung ermöglicht es, innovative und cloudbasierte Anwendungen auf eine agile, kosteneffiziente Weise umzusetzen, auszuliefern und zu warten. Ein prominentes Beispiel ist das Open-Source Projekt Docker [Dob]. Container erlauben es, die eigentliche Anwendung von der IT-Infrastruktur zu trennen. So wird es beispielsweise möglich, ein vorkonfiguriertes Betriebssystem-Image zusammen mit einer Anwendung in einen Container zu packen und in einer Testumgebung zu prüfen. Der selbe Container lässt sich dann in das Produktivsystem übertragen und bei gewachsenem Ressourcenbedarf auf einen größeren Host-Rechner umziehen. Dafür benötigt die Container-Virtualisierung eine komplexe IT-Landschaft, in der verschiedene Parteien Softwarekomponenten oder Hardwareressourcen zur Verfügung stellen, Container bereitstellen oder die Virtualisierungsumgebung betreiben [Gö17]. Container können sensible Firmendaten oder personenbezogene Informationen enthalten. Unternehmen müssen daher bei Nutzung der Container-Virtualisierung ihr IT-Sicherheitskonzept überarbeiten.

[1] Hochschule für Telekommunikation Leipzig, haar@hft-leipzig.de
[2] Hochschule für Telekommunikation Leipzig, buchmann@hft-leipzig.de

doi:10.18420/inf2019_65

Wenn das Sicherheitskonzept auf dem IT-Grundschutz [Bu11] des Bundesamts für Sicherheit in der Informationstechnik (BSI) beruhen soll oder im Rahmen einer ISO 27001 Zertifizierung auf dem IT-Grundschutz aufgebaut [Bu14] wird, ist dies schwierig. Der Baustein SYS. 1.5 Virtualisierung des aktuellen BSI Grundschutz-Kompendiums [Bu19] zielt auf eine Hypervisor-Visualisierungsschicht ab. Im Mai 2018 wurde ein Community Draft für einen neuen Baustein SYS. 1.6 „Container“ [Bu] für die Container-Virtualisierung mittels Docker oder alternativer Technologien veröffentlicht. Dieser Baustein hat jedoch noch immer einen vorläufigen Charakter. Es gibt daher keine Erfahrungen, ob die darin beschriebenen Gefährdungen und Maßnahmen in der Praxis ausreichen, um ein gegebenes Anwendungsszenario ausreichend abzusichern. Wir haben eine Absicherung nach dem aktuellen IT-Grundschutz für ein typisches Container-Szenario durchgeführt:

Szenario: *Ein Einzelhändler verwendet einen Web-Shop, um sein Ladengeschäft zu ergänzen. Der Web-Shop verfügt über eine eigene Datenbank mit Produktbeschreibungen und Kundenkonten. Darüber hinaus ist der Web-Shop mit einem Internet-Zahlungssystem ausgestattet, das über einen Dienstleister Bezahlvorgänge über unterschiedliche Kanäle sicher abwickelt. Anwendung, Datenbank und Zahlungssystem sind auf verschiedene Docker-Container aufgeteilt. Die Container werden auf einem eigenen Rechner in einer On-Premise-Umgebung ausgeführt, bei der der Einzelhändler nicht nur für die Container verantwortlich ist, sondern auch die Infrastruktur und die Container-Plattform betreibt.*

In dieser Arbeit geben wir in geraffter Form unsere Erkenntnisse aus der Anwendung des IT-Grundschutzes wieder und zeigen anhand unseres Szenarios, dass der neue Baustein SYS.1.6 in der Praxis gut angewendet werden kann. Eine ausführliche Version ist als Preprint [HB18] verfügbar. Aufbauend auf [Ba17] und [BHB18] haben wir uns auf die Container-Virtualisierung konzentriert. Das heißt, wir haben für die bereits sehr gut untersuchte Absicherung [Dä18; Ec13] der Infrastruktur sowie der organisationsübergreifenden Aspekte ein bestehendes Sicherheitskonzept nach IT-Grundschutz vorausgesetzt. Wir haben nach BSI-Standard 200-2 [Bu17a] den Informationsverbund für unser Docker-System modelliert, dafür eine Schutzbedarfsfeststellung durchgeführt, und die in den BSI-Bausteinen SYS. 1.5 Virtualisierung und SYS. 1.6 Container beschriebenen Elementargefährdungen analysiert. Da einige Daten den Schutzbedarf „hoch“ erfordern, haben wir eine Risikoanalyse nach BSI-Standard 200-3 [Bu17b] zur Identifikation und Behandlung von zusätzlichen Gefährdungen für unser Docker-Szenario durchgeführt. Im Anschluss haben wir analysiert, inwiefern sich die dabei identifizierten Gefährdungen und Maßnahmen von denen des IT-Grundschutz-Kompendiums unterscheiden. Dabei hat sich gezeigt, dass der BSI-Baustein SYS. 1.6 das BSI-Grundschutz-Kompendium für die praktische Absicherung der Container-Virtualisierung gut anwendbar ist. Wir haben jedoch zwei zusätzliche Gefährdungen identifiziert, die vom Baustein nicht ausreichend berücksichtigt werden.

