Technische Hochschule
Brandenburg
University of
Applied Sciences
Fachbereich
Informatik und Medien

IT-Grundschutz für Container-Plattformen

Bachelorarbeit

Ingo Struck
Technische Hochschule Brandenburg
Online-Studiengang Medieninformatik

Technische Hochschule Brandenburg
University of Applied Sciences
Fachbereich Informatik und Medien

IT-Grundschutz für Container-Plattformen

BSI konforme SaaS-Plattform für Container-Nutzlasten

Bachelorarbeit

zur Erlangung des akademischen Grades
Bachelor of Science (B.Sc.)

Ingo Struck

Technische Hochschule Brandenburg
Fachbereich Informatik und Medien
Online-Studiengang Medieninformatik

Prof. Dr.-Ing. habil. Michael Syrjakow (Erster Gutachter)
Prof. Dr.-Ing. Thomas Preuß (Zweiter Gutachter)

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit zum Thema
BSI konforme SaaS-Plattform für Container-Nutzlasten
vollkommen selbstständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Die Arbeit wurde in dieser oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt.

Berlin, den 6. April 2021

(Ingo Struck)

Danksagung

Bei der Entstehung dieser Arbeit habe ich vielfältige Unterstützung erfahren, für die ich mich an dieser Stelle herzlich bedanken möchte.

Zuerst danke ich meinen Betreuern und Gutachtern, Prof. Dr.-Ing. habil. Michael Syrjakow und Prof. Dr.-Ing. Thomas Preuß, die mir jede Frage umgehend beantwortet und wertvolle Hinweise zur Schärfung des Gegenstands der Arbeit gegeben haben.

Meinem Kollegen Roland Osterode danke ich dafür, dass er meine Aufmerksamkeit auf das Thema Informationssicherheit gelenkt und mir eine gute Einführung in die Denkweise des BSI IT-Grundschutzes gegeben hat. Er ist mir zum Freund und Mentor geworden, dessen kritische und immer konstruktive Anmerkungen in vielen Gesprächen maßgeblich zum Gelingen dieser Arbeit beigetragen haben.

Den Vorständen Jochen Pielage und Martin Bockelmann danke ich dafür, dass ich während der Erstellung dieser Arbeit meine reguläre Arbeitszeit auf ein Minimum reduzieren konnte.

Den Kollegen in meinem Team danke ich, dass sie während dieser Zeit Teile meiner Aufgaben zusätzlich übernommen und erfolgreich selbstständig koordiniert haben.

Dem erfahrenen Psychologen und Coach Roland Kopp-Wichmann danke ich dafür, dass er mir vor Augen geführt hat, welchen persönlichen Wert die Anfertigung dieser Arbeit für mich hat. Dem ebenso weisen wie einfühlsamen Psychologen und Coach Detlef Scheer danke ich für die Erkenntnis, dass diese Arbeit eher ein Anfang als ein Abschluss ist und es sich immer lohnt, einen frei gewählten Schritt in eine neue Richtung zu gehen.

Meiner Lebenspartnerin danke ich für ein sorgfältiges Lektorat und die langjährige Ermutigung, meinem immerwährenden wissenschaftlichen Interesse auch durch einen Abschluss sichtbar Ausdruck zu verleihen.

Nicht zuletzt danke ich meinen Eltern, die meinen Drang zu lebenslangem Lernen gefördert und die frühe, uneingeschränkte Erkundung der Perspektiven im universitären Umfeld finanziell unterstützt haben.

Kurzfassung

Im Rahmen dieser Arbeit wird die Architektur einer Cloud-Infrastruktur als Betriebsplattform für Software-as-a-Service-Angebote entworfen, die auf Basis von Software-Containern funktioniert. Sie soll nach den Anforderungen des Bundesamts für Sicherheit in der Informationstechnik abgesichert werden.

Angelehnt an eine der drei Vorgehensweisen des BSI-Standards 200–2 werden die wesentlichen Komponenten dieser Plattform als Informationsverbund definiert. Zudem werden für diese Teile Auswahlkriterien aus den Bausteinen des IT-Grundschutz-Kompendiums und den allgemeinen Empfehlungen der Fachliteratur nach *Stand der Technik* abgeleitet.

Optionen für die einzelnen Komponenten und Strukturen werden vorgestellt. Anhand der vorher definierten Kriterien wird eine Auswahl getroffen und ein Entwurf konstruiert.

Abschließend wird dieser Entwurf hinsichtlich des erzielbaren Schutzniveaus bewertet und als Ausblick die Möglichkeit betrachtet, eine Umsetzung des Entwurfs nach ISO 27001 basierend auf dem IT-Grundschutz zertifizieren zu lassen.

Schlüsselwörter

Container, Software-as-a-Service, SaaS, IT-Grundschutz, IT-Grundschutz-Kompendium, BSI, Bundesamt für Sicherheit in der Informationstechnik, ISO 27001

Abstract

Within the scope of this work, the architecture of an operating platform for software-as-a-service offerings is designed, which functions based on software containers and is secured according to the requirements of the German Federal Office for Information Security.

Based on one of the three approaches of BSI Standard 200–2, the essential components of this platform are defined as an information domain and selection criteria for these parts are derived from the building blocks of the IT-Grundschutz Compendium and the general recommendations of the specialist literature according to the *state of the art*.

Options for the individual components and structures are presented. Based on the previously defined criteria, a selection is made and a design proposal is constructed.

Finally, this draft is evaluated with regard to the achievable level of protection and, as an outlook, the possibility of having an implementation of the draft certified in accordance with ISO 27001 based on IT-Grundschutz is considered.

Keywords

Containers, Software as a Service, SaaS, IT-Grundschutz, IT-Grundschutz Compendium, BSI, German Federal Office for Information Security, ISO 27001

Aufbau der Arbeit

Zunächst wird die Problemstellung dieser Arbeit vermittelt (Abschnitt 1.1). Nach einer Begriffsklärung (Abschnitt 1.2) wird der im Rahmen dieser Arbeit behandelte Teil einer Container-Plattform zur Bereitstellung von Software-as-a-Service-Angeboten grob geklärt und eingegrenzt (Abschnitt 1.3)

Es folgt eine kurze Einführung in die Struktur von Container-Plattformen (Abschnitt 2.1), die Prinzipien der Entwicklung sicherer IT-Systeme (Abschnitt 2.2) und die Elemente des IT-Grundschutz-Kompendiums (Abschnitt 2.3).

Der *Stand der Technik* der Absicherung von Container-Plattformen wird geschildert und die weitere Vorgehensweise erläutert (Abschnitt 3.1). Relevante Bausteine des IT-Grundschutz-Kompendiums werden identifiziert (Abschnitt 3.2). Anhand der System-Bausteine werden nachfolgend Kriterien für den Entwurf geeigneter Strukturen (Abschnitt 3.3) und die Auswahl der Komponenten (Abschnitt 3.4) abgeleitet.

Aus den verfügbaren Optionen und angelehnt an die Referenz-Architekturen des IT-Grundschutz-Kompendiums wird anhand der vorher definierten Kriterien ein Lösungsvorschlag erarbeitet (Abschnitt 4.1 und Abschnitt 4.2). Die für den Betrieb der Plattform wesentlichen Prozesse werden aus den relevanten Prozessbausteinen abgeleitet und skizziert (Abschnitt 4.3).

Abschließend folgt eine Bewertung des Entwurfs (Abschnitt 5.1) und eine Einstufung in ein Schutzniveau gemäß IT-Grundschutz-Kompendium (Abschnitt 5.2). Schlussfolgerungen aus den Betrachtungen werden gezogen (Abschnitt 6.1) und als Ausblick wird die ISO 27001-Zertifizierung des Systems auf Basis des IT-Grundschutzes betrachtet (Abschnitt 6.2).

Nach Verzeichnissen der verwendeten Literatur, Abkürzungen, Abbildungen und Tabellen sowie einem Glossar (Anhang A) folgt Material zum IT-Grundschutz-Kompendium, insbesondere die zweite Fassung des Entwurfs des Bausteins SYS.1.6 *Container* (Anhang B).

Verweise auf Abschnitte, Tabellen, Abbildungen und Definitionen sind durchgängig mit einer rot eingefärbten Referenz markiert. In der PDF-Version sind sie aktive Sprungmarken. Abkürzungen und Einträge im Glossar sind gleichermaßen markiert und verknüpft. Um den Lesefluss bei wiederholtem Gebrauch dieser Verweise weniger zu stören, beschränkt sich dies bei beiden auf das jeweils erste Vorkommen pro Abschnitt. Abkürzungen werden zusätzlich beim *ersten Gebrauch im Dokument* innerhalb einer Fußnote erläutert. Links zu Webseiten sind in einer anderen Schriftart gesetzt und ebenfalls farbig markiert. Sie sind in der PDF-Version direkt aufrufbar, sofern die Einstellungen der Lesesoftware und eventuell geltende Sicherheitsvorgaben dies erlauben.

Inhaltsverzeichnis

1. Gegenstand und Ziel

1.1. Problematik und Hintergrund

IT-Grundschutz

Seit 1994 veröffentlicht das BSI[1] unter dem Schlagwort *IT-Grundschutz* Leitlinien und Standards, die eine Orientierung für den sicheren Einsatz von Informationssystemen ermöglichen sollen. Auf Basis des IT-Grundschutzes können Organisationen ihr ISMS[2] nach ISO[3] 27001 normiert zertifizieren lassen. Der methodische Ansatz des IT-Grundschutz-Kompendiums besteht darin, über einen modularisierten Kriterienkatalog (Bausteine) die Sicherheit technischer und organisatorischer Entitäten von IT-Systemen zu bewerten.

Container

Trotz der Modularität dieses Ansatzes lassen sich neuere Modelle des Betriebs von Software nur schwer mit dem verfügbaren Kriterienkatalog erfassen, weil sie sich strukturell stark von dessen impliziten Annahmen unterscheiden. Dies trifft besonders auf *Container-Plattformen* zu, bei denen die ausführbaren Teilsysteme der Software in binären Artefakten (Container-Images) gebündelt, vollautomatisch instanziiert und orchestriert werden. Instanzen und Netzwerkstrukturen werden dort regelbasiert und automatisiert der jeweiligen Systemlast angepasst und laufend aktualisiert.

Virtualisierung

Seit der ersten Ausgabe (Edition 2018) des IT-Grundschutz-Kompendiums (nachfolgend kurz: *Kompendium*) erscheint der Baustein SYS.1.5: *Virtualisierung*, der auf die Absicherung klassischer Virtualisierungslösungen abzielt. Dabei werden moderne Container-Ansätze ausdrücklich nicht behandelt, denn „andere Techniken, die teilweise ebenfalls mit dem Wort Virtualisierung in Verbindung gebracht werden (Anwendungsvirtualisierung mittels Terminalservern, Storage-Virtualisierung, Container etc.), sind nicht Gegenstand dieses Bausteins.“ [BSI18]

Baustein-Entwurf

Der Baustein SYS.1.6: *Container* ist auch in der 4. Ausgabe (Edition 2021) noch nicht Bestandteil des *Kompendiums*, sondern erneut als *Community Draft 2* publiziert. [BSI20b]

SaaS

Insgesamt lassen sich also die verbindlich veröffentlichten Bausteine nicht ohne Weiteres auf den Betrieb von Software in Container-Plattformen anwenden, die vor allem für *SaaS*[4]*-Angebote* immer bedeutender werden.[5]

Bedarf

Dennoch gibt es einen großen Bedarf bei Anbietern von SaaS-Diensten, deren Kunden hohe Anforderungen an den Datenschutz und die IT-Sicherheit haben, wie beispielsweise Finanzdienstleister, Banken, Versicherungsunternehmen oder Verarbeiter von Gesundheitsdaten. Diese wollen die Vorteile moderner Container-Plattformen nutzen und gleichzeitig standardisierte, zertifizierte Sicherheitsniveaus nachweisen können.

[1] Bundesamt für Sicherheit in der Informationstechnik (https://www.bsi.bund.de/)
[2] Information Security Management System
[3] International Organization for Standardization (https://www.iso.org/)
[4] Software-as-a-Service
[5] SaaS-Angebote sind dadurch charakterisiert, dass einer Vielzahl von Nutzern eine mandantenfähige Software über ein Netzwerk bereitgestellt wird, die auf den Systemen des Anbieters vorgehalten, aktualisiert und gesichert wird, sodass eine Nutzer-seitige Installation entfällt [Dun08, S. 265–266].

1.2. Fachbegriffe und Definitionen

Der Gegenstand dieser Arbeit wird anhand klarer Definitionen präzisiert, denn: „Für das Paradigma des Cloud Computings existiert keine etablierte und universal gültige Definition, vielmehr ist ein breites Spektrum an Beschreibungsversuchen und Begriffsabgrenzungen zu finden.“ [Ade+18, S. 3]

Um dieses Spektrum nicht weiter zu vergrößern, wird der vom NIST[6] publizierte und vielfach zitierte[7] Versuch einer Definition herangezogen [MG11]. Dabei werden die später vom NIST veröffentlichten Erläuterungen und Klärungen besonders berücksichtigt [Sim18].

Definition 1 (Cloud-Service) Ein *Cloud-Service* hat fünf wesentliche Eigenschaften:

1. *Selbstbedienung auf Abruf:* Die Nutzer können Rechenkapazitäten nach Bedarf automatisch und einseitig bestimmt abrufen.
2. *breitbandiger Netzwerkzugriff:* Die Funktionen sind über das Netzwerk verfügbar und werden über Standardmechanismen aufgerufen.
3. *Ressourcenbündelung:* Die Rechenressourcen des Anbieters werden gebündelt, um viele Nutzer über ein Mandanten-Modell zu bedienen.
4. *schnelle Elastizität:* Ressourcen können dynamisch bereitgestellt und freigegeben werden.
5. *gemessener Dienst:* In Cloud-Systeme wird die Ressourcennutzung automatisch gesteuert und optimiert, indem Messfunktionen eingesetzt werden.

[MG11, S. 2]

Definition 2 (Cloud-Anbieter) Ein *Cloud-Anbieter* (CSP[8]) bietet mehreren unabhängigen Dritten mindestens einen Cloud-Service (Definition 1) an [MG11, S. 1].

Definition 3 (Cloud-Nutzer) Ein *Cloud-Nutzer* (CSC[9]) nutzt mindestens einen Cloud-Service (Definition 1) mindestens eines Cloud-Anbieters (Definition 2) [MG11, S. 1].

Definition 4 (Cloud-Infrastruktur) Eine *Cloud-Infrastruktur* ist eine Sammlung von Hardware und Software, mit der die fünf wesentlichen Eigenschaften (Definition 1) von Cloud-Services ermöglicht werden. Die Cloud-Infrastruktur kann so betrachtet werden, dass sie sowohl eine physische Schicht als auch eine Abstraktionsschicht enthält. Aus den Hardwareressourcen, auf denen die bereitgestellten Cloud-Services betrieben werden, besteht die physische Schicht; sie umfasst typischerweise Server-, Speicher- und Netzwerkkomponenten. Die Abstraktionsschicht besteht aus Software, die auf der physischen Schicht eingesetzt wird. Hier werden die wesentlichen Cloud-Eigenschaften realisiert. Konzeptionell sitzt die Abstraktionsschicht über der physischen Schicht [MG11, S. 2].

Definition 5 (Servicemodell) Das *Servicemodell* bezeichnet die höchste Kategorie von Cloud-Services basierend auf der Art der bereitgestellten Rechenleistung. Jeder Cloud-Service kann einem von drei Servicemodellen zugeordnet werden, nämlich Software als Service (SaaS), Plattform als Service (PaaS[10]) oder Infrastruktur als Service (IaaS[11]). [MG11, S. 2–3]

[6] National Institute of Standards and Technology (https://www.nist.gov/)
[7] laut https://scholar.google.com 18269 und laut https://www.researchgate.net 8365 Zitierungen
[8] Cloud Service Provider
[9] Cloud Service Consumer
[10] Plattform-as-a-Service
[11] Infrastructure-as-a-Service

Definition 6 (SaaS) Bei *Software als Service* (SaaS) hat der Nutzer die Möglichkeit, die Anwendungen des Anbieters zu nutzen, die auf einer Cloud-Infrastruktur (Definition 4) betrieben werden. Der Zugriff auf die Anwendungen erfolgt von verschiedenen Client-Geräten entweder über eine Thin-Client-Schnittstelle (beispielsweise einen Webbrowser) oder über eine Programmschnittstelle (API[12]). Der Nutzer verwaltet oder kontrolliert weder die zugrundeliegende Cloud-Infrastruktur, die Netzwerk, Server, Betriebssysteme und Speicher umfasst, noch einzelne Anwendungsfunktionen. Eine mögliche Ausnahme stellen begrenzte benutzerspezifische Konfigurationseinstellungen der Anwendung dar. [MG11, S. 2]

Definition 7 (Bereitstellungsmodell) Das *Bereitstellungsmodell* bezeichnet, in welchem Umfang die Cloud-Services von mehreren Parteien genutzt werden und hat eine der vier Ausprägungen:

- *private Cloud:* Die Cloud-Infrastruktur wird ausschließlich durch eine einzelne Organisation mit mehreren Nutzern verwendet.
- *gemeinschaftliche Cloud:* Die Cloud-Infrastruktur wird ausschließlich durch eine bestimmte Gemeinschaft von Kunden aus Organisationen genutzt, die gemeinsame Anliegen haben, beispielsweise Mission, Sicherheitsanforderungen, Richtlinien oder Compliance-Überlegungen.
- *öffentliche Cloud:* Die Cloud-Infrastruktur kann offen von der Allgemeinheit genutzt werden.
- *hybride Cloud:* Die Cloud-Infrastruktur ist eine Komposition aus zwei oder mehr unterschiedlichen, privaten, gemeinschaftlichen oder öffentlichen Cloud-Infrastrukturen, die eigenständige Einheiten bleiben. Diese Einheiten sind durch standardisierte oder proprietäre Technologien miteinander verbunden, mit denen die Portabilität von Daten und Anwendungen ermöglicht wird.

[MG11, S. 3]

Definition 8 (Container) Ein *Container* ist eine einzelne Instanz eines abgeschlossenen Software-Artefakts (Container-Image), die gemeinsam mit anderen Containern auf einem virtuellen oder physischen Container-Host läuft und dessen Betriebssystem nutzt. Der Container enthält alle notwendigen binären Artefakte und Bibliotheken, um die Software auszuführen. Im Gegensatz zur virtuellen Maschine umfasst dies weder den Betriebssystemkern noch Bibliotheken zum direkten oder indirekten Zugriff auf zugrunde liegende Hardware [ADD19, S. 21] [Her17, S. 54].

Definition 9 (Dediziertes Hosting) Beim *dedizierten Hosting* wird innerhalb einer vertraglich definierten, besonders geschützten Rechenzentrumsumgebung IT-Ausstattung geschäftsmäßig bereitgestellt. Diese besteht aus Server-, Speicher- und Netzwerkkomponenten sowie der breitbandigen Anbindung der Hardware an öffentliche Netzwerke. Diese wird durch einzelne Kunden jeweils exklusiv genutzt. Anbieter von dediziertem Hosting werden nachfolgend kurz *Hosting-Anbieter* genannt.