Aufbau der Arbeit: Abschnitt 2 beschreibt die Grundlagen dieser Arbeit. In Abschnitt 3 und 4 führen wir eine Risikoanalyse für Docker nach BSI-Standard durch und vergleichen unsere Erkenntnisse mit denen des BSI. In Abschnitt 5 verallgemeinern wir unsere Erkenntnisse. Die Arbeit schließt mit einer Zusammenfassung in Abschnitt 6.

2 Grundlagen

In diesem Abschnitt stellen wir das Docker-System, die BSI-Bausteine SYS. 1.5 und SYS. 1.6 [Bu] sowie die Vorgehensweisen zur Standardabsicherung und Risikoanalyse nach den aktuellen BSI-Standards [Bu17a; Bu17b] vor.

2.1 Docker-Container

Die Container-Virtualisierung hat sich aus der Hypervisor-Virtualisierung [Ch07] entwickelt. Ein Hypervisor zieht eine Abstraktionsschicht zwischen Host-System und den darauf ablaufenden Gast-Systemen ein. Dies hat unter anderem den Nachteil, dass für jeden Gast ein vollständiges Betriebssystem aufzusetzen ist. Im Gegensatz dazu werden bei der leichtgewichtigen Container-Virtualisierung Container zusammengestellt, die nur die Anwendung und eine leichtgewichtige Ablaufumgebung enthalten. Die Container nutzen also den Kernel des Host-Betriebssystems mit. Dies ermöglicht es, Systemressourcen wie Prozessor, Netzwerk oder Speicher effizient zu nutzen, und Applikationen über Systeme hinweg zu verschieben, ohne dabei komplette Betriebssysteme mit zu migrieren. Auf der anderen Seite wird es jedoch schwerer, mehrere Container, die auf dem selben Host-System ablaufen, zuverlässig voneinander zu isolieren.

Eine sehr häufig eingesetzte Lösung für die Container-Virtualisierung ist das auf einem Linux-Betriebssystem aufsetzende Docker. Linux-typisch besteht die Docker Architektur [Dob] aus Docker Client, Docker Daemon, Docker Registry und den Docker Objekten (Images, Docker Files, Container). Der Docker Client und Docker Daemon bilden zusammen die Docker Engine. Ein Container enthält zwei Hauptverzeichnisse: /bin enthält die Binärdateien und /lib die dynamischen Bibliotheken und Kernel-Module, die für die Funktionalität eines Containers benötigt werden. Client und Daemon können auf dem gleichen Host-System laufen oder der Client wird mit einem Remote Daemon verbunden. Die externe Kommunikation findet über eine REST API, ein UNIX Socket oder eine andere Netzwerkschnittstelle statt. Docker ist in seinen Grundeinstellungen so konfiguriert, dass nach Images aus dem Docker Hub gesucht wird. Es ist auch möglich, eine private Registry für Images anzulegen (Docker Trusted Registry).

Abbildung 1 beschreibt eine typische Docker-Installation, wie wir sie auch für unser Anwendungsszenario zugrunde gelegt haben: Auf dem Linux-Kernel setzt die Docker Engine auf. Für jede sachlogisch voneinander getrennte Aufgabe wird ein eigener Container betrieben. In unserem Fall sind dies drei Container, die voneinander isoliert eine Datenbank, ein Zahlungssystem und eine Web-Anwendung für den Online-Shop bereitstellen. Die Container kommunizieren über Linux-übliche Netzwerkschnittstellen miteinander und mit dem Internet (schwarze Pfeile). Zu diesem Zweck nutzen sie Funktionen der Docker Engine (graue Pfeile). Im Folgenden konzentrieren wir uns auf die Absicherung des in der Abbildung gestrichelt dargestellten Bereichs nach dem Ende 2017 überarbeiteten IT-Grundschutz. Eine Risikoanalyse nach dem alten IT-Grundschutz ist Teil unserer Vorarbeiten [BHB18].

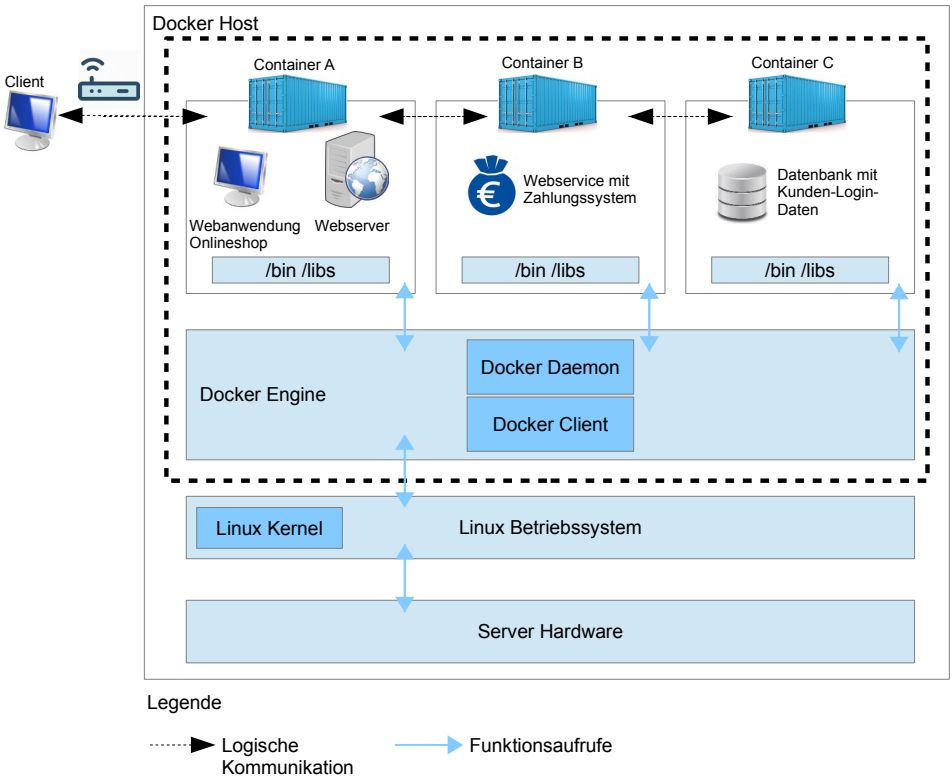

Abb. 1: Beispiel-System mit Dockerarchitektur

2.2 Standard-Absicherung und Risikoanalyse nach BSI

Der BSI-Standard 200-2 beschreibt, in welchen Schritten eine Standard-Absicherung eines Systems durchzuführen ist [Bu17a].

1. Zunächst ist der **Geltungsbereich** („Informationsverbund“) festzulegen, für den das Sicherheitskonzept realisiert werden soll. Der Geltungsbereich für unser Szenario ist in Abb. 1 gestrichelt dargestellt.
2. Bei der **Strukturanalyse** werden die Prozesse, Anwendungen, IT-Systeme, Infrastrukturen, etc. im Geltungsbereich aufgelistet.
3. Mit Hilfe der **Schutzbedarfsfeststellung** wird ein angemessener Schutz für die

Geschäftsprozesse, die darin verarbeiteten Informationen und die verwendete Informationstechnik ermittelt.
4. Bei der **Modellierung** werden Sicherheitsanforderungen und umzusetzende Maßnahmen mit den **Bausteinen** des IT-Grundschutz-Kompendiums [Bu19] identifiziert.
5. Mit dem **IT-Grundschutz-Check** wird geprüft, ob bereits umgesetzte Maßnahmen zur Absicherung des Informationsverbunds ein ausreichendes Schutzniveau bieten.
6. Wenn für ein Zielobjekt ein hoher oder sehr hoher Schutzbedarf besteht, kein passender BSI-Baustein existiert, oder das Zielobjekt auf eine Art und Weise betrieben wird, die der existierende Baustein nicht berücksichtigt, ist eine **Risikoanalyse** durchzuführen.

2.3 SYS. 1.5 Virtualisierung und SYS. 1.6 Container

Im BSI-Grundschutz-Kompendium [Bu19] werden abzusichernde Zielobjekte in Form von Bausteinen beschrieben. Für jedes Zielobjekt wird im Rahmen des Bausteins eine Zielstellung definiert, die das Ergebnis der Absicherung des Zielobjektes beschreibt. Darüber hinaus wird in jedem Baustein festgelegt, welche Bestandteile zur Absicherung des Zielobjektes zum Baustein gehören und welche nicht. Im Weiteren werden in jedem Baustein spezifische Gefährdungen für das Zielobjekt beschrieben. Zur Abwendung dieser Gefährdungen werden in jedem Baustein Anforderungen definiert. Zuletzt werden in jedem Baustein zusätzliche Informationen zu Gefährdungen und Sicherheitsmaßnahmen bereitgestellt.

Zur Absicherung des Docker-Systems benötigen wir zwei Bausteine. Der Virtualisierungs-Baustein SYS. 1.5 des BSI-Kompendiums [Bu19] behandelt die Gefährdungslage für Virtualisierungs-Systeme. Zwar adressiert der Baustein explizit nicht die Container-Virtualisierung. Da diese jedoch auf eine klassische Virtualisierung aufsetzt, haben wir SYS. 1.5 in unsere Analyse mit einbezogen. Der Baustein identifiziert folgende Gefahren:

- Fehlerhafte Planung der Virtualisierung
- Fehlerhafte Konfiguration der Virtualisierung
- Unzureichende Ressourcen für virtuelle IT-Systeme
- Informationsabfluss oder Ressourcen-Engpass durch Snapshots
- Ausfall des Verwaltungsservers für Virtualisierungs-Systeme
- Missbräuchliche Nutzung von Gastwerkzeugen
- Kompromittierung der Virtualisierungssoftware

Der im Mai 2018 als Community Draft veröffentlichte Baustein SYS. 1.6 [Bu] beschreibt folgende Gefährdungen für die Container-Virtualisierung:

- Schwachstellen in Images
- Administrative Zugänge ohne Absicherung
- Tool-basierte Orchestrierung ohne Absicherung
- Datenverluste durch fehlende Persistenz
- Vertraulichkeitsverlust von Zugangsdaten

Beide Bausteine sind hersteller- und produktneutral verfasst. Damit bleibt die Frage offen, ob die in den Bausteinen aufgeführten Elementargefährdungen und spezifischen Gefährdungen auch die für Docker spezifischen Bedrohungen für die IT-Sicherheit mit abdecken, und wie gut es möglich ist, diese Bedrohungen mit den Bausteinen als Handlungsunterstützung zu identifizieren und ihnen die passenden Maßnahmen gegenüberzustellen.