Definition 10 (Schutzbedarf) „Der Schutzbedarf beschreibt, welcher Schutz für die Geschäftsprozesse, die dabei verarbeiteten Informationen und die eingesetzte Informationstechnik ausreichend und angemessen ist.“ [BSI21, S. 37]

[12] Application Programming Interface

1.3. Ziel der Arbeit

Zielsetzung

Bezogen auf die vorigen Definitionen wird als *Ziel* dieser Arbeit die Architektur einer Cloud-Infrastruktur (Definition 4) für ein SaaS-Betriebsmodell (Definitionen 5 und 6) entworfen, in der die Software basierend auf Containern (Definition 8) privat, gemeinschaftlich oder öffentlich bereitgestellt wird (Definition 7). Diese Architektur soll entsprechend den Anforderungen des *Kompendiums* für normale oder erhöhte Schutzbedarfe (Definition 10) abgesichert sein.

Dabei werden der Betrieb im Rahmen einer hybriden Cloud-Architektur (Definition 7) und die dafür zusätzlich nötigen Maßnahmen nicht betrachtet. Ebenfalls nicht Gegenstand der Arbeit sind die notwendigen organisatorischen Rahmenbedingungen, die gemäß *Kompendium* zwingend für einen sicheren IT-Betrieb erforderlich sind. Sie werden gleichermaßen als gegeben und erfüllt vorausgesetzt.

externes Hosting

Um den Umfang dieser Arbeit weiter zu begrenzen, wird davon ausgegangen, dass die Hardware der Cloud-Infrastruktur im Rahmen eines *Hosting-Vertrags*[13] für den Cloud-Anbieter (Definition 2) von einem Hosting-Anbieter (Definition 9) betrieben wird. Deshalb brauchen die Bausteine für die (Gebäude-) Infrastruktur nicht betrachtet werden, die für den gemäß IT-Grundschutz abgesicherten Rechenzentrumsbetrieb umgesetzt werden müssen.[14] Dies setzt voraus, dass der Hosting-Anbieter die IT-Einrichtungen seines Verantwortungsbereichs entsprechend den Vorgaben des Cloud-Anbieters und gemäß BSI IT-Grundschutz bereitstellt, möglichst nach ISO 27001 zertifiziert.

exklusive Hardware

Die Hardware wird aus Sicht des SaaS-Anbieters *nicht* mit weiteren Parteien geteilt, sodass der SaaS-Anbieter der einzige Mandant ist, der sie nutzt. Die Mandantenfähigkeit bezüglich der Cloud-Nutzer (Definition 3) ist eine Eigenschaft der Software selbst, die auf der entwickelten Plattform betrieben werden kann.

Dieses Vorgehen folgt Liebel, der anmerkt, „dass Public-Cloud-Implementierungen niemals die gleichen Sicherheitskriterien erfüllen können wie On-Prem-Systeme – und dies gilt umso mehr, wenn letztere *air gapped* bzw. *disconnected* betrieben werden." [Lie21, S. 68][15]

[13] Aus Sicht der Datenschutzgrundverordnung (DSGVO) stellt dieses Vertragsverhältnis eine *Auftragsverarbeitung* dar, die spezifischen gesetzlichen Anforderungen genügen muss.

[14] Anderson erläutert ausführlich, dass die physische Sicherheit der Ausrüstung an sich ein schwieriges Problem darstellt [And20].

[15] Liebel verschärft dabei den Grad der Isolation weiter: *On-Prem* kurz für *on premises* bezeichnet im Gegensatz zum Hosting den Betrieb von IT-Systemen in den eigenen Räumlichkeiten einer Organisation und *air gapped* einen Betrieb von Systemen ohne jegliche Verbindung zu öffentlichen Netzwerken.

2. Strukturen und Prinzipien

2.1. Struktur von Container-Plattformen

Schichtenmodell einer Container-Plattform

Analog zum OSI[1]-Modell der Schichtenarchitektur von Netzwerkprotokollen schlägt Liebel ein Modell aus sechs übereinanderliegenden Schichten vor [Lie21, S. 67].

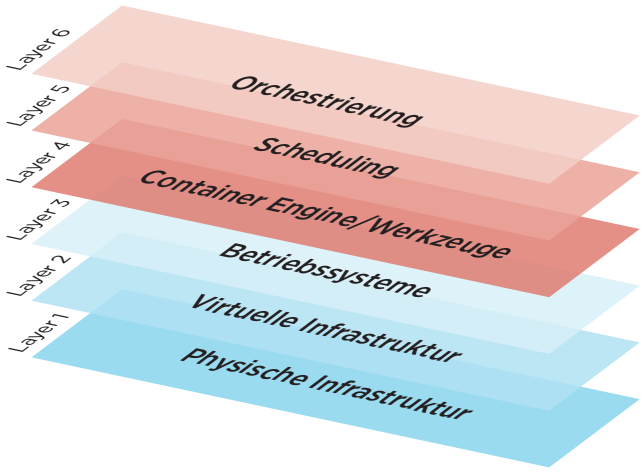


Abbildung 2.1.: Schichtenmodell einer Container-Plattform

Layer 1–3

Die Komponenten auf den ersten drei *Layern* entsprechen denen klassischer IT-Infrastrukturen — wie auch dort ist der zweite Layer optional. Es ist angeraten, auf der Ebene der Betriebssysteme spezifisch für den Betrieb einer Container-Plattform angepasste Komponenten zu wählen.

Layer 4

Auf dem *vierten Layer* sind Komponenten angesiedelt, mit denen die Artefakte (Container-Images) erstellt und verwaltet sowie die Container instanziiert werden.[2] In frühen Container-Implementierungen wurden diese unterschiedlichen Aufgaben durch die Container-Engine als einzige Komponente wahrgenommen. Allerdings werden Artefakte zunehmend durch separate Komponenten erzeugt (build), lokal aktualisiert (pull) und ausgeführt (run): „[...] eine Container-Engine in Container-Clustern ist in der Regel ohnehin auf die beiden elementarsten Funktionen reduziert (*pull* und *run*).“ [Lie21, S. 72]

Layer 5

Mit dem *fünften Layer* wird die Eigenschaft *schnelle Elastizität* nach Definition 1 sichergestellt: Hier wird dafür gesorgt, dass Container entsprechend der Auslastung der darunterliegenden Schichten nach Bedarf automatisch bereitgestellt und wieder beseitigt werden.

Layer 6

Über den *sechsten Layer* werden schließlich das koordinierte Zusammenspiel und die Überwachung der einzelnen Komponenten organisiert.[3] Die Eigenschaft *gemessener Dienst* nach Definition 1 wird dort realisiert. Es werden die Services definiert, Netzwerk- und Namensraum dynamisch verwaltet, die Last gemessen, die Instanzen aktualisiert sowie die verfügbaren Ressourcen überwacht, kontrolliert und skaliert.

[1] Open Systems Interconnection

[2] Dabei werden in einem automatisierten Prozess versionierte Artefakte erstellt, die als binäres und aus Sicht der laufenden Anwendung unveränderliches Abbild zentral gespeichert werden. Die aktiven Instanzen (Container) werden nach Bedarf auf den einzelnen Knoten des Clusters aus diesen Artefakten erzeugt, nachdem eine lokale Kopie auf dem Knoten zwischengespeichert wurde.

[3] Je nach Automatisierungsgrad umfasst dies meist die Layer 4–5, seltener auch die Layer 2–3.

In typischen aktuellen Installationen sind die Komponenten der obersten Layer nicht strikt getrennt. So kann Kubernetes („nach wie vor die populärste Container-Cluster- und Orchestrierungslösung für Self-Hosted- und Cloud-Umgebungen“ [Lie21, S. 269]) mit einem geeigneten Satz von Plug-ins als alleiniges Werkzeug für Scheduling und Orchestrierung eingesetzt werden.

Durch die Gliederung in Schichten können im weiteren Verlauf die Komponenten einfacher den entsprechenden Bausteinen im IT-Grundschutz-Kompendium zugeordnet werden.

Komponenten einer Container-Plattform

lose Kopplung

Die Abb. 2.2 zeigt Komponenten einer Container-Plattform. Im Sinne der Systemarchitektur bezeichnet eine *Komponente* in diesem Kontext ein in sich geschlossenes Teilsystem,[4] das über wohldefinierte Schnittstellen mit den anderen Komponenten *(lose) gekoppelt* ist. Üblicherweise lässt sich eine Komponente durch ein spezifisches Produkt eines Herstellers implementieren.

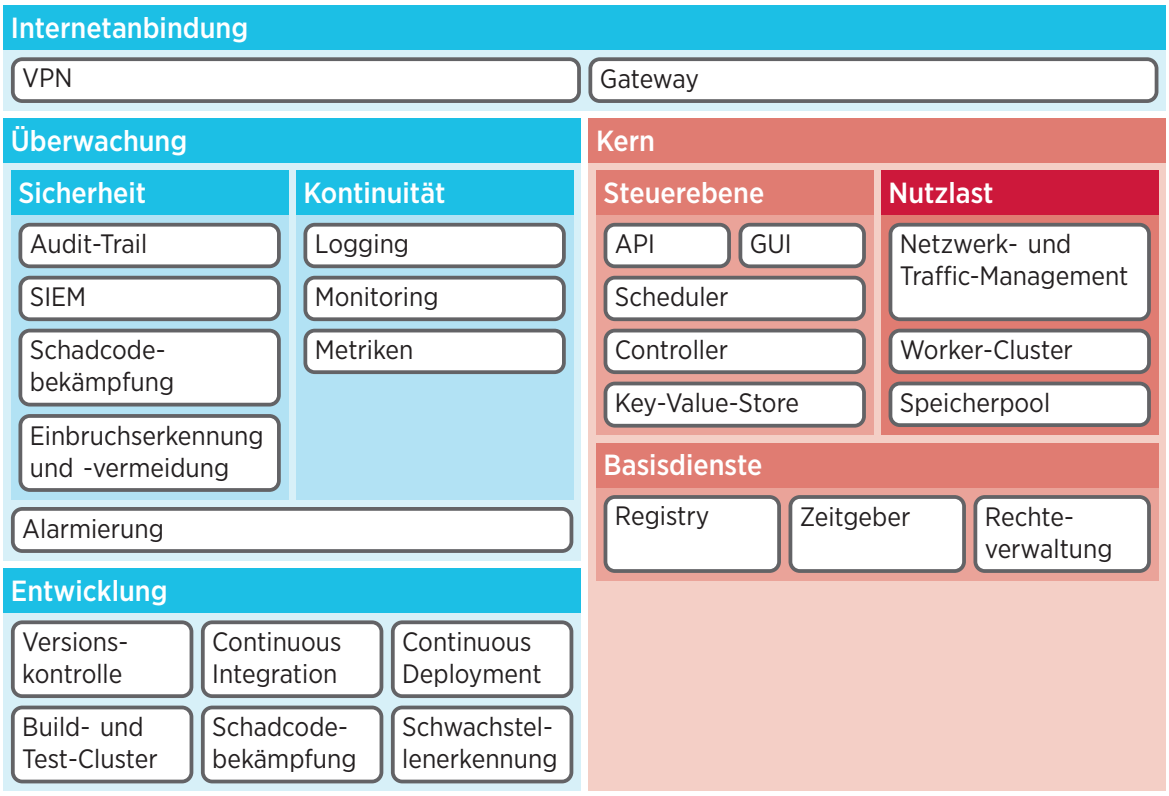


Abbildung 2.2.: Teilsysteme und Komponenten einer Container-Plattform

Teilsysteme

Grundsätzlich lassen sich die Komponenten den vier Bereichen Kernsystem (nachfolgend kurz: *Kern*), Überwachung, Entwicklung und Internetanbindung zuordnen. Der *Kern* lässt sich weiter aufteilen in die *Teilsysteme* Basisdienste, Steuerebene und Nutzlast. Im Teilsystem der Überwachung lassen sich Komponenten den untergeordneten Teilsystemen Sicherheit und Kontinuität zuordnen. Das Teilsystem der Entwicklung umfasst stärker gekoppelte Komponenten, deshalb wird es nicht weiter untergliedert. Im Rahmen eines SaaS-Angebots wird über die Internetanbindung der *Kern* an ein öffentliches Netzwerk (Internet) angeschlossen.

[4] Systeme lassen sich hierarchisch in logisch kleinere Teilsysteme (echte Teilmengen) gliedern. Als Baumstruktur stellen Komponenten die Blätter, Teilsysteme die inneren Knoten und das Gesamtsystem die Wurzel dar.

Internetanbindung

In der überwiegenden Anzahl der Anwendungsfälle stellt eine Container-Plattform Dienste in einem öffentlichen Netz bereit. Typischerweise wird sie in einem externen Rechenzentrum betrieben und aus der Ferne administriert. Den Zugriff auf die Dienste der Plattform vermittelt ein *Gateway*, den administrativen Fernzugriff meistens ein VPN[5]. Gateway, VPN

Kern: Basisdienste, Nutzlast und Steuerebene

In der *Registry* werden die Software-Artefakte (Container-Images) gespeichert und verwaltet. Dies kann neben den Komponenten der Nutzlast auch solche der Steuerebene umfassen. Die Registry selbst kann für Entwicklungszwecke als externer Cloud-Service ausgeprägt sein, ist im produktiven Betrieb aber grundsätzlich eine lokale Komponente. Ein zentraler *Zeitgeber* ist für den hochverfügbaren Cluster-Betrieb einiger Komponenten und für eine konsistente Protokollierung von Ereignissen unabdingbar.[6] Die zentrale *Rechteverwaltung* steuert sowohl den administrativen Zugriff als auch den Zugriff der Komponenten untereinander. Registry, Zeitgeber, Rechteverwaltung

Die *Nutzlast* stellt den Teil der eigentlichen Anwendung dar, die dem Cloud-Nutzer als Dienst bereitgestellt wird. Dieser Bereich muss die ersten drei wesentlichen Eigenschaften des Cloud-Services (Definition 1) bereitstellen. Die Nutzlast liefert im Wesentlichen drei *Ressourcentypen* für die betriebene Software: *Netzwerkverbindungen*, *Rechenressourcen* (CPU/RAM) innerhalb eines *Worker-Clusters* sowie *persistenten Speicher* in einem *Speicherpool*. Nutzlast, Ressourcentypen

In der *Steuerebene* wird dafür gesorgt, dass die Software im Bereich der Nutzlast aus den benötigten Artefakten automatisch instanziiert wird. Die Auslastung der Ressourcen der Nutzlast wird gemessen und diese entsprechend skaliert. So werden die übrigen wesentlichen Eigenschaften sichergestellt. Steuerebene

Ein *Key-Value-Store* speichert als zentrale, hochverfügbar ausgelegte Komponente der Steuerebene den Systemzustand:[7] „Key/Value-Stores und die über sie zur Verfügung gestellten Service-Registry/Discovery-Funktionen stellen [...] das definitive Schaltzentrum aller Container-Cluster-Dienste dar." [Lie21, S. 257] Key-Value-Store

Der *Controller* verwaltet alle in der Nutzlast vorhandenen und genutzten Ressourcen und führt Buch über sie. In konkreten Implementierungen besteht der Controller meist aus mehreren Komponenten, die je für einen Ressourcentyp verantwortlich sind. Liebel übernimmt den Begriff *Controller Manager* für diese Komponente aus der Implementierung von Kubernetes und bezeichnet als *Controller* nur die „*Regelkreise* für bestimmte Ressourcentypen." [Lie21, S. 277] Abhängig vom Automatisierungsgrad können auch Ressourcen aus den Layern 1–2 gesteuert werden. Controller

Ähnlich wie in einem Betriebssystemkernel teilt der *Scheduler* abhängig von der Auslastung die Ressourcen in einem eigenen Regelkreis zu und verteilt bei Bedarf auch neu. Im Gegensatz zu einem klassischen Scheduler werden die nötigen Ressourcen zusätzlich dynamisch skaliert, beispielsweise die Zahl der gleichartigen Container, mit denen Anfragen verarbeitet werden. Scheduler

Die Komponenten *API und GUI*[8] stellen die von außerhalb des *Kern*s erreichbaren Schnittstellen zur Konfiguration seiner Parameter dar. Mit dem API können Controller und Scheduler programmatisch und automatisiert gesteuert werden, insbesondere durch die Komponente *Continuous Deployment* aus dem Bereich der Entwicklung. Das optionale GUI liefert eine interaktive Schnittstelle, um notfalls Parameter manuell steuern und überprüfen zu können. API und GUI

[5] Virtual Private Network

[6] Traditionelle Cluster-Verfahren brauchen zur Aufrechterhaltung konsistenter Datenhaltung eine exakte gemeinsame absolute Uhrzeit. Moderne Ansätze kommen mit Vektoruhren ohne zentrale Taktgeber aus.

[7] Der Begriff *Service-Registry* ist unglücklich gewählt. Während die *Registry* Artefakte speichert, führt die *Service-Registry* (hier: Key-Value-Store) ein Verzeichnis der im System verfügbaren Instanzen und Dienste.

[8] Graphical User Interface

Überwachung: Sicherheit und Kontinuität

Die Komponenten der Überwachung gewährleisten die *Kontinuität* des System- und Anwendungsbetriebs sowie die *Sicherheit* des *Kern*s. Dabei findet der Aspekt der Kontinuität in der Literatur recht häufig einen Platz, während Sicherheit oft nur nachgelagert oder überhaupt nicht als integraler Bestandteil von Container-Plattformen betrachtet wird. Im Sinne des Security-by-Design-Ansatzes muss aber auch der Aspekt der Absicherung des Systems von vornherein mit geplant und entwickelt werden.

Logging

Eine zentrale *Logging*-Komponente erfasst, aggregiert und komprimiert die erzeugten Log-Einträge *aller* anderen Komponenten des Systems. In der Praxis ist eine vollständige Zentralisierung oft sehr aufwendig, da viele Komponenten in ihrer Standardkonfiguration Log-Einträge nur in lokal vorgehaltene Dateien ausgeben.

Monitoring

Das interne *Monitoring* überwacht die Funktionsfähigkeit der anderen Komponenten laufend, einerseits über die regelmäßige Abfrage von Systemparametern, wie CPU-, Arbeitsspeicher- und Datenträgerauslastung, andererseits über regelmäßiges Absetzen von Prüfanfragen an Services und Auswerten der zurückgegebenen Werte. Angelehnt an physische Sonden zur Messdatenerfassung wird eine einzelne Prüfregel in dem Kontext *Sensor* genannt.

Alarmierung

Eine *Alarmierung* des zuständigen Personals wird durch das Monitoring und die Komponenten aus dem Bereich der Sicherheit ausgelöst, wenn Sensoren ihren als funktionstüchtig definierten Wertebereich verlassen. Der Meldeweg für einen Alarm kann von einer einfachen E-Mail bis zu einem hochkomplexen, mehrstufigen und feingranularen Alarmierungsschema mit mehreren Nutzergruppen und Kanälen (wie E-Mail, SMS, mobile App, Pager) reichen. Im Sinne einer guten Systemarchitektur sollte die Komponente zur Alarmierung nur einmal existieren. Um Aufgaben stärker zu trennen, kann es aber auch sinnvoll sein, die Alarmierungen für die Bereiche Kontinuität und Sicherheit unabhängig zu betreiben.