2.4 Identifikation zusätzlicher Gefährdungen

In einer Risikoanalyse sollen zusätzliche Gefährdungen identifiziert, eingeschätzt und um Maßnahmen ergänzt werden, die über die in den Bausteinen aufgeführten Gefährdungen hinausgehen. Zu diesem Zweck gibt der BSI-Standard 200-3 [Bu17b] folgenden Rahmen zur Ermittlung zusätzlicher Gefährdungen vor:

- Welche Gefahren aus dem Bereich „höhere Gewalt“ sind besonders relevant?
- Bestehen organisatorische Mängel, die die Informationssicherheit beeinträchtigen?
- Kann die Sicherheit durch menschliche Fehlhandlungen beeinträchtigt werden?
- Welche speziellen Sicherheitsprobleme kann technisches Versagen hervorrufen?
- Welche Gefährdungen können durch externe Angriffe entstehen?
- Ist es Mitarbeitern möglich, den Betrieb des Zielobjekts mutwillig zu beeinträchtigen?
- Können von Objekten außerhalb des Informationsverbunds Gefahren ausgehen?
- Welche Hinweise geben Herstellerdokumentationen sowie Informationen von Dritten?

Diese Fragen sollen von Experten, Mitarbeitern, Administratoren und Benutzern gemeinsam bearbeitet werden.

3 Schutzbedarfsfeststellung und Elementargefährdungen für Docker

In diesem Abschnitt entwickeln wir nach BSI-Standard 200-2 Abschnitt 8 [Bu17a] eine Standard-Absicherung für unser Anwendungsszenario. Wir beginnen mit der Modellierung des Informationsverbunds und einer Schutzbedarfsfeststellung. In einem nächsten Schritt untersuchen wir die in den Bausteinen SYS. 1.5 und SYS. 1.6 vorgegebenen Elementargefährdungen auf ihre Anwendbarkeit für Docker.

3.1 Das Docker-System

Unser in Abbidlung 1 dargestelltes Docker-Szenario besteht aus einem Online-Shop, dessen Komponenten auf drei Container aufgeteilt sind. In Container A läuft eine Webanwendung auf einem Webserver, die ein Shop-System incl. Einkaufskorb, Kundenrezensionen etc. umsetzt. In Container B wird ein Webservice betrieben, der die Zahlungsabwicklung für unseren Onlineshop realisiert. Container C enthält eine Datenbank, die Produkt-, Kunden-

und Bestelldaten beinhaltet. Diese drei Container werden isoliert voneinander betrieben und bilden zusammen mit der Docker Engine das Host-System. Der Informationsverbund ist nachfolgend zusammengefasst:

Nr.	Datenobjekt	Beschreibung
D1	Personendaten	Einzelangaben zu einer natürlichen Person
D2	Nutzdaten	Fachdaten der Anwendungen und Services
D3	Accountdaten	Anmelde- und Berechtigungsdaten der Anwender
D4	Konfigurationsdaten	Daten zur Änderung, Einstellung und Anpassung
D5	Protokolldaten	Statusinformationen und Funktionen

Nr.	Beschreibung	verarbeitete Daten	Software
A1	Webanwendung	D1, D2, D3, D4, D5	Allgemeine Anwendung z.B. PHP
A2	Webserver	D1, D2, D4, D5	Apache Webserver
A3	Webservice	D2, D3, D4, D5	REST-basierter Dienst
A4	Datenbank	D1, D2, D3, D4, D5	Allgemeine Datenbank z.B. MySQL

Nr.	Beschreibung	verarbeitete Daten	IT-System
SSW1	Docker Software	D4, D5	S1 und S1

Nr.	Beschreibung	verarbeitete Daten	Plattform	Ort
S1	Host-System	D1, D2, D3, D4, D5	x86 Linux-Server	RZ 1

Dieser Informationsverbund ist typisch für viele Docker-Installationen. Da wir uns auf die Absicherung von Docker konzentrieren wollen, haben wir unter S1 „Host-System“ die gesamte Host-Umgebung zusammengefasst, d.h., das Rechenzentrum mit dem Host-Rechner und dem darauf installierten Host-Betriebssystem.

3.2 Schutzbedarfsfeststellung

Um herauszufinden, welche Maßnahmen für den Schutz der Objekte in unserm Informationsverbund angemessen sind, haben wir eine Schutzbedarfsfeststellung durchgeführt. Wir verwenden die im Standard 200-2 [Bu17a] definierten Schutzbedarfskategorien „normal“, „hoch“ und „sehr hoch“.

Bei der Schutzbedarfsfeststellung vererben sich die Schutzbedarfe einzelner Datenobjekte (D1-D4) auf die Anwendungen (A1-A4), die diese Daten verarbeiten, und von dort auf die Systeme (S1), auf denen diese Anwendungen ablaufen. Speichert ein System Daten mit unterschiedlichen Schutzbedarfen, so wird dem System der höchste dieser Schutzbedarfe zugewiesen.

Daraus ergibt sich eine Besonderheit für die Container-Virtualisierung: Sämtliche Container laufen möglicherweise auf der gleichen physischen Maschine (S1). Funktionen des Betriebssystem-Kernels des Hosts werden von allen Containern gleichermaßen verwendet. Zudem funktioniert das Gesamtsystem – in unserem Falle der Online-Shop – nur, wenn sämtliche Container betriebsbereit sind. Deswegen vererbt sich der höchste Schutzbedarf jedes einzelnen Containers automatisch auf den gesamten Informationsverbund. Für die Schutzbedarfsfeststellung genügt es deswegen, über alle Container hinweg nach den Daten oder Diensten mit dem höchsten Schutzbedarf bezüglich Vertraulichkeit, Integrität und Verfügbarkeit zu suchen und diesen dann für das Gesamtsystem zu übernehmen. Für unser Anwendungsszenario bedeutet dies:

- **Vertraulichkeit:** In Container C wird eine Datenbank betrieben, die Kundendaten mit Personenbezug speichert. Daher besteht für den Informationsverbund ein hoher Schutzbedarf für den Grundwert Vertraulichkeit.
- **Integrität:** In Container B werden die Zahlungsvorgänge der Kunden abgewickelt. Der Schutzbedarf des Informationsverbunds bezüglich der Integrität ist deshalb hoch.
- **Verfügbarkeit:** Der Web-Shop ist geschäftskritisch, funktioniert aber nur, wenn alle drei Container sowie das Betriebssystem und die Hardware verfügbar sind. Deswegen besteht für den gesamten Informationsverbund ein hoher Schutzbedarf für die Verfügbarkeit.

Unsere zentrale Erkenntnis aus der Schutzbedarfsfeststellung ist, dass die Schutzbedarfe für Vertraulichkeit, Integrität und Verfügbarkeit für typische Einsatzszenarien mindestens „hoch“ sind. Dies gilt beispielsweise für alle von Docker aufgeführten Kundenprojekte [Doa]. Bezogen auf den BSI-Grundschutz bedeutet dies, dass in jedem Fall nach der Standard-Absicherung eine Risikoanalyse durchzuführen ist (s. Abschnitt 4).

3.3 Analyse der Elementargefährdungen

Nach der Schutzbedarfsfeststellung sieht das BSI die Modellierung eines Grundschutzkonzepts auf der Basis der im Grundschutz-Kompendium definierten Bausteine vor. Der erste Schritt besteht dabei in der Prüfung der in den Bausteinen genannten Elementargefährdungen. Wir haben bereits festgestellt, dass für das Docker-System die Bausteine SYS. 1.5 und SYS. 1.6 relevant sind. Zusammen listen die beiden Bausteine 25 Elementargefährdungen auf. Für eine detaillierte Auseinandersetzung mit Elementargefährdungen wie Datenverlust oder Ressourcenmangel im Docker-Informationsverbund verweisen wir auf den Preprint [HB18] dieser Arbeit. Im nächsten Abschnitt legen wir unseren Fokus auf den Umgang mit Docker-spezifischen Bedrohungen, die über Elementargefährdungen hinausgehen.

4 Docker-spezifische Gefährdungen

Da in unserem Informationsverbund mehrere Objekte einen über „normal“ hinausgehenden Schutzbedarf ausweisen, ist eine Riskioanalyse zur Identifikation und bewertung zusätzlicher Gefährdungen erforderlich, gefolgt von einer Analyse der Risikobehandlungsoptionen.

4.1 Identifikation zusätzlicher Gefährdungen

Gemeinsam mit Experten der Open Telekom Cloud haben wir eine Risikoanalyse durchgeführt (vgl. Abs. 2.2). Dabei haben wir 14 Gefährdungen identifiziert (s. Abbildung 2). 12 dieser Gefährdungen sind auch in den Bausteinen SYS. 1.5 und SYS. 1.6 als spezifische Gefährdungen enthalten. Darüber hinaus konnten wir zwei zusätzliche Gefährdungen identifizieren (gestrichelt in Abb. 2 dargestellt).

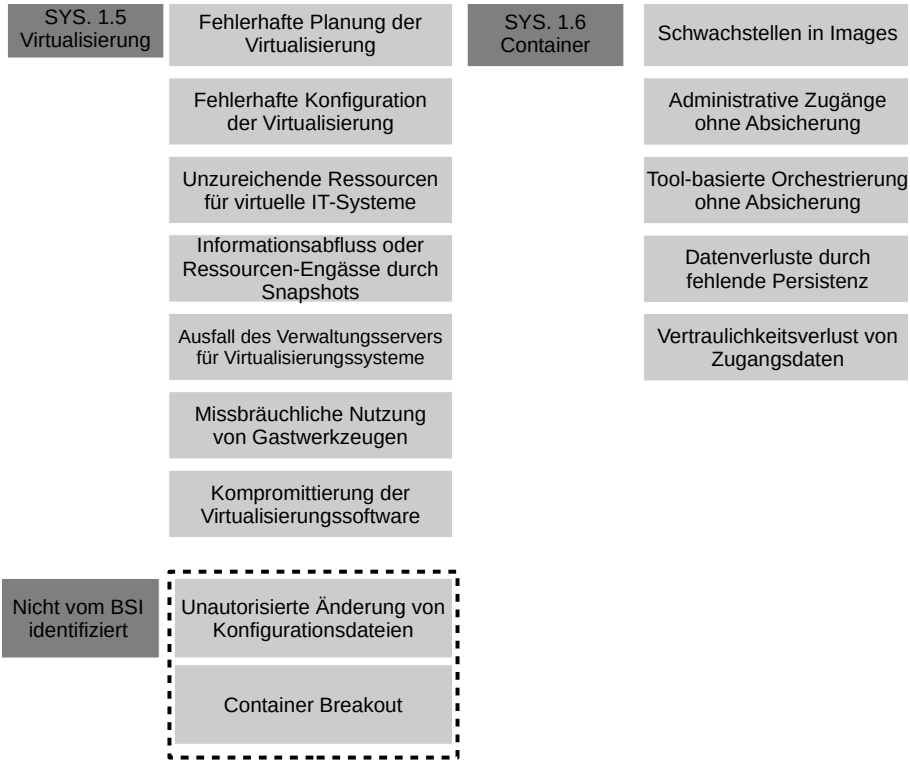

Abb. 2: Spezifische Gefährdungen für das Docker-System

Ein **Container Breakout** [Ro] wird möglich, wenn die Container durch Schwachstellen in der Implementierung nicht lückenlos voneinander isoliert sind. Bei einem erfolgreichen Breakout kann der Angreifer mit den Privilegien des Containers, aus dem er ausgebrochen ist,