Metriken

Anhand von *Metriken* können Trends erkannt werden. Daraus lässt sich der Bedarf an Kapazitäten übergeordnet ableiten und längerfristig planen. Dabei werden sowohl Sensordaten des Monitorings als auch die Auslastung und Nutzung der Ressourcen der Nutzlast laufend ausgewertet und aggregiert. Das betrifft beispielsweise extern übertragene Datenvolumina sowie Zahl der Anfragen, der Schreib- und Lesezugriffe auf Speicher, der verwendeten Knoten im Worker-Cluster und der laufenden Container-Instanzen.

Audit-Trail

Ein *Audit-Trail* erfüllt eine ähnliche Aufgabe wie das Logging: Es werden wesentliche Ereignisse der Komponenten fortlaufend erfasst und persistiert. Eine zentrale Anforderung an ein Audit-Trail ist die hohe Integrität der Aufzeichnungen. Es muss gewährleistet sein, dass diese nicht nachträglich verändert werden können.

Dieser Anspruch ergibt sich aus dem Bedarf, unbeabsichtigte oder auch unrechtmäßige Zugriffe auf Ressourcen des Systems rechtssicher und eindeutig attribuierbar nachvollziehen zu können. Ein Audit-Trail kann beim *Audit* zum Nachweis bestimmter Eigenschaften des Systems oder während *forensischer Untersuchungen* zum Nachweis missbräuchlicher Nutzung herangezogen werden. Weil ein Audit-Trail aufwendig zu implementieren ist und die anfallenden Datenmengen groß sind, werden die erfassten Ereignisse gegenüber einem Logging meistens stark eingeschränkt. Im Fokus stehen systemkritische Ereignisse wie geänderte Zugriffsrechte sowie schreibende (und seltener lesende) Zugriffe auf die persistenten Daten der Nutzlast.

Audit
Forensik

Um sicherheitsrelevante Ereignisse und Informationen ganzheitlich zu erfassen und auszuwerten, kann ein SIEM[9]-System eingesetzt werden. Eine solche Komponente einzurichten und nutzbringend zu betreiben, stellt hohe technische und organisatorische Anforderungen und ist kostspielig. Sie wird daher zumeist bei höchsten Sicherheitsanforderungen und in personell und finanziell angemessen ausgestatteten Organisationen umgesetzt. In den meisten Fällen lassen

SIEM

[9] Security Information and Event Management

sich die entsprechenden Aufgaben mit geeigneten organisatorischen Maßnahmen basierend auf den anderen Komponenten des Bereichs Sicherheit realisieren. Eine Auslagerung der Funktion an spezialisierte externe Dienstleister (externes SOC[10]) ist ebenfalls möglich und üblich.

Schadcode

Die Erkennung und Bekämpfung von *Schadcode* (oft vereinfachend *Virenscanner* genannt) bezieht sich sowohl auf die Untersuchung der Datenströme als auch der persistierten Daten im *Kern* oder im gesamten System. Dabei wird oft eine einzige Komponente verwendet, die basierend auf Mustererkennung, Heuristiken und zunehmend Machine-Learning-Algorithmen sowohl Datenströme als auch Datenobjekte als harmlos oder schädlich klassifiziert. In letzterem Fall werden Datenströme unterbrochen oder modifiziert und Datenobjekte zur näheren Untersuchung in Quarantänezonen verschoben oder gelöscht.

Einbrüche (IDS, IPS)

Einbrüche sollen mit einem IDS[11] erkannt und mit einem IPS[12] aktiv verhindert oder verlangsamt werden. Diese beiden Funktionen sind hier, wie auch allgemein üblich[13], zusammengefasst worden — während IDS größere Verbreitung finden, ist die Umsetzung aktiver und vor allem effektiver IPS schwieriger; hohe Fehlerraten in der Erkennung machen deren stabilen Betrieb aufwendig. In der Fachliteratur werden IPS auch in aktuellen Ausgaben teilweise überhaupt nicht ([Eck18], [And20]) oder nur am Rande ([Pog17, S. 203], [Mül18, S. 525], [Har18, S. 252]) erwähnt. Meist beschränkt sich der Wirkungsbereich von IDS und IPS auf die Layer 2–3, sollte aber möglichst auch die Layer 4–6 umfassen und kann bei sehr hohen Sicherheitsanforderungen auf den Layer 1 ausgeweitet werden. Es ist allerdings anzumerken, dass den Möglichkeiten von IDS, und damit erst recht von IPS, nach wie vor enge Grenzen gesetzt sind. Anderson erklärt diese Einschränkungen sehr ausführlich [And20, S. 744].

Entwicklung

Der letzte Bereich, der charakteristisch für moderne Container-Plattformen ist, umfasst Komponenten zur Automatisierung der Softwareentwicklung und -aktualisierung. Ihren eigentlichen Vorteil gegenüber klassischen Betriebsmodellen spielen Container-Plattformen nur durch massive Nutzung von Automatisierung aus: Neue Softwareversionen können in kleinen Inkrementen überprüft und in Form vieler spezifischer, lose gekoppelter Services bereitgestellt werden. Dies wird mit einer angepassten Organisationsform der Softwareentwicklung verbunden: „agile development, which may be summed up in the slogan: ‘Solve your worst problem. Repeat’.“ [And20, S. 984]

Versions-kontrolle

Die *Versionskontrolle* verwaltet die Historie der Software- und Konfigurationsobjekte in einer für Menschen und Maschinen lesbaren und zu bearbeitenden Form. Binäre Bibliotheken und Kompilate zur Wiederverwendung in den erzeugten Artefakten können dort ebenfalls versioniert abgelegt werden. Meistens verwalten separate Werkzeuge den Quellcode und die Bibliotheken. Es ist jedoch entscheidend, dass beides mit einer geeigneten Historie und Versionierung vorgehalten wird, um notfalls schnell ältere Versionen der Software in den produktiven Betrieb überführen zu können (*Rollback*). Auch für Anwendungsfälle des Parallel- und Probebetriebs neuer Versionen (A/B- und Canary-Deployments) ist dies unabdingbar.

Continuous Integration

Der automatische Prozess, in dem Artefakte aus Quellcode und Bibliotheken erzeugt und anschließend technisch wie fachlich geprüft werden, heißt *Continuous Integration*, ebenso wie die Komponente, die diesen Prozess steuert. Dabei wird die Versionskontrolle aktiv oder passiv auf Änderungen überwacht. Regelbasiert werden daraufhin Artefakte in einem meist mehrstufigen Verfahren neu erzeugt und den drei nachfolgend beschriebenen Komponenten zur Prüfung

[10] Security Operation Center
[11] Intrusion Detection System
[12] Intrusion Prevention System
[13] Aktuelle Marktstudien und Trend-Monitore nennen IDS/IPS nahezu ausschließlich zusammengefasst.

übergeben. Die Reihenfolge der Prüfungen kann variieren und wird hinsichtlich minimaler Ressourcennutzung und minimaler Laufzeiten optimiert (*fail-fast*). Nach vollständig erfolgreichen Prüfungen wird das Artefakt mit einer eindeutigen Version gekennzeichnet und in der Registry abgelegt. Die oft zahlreichen Prozessschritte in komplizierten Abfolgen werden zu sogenannten *Pipelines* verknüpft und in Form von Konfigurationsskripten ebenfalls in der Versionskontrolle gehalten. Das meist speicher- und rechenintensive Erzeugen der Artefakte wird häufig von einem *Build-Cluster* und seltener von einzelnen Servern übernommen.

Build-Cluster

Zur Überprüfung der korrekten Funktionalität und zur Vermeidung von Rückfällen auf bereits früher behobene Fehler (Regressions) werden die neu erzeugten Artefakte in einem *Test-Cluster* wohldefinierten und reproduzierbaren Testszenarien auf Basis synthetisch generierter Daten und Anfragen unterzogen. Dabei wird meistens nur ein einzelnes Artefakt in neuer Version gegen andere Artefakte der produktiv eingesetzten Version getestet. Dieses Verfahren motiviert die Bezeichnung *Integration*: Eine einzelne Komponente wird in neuer Version mit dem bestehenden System *integriert*.

Test-Cluster

Eine *Schwachstellenerkennung* kann den Quellcode, die eingesetzten Bibliotheken, die erzeugten Artefakte oder im Idealfall alle drei Aspekte berücksichtigen. Dabei wird der Quellcode statisch analysiert (sowohl syntaktisch als auch semantisch) und die Versionen eingesetzter Bibliotheken werden anhand von Datenbanken auf bekannte Verwundbarkeiten geprüft. Dynamische Methoden, bei denen Instanzen der Artefakte in Testumgebungen ungültige oder manipulierte Anfragen und Eingabewerte verarbeiten, beispielsweise Fuzzing, erkennen mögliche Fehlfunktionen im produktiven Betrieb.

Schwachstellen-erkennung

Die *Schadcodebekämpfung* adressiert fast immer die fertig erzeugten Artefakte oder deren binäre Vorstufen. Es ist leichter, schädliche Software in kompiliertem Binärcode zu verstecken, weshalb die weitaus meiste Schadsoftware in binärer Form verbreitet wird. Obwohl Schadcode auch in Form von Quelltext eingeschleust werden kann und entsprechende Angriffe bereits häufiger durchgeführt wurden, basieren daher die gängigen Verfahren auf der Erkennung von Mustern (*Signaturen*) binärer Schadsoftware. Im Bereich der Entwicklung können die gleichen Komponenten zur Schadcodebekämpfung verwendet werden wie im Bereich der Überwachung.

Schadcode-bekämpfung

Am engsten an den *Kern* der produktiven Software gekoppelt ist das *Continuous Deployment*. Diese Komponente veranlasst „das automatische Deployen erfolgreicher Builds in die Produktiv-Umgebung“ [ADD19, S. 271] mit möglichst keiner oder nur minimaler Interaktion: „Entwickler sollten neue Versionen deployen können, indem sie entweder einen Button anklicken, einen Merge-Request zusammenführen oder ein Git-Release-Tag pushen.“ [ADD19, S. 271]

Continuous Deployment

Das Continuous Deployment wird aktiv als Prozessschritt der Continuous Integration angestoßen oder durch eine überwachte Änderung der Inhalte der Registry ausgelöst. Über das API der Steuerebene werden die Rahmenparameter der Orchestrierung bereitgestellt und aktualisiert. Diese Parameter, die Service- und Strukturdefinition sowie die Versionen der einzelnen Komponenten der Anwendung, sollten ebenfalls in Form von Skripten in der Versionskontrolle gehalten werden. Oft werden Continuous Deployment und Continuous Delivery in einer Komponente zusammengefasst.

2.2. Security-by-Design-Ansatz

Dem Security-by-Design-Ansatz liegt die Frage zugrunde: *Welche Strukturen, Prinzipien, Systeme und Prozesse werden benötigt, um einen Geschäftsprozess mit gegebenem Schutzbedarf hinreichend sicher nach Stand der Technik zu unterstützen oder umzusetzen?* Die Antwort zielt darauf ab, schon in der Planungsphase systematisch Eigenschaften einzuarbeiten, durch die Systeme und Prozesse *inhärent sicher* gestaltet werden.

inhärente Sicherheit

Der klassische Ansatz der Absicherung stellt hingegen eher die Frage: *Wie können vorliegende Systeme, vorliegende Designs oder bestehende Prozesse und deren technologische Unterstützung oder Umsetzung entsprechend der identifizierten Schutzbedarfe nach Stand der Technik hinreichend sicher geschützt werden?* Die Antworten ergeben eine Bewertung des vorgefundenen Grades der Absicherung (Schutzniveau) sowie Maßnahmen zu dessen iterativer Verbesserung.

Motivation von Security-by-Design

Die Idee, dass Sicherheit nachträglich durch zusätzliche Maßnahmen erzielt werden kann, ist verbreitet und alt. Analog zur Absicherung von Gebäuden[14] hat sich dieses Konzept in den letzten Jahrzehnten auch in der IT-Branche verbreitet und verankert.[15] Es setzt sich aber langsam die Erkenntnis durch, dass dieser Ansatz nur wenig Schutz vor gezielten Angriffen bietet und nicht mehr als Stand der Technik gelten kann. So bemerkt Eckert in der Einleitung ihres Lehrbuchs: „Die Qualität eines sicheren IT-Systems hängt wesentlich davon ab, dass seine Konstruktion methodisch und systematisch erfolgt." [Eck18, S. V] Auch Abolhassan erwähnt diesen Wandel bereits im Geleitwort: „Bisher war die Sicherheit von Lösungen und Produkten eher eine Zusatzfunktion, die ins fertige Produkt integriert wurde. Zunehmend wird sie von Anfang an mitgedacht und so besser integriert." [Abo17b, S. VII] Im *Kompendium* wird diese Entwicklung ebenfalls sichtbar: in der 4. Ausgabe (Edition 2021) wurde der Bereich der Softwareentwicklung um zwei Bausteine zu den Themen *Bedrohungsmodellierung* und *sicherer Software-Entwurf* ergänzt [BSI20a; BSI21].

Sicherheit per Konstruktion

Gleichzeitig erweisen sich tatsächliche Sicherheitsdefizite und mangelndes Vertrauen in die IT-Sicherheit insbesondere von Cloud-Diensten als eine der größten Hürden, die Technologie zu nutzen [Bin20, S. 22, 48]. Angriffe werden weiter professionalisiert und häufiger durchgeführt: „Laut einer aktuellen Sicherheitsstudie von IBM gehen rund 70 Prozent der Datenverluste auf Hackeraktivitäten und Cyberangriffe zurück. Im Vergleich zum vergangenen Jahr entspricht dies einem Anstieg um 556 Prozent." [Bin20, S. 36] Durch unzulänglich geschützte IT-Systeme ausgelöste Schäden wachsen ebenfalls stetig: „2017 verloren 37 Prozent der mittelständischen deutschen Unternehmen Geschäftsdaten durch Cyberkriminalität, menschliches Versagen oder Hardware-Ausfälle. Der wirtschaftliche Schaden belief sich pro Fall auf rund 560 000 Euro." [Bin20, S. 31]

Anstieg von Angriffen

Daher ist für die Konstruktion neuer IT-Systeme ein systematischer, *umfassender Schutz* erforderlich, der sowohl die Produkte als auch deren Nutzungskontext erfasst: „Das Security-by-Design-Prinzip muss im Zentrum jeder Produktentwicklung und Implementierung stehen. Dies umfasst die Software-Sicherheit über den gesamten Lebenszyklus hinweg, aber auch die gesamte Infrastruktur und die Prozesse." [Abo17b, S. 6]

umfassender Schutz

Durch diesen Ansatz werden mit den Systemen auch deren Schutzmaßnahmen weiter automatisiert: „[…] *Manuelle Ansätze können und dürfen hier nicht mehr greifen.* […] Komplexe Container-Cluster erfordern Security-technisch eine Gesamtlösung." [Lie21, S. 1080]

Automatisierung

Schaar verweist auf die gesellschaftliche, wirtschaftliche und politische Dimension dieser Entwicklung [Sch17, S. 24] und schließt: „Deshalb gewinnen Fragen nach dem Design, der Funktionsweise und der Einbettung der Informationstechnik existenzielle Bedeutung für die Zukunft der Gesellschaft und für die Entfaltungsmöglichkeiten des Einzelnen. Je stärker der ‚Code' unser Leben beeinflusst, desto bedeutsamer wird die Frage, wer den Code bestimmt und welchen Regeln er folgt." [Sch17, S. 25]

übergeordnete Relevanz

[14] Alarmanlagen für Schaufenster, *Sicherheitsriegel*, *Sicherheitszylinder* mit Standardmaßen, *Fensterschlösser* und ähnliches

[15] *Virenscanner* für Betriebssysteme ohne *strukturellen* Schutz gegen Schadsoftware; *Security-Appliances*; *Security-as-a-Service*-Angebote

Vorgehensmodelle

plan, do, check, act

Ein bekanntes Vorgehensmodell, mit dem Systeme entwickelt und laufend verbessert werden können, ist der PDCA[16]-Zyklus (auch Demingkreis genannt).[17] Es wird nahezu unverändert[18] in mehreren Quellen zitiert und als grundlegende Vorgehensweise zur Konstruktion sicherer Systeme empfohlen [Pog17, S. 26–27]. Im Vordergrund steht die Auffassung, dass ein Vorgehensmodell den Lebenszyklus der zu schützenden Systeme komplett erfassen muss: „Die Aufrechterhaltung eines geforderten Sicherheitsniveaus erfordert einen dynamischen, iterierenden Prozess der kontinuierlichen Überwachung der Einhaltung der Schutzziele [...]." [Eck18, S. 174]

konkretes Vorgehen

Im Bereich der Softwareentwicklung wurde diese abstrakt formulierte Vorgehensweise mehrfach konkretisiert, beispielsweise im SDL[19] von Microsoft [Eck18, S. 207–212], im mittlerweile archivierten CLASP[20] [CLASP06] und im aktuellen SAMM[21] [SAMM2020] des OWASP.[22]

Deskriptive, adaptive und empirisch fundierte Ansätze wie das BSIMM[23] [BSIMM20] der Synopsis Inc. sind seltener zu finden. Dieses Verfahren beruht darauf, tatsächlich durchgeführte Maßnahmen freiwillig teilnehmender Organisationen zu erheben und auszuwerten.

Im *BSI-Standard 200–2* werden Verfahren formalisiert, um IT-Systeme abzusichern [BSI17]. Mit dieser Vorgehensweise werden in einem „Informationsverbund sowohl bereits realisierte als auch in Planung befindliche Anteile" [BSI21, S. 21] realisiert, wenngleich sie laut Eckert primär „[...] darauf abzielt, [...] bestehende IT-Infrastrukturen abzusichern." [Eck18, S. 174]

Planen

Bei der *Planung* sicherer IT-Systeme ist es unverzichtbar, Struktur, Bedrohungen und Risiken zu analysieren, den Schutzbedarf zu ermitteln sowie eine Sicherheitsstrategie und ein Sicherheitsmodell aufzustellen [Pog17, S. 27–28]. Eine gute und detaillierte Erklärung dieser Teilschritte bietet Eckert im Kapitel *Security Engineering* [Eck18, S. 171–212]. Da sie insgesamt umfangreich und damit in der Durchführung aufwendig sind, sollten sie vereinfacht und standardisiert werden. Hier setzt das Konzept des BSI IT-Grundschutzes an. Beispielsweise wird im IT-Grundschutz-Kompendium eine separate Risikoanalyse nur für hohe Schutzbedarfe empfohlen oder für Komponenten, die nicht bereits von seinen Bausteinen erfasst werden [Eck18, S. 195].

Ausführen

An die *Umsetzung* werden in den Bausteinen des IT-Grundschutz-Kompendiums ganz konkrete Anforderungen gestellt. Im Kontext der Zertifizierung nach der Norm ISO 27001 werden die Handlungsempfehlungen unter ISO 27031–27040 in Form von Kontrollelementen (Controls) konkretisiert [Mül18, S. 126–138]. Darüber hinaus gibt es allerdings nach wie vor nur wenige standardisierte Handreichungen für die Umsetzung: „Dedizierte Methodiken zur Konstruktion sicherer IT-Systeme im Sinne eines systematischen Security-Engineerings wurden bislang kaum entwickelt." [Eck18, S. 172] Anderson orientiert sich mit seinem interdisziplinären Ansatz an den Methoden des *Safety-Engineerings* (d. h. an der systematischen Konstruktion von Betriebssicherheit und Unfallvermeidung). Er bezeichnet Security-Engineering als das schwierigste und zugleich bedeutendste Forschungsfeld der IT-Sicherheit, das vor allem wegen der interdisziplinären Fragestellungen, die sich auf Software-Engineering, angewandte Psychologie, Ökonomie und Management beziehen, nach wie vor wenig Beachtung findet [And20, S. 1011].