auf Daten oder Dienste des Host-Systems oder anderer Container zugreifen. Ein Container Breakout beeinträchtigt nicht nur die Vertraulichkeit, sondern auch die Verfügbarkeit und die Integrität von anderen Objekten im Informationsverbund. Der Baustein SYS. 1.6 adressiert diesen Punkt nur indirekt durch die Basis-Anforderung SYS. 1.6. A2 zur Planung eines Netzzonenkonzepts.

Durch **unautorisierte Änderungen an Konfigurationsdateien** der virtuellen Infrastruktur können erhebliche und tiefgreifende Schäden entstehen, ebenso wie durch vorsätzliche oder versehentliche Fehlkonfigurationen der Netzzuordnung. Hier stellt insbesondere der Docker Daemon eine Angriffsoberfläche dar, da dieser root-Privilegien besitzt und die Funktionsfähigkeit aller Container beeinflussen kann. Für Vertraulichkeit, Integrität oder Verfügbarkeit der Objekte im Informationsverbund ist die Integrität von Konfigurationsdaten daher ausschlaggebend. Auch diese Gefährdung wird nur mittelbar durch die Standard-Anforderungen A17 in SYS. 1.5 (Netzzuordnungen im Virtualisierungslayer) und A12 in SYS. 1.6 (Freigabe von Images) adressiert. Details zu allen von uns identifizierten Gefährdungen finden sich in [HB18].

4.2 Risikoeinstufung und Risikobewertung

Im nächsten Schritt muss das Risiko ermittelt werden, welches von der jeweiligen Gefährdung ausgeht. Dazu wird nach BSI-Standard 200-3 [Bu17b] eine qualitative Risikobewertung herangezogen. Unsere zentrale Erkenntnis ist hier, dass sich alle spezifischen Gefährdungen für das Docker-System auf technische Schwachstellen beziehen, für die sich Angriffe automatisieren lassen. Wird beispielsweise ein Exploit bekannt, durch den sich ein Container Breakout durchführen lässt, so kann dieser Exploit auch automatisiert auf eine große Zahl von anfälligen Containern angewendet werden. Wir gehen aus diesem Grund davon aus, dass die Eintrittshäufigkeit für jede Gefährdung für Docker „sehr häufig“ ist. Für die Risikobewertung nach BSI-Standard 200-3 genügt es also, die Schadenshöhen der Gefährdungen zu ermitteln.

Im Folgenden stellen wir die Risikobewertung für die Gefährdung „Container Breakout“ beispielhaft dar. Für die „Unautorisierte Änderung von Konfigurationsdateien“ gelten die gleichen Eintrittshäufigkeiten, Auswirkungen und Risiken wie für den Container Breakout. Es kommen noch die Beeinträchtigung der Integrität und Verfügbarkeit hinzu.

Docker-System	Vertraulichkeit: hoch Integrität: hoch Verfügbarkeit: hoch	
Gefährdung **Container Breakout**	Beeinträchtigte Grundwerte: Vertraulichkeit	
Eintrittshäufigkeit ohne zus. Maßnahme: sehr häufig	**Auswirkungen** ohne zusätzliche Maßnahmen: beträchtlich	**Risiko** ohne zusätzliche Maßnahme: hoch
Beschreibung: Ein Gefährdungsszenario ist der Container Breakout, der einem Angreifer Zugriff auf das Host-System oder auf weitere Container im gleichen System erlaubt, und zwar mit den Privilegien des Containers, aus dem der Ausbruch erfolgte.		
Bewertung: Ein Container Breakout würde zum Verlust der Vertraulichkeit von z.B. Kundendaten führen. Diese gelten als hoch schutzbedürftig, sodass das Schadensausmaß bei einem Container Breakout beträchtlich und das Risiko als hoch einzustufen ist.		

4.3 Risikobehandlung

Grundsätzlich stehen als Risikobehandlungsoptionen die Risikoreduktion durch zusätzliche Maßnahmen oder durch Umstrukturierung der Prozesse, der Risikotransfer oder die Risikoakzeptanz zur Verfügung [Bu17b]. Aufgrund der Risikoeinstufung „hoch“ scheidet die Risikoakzeptanz aus. Risikotransfer oder Umstrukturierung liegen außerhalb unseres Geltungsbereichs. Im Folgenden gehen wir auf die Risikoreduktion durch zusätzliche Maßnahmen für die von uns identifizierten zusätzlichen Gefährdungen ein. Für die Gefährdungen, die auch das BSI identifiziert hat, verweisen wir auf die Bausteine SYS.1.5 und SYS.1.6.

Container Breakout (Risikokategorie: hoch)

Definition der Systembenutzer Docker-Container sind nicht als privilegierte Container zu betreiben, damit Angreifer im Erfolgsfall nur unprivilegierten Zugriff auf andere Ressourcen erhalten.