Prüfen

Kriterienkataloge sind ein gängiges Mittel, um die IT-Sicherheit zu *prüfen*. „Kriterien definieren [...] eine Art Metrik zur Bewertung der Sicherheit eines Systems [...]." [Eck18, S. 213]

[16] Plan — Do — Check — Act
[17] Der Aufbau dieser Arbeit lässt sich als ein einzelner Durchlauf durch diesen Zyklus auffassen.
[18] Poguntke greift den Punkt *Act* sinngemäß aber treffend als *Anpassen* (adapt) auf [Pog17, S. 26]
[19] Security Development Lifecycle
[20] Comprehensive Lightweight Application Security Process
[21] Software Assurance Maturity Model
[22] Open Web Application Security Project (https://owasp.org/)
[23] Building Security In Maturity Model

Eckert nennt die TCSEC[24] als die „ältesten Kriterien zur Bewertung der Sicherheit von IT-Systemen“ [Eck18, S. 213], die deutschen IT-Kriterien, die europäischen ITSEC,[25] die CTCPEC[26] sowie als modernen und vereinheitlichten internationalen Nachfolger die CC[27] [Eck18, S. 213–238]. Im Bereich der Softwareentwicklung sei noch der ASVS[28] genannt. Hier werden Prüfschemata für drei *Sicherheitsüberprüfungsstufen* (security verification levels) von Web-Applikationen definiert.

Durch Audits wird das erreichte Sicherheitsniveau regelmäßig und systematisch bewertet. Kommen als Ergebnis Differenzen zu den Vorgaben zum Vorschein, werden entsprechende Handlungen und Korrekturen empfohlen. Es wird dann nachgeprüft und erneut bewertet, ob diese in angemessenem Umfang und ausreichender Güte umgesetzt wurden. Audits finden beispielsweise im Rahmen von Zertifizierungen statt, mit denen offiziell (durch staatliche oder unabhängige Prüfstellen) bestätigt wird, dass ein System (wie ein ISMS) mit formal festgelegten Prüfkriterien oder Standards übereinstimmt.

Anpassen

Damit schließt sich der Zyklus, der sehr ausführlich und umfassend von Poguntke [Pog17], Eckert [Eck18] und Müller [Mül18] beschrieben wird.

Grundfunktionen der IT-Sicherheit

IT-Sicherheit soll gewährleisten, dass bestimmte schützenswerte Eigenschaften von IT-Systemen dauerhaft und nachprüfbar aufrechterhalten werden. Meistens werden als primäre Eigenschaften *Vertraulichkeit, Integrität und Verfügbarkeit* genannt, die häufig als CIA[29] referenziert werden. Sie sind im Rahmen der DSGVO[30] mittlerweile gesetzlich verankert. Oft werden sie ergänzt um Authentizität, Verbindlichkeit und Anonymität. In der DSGVO kommen noch Belastbarkeit bzw. Resilienz hinzu. Eckert liefert präzise Definitionen der *Schutzziele*; der Begriff *Vertrauen* wird bewusst nicht verwendet [Eck18, S. 7–15]. Die Schutzziele werden über spezifische Grundfunktionen erreicht, die Eckert ausführlich beschreibt: Identifikation und Authentifikation; Rechteverwaltung; Rechteprüfung; Beweissicherung; Wiederaufbereitung; Gewährleistung der Funktionalität (Verfügbarkeit) [Eck18, S. 201–204].

Vertraulichkeit, Integrität und Verfügbarkeit

Schutzziele

Dabei beantwortet die Identifikation die Frage nach dem Subjekt: *Wer oder was operiert auf dem System?* Die Authentifikation stellt sicher, dass die Subjekte die sind, die sie vorgeben zu sein. Mit der Rechteverwaltung wird festgelegt, welche Subjekte unter welchen Rahmenbedingungen welche Ressourcen des Systems in welcher Weise nutzen dürfen. Die Rechteprüfung gewährleistet, dass nur die innerhalb der Rechteverwaltung festgelegte Nutzung stattfindet. Um missbräuchliche Nutzung erkennen, nachverfolgen und ahnden zu können, ist eine Beweissicherung notwendig. Nur nach einer angemessenen Wiederaufbereitung lassen sich Ressourcen sicher mehrfach verwenden. Beispielsweise werden Speicherbereiche gezielt überschrieben, in denen sicherheitskritische Daten temporär abgelegt wurden, etwa im Rahmen der Eingabe von Passwörtern oder um Datenträger zu entschlüsseln [Eck18, S. 204].

Kurzfassung der Grundfunktionen

Die *Verfügbarkeit* von IT-Systemen und die Gewährleistung ihrer Funktionalität entwickelt sich im Zuge der *Digitalisierung* von Prozessen und Wertschöpfungsketten zunehmend zu einem der zentralen Sicherheitsaspekte. Für alle Teilsysteme und -funktionen sollte festgelegt sein, welche Priorität deren Verfügbarkeit in einem IT-Verbund hat [Eck18, S. 204].

Verfügbarkeit ist essenziell

[24] Trusted Computer System Evaluation Criteria
[25] Information Technology Security Evaluation Criteria
[26] Canadian Trusted Computer Product Evaluation Criteria
[27] Common Criteria for Information Technology Security Evaluation
[28] Application Security Verification Standard
[29] Confidentiality, Integrity, Availability
[30] Datenschutz-Grundverordnung

Prävention, Erkennung und Reaktion

Um den Verlust der schützenswerten Eigenschaften durch technische oder menschliche Fehler, gezielte Angriffe oder höhere Gewalt zu vermeiden, setzt man im Bereich der IT-Sicherheit auf die „drei Säulen“ [Abo17b, S. 5] *Prävention, Erkennung und Reaktion*. Eckert identifiziert den Security-by-Design-Ansatz als wesentlichen Bestandteil der Prävention und empfiehlt, hier den Schwerpunkt zu setzen. „Techniken der Angriffserkennung und -Reaktion (sic!) gehören deshalb ebenso zur IT-Sicherheit, wie methodische Grundlagen, um IKT-Systeme so zu entwickeln, dass sie qua Design ein hohes Maß an Sicherheit bieten. Man spricht in diesem Zusammenhang auch oft von *Secure by Design*.“ [Eck18, S. 1] „Es ist somit wichtig, Sicherheitskonzepte und Maßnahmen in die Anwendungen selber zu integrieren anstatt auf den Schutz von Firewalls und Intrusion Detection Systemen zu vertrauen.“ [Eck18, S. 738]

Konstruktionsprinzipien

Ein wesentlicher Aspekt von Design sind Konstruktionsprinzipien. Die Liste der in der Literatur erwähnten Prinzipien, die beim Entwurf sicherer IT-Systeme zu berücksichtigen sind, ist sehr umfangreich. Es gibt aber einen gemeinsamen Kern von Konstruktionsprinzipien, die häufig genannt werden [Pog17, S. 28–29] [Eck18, S. 172–173].

- *Erlaubnis-Prinzip:* Alles ist verboten, außer es ist explizit erlaubt.
- *Vollständigkeits-Prinzip:* Jeder Zugriff ist zu prüfen.
- *Minimale Rechte (need-to-know):* Jede Instanz[31] verfügt nur über die nötigen Rechte zur Erfüllung ihrer eigenen Aufgaben.
- *Funktionstrennung:* Jedes Teilsystem lässt sich eineindeutig einer Funktion zuordnen.
- *Trennung von Code und Daten:* Ausführbarer Programmcode ist von den verarbeiteten Daten getrennt [Eck18, S. 44].
- *sicheres Versagen:* Die Sicherheit des Systems bleibt erhalten, wenn Teilsysteme oder einzelne Mechanismen ausfallen, versagen oder umgangen werden [And20, S. 271].
- *sichere Voreinstellung:* Konfigurierbare Mechanismen werden mit einer sicheren Werks- oder Grundeinstellung versehen [Pog17, S. 181].
- *Benutzerakzeptanz:* Mechanismen müssen einfach zu nutzen sein und automatisch bzw. routinemäßig angewendet werden [And20, S. 89].
- *offener Entwurf:* Die Sicherheit darf nicht davon abhängig sein, dass Verfahren geheim sind.
- *Sicherheitskern:* Sicherheitsrelevante Dienste und Maßnahmen sind zusammenzufassen und abgetrennt von anderen Systemen zu realisieren.
- *Reduktion der Angriffsoberfläche:* Der Bedarf sowie die Dienste, Rechte und Nutzung werden minimiert [Mou17, S. 149] [Mül18, S. 303–307].
- *Verteidigung in der Tiefe:* Das System wird in Schichten und Segmente mit eigenen Sicherheitsmaßnahmen aufgeteilt, sodass beim Versagen einzelner Maßnahmen weitere greifen können [Abo17a, S. 8].

Zur Realisierung dieser Prinzipien haben sich bestimmte Strukturen und Komponenten etabliert, die nachfolgend beschrieben werden.

Komponenten

Einige typische Komponenten, um die Grundfunktionen entsprechend den Konstruktionsprinzipien zu erfüllen, wurden bereits in Abschnitt 2.1 näher erläutert (Schadcodebekämpfung, IDS/IPS, SIEM, Audit-Trail). Im Bereich Einbruchserkennung wird zwischen HIDS[32] und NIDS[33]

[31] Instanz bezeichnet in diesem Kontext technische Systeme, Teilsysteme, aber auch handelnde Personen, Gruppen und Organisationen.
[32] Host based Intrusion Detection System
[33] Network based Intrusion Detection System

unterschieden. Das sind einerseits Systeme, die Veränderungen auf einzelnen Servern erkennen und andererseits solche, die Anomalien in den Datenströmen des Netzwerks identifizieren. Zur *Einbruchsvermeidung* werden diese um Funktionen ergänzt, mit denen Dienste oder Netzwerkströme als Reaktion auf die Erkennung unterbrochen oder eingeschränkt werden (IPS).

Einbruchsvermeidung

Eine zentrale Komponente übernimmt die Grundfunktionen Identifikation, Authentifikation und Rechteverwaltung. Sie dient dazu, das Erlaubnis- und Vollständigkeits-Prinzip umzusetzen und kann als Teil eines Sicherheitskerns aufgefasst werden. Die Funktionen dieser Komponente werden häufig in Form eines *Verzeichnisdienstes* realisiert, auf den mit dem Standardprotokoll LDAP[34] zugegriffen werden kann. Für eine zuverlässige Authentifizierung werden nach Stand der Technik *mehrere Faktoren* eingesetzt: geheimes Wissen, Besitz und individuelle Eigenschaften. Der Faktor *Besitz* kann durch spezifische Anwendungen auf mobilen Endgeräten (*Authenticator Apps*) oder spezielle Hardware (*Token*) realisiert werden. Immer häufiger übernehmen solche Geräte auch die Funktion der sicheren Identifizierung [Eck18, S. 503].

Verzeichnisdienst

Mehrfaktor-Verfahren

Die Internetanbindung umfasst zwei Komponenten zur *Segmentierung* des Systems. Dabei vermittelt das *Gateway* zwischen Netzsegmenten niedrigster und hoher Vertrauensstufe (öffentliches Netz / Internet und privates Netz des Systems). In der einfachsten Ausbaustufe kommt ein einzelner Paketfilter zum Einsatz, der oft als *Firewall* bezeichnet wird. Stand der Technik für normale Schutzbedarfe ist ein *dreistufiges Gateway,* das aus einem äußeren und einem inneren Paketfilter (auf den OSI-Layern 3 und 4) und einem zwischengeschalteten ALG[35] (auf den OSI-Layern 5–7) besteht. Diese Struktur wird vom BSI als P-A-P-Struktur bezeichnet.

Segmentierung

mehrstufiges Gateway

Um den unerwünschten Abfluss von Daten bei höheren Schutzbedarfen der Vertraulichkeit zu vermeiden (DLP[36]), kommen zusätzlich Komponenten zum Einsatz, die den Datenverkehr auf Protokollebene normalisieren[37] (Circuit Gateways [And20, S. 718]) oder den Datenfluss vollständig auf eine Richtung beschränken (Datendioden [And20, S. 327]).

Ein wesentlicher Bestandteil des Sicherheitskerns sind Komponenten, mit denen *private Schlüssel* kryptografischer Verfahren und Zugangsdaten verwaltet werden. Dies sind entweder speziell entwickelte, meist clusterfähige Datenbanken, deren Inhalte mit starker Kryptografie verschlüsselt sind, oder spezielle Geräte, die auch physischen Schutz gegen den Angriff auf ihre verschlüsselten Inhalte bieten (HSM[38]).

Schlüsselverwaltung

Schwachstellen-Scanner überprüfen die gesamte bekannte Angriffsoberfläche regelmäßig, regelbasiert und automatisch. Sie basieren auf öffentlich verfügbaren oder entgeltlich angebotenen Datenbanken, in denen bekannte Schwachstellen und zugehörige Erkennungs- und Angriffsmuster veröffentlicht werden. Verbreitet sind rein netzbasierte Scanner wie OpenVAS, mit denen über das Netzwerk erreichbare Schnittstellen und Komponenten geprüft werden. Es gibt aber auch Lösungen, die über Software auf den Servern (*Agents*) die installierte Software jeweils lokal untersuchen können.

Schwachstellen-Scanner

Zur sicheren Löschung (*Wiederaufbereitung*) von Datenträgern wird meist Software eingesetzt, für größere Volumina gibt es auch spezielle Hardware.

Wiederaufbereitung

Eine sehr viel detailliertere Beschreibung gängiger Komponenten und Strukturen, mit denen die Grundfunktionen umgesetzt werden können, liefern Anderson [And20], Poguntke [Pog17], Wendzel [Wen18] und Eckert [Eck18]. Verfahren zum Schutz der *Supply-Chain* und zur sicheren Beschaffung von Hard- und Software beleuchtet Piller [Pil17].

[34] Lightweight Directory Access Protocol

[35] Application-Level-Gateway

[36] Data Loss Prevention

[37] Primäres Gegenmittel für die Ausnutzung von Seitenkanälen (covert channels), d. h. nicht geplanten Kommunikationsmöglichkeiten, ist Normalisierung: die von Wendzel vorgestellten *praktikablen* Techniken der Modulation erfordern alle den nicht normalisierten Gebrauch von Transportprotokollen [Wen18, S. 279–282].

[38] Hardware Security Module

2.3. IT-Grundschutz-Kompendium

Kontext IT-Grundschutz

Die vom BSI unter dem Begriff *IT-Grundschutz* seit 1994 veröffentlichten Dokumente gliedern sich in die Bereiche *Standards*, die Vorgehensweisen und Methodologie beschreiben, und *Sicherheitsanforderungen*, mit konkreten technischen und organisatorischen Vorgaben, um Informationsverbunde abzusichern. Die Erstveröffentlichung hieß IT-Grundschutzhandbuch. Es wurde 2005 durch die Standards 100–1, 100–2 und 100–3 sowie die IT-Grundschutz-Kataloge ersetzt. 2008 wurde der Standard 100–4 (Notfallmanagement) veröffentlicht. Seit 2017 wird der IT-Grundschutz erneut schrittweise modernisiert. Dabei wurden die Leitlinien der IT-Grundschutz-Kataloge in sogenannte *Bausteine* aufgegliedert, die im jährlich erscheinenden IT-Grundschutz-Kompendium zusammengefasst sind. Die Standards 100–1 bis 100–3 wurden durch die Standards 200–1 bis 200–3 ersetzt.[39] „Zudem enthalten die Umsetzungshinweise, die ergänzend zu den meisten Bausteinen veröffentlicht werden, Best Practices sowie ergänzende Hinweise, wie die Anforderungen erfüllt werden können." [BSI21, S. 15] Dort und im *Kompendium* werden ergänzend ausgewählte Teile des BSI-Standards zur Internet-Sicherheit (ISi-Reihe) referenziert. Der IT-Grundschutz ist wie in Abb. 2.3 gegliedert [BSI17, S. 25].

laufende Modernisierung

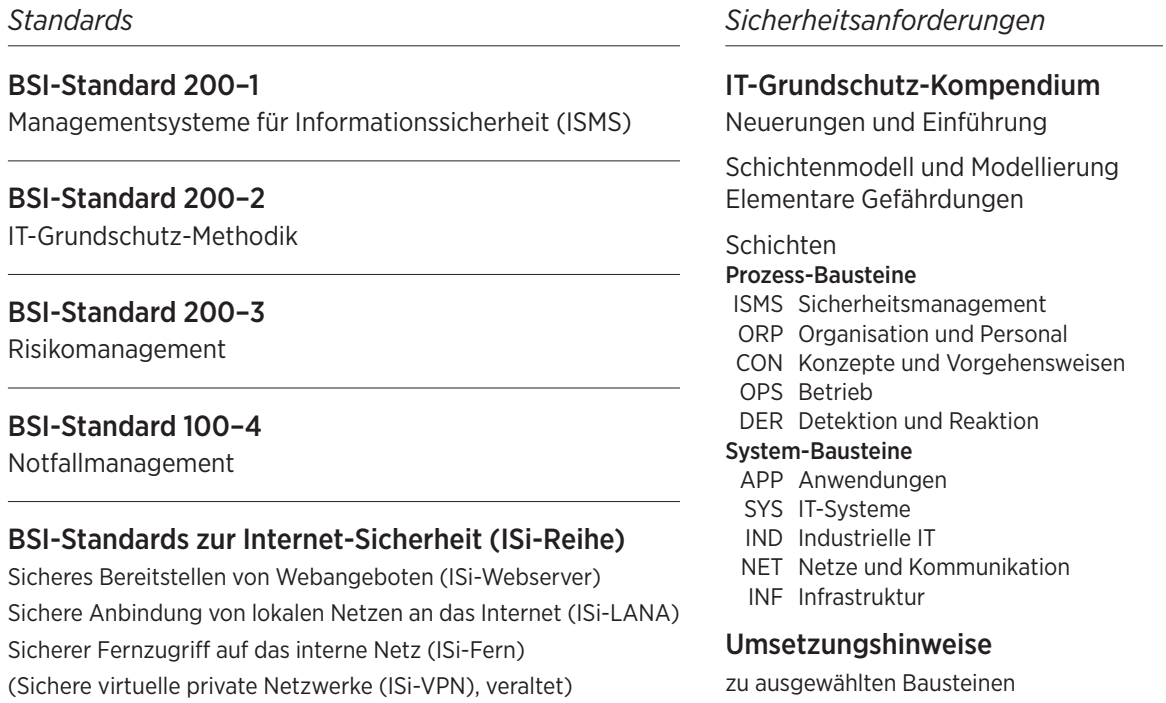


Abbildung 2.3.: Gliederung des BSI IT-Grundschutzes 2021

Motivation des IT-Grundschutz-Kompendiums

Die vorher dargelegte Idee eines umfassenden Schutzes von IT-Systemen (Abschnitt 2.2) wird im *Kompendium* aufgegriffen: „Ziel des IT-Grundschutzes ist es, einen angemessenen Schutz für alle Informationen einer Institution zu erreichen. Die IT-Grundschutz-Methodik zeichnet sich dabei durch einen ganzheitlichen Ansatz aus." [BSI21, S. 14]

ganzheitlicher Ansatz

Aufbau, Vernetzung und Betrieb von IT-Systemen bestehen oft aus ähnlichen, in der jeweils konkreten Umsetzung aber deutlich unterschiedlichen Komponenten, Strukturen und Prozes-

[39] Der Standard 200–4 wird den älteren Standard 100–4 unter dem weiter gefassten Begriff *Business Continuity Management* ablösen. Der *Community Draft* kann bis zum 30.06.2021 kommentiert werden.

sen. Daher verwendet der IT-Grundschutz „[...] das Baukastenprinzip, um den heterogenen Bereich der Informationstechnik einschließlich der Einsatzumgebung besser strukturieren und planen zu können." [BSI21, S. 14][40]

Baukasten-prinzip

Ein primäres Anliegen des *Kompendium*s ist die Reduktion des hohen Aufwands klassischer Risikoanalysen [BSI21, S. 15]. Dies wird in Form von Bausteinen realisiert, die das modularisierte Ergebnis einer generalisierten Risikoanalyse sind, die das BSI für häufig wiederkehrende Prozesse und verbreitet eingesetzte Systeme und Strukturen durchgeführt hat. So „[...] reduziert sich die Analyse auf einen Soll-Ist-Vergleich zwischen den [...] empfohlenen und den bereits umgesetzten Sicherheitsanforderungen." [BSI21, S. 15] Für erhöhte Schutzbedarfe und für Bereiche, die nicht von den Bausteinen abgedeckt werden, ist aber nach wie vor eine spezifische Risikoanalyse erforderlich.

Soll-Ist-Vergleich

Gleichzeitig soll die großflächige Verwendung „auch international" [BSI21, S. 15] erreicht werden. Die resultierenden unterschiedlichsten Schutzbedarfe sind daher dreistufig formuliert: „Die Anforderungen sind in Basis- und Standard-Anforderungen sowie Anforderungen für erhöhten Schutzbedarf unterteilt. Die Basis-Anforderungen stellen das Minimum dessen dar, was vernünftigerweise an Sicherheitsvorkehrungen umzusetzen ist. [...] Eine angemessene Sicherheit wird allerdings erst mit der Umsetzung der Standard-Anforderungen erreicht." [BSI21, S. 15] Im Gegensatz zum Standard 200–2 unterscheidet das *Kompendium* nur zwischen *normalen* und *erhöhten* Schutzbedarfen. Die Anforderungen für *hohe und sehr hohe* Schutzbedarfe sind als Empfehlungen für *erhöhte Schutzbedarfe* zusammengefasst [BSI21, S. 17]. Deshalb wird im Rahmen dieser Arbeit binär zwischen Standard-Anforderungen und Anforderungen für erhöhten Schutzbedarf unterschieden. Auch eine etwaige Zertifizierung kann nur auf diesen beiden Niveaus durchgeführt werden, wobei grundsätzlich *alle* relevanten Basis-Anforderungen erfüllt werden müssen [BSI21, S. 15]. Kurz gefasst werden Organisationen und Systeme nachfolgend als konform bezeichnet, wenn sie mindestens die Basis- und Standard-Anforderungen der für sie relevanten Bausteine des *Kompendium*s vollständig erfüllen. Im Gegensatz zur SOLL-Formulierung der Standard-Anforderungen im *Kompendium* wird im Sinne des Security-by-Design-Ansatzes keine dieser Anforderungen als verzichtbar gewertet.

universelle Verwendung

verschiedene Schutzbedarfe

konforme Systeme

Angesichts der „rasanten Entwicklungen in der Informationstechnik" [BSI21, S. 15] bleibt es schwierig, geeignete Sicherheitsmaßnahmen zu formulieren und aktuell zu halten. Daher können die Bausteine voneinander unabhängig aktualisiert und ergänzt werden. Die Gemeinschaft ist aufgefordert, fehlende Bausteine selbst zu erstellen und dem BSI zu übereignen [BSI21, S. 15]. Damit öffnet das BSI einen einfachen und effektiven Weg für Interessierte, den IT-Grundschutz aktiv weiterzuentwickeln und den tatsächlichen Bedürfnissen anzupassen.

eigene Bausteine

Grundlegender Aufbau des IT-Grundschutz-Kompendiums

Der Aufbau des *Kompendium*s wird in seinem einleitenden Abschnitt *Aufbau des IT-Grundschutz-Kompendiums* [BSI21, S. 16] erläutert. Nach Vorwort, Dankesworten und Gesamtinhaltsverzeichnis beginnen die aktuellen Ausgaben mit einem Abschnitt, der die *Änderungen* zur jeweils vorigen Ausgabe zusammenfasst.

Präambel

Synopse

Es folgt eine *Einführung*, in der die Relevanz von Informationssicherheit dargelegt wird, Aufbau und Idee des IT-Grundschutzes erläutert werden sowie der Zusammenhang zwischen dem BSI Standard 200–2 und dem *Kompendium* erklärt wird. Das Konzept der Bausteine und ihre konsequent eingehaltene Struktur wird erläutert. Die zusätzlichen Umsetzungshinweise werden eingeführt.

Konzept

[40] Das *Kompendium* ist streng modular konzipiert und als Lose-Blatt-Sammlung publiziert. Jeder Teil hat eine eigene Nummerierung, jeweils mit Seite 1 beginnend. Es gibt *keine gedruckten fortlaufenden Seitenzahlen*. Trotzdem wird in Zitaten die fortlaufende Seitenzahl der jeweils zitierten PDF-Version referenziert.

Modell

Die Gliederung der Bausteine in ein *Schichtenmodell* und ihre Einordnung als Prozess- oder System-Baustein erleichtern die *Modellierung*, bei der pro Baustein entschieden wird, ob er auf ein Zielobjekt des Informationsverbunds anzuwenden ist.

Rollen

Organisatorische *Rollen*, die für die Umsetzung ganzer Bausteine oder einzelner Anforderungen zuständig sind, werden tabellarisch aufgeführt und beschrieben [BSI21, S. 27].

In einem *Glossar* „[...] werden die wichtigsten Begriffe rund um Informationssicherheit und IT-Grundschutz erläutert." [BSI21, S. 31] Die wesentlichen für den IT-Grundschutz spezifischen Begrifflichkeiten werden im Glossar dieser Arbeit zitiert.

Elementare Gefährdungen

Elementare Gefährdungen, die Informationen im Allgemeinen und spezifisch IT-Systeme bedrohen, werden detailliert erläutert [BSI21, S. 41].

Abschließend werden die Bausteine gruppiert nach Schichten in der Reihenfolge ISMS, ORP, CON, OPS, DER (Prozesse) und APP, SYS, IND, NET, INF (Systeme) aufgeführt.

Das *Kompendium* umfasst in der 4. Ausgabe (Edition 2021) insgesamt 97 Bausteine, in denen 1522 einzelne Anforderungen beschrieben werden. Dies verdeutlicht einerseits den Anspruch, alle Bereiche der Informationssicherheit modularisiert zu erfassen, zeigt aber andererseits den Umfang der Arbeiten, die für die Absicherung komplexer Informationsverbunde nach Einschätzung des BSI notwendig sind.

Die Abb. 2.4 zeigt das Schichtenmodell des *Kompendiums* ([BSI21, S. 21]) ergänzt um die Zahl der enthaltenen Teilschichten, Bausteine und Anforderungen.

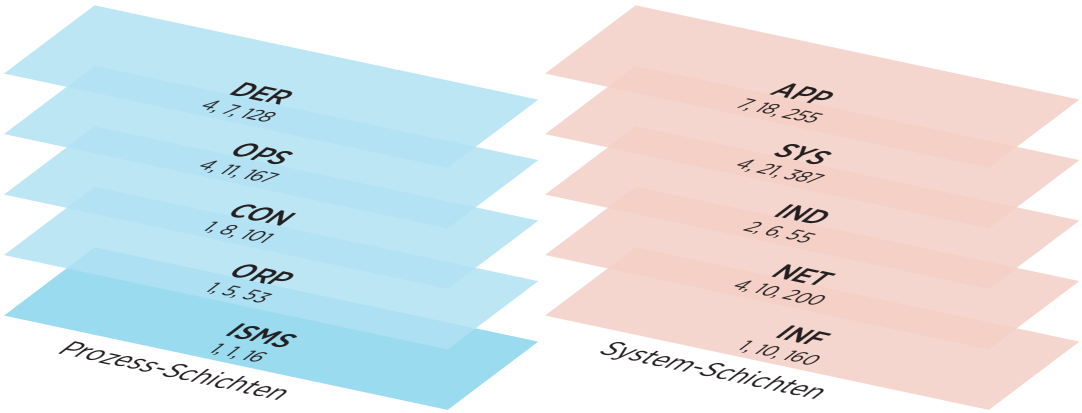


Abbildung 2.4.: Schichten mit Anzahl der Teilschichten, Bausteine und Anforderungen

Aufbau der Bausteine

In der Einführung des *Kompendiums* wird der einheitliche Aufbau der Bausteine beschrieben [BSI21, S. 17]. Einleitend wird das Zielobjekt erläutert und die Anwendbarkeit eingegrenzt, jeweils mit Verweis auf Bausteine, die nicht behandelte Aspekte aufgreifen. Die *Gefährdungslage* wird anhand von spezifischen Bedrohungen und Schwachstellen erläutert. Es folgen nach Kategorien gruppierte Anforderungen (Basis, Standard, erhöht).[41] Am Schluss stehen Verweise auf weiterführende Informationen und als Anhang eine Kreuzreferenztabelle, die den elementaren Gefährdungen die für sie relevanten Anforderungen zuordnet.

Gefährdungslage

zuständige Rolle

Jeder Baustein beschreibt eine für seine Umsetzung insgesamt zuständige organisatorische Rolle. Abweichende Zuständigkeiten werden im Titel einzelner Anforderungen vermerkt. Die Anforderungen kennzeichnen mittels der gängigen Modalverben MUSS und SOLLTE (und deren

[41] Dabei müssen die Basis-Anforderungen für alle Schutzbedarfe vollständig erfüllt sein. Die Standard-Anforderungen adressieren einen normalen Schutzbedarf. Die Anforderungen für erhöhte Schutzbedarfe sind Vorschläge nach dem Stand der Technik.[BSI21, S. 17]

Verneinungen) gemäß RFC[42] 2119, welche Teile der jeweiligen Anforderung verpflichtend oder empfohlen sind. Dabei werden alle Anforderungen für erhöhte Schutzbedarfe als Empfehlung ausgesprochen und sollten im Rahmen von *Risikoanalysen* verwendet werden [BSI21, S. 19].

Risikoanalyse

Die Bausteine und Anforderungen sind systematisch und unveränderlich nummeriert. So sind diese über alle Editionen hinweg *eindeutig* identifizierbar. Anforderungen werden weiterhin mit der erstmalig veröffentlichten Kennung aufgeführt, auch wenn sie in überarbeiteten Editionen entfallen sind oder die Zuordnung zu einer Schutzbedarfskategorie geändert wurde.

eindeutige Kennung

Anwendung des IT-Grundschutz-Kompendiums

Das *Kompendium* kann eigenständig ohne weitere Dokumente verwendet werden. Zu ausgewählten Bausteinen veröffentlicht das BSI ergänzende *Umsetzungshinweise* separat.[43]

Umsetzungshinweise

Es ist empfehlenswert, zum Erstellen einer Sicherheitskonzeption die vorgegebenen Verfahren des BSI-Standards 200–2 zu verwenden. Dieser definiert drei unterschiedliche Vorgehensmodelle, die aufeinander aufbauen und für unterschiedliche Schutzbedarfe geeignet sind.

Die *Basis-Absicherung* ermöglicht die „[…] grundlegende Erst-Absicherung über alle Geschäftsprozesse und Fachverfahren einer Institution hinweg […].“ [BSI17, S. 74] Sollen zunächst nur die schützenswertesten Zielobjekte oder nur ein kleiner Aspekt einer gesamten Organisation oder eines Informationsverbunds gezielt abgesichert werden, empfiehlt sich die *Kern-Absicherung*. Sie ähnelt vom Vorgehen der *Standard-Absicherung*, mit der ein Verfahren beschrieben wird, um für einen Informationsverbund ein normales Sicherheitsniveau zu erzielen und in einem Zyklus dauerhaft aufrechtzuerhalten [BSI17, S. 147].

Basis-Absicherung

Kern-Absicherung

Standard-Absicherung

Weder Basis- noch Kern-Absicherung bieten einer Organisation umfassenden Schutz. Auf beide folgt zeitlich nachgelagert eine Standard-Absicherung. Nach einer Basis-Absicherung kann als Zwischenschritt eine Kern-Absicherung stehen und umgekehrt [BSI17, S. 132].

abgestuftes Vorgehen

Standard- und Kern-Absicherung lassen sich nach ISO 27001 zertifizieren. Den Zusammenhang zwischen Schutzniveau, Schutzumfang, Vorgehensweise und Zertifizierung zeigt Abb. 2.5.

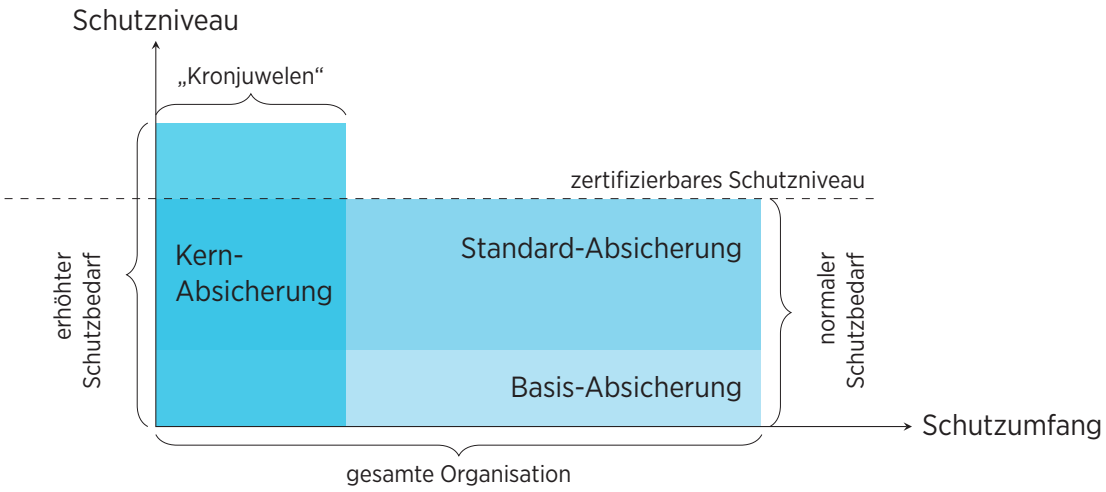


Abbildung 2.5.: Schutzniveau und -umfang der Basis-, Standard- und Kern-Absicherung

Der Standard 200–2 gibt eine umfassende *Entscheidungshilfe*, welche Vorgehensweise in welchen Situationen am besten geeignet ist. Die *Managemententscheidung* wird mit kurzen Pro- und Contra-Argumenten unterstützt [BSI17, S. 99–100]. Demnach ist für diese Arbeit die

Entscheidungshilfe

[42] Request for Comment

[43] Mit der jährlichen Überarbeitung des *Kompendiums* hält die Aktualisierung der Umsetzungshinweise leider nicht Schritt. Deren Anpassung liegt oft eine oder zwei Editionen hinter der des *Kompendiums*.

Basis-Absicherung wenig geeignet, denn sie zielt auf einen Schutz in der Breite einer Organisation ab und kann nicht nach ISO 27001 zertifiziert werden. Umfang und Aufwand einer Standard-Absicherung sind hoch, ihr angestrebtes Ergebnis ist „[...] ein gleichmäßiges Sicherheitsniveau über die gesamte Institution [...]." [BSI17, S. 100].

schrittweise Absicherung

Am besten geeignet scheint daher das Vorgehen zur Kern-Absicherung, die sich grundsätzlich nach ISO 27001 zertifizieren lässt. Es „[...] kann in einem ersten Schritt zunächst der kritischste Geschäftsprozess abgesichert werden, um in weiteren Schritten wahlweise die nächsten kritischen Geschäftsprozesse abzusichern oder für alle Bereiche der Institution die Basis- oder Standard-Absicherung zu beginnen." [BSI17, S. 97]

Dies kommt einerseits der Planungssituation entgegen, die primär auf den strukturellen Entwurf einer Systemarchitektur abzielt, andererseits der skizzierten Struktur von Container-Plattformen, deren Absicherung stufenweise erfolgen kann.

Kern-Absicherung nach Standard 200–2

Kronjuwelen

Die Kern-Absicherung zielt auf die besonders schützenswerten Informationen und Geschäftsprozesse einer Organisation ab. Diese werden als *Kronjuwelen* bezeichnet. Ihr Verlust oder ihre Beeinträchtigung sind definitionsgemäß mittelbar oder direkt existenzbedrohend für die Organisation [BSI17, S. 140]. Daher wird davon ausgegangen, dass der im Rahmen einer Kern-Absicherung betrachtete Schutzbedarf hoch oder sehr hoch ist [BSI17, S. 139].

Der grundlegende Ablauf einer Kern-Absicherung besteht aus folgenden Schritten:

- Festlegung des Geltungsbereichs
- Identifikation und Festlegung der kritischen Assets (Kronjuwelen)
- Strukturanalyse
- Schutzbedarfsfeststellung
- Modellierung
- IT-Grundschutz-Check
- Risikoanalyse und weiterführende Sicherheitsmaßnahmen
- Umsetzung und weitere Schritte

Diese Abfolge liest sich so, als wäre sie nur auf *bestehende* Informationsverbunde anwendbar. Der Standard 200–2 zeigt jedoch auf, dass bei einem Planungsvorhaben das Ergebnis des Schritts *Modellierung* kein Prüfplan, sondern ein Entwicklungskonzept ist [BSI17, S. 209].

IT-Grundschutz-Check

In diesem Szenario entfällt der *IT-Grundschutz-Check*. Dieser bezeichnet einen Soll-Ist-Vergleich, mit dem die Differenz zwischen erforderlichen und erfüllten Sicherheitsanforderungen ermittelt wird.

Für erhöhte Schutzbedarfe und Teile des Informationsverbunds, die nicht vom *Kompendium* abgedeckt sind, sind eine Risikoanalyse und die Entwicklung weiterer Sicherheitsmaßnahmen notwendig. Da dies trotz der Vereinfachung beim Vorgehen nach dem BSI-Standard 200–2 aufwendig und langwierig sein kann, ist es oft ratsam, erst nach der Umsetzung weiterer Schritte die Risikoanalyse durchzuführen [BSI17, S. 230]. Wie diese für eine Container-Plattform aussehen kann, zeigen im Schnelldurchlauf Haar und Buchmann anhand der ersten Fassung des Bausteins SYS.1.6: *Container* und erarbeiten die beiden Gefährdungen[44] *Containerausbruch* und *manipulierte Konfigurationsdateien* [HB19, S. 487].