Rechtemanagement Es sind die Berechtigungen für alle definierten Benutzergruppen auf Minimalität zu prüfen.

Rollenaufteilung Es ist auch für virtuelle IT-Systeme eine Aufteilung in verschiedene Rollen notwendig. Linux-Schutzmaßnahmen wie apparmor, selinux, seccomp, Filter und namespaces auf dem Host-System können das Risiko eines Ausbruchs reduzieren.

Unautorisierte Änderung von Konfigurationsdateien (Risikokategorie: hoch)

Prüfsummen Die Prüfung auf unautorisierte Änderungen der Konfigurationsdateien kann beispielsweise mittels Werkzeugen wie OS-SEC erfolgen [OS].

Docker Bench for Security Docker ab Version 1.10.0 bietet das Docker Bench for Security Script [Ce] an, welches die eigene Docker Konfiguration prüft.

Konfiguration der Netzfunktionen Bekannte Linux-Werkzeuge wie beispielsweise Puppet [AJ17] können die Netzkomponenten zentral überwachen.

Benennung virtueller Netze Eine aussagekräftige Benennung der Netze anhand ihrer Funktion vermeidet ein versehentliches Verbinden mit dem falschen Netzwerk [AJ17].

Speicher-Zentralisierung Wenn ein Dateiverzeichnis des Containers mit dem Host- System verknüpft wird, muss dieses die Isolation von Betriebssystem, Systembibliotheken [Va17] und gemeinsamen Anwendungen sicherstellen.

Monitoring Das Monitoring lässt sich durch den Einsatz eines Linux-Servers mit den systemeigenen Monitoring Systemen wie Nagios bewerkstelligen [AJ17].

Kommunikation zwischen Containern Bei aktivem Container Linking [Ja] müssen Container, die nicht miteinander kommunizieren dürfen, auf separaten Hosts ablaufen.

In einem letzten Schritt müssen nun die von uns vorgeschlagenen Maßnahmen mit den Anforderungen der bestehenden BSI-Bausteine konsolidiert werden. Beispielsweise findet sich die von uns vorgeschlagene Definition der Systembenutzer als Maßnahme gegen den Container Breakout im Baustein SYS. 1.6 in der Anforderung A17 wieder. Andere Maßnahmen wie die Berücksichtigung der Kommunikation zwischen Containern finden sich in den BSI-Bausteinen nicht wieder. Aus Platzgründen verweisen wir für eine vollständige Übersicht auf unser Preprint [HB18].

5 Diskussion

In diesem Abschnitt diskutieren wir, inwiefern sich unsere Erkenntnisse auf die Container-Virtualisierung insgesamt sowie auf andere Anwendungsszenarien verallgemeinern lassen.

Container-Virtualisierung Der BSI-Baustein SYS. 1.6 ist bereits von der eingesetzten Technologie unabhängig definiert. Die von uns identifizierten zusätzlichen Gefährdungen sind jedoch ebenfalls nicht Docker-spezifisch. Im Gegensatz zur traditionellen Hypervisor-Virtualisierung [Ch07] nutzen die leichtgewichtigen Container Funktionen aus dem Kernel Host-Betriebssystem [Dob], beispielsweise um zu kommunizieren oder um Ressourcen zu allokieren. Diese Funktionen öffnen potentielle Zugriffspfade für einen Angreifer, um aus der isolierten Container-Umgebung auszubrechen. Auch unautorisierte Änderungen der Konfigurationsdateien stellen für Container eine Gefährdung dar. Jeder Container wird entsprechend seiner benötigten Berechtigungen konfiguriert. Unautorisierte Änderungen an den Konfigurationsdateien können daher einen erheblichen Einfluss auf die Integrität, Verfügbarkeit und Vertraulichkeit sowohl der Container als auch des Host-Systems haben. Es würde sich daher anbieten, diese beiden Gefährdungen explizit in den neuen Container-Bausteins SYS. 1.6 aufzunehmen.

Allgemeine Anwendungsszenarien Unsere Risikoanalyse hat ergeben, dass sich die von uns identifizierten zusätzlichen Gefährdungen für automatisierbare Angriffe eignen, sobald eine entsprechende Schwachstelle für eine Container-Technologie entdeckt wird.

Daher sind unsere Erkenntnisse über unser Anwendungsszenario und dessen konkrete Schutzbedarfe hinaus wichtig. Wir haben unsere Risikoanalyse auf der Basis einer Schutzbedarfsfeststellung durchgeführt, bei der die Bedarfe für Vertraulichkeit, Integrität und Verfügbarkeit für das Gesamtsystem mit „hoch“ festgesetzt wurde. Wir haben festgestellt, dass dies aufgrund des Maximumprinzips typisch ist für viele kommerzielle Anwendungen der Container-Virtualisierung. Für Anwendungsfälle, bei denen die Schadensauswirkungen ein „existenziell bedrohliches, katastrophales Ausmaß erreichen“ [Bu17a] können, ist jedoch eine umfassendere Risikoanalyse erforderlich. Ein Beispiel für so ein Anwendungsszenario könnte ein Krankenhaus sein, das medizinische Geräte über eine Container-Lösung steuert.