[44] Entsprechende Anforderungen, um mit den Gefährdungen umzugehen, wurden in der zweiten Fassung des Bausteins SYS.1.6: *Container* (Community Draft) aufgenommen.

3. Kriterien für die Systemarchitektur

3.1. Stand der Technik und Vorgehen

Gegenüber dem Stand der Technik bei der Absicherung klassischer SaaS-Architekturen (oftmals Web-Anwendungen nach dem 3-Tier- oder n-Tier-Modell) sind bei aktuellen Container-Plattformen deutliche Defizite festzustellen. Dies liegt einerseits in ihrer deutlich komplizierteren Struktur begründet. „Der Grad der Komplexität auf jedem einzelnen Layer des Turms ist gewaltig. Und wie wir hinlänglich wissen, ist Komplexität der Feind jedweder Zuverlässigkeit." [Lie19, § 25.1] Andererseits wurden die Container-Plattformen in den letzten Jahren rasant weiterentwickelt. Dabei liegt der Fokus der Entwicklung nach wie vor auf Funktionalität und Skalierung. Sicherheit und Stabilität werden fast ausschließlich nachgelagert betrachtet. „Was die Schnelllebigkeit angeht: Sicher, die Kubernetes-Entwickler haben hier und da endlich einmal die Handbremse angezogen [...] Aber es ist immer noch einiges an Bewegung unter der Haube, [...] zu volatil und damit potenziell fehleranfällig und zeitraubend." [Lie21, S. 1245]

Dies betrifft ganz besonders frei verfügbare Varianten, mit teilweise verheerenden wirtschaftlichen Folgen. „Was fehlt, ist bei vielen Unternehmen leider immer noch das elementare Verständnis dafür, dass ein »kostenloser« Vanilla-Kubernetes-Cluster [...] nach wie vor die ultimative Zeit-, HR- und damit Geldverbrennungsmaschine ist [...]." [Lie21, S. 1246]

Die Defizite betreffen folgende Bereiche:

- *Struktur:* Die bewährte Aufteilung in Zonen nach Aufgabe und Schutzbedarf sowie die Segmentierung auf physischer Ebene (Layer 1) ist aufgeweicht oder verschwindet im einheitlichen Container-Cluster-Konzept vollends.
- *Herkunft:* Das an sich sinnvolle Konzept der Wiederverwendbarkeit und *Stapelung* von Container-Images führt oft allein aus Bequemlichkeit (seltener aufgrund gezielter wirtschaftlicher Überlegung) zur unkontrollierten Nutzung öffentlicher Registries und Container-Images. Angriffe auf die *Supply-Chain* und Skandale um bösartige Container, mit denen fremde Infrastrukturen gekapert oder illegal und unbemerkt genutzt werden, sind die Folge.
- *Stabilität:* Die Entwicklungszyklen weitverbreiteter Software zur Orchestrierung sind extrem kurz[1]; zentrale Werkzeuge haben immer noch keine vollständig stabilen APIs. Nicht nur zwischen Hauptversionen gibt es oft Änderungen, die nicht abwärtskompatibel sind (*breaking changes*).
- *Verschlüsselung:* Die Verschlüsselung der Datenströme ist oft mit erheblichem Mehraufwand verbunden und fast nie in der Standardeinstellung aktiviert, obwohl sie spätestens seit Inkrafttreten der DSGVO regelmäßig erforderlich ist. Die meisten der gängigen Overlay-Netzwerke unterstützen die Verschlüsselung auf unteren Netzwerk-Layern nicht oder nur bedingt. Manche Lösungen terminieren die Verschlüsselung außerhalb des Clusters und lassen die internen Datenströme ungeschützt [Lie21, S. 1044].
- *Vorgehensmodell:* Die Software wird nicht nach erkennbar sicheren Vorgehensmodellen entwickelt: Funktionalität (*Features*) und Skalierbarkeit haben Vorrang vor Sicherheit und Stabilität.

Mit dem Security-by-Design-Ansatz und dem IT-Grundschutz als Werkzeug wird versucht, eine Architektur zu entwerfen, die einige der genannten Probleme überwindet.

Leser, die mit der Absicherung nach BSI-Grundschutz vertraut sind, werden die Reihenfolge des weiteren Vorgehens ungewöhnlich finden. Der Standard 200–2 geht implizit von einer Abfolge aus, die primär für die Absicherung *bestehender* Systeme oder Entwürfe geeignet ist:

[1] Kubernetes hat seit der Erstveröffentlichung am 10. Juli 2015 bis zum 8. Dezember 2020 zwanzig Hauptversionen veröffentlicht. Der Support pro Version wurde erst mit der Version 1.19 von neun Monaten auf ein Jahr erhöht.

Informationsverbund

Strukturanalyse

Modellierung

Risikoanalyse

Für ein vorliegendes System oder Design wird festgelegt, welcher Teil der Struktur und der zugehörigen Prozesse abzusichern ist (*Informationsverbund*). Daraufhin wird eine detaillierte *Strukturanalyse* des Informationsverbunds vorgenommen. Basierend auf der Strukturanalyse werden die Schutzbedarfe für die identifizierten Prozesse und Systeme des Informationsverbunds definiert. Anschließend werden die Anforderungen entsprechend der vorher festgelegten Schutzbedarfe, der *vorgegebenen* Strukturen und Prozesse anhand der Bausteine *modelliert.* Als Nächstes wird überprüft, ob die Anforderungen erfüllt sind und als Ergebnis Anforderungen entweder als verzichtbar gewertet (ausgeschlossen) oder entsprechende Maßnahmen zu ihrer Umsetzung formuliert. Abschließend folgt für höhere Schutzbedarfe eine zusätzliche *Risikoanalyse* und die Ableitung weiterer Sicherheitsmaßnahmen.[2] Falls Teile des Informationsverbunds nicht hinreichend durch Bausteine beschrieben sind, müssen diese beiden Schritte in jedem Fall durchgeführt werden.

vorgegebenes Schutzniveau

geeignetes Design

Dieses Vorgehen steht in direktem Widerspruch zum Security-by-Design-Ansatz. Hier geht es darum, *zuerst* festzulegen, welches angestrebte *Schutzniveau* die Funktionen und Geschäftsprozesse benötigen, die durch das entwickelte System erfüllt oder unterstützt werden sollen. In einem zweiten Schritt werden die notwendigen funktionalen Komponenten und Prozesse identifiziert. Für diese werden *alle* Anforderungen und bewährten Verfahren gesammelt, mit denen das angestrebte Schutzniveau erreicht werden kann. Aus dieser Information wird dann ein Design abgeleitet und es werden *geeignete* Implementierungen für die notwendigen Komponenten ausgewählt.

Basierend auf der Abfolge aus dem Standard 200–2 wird ein ähnlicher Ablauf verwendet, bei der die Strukturanalyse durch ein gezieltes Strukturdesign ersetzt wird und der Entwurf nicht Ausgangspunkt, sondern Ergebnis des Verfahrens ist (Abb. 3.1). Die einzelnen Teilschritte orientieren sich an der Methodik der Kern-Absicherung.

Standard 200–2	Security-by-Design-Ansatz
• bestehender Entwurf • Festlegung des Geltungsbereichs • Strukturanalyse • Schutzbedarfsfeststellung • Modellierung der Anforderungen • IT-Grundschutz-Check • Risikoanalyse • Maßnahmen	• Festlegung des Geltungsbereichs • Schutzbedarfsfeststellung • Modellierung der Anforderungen • Strukturdesign • Entwurf • Überprüfung

Abbildung 3.1.: Vorgehensweise BSI Standard 200–2 und Security-by-Design-Ansatz

[2] Auf diese Art des Vorgehens bezieht sich die Bemerkung von Eckert, dass die gängigen Verfahren vorwiegend auf die Absicherung bestehender Systeme abzielen, aber *nicht* auf die systematische Konstruktion inhärent sicherer Strukturen und Systeme [Eck18, S. 172, 174].

3.2. Modellierung gemäß Kern-Absicherung

Kritische Assets (Kronjuwelen)

Es kann regelmäßig davon ausgegangen werden, dass der Betrieb vermieteter Software die einzige Einnahmequelle eines SaaS-Anbieters darstellt. Weiterhin kann unterstellt werden, dass die Daten der Nutzer sensibel sind und regelmäßig auch personenbezogene Daten verarbeitet werden. Die zum Betrieb des SaaS-Angebots gehörigen Informationen und Prozesse sind daher die *Kronjuwelen* eines SaaS-Anbieters und werden wie in Abb. 3.2 modelliert.

Prozesse	Informationen
• Einrichtung und Aufbau der Plattform auf gegebener Hardware-Infrastruktur • Aktualisierung der Plattform • Entwicklung der Software • Aktualisierung der Software	• Quellcode der angebotenen Software • zum Betrieb notwendige Dokumentation[3] • eingesetzte Fremdsoftware (Bibliotheken, Betriebssysteme) • Konfigurationsdaten • administrative Zugangsdaten • vom System verarbeitete Kundendaten

Abbildung 3.2.: Kritische Assets (Kronjuwelen)

Geltungsbereich und Informationsverbund

Der Informationsverbund wird für diese Arbeit möglichst klein gefasst, um ihren Umfang im vorgegebenen Rahmen zu halten. Der Fokus liegt auf den Komponenten, die spezifisch für Container-Plattformen sind, auch wenn in einem realen Betrieb weitere Komponenten zwingend erforderlich und dem Informationsverbund üblicherweise zuzurechnen sind. Dieser wird in Abb. 3.3 umrandet dargestellt und folgendermaßen formuliert:

Informationsverbund

Der Informationsverbund im Rahmen dieser Arbeit umfasst die für den Betrieb eines SaaS-Angebots erforderlichen Teile einer Plattform, die spezifisch das Betriebsmodell Container-Plattform betreffen. Das sind alle benötigten Prozesse und Systeme, um Container-Images aus Quellcode zu erzeugen, diese Container-Images zu prüfen und zu speichern und sie innerhalb eines Rechnerverbunds (Container-Cluster) als Gesamtsystem lauffähig zu halten und regelmäßig zu aktualisieren. Die Anbindung des Systems an öffentliche Netzwerke, um den angebotenen Dienst Dritten bereitzustellen, ist ebenfalls Teil des Informationsverbunds.
Der Informationsverbund umfasst weder die Hardware, auf der das System betrieben wird, noch die zur Überwachung oder Fernwartung des Systems notwendigen Komponenten und Prozesse.

Im vorgeschlagenen Aufbau wird die komplette Hardware-Infrastruktur von einem entsprechend zertifizierten internen oder externen Dienstleister betrieben, der die Kriterien nach den Bausteinen der Schicht INF vollständig erfüllen kann. Die physische Infrastruktur ist insgesamt *nicht Gegenstand* dieser Arbeit. In diesem Betriebsmodell sind VPN-Anbindungen üblich, um auf die administrativen Schnittstellen des Systems zugreifen zu können. Dieser Bereich ge-

Abgrenzung

[3] Die erforderliche Dokumentation kann innerhalb des Informationsverbundes identisch zum Quellcode verwaltet werden und wird entsprechend behandelt.

hört nicht zum betrachteten Informationsverbund.[4] Die hinreichende Absicherung mobiler Arbeitsplätze und Arbeitsgeräte wird ebenfalls stillschweigend vorausgesetzt, obwohl das Thema durchaus nicht trivial ist.

Generische Komponenten, die auch für andere Betriebskonzepte notwendig wären, werden aus der Betrachtung ausgeschlossen. Dies betrifft den gesamten Bereich *Überwachung*. In der Modellierung wird unterstellt, dass der Informationsverbund über zwei rein schreibende

Schnittstellen

Schnittstellen die notwendigen Informationen dorthin übermittelt: über eine zentrale Logging-Schnittstelle wird die gesamte Protokollierung erfasst, an einer Monitoring-Schnittstelle werden laufend Sensordaten aus dem Informationsverbund übernommen.[5] Dadurch braucht auch

Archivierung

die sichere und ordnungsgemäße *Archivierung* der Protokolle nicht betrachtet werden, die im Baustein OPS.1.1.5: *Protokollierung* empfohlen wird.

Die Softwareentwicklung selbst ist ebenfalls nicht Gegenstand dieser Arbeit.

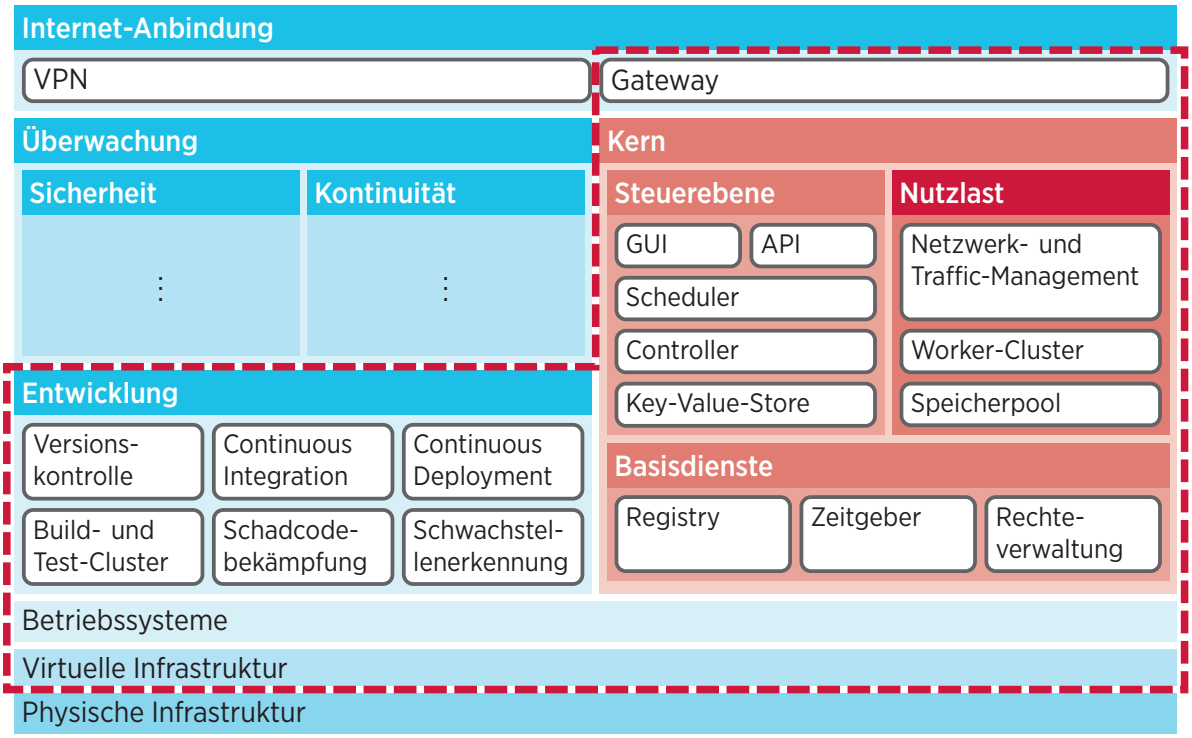


Abbildung 3.3.: Informationsverbund

Ermittlung der relevanten Bausteine

Es folgt die Bestimmung der relevanten Bausteine über ein Ausschlussverfahren. Im Gegensatz zur Vorgehensweise des Standards 200–2, bei der die Modellierung mit den Bausteinen auf einer Strukturanalyse des Systems aufbaut, werden hier irrelevante Bausteine systematisch ausgeschlossen, um dann aus den verbleibenden Anforderungen geeignete Prozesse, Strukturen und Systeme abzuleiten. In der tabellarischen Übersicht werden nur die Kürzel der Bausteine genannt. Eine vollständige Tabelle befindet sich im Anhang (Abschnitt B.3). Die Tabellen 3.1 und 3.2 zeigen, ob Bausteine nicht (n) oder mindestens teilweise (j) anwendbar sind.

[4] Oft bieten Hosting-Provider eine VPN-Anbindung als *managed Service* und integralen Bestandteil der Betriebsumgebung mit an.

[5] Diese starke Trennung besitzt gleichzeitig Vorteile für die Anpassung auf höhere Schutzbedarfe: das logische Netzsegment kann einfach in eine oder zwei physisch abgesetzte Zonen umgewandelt werden; im Extremfall für sehr hohe Schutzbedarfe kann die unidirektionale Kopplung über eine *Datendiode* erfolgen.

Bausteine	j/n	Erläuterung
ISMS.1 ORP.1–3,5 DER.2–4	n	Es wird als gegeben vorausgesetzt, dass der Betrieb der Plattform in ein funktionierendes und konformes ISMS eingebettet ist, das heißt, dass die organisatorischen Voraussetzungen für einen sicheren Systembetrieb und die entsprechenden Anforderungen vollständig erfüllt sind.
ORP.4	j	Wird herangezogen, um Komponenten und Strukturen des Identitäts- und Berechtigungsmanagements (nicht aber organisatorische Maßnahmen) zu bewerten und umzusetzen.
CON.1	j	Erhöhte Schutzbedarfe erfordern kryptografische Absicherung der Daten.
CON.2	n	Die einzige Anforderung an den Datenschutz ist organisatorischer Art. Obwohl im Rahmen eines SaaS-Angebots regelmäßig davon ausgegangen werden kann, dass personenbezogene Daten verarbeitet werden, wird der Baustein daher nicht betrachtet.
CON.3	j	Eine Datensicherung ist im Anwendungsszenario unverzichtbar.
CON.4–5	n	Die Bausteine wurden umbenannt.
CON.6	n	Löschen und Vernichten von Datenträgern obliegt dem Hosting-Anbieter.
CON.7,9	n	Die Bausteine definieren rein organisatorische Anforderungen.
CON.8	j	Nur technische Anforderungen der Softwareentwicklung werden verwendet.
CON.10	n	Softwareentwicklung selbst ist nicht Gegenstand dieser Arbeit.
OPS.1.1.1	n	Der Baustein wurde nicht ins *Kompendium* aufgenommen.
OPS.1.1.2–4	j	Betrifft die Prozesse der *Kronjuwelen*.
OPS.1.1.5	n	Protokollierung ist kein Bestandteil des Informationsverbunds.
OPS.1.2.1,3	n	Die Bausteine wurden nicht ins *Kompendium* aufgenommen.
OPS.1.2.2	n	Langzeit-Archivierung ist kein Bestandteil des Informationsverbunds.
OPS.1.2.4–5	n	Telearbeit und Fernwartung sind nicht Bestandteil des Informationsverbunds.
OPS.2.1	n	Das ausgelagerte Hosting ist kein Bestandteil des Informationsverbunds.
OPS.2.2	n	Der Nutzungskontext *hybride Cloud* wurde von vornherein ausgeschlossen.
OPS.3.1	j	Die angebotene Dienstleistung stellt für die Nutzer Outsourcing dar.
DER.1	n	Überwachung ist kein Bestandteil des Informationsverbunds.

Tabelle 3.1.: Verwendete und verworfene Prozess-Bausteine

Ohne die rein die organisatorischen Anforderungen zu berücksichtigen, wären für alle skizzierten Komponenten einer Container-Plattform (Abb. 2.2) insgesamt 268 Anforderungen aus 27 Bausteinen relevant. Aus dem ebenfalls angewendeten *Community Draft 2* des Bausteins SYS.1.6: *Container* kommen weitere 36 Anforderungen hinzu.