6 Zusammenfassung

Das Ziel dieser Arbeit bestand darin, zu untersuchen, wie gut die aktuellen BSI-Standards und der neue Baustein SYS. 1.6 auf einen typischen Anwendungsfall der Container-Virtualisierung angewendet werden können. Dazu haben wir eine Standard-Absicherung und eine Risikoanalyse nach BSI IT-Grundschutz für Docker Container in einer On-Premise-Umgebung durchgeführt. Wir haben festgestellt, dass der neue Baustein SYS. 1.6 in Verbindung mit dem Virtualisierungs-Baustein SYS. 1.5 ein wertvolles Werkzeug bei der Erstellung eines Sicherheitskonzepts für Docker darstellt. In unserem konkreten Anwendungsfall hat sich jedoch gezeigt, dass zwei zusätzliche Gefährdungen für Docker existieren, die im Rahmen des neuen Bausteins SYS. 1.6 noch nicht berücksichtigt wurden. Wir haben gezeigt, dass sich unsere gewonnenen Erkenntnisse nicht nur auf Docker-Szenarien beschränken sondern im Allgemeinen für Container-Technologien gelten. Daher ist eine Ergänzung des Baustein SYS. 1.6 um unsere zusätzlichen Gefährdungen sowie der dazugehörigen Maßnahmen und Anforderungen zu überlegen.

Literatur

[AJ17] Atug, M.; Jedecke, D.: iX Kompakt - Container und Virtualisierung. Heise Medien, 2017.

[Ba17] Bauer, S.: Erarbeitung eines Informationssicherheitskonzepts nach IT-Grundschutz für Docker Container. Bachelor-Arbeit, Hochschule für Telekommunikation Leipzig, Kopie s. http://www.webcitation.org/6xAkE4g1l/, 2017.

[BHB18] Buchmann, E.; Hartmann, A.; Bauer, S.: Informationssicherheitskonzept nach IT-Grundschutz für Containervirtualisierung in der Cloud. SICHERHEIT 2018/, 2018.

[Bu] Bundesamt für Sicherheit in der Informationstechnik: SYS.1.6 Container, https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/IT-Grundschutz-Modernisierung/BS_Container.html, abgerufen Sept. 2018.

[Bu11] Bundesamt für Sicherheit in der Informationstechnik: Webkurs IT-Grundschutz, IT -Grundschutz im Selbststudium. https://www.bsi.bund.de/, 2011.

[Bu14] Bundesamt für Sicherheit in der Informationstechnik: Zertifizierung nach ISO 27001 auf der Basis von IT-Grundschutz. https://www.bsi.bund.de/, 2014.

[Bu17a] Bundesamt für Sicherheit in der Informationstechnik: BSI-Standard 200-2, IT-Grundschutz-Methodik. https://www.bsi.bund.de/, 2017.

[Bu17b] Bundesamt für Sicherheit in der Informationstechnik: BSI-Standard 200-3, Risikomanagement. https://www.bsi.bund.de/, 2017.

[Bu19] Bundesamt für Sicherheit in der Informationstechnik: IT-Grundschutz-Kompendium - Edition 2019. https://www.bsi.bund.de/, 2019.

[Ce] Center for Internet Security: Docker Community Edition Benchmark, https://www.cisecurity.org, abgerufen Sept. 2018.

[Ch07] Chisnall, D.: The Definitive Guide to the Xen Hypervisor. Prentice Hall, 2007.

[Dä18] Dännart, S.; Diefenbach, T.; Hofmeier, M.; Rieb, A.; Lechner, U.: IT-Sicherheit in Kritischen Infrastrukturen–eine Fallstudien-basierte Analyse von Praxisbeispielen. Multi-Konferenz Wirtschaftsinformatik (MKWI'18)/, 2018.

[Doa] Docker Inc.: Docker Customers, https://www.docker.com/customers, abgerufen Sept. 2018.

[Dob] Docker Inc.: Docker Overview, https://docs.docker.com/engine/docker-overview, abgerufen Sept. 2018.

[Ec13] Eckert, C.: IT-Sicherheit: Konzepte-Verfahren-Protokolle. Walter de Gruyter, 2013.

[Gö17] Göbel, L.: Container-as-a-Service - Die Zukunft der Virtualisierung, https://www.cloudcomputing-insider.de/container-as-a-service-die-zukunft-der-virtualisierung-a-576244, abgerufen Sept. 2018, 2017.

[HB18] Haar, C.; Buchmann, E.: IT-Grundschutz für die Container-Virtualisierung mit dem neuen BSI-Baustein SYS. 1.6. In. Quality Content Of Saxony, 2018.

[Ja] Jacqueline von Ogden: The Top 5 Security Risks in Docker Container Deployment, https://www.cimcor.com/blog/the-top-5-security-risks-in-docker-container-deployment, abgerufen Sept. 2018.

[OS] OSSEC Project Team: OSSEC's Documentation, https://ossec-docs.readthedocs.io/en/latest, abgerufen Sept. 2018.

[Ro] Rob Shapland: Eine Schwachstelle in Container-Techniken erlaubt Angriffe auf den Host, https://www.searchsecurity.de/tipp/Eine-Schwachstelle-in-Container-Techniken-erlaubt-Angriffe-auf-den-Host, abgerufen Sept. 2018.

[Va17] Vasily Tarasov, L. R.: In Search of the Ideal Storage Configuration for Docker Containers. IEEE 2nd International Workshops on Foundations and Applications of Self* Systems (FAS*W)/, 2017.