Für den reduzierten Informationsverbund müssen aus 24 Bausteinen 245 Anforderungen erfüllt werden, davon 97 Basis-Anforderungen, 94 Standard-Anforderungen und 54 Anforderungen für erhöhte Schutzbedarfe (Abb. 3.4). Eine detaillierte Liste der verwendeten Anforderungen ist im Anhang aufgeführt (Abschnitt B.3). Um dem geplanten Umfang dieser Arbeit gerecht zu werden, werden die Anforderungen für Modellierung und Entwurf weiter zusammengefasst.

Schutzbedarfsanalyse

Nach dem Ablauf der Kern-Absicherung erfolgt erst *nach* einer eingehenden Strukturanalyse des Informationsverbunds seine Schutzbedarfsanalyse. Diese wird hier vorgezogen, weil innerhalb der verwendeten Vorgehensweise aus den Anforderungen und den Schutzbedarfen geeignete Systeme, Strukturen und Prozesse abgeleitet werden sollen.

Bausteine	j/n	Erläuterung
APP.1.1,2,4	n	Die Bausteine adressieren Anwendungen auf Endgeräten.
APP.1.1.3	n	Der Baustein wurde nicht veröffentlicht.
APP.2.1	j	Ein zentraler Verzeichnisdienst ist für die Standardabsicherung erforderlich.
APP.2.2–3	n	Adressiert spezifische Produkte (Verzeichnisdienste).
APP.3.1–2	j	Typische SaaS-Angebote basieren auf Web-Services.
APP.3.3–4	j	Fileserver und Samba werden im Informationsverbund nicht verwendet.
APP.3.5	n	Der Baustein wurde nicht veröffentlicht.
APP.3.6	j	Der Controller der Steuerebene benötigt interne Namensauflösung (DNS).
APP.4.1,4,5	n	Die Bausteine wurden nicht veröffentlicht.
APP.4.2,6	n	SAP ERP-Software wird nicht eingesetzt.
APP.4.3	j	Es ist davon auszugehen, dass Komponenten relationale Datenbanken verwenden.
APP.5	n	Die Bausteine adressieren E-Mail-Clients und -Server, nicht MTA.
APP.6	j	Allgemeine Anforderungen an Software.
APP.7	n	Vereinfachend wird angenommen, dass nur eigenes Personal die Software des SaaS-Anbieters erstellt.
SYS.1.1	j	Es werden Server verwendet.
SYS.1.2	n	Windows-Server sind keine geeignete Wahl für Container-Plattformen.
SYS.1.3	j	Linux-Server werden verwendet (andere UNIX-Server sind möglich).
SYS.1.4	n	Der Baustein wurde nicht veröffentlicht.
SYS.1.5	j	Die Fachliteratur empfiehlt eine klassische Virtualisierungsschicht.
SYS.1.6	j	Der Entwurf des Bausteins adressiert Container-Plattformen.
SYS.1.7	n	IBM Z-Systeme sind für kleinere SaaS-Anbieter in der Anschaffung zu teuer.
SYS.1.8	j	Eine zentrale Speicherlösung ist Bestandteil des *Kern*s.
SYS.2	n	Die Bausteine adressieren Client-Systeme.
SYS.3	n	Die Bausteine adressieren mobile Endgeräte.
SYS.4	n	Die Bausteine adressieren sonstige Systeme, z. B. Drucker und IoT-Geräte.
IND	n	Diese Schicht adressiert industrielle IT für Fertigungsprozesse.
NET.1.1	j	Die Netzarchitektur hat wesentlichen Einfluss auf die Sicherheit des Systems.
NET.1.2	j	Netzmanagement ist Teil der Betriebsprozesse im Informationsverbund.
NET.2.1–2	n	Drahtlose Netze werden im Informationsverbund nicht eingesetzt.
NET.3.1	n	Ein zertifizierter Dienstleister betreibt Hardware-Switches und -Router.
NET.3.2	j	Firewalls sind Teil der Netzsegmentierung.
NET.3.3	n	Fernwartung und VPN sind nicht Teil des Informationsverbunds.
NET.4.1–3	n	Die Bausteine adressieren Telekommunikationsanlagen.
INF	n	Ein zertifizierter Dienstleister betreibt die Hardware-Infrastruktur.

Tabelle 3.2.: Verwendete und verworfene System-Bausteine

Die Schutzbedarfe für die einzelnen Objekte werden aus denen der Informationen und Prozesse abgeleitet. Der jeweils festgestellte Schutzbedarf vererbt sich auf die zur Verarbeitung genutzten Objekte. Gelten für ein Objekt unterschiedliche Schutzbedarfe, weil es verschiedene Informationen oder Prozesse verarbeitet, wird das Maximum-Prinzip angewendet [BSI17, S. 181].

Gemäß der Abgrenzung der Schutzbedarfe *normal*, *hoch* und *sehr hoch* nach BSI-Standard 200–2 ist der Schutzbedarf für die modellierten Kronjuwelen als *hoch* einzustufen. „Insgesamt gilt: Im Schadensfall tritt Handlungsunfähigkeit zentraler Bereiche der Institution ein. Schäden haben erhebliche Beeinträchtigungen der Institution selbst oder betroffener Dritter zur Fol-

ge.“ [BSI17, S. 91] Nach den beiden Prinzipien der Vererbung und der Maximierung ergibt sich für alle Komponenten (Hardware, Software) und Prozesse insgesamt ein hoher Schutzbedarf bezüglich Vertraulichkeit, Integrität und Verfügbarkeit.

Es folgen die Modellierung mit den Bausteinen und der Entwurf des Systems mit einem einheitlich als *hoch* identifizierten Schutzbedarf.

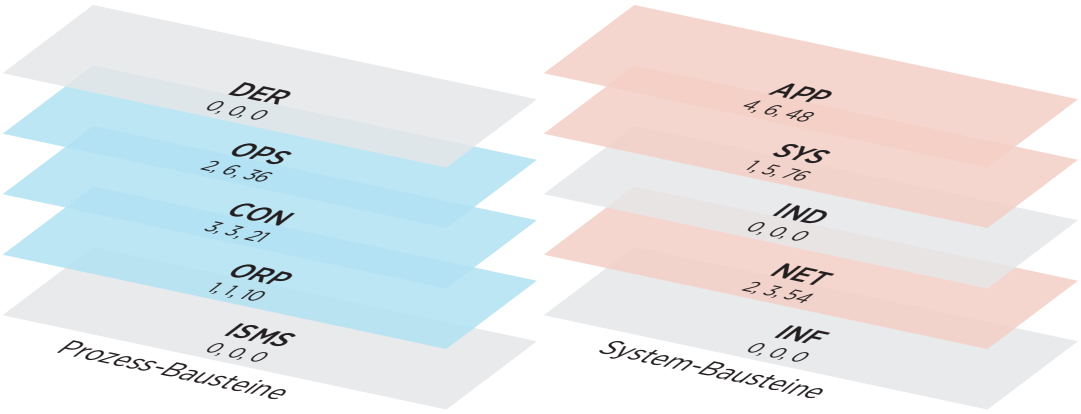


Abbildung 3.4.: Relevante Bausteine und Anforderungen

3.3. Kriterien für die Struktur

Ausgehend von in der Literatur einschlägigen Entwurfsmustern und den relevanten Bausteinen werden für die Auslegung der Systemstruktur Kriterien erarbeitet, nach denen unterschiedliche Ansätze und Referenz-Architekturen bewertet und ausgewählt werden können. Die Kriterien aus dem *Kompendium* werden sinngemäß zusammengefasst, falls dies möglich und sinnvoll scheint.

Segmentierung

Segmentierung

Zwei wesentliche Aspekte der Strukturierung eines Informationsverbundes sind dessen technische *Segmentierung* auf allen Layern sowie die organisatorische Trennung in Teilaufgaben. Beide stellen zentrale Prinzipien von *Security-by-Design* dar und werden im BSI-Grundschutz aufgegriffen und eingehend behandelt.

Dies steht in diametralem Widerspruch zu einigen zentralen Ideen von Container-Plattformen und den damit verbundenen Entwicklungs- und Betriebskonzepten.[6]

Ein Grundproblem der gängigen Netzwerk-Stacks und -Architekturen ist mangelnde Authentifizierung der Kommunikationspartner auf allen Ebenen der OSI- bzw. TCP[7]/IP[8]-Netzwerk-Layer. Eine Kern-Anforderung an die Struktur ist daher, dass die Netzwerkübergänge zahlenmäßig klein und durchgängig beidseitig authentifiziert sind. „Die Absicherung von Netzwerken muss schichten- und protokollweise erfolgen. [...] Viele Angriffe basieren darauf, dass Angreifer die Netzwerkidentität (etwa Sender-MAC-Adresse oder Sender-IP-Adresse) eines Dritten übernehmen, um sich somit über Beschränkungen hinwegsetzen oder Daten mitlesen zu können, die nicht für sie bestimmt sind. Derlei Angriffe können mit Authentifizierungsverfahren und signierten Adressen erschwert werden.“ [Wen18, S. 269]

[6] Beispielhaft: der *Dev-Ops*-Ansatz oder auch *you build it, you run it*, bei dem die Entwickler gleichzeitig wesentliche Teile des Betriebs übernehmen sollen; große Container-Cluster, in denen Komponenten mit sehr unterschiedlichen Schutzbedarfen auf gleicher Hardware und in gleichen Kontexten laufen; *Infrastructure as Software*, bei der die kontrollierte physische Trennung teilweise schon konzeptionell aufgehoben wird.

[7] Transmission Control Protocol

[8] Internet Protocol

Das *Kompendium* definiert viele Anforderungen an die Strukturierung des Netzwerks von Informationsverbunden und gibt klare Vorgaben, die gerade im Bereich von Container-Clustern wenig oder gar nicht berücksichtigt werden. Daher liegt ein starker Fokus auf der Frage, wie die Anforderungen an das Netzwerkdesign im Rahmen einer Container-Plattform umgesetzt werden können (Tabelle 3.3).

Details zur Strukturierung und Absicherung von Netzwerken sowie Verweise auf weiterführende Literatur liefert Wendzel [Wen18].

Hochverfügbarkeit

Anforderungen an die Hochverfügbarkeit werden insgesamt nur für erhöhte Schutzbedarfe gestellt. Sie betreffen die Bereiche Netzwerke, Systeme und (Geo-)Redundanz (Tabelle 3.4).

Kürzel	Beschreibung	Basis	Standard	erhöht
SEG.A1	sorgfältige Planung, Dokumentation und Review	*NET.1.1.A2,A13*	*NET.1.1.A16,A17* *NET.1.1.A22,25*	*keine*
SEG.A2	physische, mit Firewalls realisierte Trennung in mindestens drei Zonen	*NET.1.1.A4,A8* *NET.1.1.A10*	*keine*	*keine*
SEG.A3	Internetanbindung mit einer P-A-P-Struktur	*NET.1.1.A4,A12*	*APP.3.6.A16* *NET.1.1.A18* *NET.3.2.A16*	*keine*
SEG.A4	Strikte Trennung administrativer und datenführender Datenströme	*OPS.1.1.2.A6* *SYS.1.5.A5* *SYS.1.8.A4* *NET.1.2.A9* *NET.3.2.A6*	*SYS.1.6.A23* *NET.1.1.A21* *NET.1.2.A21* *NET.3.2.A18*	*OPS.1.1.2.A16*
SEG.A5	starke funktionale Trennung in Netzsegmente	*SYS.1.6.A2* *NET.1.1.A9,A11*	*APP.3.1.A11* *APP.3.6.A14* *SYS.1.5.A9* *NET.1.1.A19–A21* *NET.1.1.A23,A24* *NET.1.2.A29*	*NET.1.1.A33* *NET.3.2.A30*
SEG.A6	Verschlüsselung und Authentifizierung der Datenströme zwischen Komponenten	*NET.1.1.A7* *NET.1.2.A10*	*APP.3.1.A11*	*keine*
SEG.A7	Aufgabentrennung der Komponenten	*SYS.1.6.A4*	*SYS.1.6.A12*	*NET.1.1.A35*

Tabelle 3.3.: Anforderungen an die Segmentierung

Kürzel	Beschreibung	erhöht
HAV.A1	hochverfügbare Netzwerke	*OPS.1.1.2.A19*, *NET.1.1.A28*, *NET.1.1.A29*, *NET.1.2.A30*, *NET.3.2.A29*
HAV.A2	hochverfügbare Komponenten	*OPS.1.1.2.A19*, *SYS.1.5.A20*, *NET.1.1.A28*, *NET.1.2.A30*, *NET.3.2.A29*
HAV.A3	Redundanz	*APP.3.2.A15*, *SYS.1.1.A28*, *SYS.1.6.A34*

Tabelle 3.4.: Anforderungen an die Hochverfügbarkeit

3.4. Kriterien für die Komponenten

Ausgehend von in der Literatur einschlägigen Empfehlungen und den relevanten Anforderungen des *Kompendiums* werden für alle System-Komponenten des Informationsverbunds Auswahlkriterien erarbeitet, nach denen unterschiedliche Implementierungen bewertet und ausgewählt werden können. Mit diesem Vorgehen werden die Anforderungen *APP.3.1.A8 (S)*, *APP.6.A6 (S)*, *APP.6.A7 (S)*, *SYS.1.6.A1 (B)* erfüllt.[9]

Allgemeine Kriterien

Für alle Komponenten im Informationsverbund werden aus den allgemeinen Empfehlungen aus der Fachliteratur und den in der Praxis erprobten *bewährten Verfahren* (*best practices*) gemeinsam geltende Kriterien abgeleitet (Tabelle 3.5).

Kürzel	Beschreibung	Basis	Standard	erhöht
ALL.A1	Beschaffung aus stabilen, vertrauenswürdigen Quellen, die professionellen Support anbieten	*keine*	*APP.6.A7*	*APP.6.A14*
ALL.A2	Einplanen von Sicherheitsaspekten	*SYS.1.6.A1*	*CON.8.A5* *APP.3.1.A8* *APP.6.A6*	*keine*
ALL.A3	Minimalkonfiguration der Systeme, Deinstallation nicht genutzter Teile	*keine*	*APP.6.A11,A13*	*keine*
ALL.A4	Nutzung zentraler Komponenten des Sicherheitskerns und der Überwachung	*SYS.1.1.A2*	*ORP.4.A18* *OPS.1.1.5.A6* *DER.1.A11* *SYS.1.1.A23* *NET.1.2.A24*	*NET.3.2.A28* *DER.1.A15*
ALL.A5	Sachgerechter Einsatz von Kryptografie nach Stand der Technik	*ORP.4.A23*	*CON.1.A1–A5*	*keine*

Tabelle 3.5.: Allgemeine Anforderungen an die Komponenten

Herkunft

Die *Herkunft* der Komponenten muss dabei grundsätzlich bekannt und nachvollziehbar sein. „Die Lieferketten von Hard- und Software werden immer globaler und komplexer und bringen große Herausforderungen und Abhängigkeiten für Hersteller, Anbieter und Endkunden mit sich." [Pil17, S. 7] Dies ist vor allem bei gemeinschaftlich entwickelter *freier Software* (FOSS[10]) oft nicht einfach, weil die beteiligten Personen schnell wechseln oder jeweils nur wenige oder kleine Beiträge leisten. Insgesamt lassen sich viele Aspekte der Beschaffung bei FOSS nur schwer umsetzen, da kein *Anbieter* im klassischen Sinn vorhanden ist. *Professioneller Support* und verfügbare *Dual-Licence* Modelle sind daher Auswahlkriterien, um qualitative Eigenschaften der Software sicherzustellen, wie „SLA für Leistungsfähigkeit, Incident-Management und Reporting, Ansprechpartner beim Anbieter, Gewährleistung, allgemeine Haftungsbedingungen" [Pil17, S. 27–32]

Professioneller Support

Diese Anforderung nach Vertrauenswürdigkeit, besonders der Sicherheitsfunktionen, wird in *APP.6.A14 (H)* für hohe Schutzbedarfe geschärft: bei ansonsten gleichem Leistungsumfang sind *zertifizierte Produkte* (beispielsweise nach den CC) zu bevorzugen.

Zertifizierung

[9] Anforderungen aus den Bausteinen im *Kompendium* werden nachfolgend jeweils nur mit ihrer eindeutigen Nummer referenziert. In Klammern steht das Kürzel für das Schutzniveau. Eine vollständige Liste zeigt Abschnitt B.3.

[10] Free and Open Source Software

Stabilität

Ein weiteres allgemeines Kriterium ist *Stabilität*. Gerade im Bereich der Container-Plattformen ist dies wie beschrieben ein Problem. Grundsätzlich sollten daher Komponenten zum Zeitpunkt der Auswahl seit mindestens fünf Jahren öffentlich verfügbar sein und über mindestens zwei stabile Hauptversionen verfügen, die nicht älter als zwei Jahre sind.

zentrale Komponenten

Im Sinn der Abtrennung eines Sicherheitskerns (Abschnitt 2.2) müssen alle Komponenten für die Nutzung folgender *zentraler Systeme* konfigurierbar sein:

- Zugriffsberechtigung
- Verwaltung kryptografischer Schlüssel
- Logging
- Monitoring

Komponenten mit (relationalen) Datenbanken

Sofern Komponenten persistente Datenhaltung in (relationalen) Datenbanken benötigen, ist die sichere Konfiguration zu gewährleisten. Die Daten sind zu verschlüsseln und regelmäßig zu sichern (Tabelle 3.6).

Betriebssysteme

Grundsätzlich ist davon auszugehen, dass bei der Auswahl der Betriebssysteme Linux-Distributionen im Fokus stehen, denn: „Fast alle Container-Implementierungen basieren dabei auf einer Funktionalität, die der Linux-Kernel seit Version 2.6.32 mitbringt – den sogenannten Kernel Namespaces." [Lie19, § 3.2] In jüngster Zeit drängen auch andere integrierte Lösungen in den Markt, diese sind aber (noch) deutlich weniger verbreitet.[11] Der Fokus des *Kompendiums* für Anforderungen an Betriebssysteme liegt auf der sicheren Konfiguration und der möglichst umfassenden Nutzung vorhandener *Schutzmechanismen* (Tabelle 3.7).

Nutzung aller Schutzfunktionen

Virtualisierung

Obwohl eine klassische Virtualisierung für den Betrieb einer Container-Plattform nicht zwingend notwendig und für höchste Schutzbedarfe auch nicht wünschenswert ist, sollte sie in kleineren und mittelgroßen Installationen berücksichtigt werden. Besonders Komponenten, die nicht zum *Kern* gehören, profitieren oft von einer klassisch virtualisierten Umgebung. Vor dem Hintergrund einer stärkeren Isolierung wird bei einigen Container-Engines die vorhandene hardwarebasierte Virtualisierung auch zum Betrieb der Container eingesetzt. [12] Daher wird zusätzlich gefordert, dass eine verwendete Virtualisierungslösung dies unterstützt (Tabelle 3.8).

Internetanbindung

Im *Kompendium* werden an die Kopplung des Systems mit öffentlichen Netzen umfangreiche Anforderungen gestellt (Tabelle 3.9).

Obwohl sich Angriffe zunehmend gegen Personen als Einstieg in die Unterwanderung von Organisationen und Systemen richten[13], stellen automatisierte Attacken über öffentliche Netzwerke nach wie vor eine bedeutende Bedrohung dar. Deshalb wird das Prinzip *Verteidigung in der Tiefe* an diesem zentralen Übergabepunkt[14] besonders sorgfältig beachtet und detaillierte

[11] Meist setzen diese auf bestehende Virtualisierungslösungen auf, wie VMware vSphere Integrated Containers oder IBM z/OS Container Extensions, seltener auf ältere Container-Konzepte wie BSD jails oder Solaris zones.

[12] Das bekannteste Beispiel sind kata Container https://katacontainers.io/, die eine Orchestrierung auf allen gängigen Architekturen und unterschiedlichen Hypervisoren ermöglichen.

[13] „Most real attacks nowadays target the user. Various kinds of phishing are the main national-security threat, the principal means of developing and maintaining the cybercrime infrastructure, and one of the principal threats to online banking systems." [And20, S. 116]

[14] Dieser Übergabepunkt wird im *Kompendium* allgemein Firewall [BSI21, S. 701–710] oder spezifischer Sicherheitsgateway [BSI21, S. 349, 515] bezeichnet. Der Begriff Sicherheits-Gateway wird im BSI-Standard zur Internet-Sicherheit ISi-LANA verwendet [BSI14, S. 54].

Kürzel	Beschreibung	Basis	Standard	erhöht
RDB.A1	Sichere Konfiguration	*APP.4.3.A3,A4*	*APP.4.3.A12*	*keine*
RDB.A2	Datenverschlüsselung in Ruhe und in Transit	*keine*	*APP.4.3.A16*	*APP.4.3.A24*
RDB.A3	Regelmäßige und zuverlässige Datensicherung	*APP.4.3.A9*	*keine*	*keine*

Tabelle 3.6.: Anforderungen an Komponenten mit Datenbanken

Kürzel	Beschreibung	Basis	Standard	erhöht
OSY.A1	Sichere Konfiguration der Dienste und Nutzerkonten	*SYS.1.3.A2* *SYS.1.3.A5,A6*	*keine*	*SYS.1.3.A14*
OSY.A2	Nutzung aller vorhandenen Schutzmechanismen	*OPS.1.1.4.A2* *SYS.1.1.A9* *SYS.1.3.A4*	*SYS.1.1.A19* *SYS.1.3.A10*	*SYS.1.3.A17*
OSY.A3	Minimierung der Angriffsoberfläche und Dienste	*SYS.1.1.A5,A6*	*keine* *SYS.1.3.A16*	*SYS.1.1.A30,A31*
OSY.A4	Sachgerechter Einsatz von Kryptografie nach Stand der Technik	*keine*	*SYS.1.3.A8* *SYS.1.1.A34,A36*	*CON.1.A13*

Tabelle 3.7.: Anforderungen an Betriebssysteme

Kürzel	Beschreibung	Basis	Standard	erhöht
VIR.A1	Sorgfältige Planung und Dokumentation	*keine*	*SYS.1.5.A8,A10*	*SYS.1.5.A25*
VIR.A2	Sichere Konfiguration der Virtualisierung	*SYS.1.5.A2–A4*	*SYS.1.5.A14,A16*	*SYS.1.5.A21–A24*
VIR.A3	Zertifizierte Software	*keine*	*keine*	*SYS.1.5.A27*
VIR.A4	Sachgerechter Einsatz von Kryptografie nach Stand der Technik	*keine*	*keine*	*SYS.1.5.A26,A28*
VIR.A5	direkte Verwendbarkeit für Container-Nutzlasten	*keine*	*keine*	*keine*

Tabelle 3.8.: Anforderungen an die Virtualisierung

Kürzel	Beschreibung	Basis	Standard	erhöht
SGW.A1	Sichere Konfiguration der Komponenten	*APP.3.2.A1–A2* *NET.3.2.A3,A4,A8* *NET.3.2.A14*	*APP.3.2.A12* *APP.3.6.A13* *NET.3.2.A17*	*NET.3.2.A25* *NET.3.2.A31*
SGW.A2	Bündelung und Analyse der Datenströme zwischen öffentlichem und privatem Netz	*NET.3.2.A2*	*APP.3.2.A13* *SYS.1.6.A19*	*keine*
SGW.A3	Isolierung von schädlichen Inhalten	*OPS.1.1.4.A3* *APP.3.1.A4* *APP.3.2.A3*	*APP.3.2.A14*	*NET.3.2.A28*
SGW.A4	Spezifischer Schutz gängiger Protokolle (TCP, IP, UDP, ICMP, HTTP, SMTP, DNS)	*APP.3.1.A7* *NET.3.2.A10*	*APP.3.1.A21* *NET.3.2.A19,A20*	*APP.3.2.A20*
SGW.A5	Terminierung und Analyse verschlüsselter Datenströme	*APP.3.2.A5* *APP.3.2.A11*	*NET.3.2.A21*	*keine*

Tabelle 3.9.: Anforderungen an die Internetanbindung

Anforderungen spezifiziert. Das *Kompendium* sieht eine P-A-P-Struktur für die Internetanbindung bei normalen und erhöhten Schutzbedarfen zwingend vor, sie wird im Standard ISi-LANA detailliert beschrieben [BSI14, S. 54–59].

Registry

Wenig überraschend gibt es im aktuellen *Kompendium* keine Anforderungen an die Registry, die jedoch eine zentrale Komponente einer Container-Plattform darstellt. Drei Anforderungen sind dem Entwurf SYS.1.6 *Container* [BSI20b] zu entnehmen (Tabelle 3.10).

Rechteverwaltung

In diesem Kontext ist die Rechte- und Zugriffsverwaltung (kurz IAM[15]) als eine monolithische Komponente modelliert. Von Interesse sind allein zwei Funktionen: Erstens *Authentisierung* zur Bestimmung der Befugnis eines Nutzers, zum angefragten Zeitpunkt auf einer gegebenen Schnittstelle eine Anfrage zu stellen. Zweitens die *Autorisierung* zur Ermittlung der Rechte, die dem Nutzer gestatten, an einer Komponente über eine Schnittstelle spezifische Funktionen auszuüben. Wesentlich ist, dass das gewählte IAM mindestens für diese beiden Funktionen über jeweils genau eine Schnittstelle verfügt, die mit *allen* anderen Komponenten kompatibel ist. Dieser Aspekt wird als Anforderung *geeignete Authentisierung und Autorisierung* formuliert (Tabelle 3.11). Tatsächlich umfasst ein IAM viele weitere Funktionen für die Verwaltung der relevanten Daten, die üblicherweise von mehreren separaten Komponenten realisiert werden.

Schlüsselverwaltung

Ein KMS[16] dient zur Verwaltung kryptografischer Schlüssel und Zertifikate und kann Teil eines IAM sein. Um *Service-Accounts* und Zertifikate bereitzustellen, mit denen einzelne Komponenten des Systems verschlüsselt und beidseitig authentifiziert kommunizieren, ist die Funktionalität auch außerhalb eines IAM notwendig. Sie wird unter anderem vom Controller benötigt und verwendet. Daher wird diese Komponente gesondert betrachtet und modelliert. Sie muss hohen Anforderungen an die Verwendung von Kryptografie gerecht werden (Tabelle 3.12).

Zeitquelle

Die zentrale interne Zeitquelle muss von allen Komponenten über das Protokoll NTP[17] genutzt werden (Tabelle 3.13).

Netzwerk- und Traffic-Management

Das Netzwerk- und Traffic-Management in einem Container-Cluster unterscheidet sich von denen anderer Betriebsmodelle dadurch, dass es hochdynamisch auf Änderungen reagieren muss. Diese finden regelbasiert und vollständig automatisiert statt. Es stellt damit eine für Container-Plattformen charakteristische Komponente dar, die vom *Kompendium* bislang nicht adressiert wurde. Daher stammen fünf der sechs Anforderungen des BSI aus dem Entwurf SYS.1.6 *Container*. Sie betreffen im Wesentlichen die Segmentierung des Netzes (Tabelle 3.14).

Worker-Cluster

Der Worker-Cluster ist ebenfalls eine spezifische und zentrale Komponente zur Realisierung des Betriebsmodells Container-Plattform. 10 der 36 Anforderungen des Entwurfs SYS.1.6 betreffen diese Komponente. Dabei liegt der Fokus der Anforderungen darauf, Code, persistente Daten und Funktionen zu trennen, sowie Privilegien und Nutzung der Ressourcen geeignet zu beschränken (Tabelle 3.15).

[15] Identity and Access Management
[16] Key Management System
[17] Network Time Protocol

Kürzel	Beschreibung	Basis	Standard	erhöht
REG.A1	Vertrauenswürdige Registry und Images	*SYS.1.6.A6*	*keine*	*SYS.1.6.A30*
REG.A2	abgesicherte Registry	*keine*	*SYS.1.6.A28*	*keine*

Tabelle 3.10.: Anforderungen an die Registry

Kürzel	Beschreibung	Basis	Standard	erhöht
IAM.A1	Funktionstrennung, Aufgabentrennung	*ORP.4.A4,A9*	*SYS.1.6.A25*	*keine*
IAM.A2	Vollständige Identifizierung und Authentifizierung	*OPS.1.1.2.A5* *APP.3.1.A1*	*SYS.1.6.A24*	*keine*
IAM.A3	Geeignete Authentisierung und Autorisierung	*APP.2.1.A2*	*ORP.4.A13,A18* *APP.2.1.A9*	*keine*
IAM.A4	Sachgerechter Einsatz von Kryptografie nach Stand der Technik	*APP.3.1.A14*	*ORP.4.A12* *APP.2.1.A13*	*keine*
IAM.A5	Erhöhter Schutz für Zugangsdaten, administrative Zugänge und Tätigkeiten	*OPS.1.1.2.A6* *APP.2.1.A4,A6* *SYS.1.6.A10*	*ORP.4.A10* *APP.2.1.A8,A11*	*ORP.4.A21*

Tabelle 3.11.: Anforderungen an die Rechteverwaltung

Kürzel	Beschreibung	erhöht
KMS.A1	Rollen- und Schlüsselteilung, Vier-Augen-Prinzip	*ORP.4.A24*, *CON.1.A12*, *OPS.1.1.2.A17*
KMS.A2	Sachgerechter Einsatz von Kryptografie nach Stand der Technik	*CON.1.A9–A11*

Tabelle 3.12.: Anforderungen an die Schlüsselverwaltung

Kürzel	Beschreibung	Basis	Standard	erhöht
TIM.A1	Gemeinsame Zeitquelle für *alle* Komponenten	*OPS.1.1.5.A4* *SYS.1.5.A7* *NET.1.2.A8*	*keine*	*keine*
TIM.A2	Nutzung eines internen Zeitgebers mit NTP	*keine*	*NET.3.2.A22*	*NET.1.2.A37*

Tabelle 3.13.: Anforderungen an die Zeitquelle

Kürzel	Beschreibung	Standard	erhöht
NTM.A1	Segmentierung dynamischer Container-Netzsegmente	*SYS.1.6.A18–A19*	*SYS.1.6.A33*
NTM.A2	Verschlüsselte und limitierte Kommunikation	*keine*	*SYS.1.6.A36* *NET.3.2.A30*

Tabelle 3.14.: Anforderungen an das Netzwerk- und Traffic-Management

Kürzel	Beschreibung	Basis	Standard
WCL.A1	Trennung von Code, Daten und Funktionen	*SYS.1.6.A4,A5,A8,A9*	*SYS.1.6.A23*
WCL.A2	Begrenzte Privilegien und Ressourcennutzung	*keine*	*SYS.1.6.A16,A21,A26*
WCL.A3	Aktuelle und unveränderliche Artefakte	*keine*	*SYS.1.6.A14,A15*

Tabelle 3.15.: Anforderungen an den Worker-Cluster

Speicherpool

Für den Speicherpool der Nutzlast sind unterschiedliche Ausprägungen denkbar. Die Möglichkeit, diese Komponente als integralen Bestandteil des Worker-Clusters zu betreiben, beispielsweise als VSAN,[18] schließen sowohl eine Anforderung des Entwurfs SYS.1.6 als auch das Prinzip der Aufgabentrennung aus (Tabelle 3.16). Die Einbindung lokaler persistenter Speicher durch Container sollte vollständig ausgeschlossen werden. Deren temporäre Daten sollten ausschließlich im Arbeitsspeicher gehalten werden, falls notwendig in einem geeigneten RAM-Dateisystem. Der Speicherpool kann daher entweder als separates Cluster-Dateisystem, wie GlusterFS, CephFS, GFS, HDFS, oder als klassisches SAN[19] ausgeprägt sein. Er ist aber in jedem Fall über ein separates, sehr leistungsfähiges Netzwerk an den Worker-Cluster angebunden.

API und GUI

Weder an das API noch an das GUI werden spezifische Anforderungen gestellt. Sofern ein GUI eingesetzt wird, ist zu erwarten, dass dieses in Form einer Web- oder Desktop-Anwendung ausgeprägt ist. In diesem Fall müssten der Baustein APP.3.1 oder APP.6 berücksichtigt werden. Hier wird auf ein GUI als optionale Komponente verzichtet. Für das API gelten die allgemeinen Anforderungen und es wird unterstellt, dass manuelle administrative Tätigkeiten über eine Befehlszeilenschnittstelle (CLI[20]) vorgenommen werden, die das API direkt nutzen kann.

Scheduler

An den Scheduler, der die Zuordnung von Containern zu Servern im Worker-Cluster festlegt, werden keine spezifischen Anforderungen gestellt.

Controller

Die Anforderungen an den DNS[21] sind auf den Controller anzuwenden, der eine interne und dynamische Namensauflösung übernimmt. Es wird davon ausgegangen, dass DNS für öffentlich erreichbare Services über mehrere externe Dienstleister (Multi-Provider-DNS) abgebildet wird. Das SaaS-Angebot sollte über eine zentrale, stabile Domain und eine oder mehrere vorab festgelegte IP-Adressen erreichbar sein. Die Anforderungen an den Controller im Entwurf SYS.1.6. betreffen die sichere Automatisierung, die Überwachung der Betriebsfähigkeit laufender Container und die gleichmäßige Auslastung der Ressourcen (Tabelle 3.17). Die Container müssen Auskunft darüber geben, ob sie bereit sind, Anfragen zu verarbeiten (*readiness*) und ob sie in einem funktionsfähigen Betriebszustand sind (*liveness*). Der Scheduler sollte Container durch weitere Instanzen ergänzen (d. h. skalieren), wenn die *Readiness*-Prüfung häufiger fehlschlägt und Container beenden und durch neue Instanzen ersetzen, deren *Liveness*-Prüfung fehlschlägt. Im laufenden Betrieb sollten Container entsprechend der Auslastung der Server im Worker-Cluster bei Bedarf regelbasiert dynamisch verschoben werden.

Key-Value-Store

Die einzige Anforderung an den Key-Value-Store ist die hochverfügbare Auslegung.

Versionskontrolle

Der Fokus der Anforderungen an die Versionskontrolle liegt darauf, die Herkunft der Software sowie deren Änderungen laufend zu dokumentieren und möglichst lückenlos nachvollziehen zu können (Tabelle 3.18).

[18] Virtual Storage Area Network
[19] Storage Area Network
[20] Command Line Interface
[21] Domain Name Service

Kürzel	Beschreibung	Basis	Standard	erhöht
STO.A1	Geeignete Auswahl und Konfiguration	*SYS.1.8.A2*	*SYS.1.8.A8*	*keine*
STO.A2	Segmentierung der Netze und Speicherbereiche	*SYS.1.8.A4*	*SYS.1.6.A14*	*keine*
STO.A3	Verschlüsselung und Zugriffsbeschränkung	*keine*	*SYS.1.6.A17*	*SYS.1.6.A35*

Tabelle 3.16.: Anforderungen an den Speicherpool

Kürzel	Beschreibung	Basis	Standard
CTL.A1	Sicherer Betrieb des internen DNS	*APP.3.6.A1–A4,A6*	*APP.3.6.A10,A11,A13,A17*
CTL.A2	Laufende Überwachung, Verteilung, Skalierung	*keine*	*SYS.1.6.A3,A27*
CTL.A3	Sichere Automatisierung	*keine*	*SYS.1.6.A3,A20*

Tabelle 3.17.: Anforderungen an den Controller

Continuous Integration

Die Kernaufgabe der Komponente Continuous Integration besteht darin, Artefakte zu erzeugen, bei denen sichergestellt ist, dass alle Teile aus vertrauenswürdigen Quellen stammen, ihre technischen und fachlichen Spezifikationen erfüllt werden und sie frei von Schadcode und Schwachstellen sind (Tabelle 3.19). Diese Aufgabe wird in engem Zusammenspiel mit der Schadcodebekämpfung, der Schwachstellenerkennung und dem Build- und Test-Cluster erfüllt.

Continuous Deployment

Das Continuous Deployment muss die Artefakte in der Registry laufend überwachen, um bei Bedarf dem Controller neue Versionen und Parameter der Orchestrierung zuzuführen. Werden Schwachstellen in den Artefakten erkannt, sollte die Erzeugung neuer Artefakte automatisiert ausgelöst werden (Tabelle 3.20).

Build- und Test-Cluster

Zwei zentrale Anforderungen an einen Build- und Test-Cluster sind einerseits die strikte Trennung vom produktiven Betrieb und andererseits die Prüfung sowohl funktionaler als auch nichtfunktionaler Spezifikationen (Tabelle 3.21).

Schadcodebekämpfung

Für die Schadcodebekämpfung zielen die Anforderungen auf eine hohe Erkennungsrate und den sicheren Betrieb ab (Tabelle 3.22).

Schwachstellenerkennung

Bei der Schwachstellenerkennung kommt es vor allem darauf an, dass die Informationen über bekannte Schwachstellen möglichst aktuell sind und dass erkannte Probleme möglichst vollständig automatisiert behoben werden können (Tabelle 3.23).

Konsolidierte Anforderungen

Durch die geeignete Zusammenfassung der Anforderungen aus den Bausteinen konnte die Zahl der Prüfkriterien, die zu einem übersichtlichen Vergleich möglicher Optionen herangezogen werden, deutlich reduziert werden. Auf dieser Grundlage wird ein Lösungsvorschlag erarbeitet.

Kürzel	Beschreibung	Basis	Standard
RCS.A1	Sicherstellung der Integrität und Authentizität	*CON.8.A10*	*OPS.1.1.3.A10*
RCS.A2	Kontrollierte Änderungen	*keine*	*OPS.1.1.3.A6,A8 APP.3.1.A4*
RCS.A3	Laufende Dokumentation	*keine*	*OPS.1.1.3.A11*

Tabelle 3.18.: Anforderungen an die Versionskontrolle

Kürzel	Beschreibung	Basis	Standard
CIT.A1	Sicherstellen der vertrauenswürdigen Herkunft aller Quellen	*CON.8.A6*	*OPS.1.1.3.A10*
CIT.A2	Automatisiertes Testen technischer und funktionaler Anforderungen	*OPS.1.1.6.A2–A5*	*OPS.1.1.6.A10,A12,A13*
CIT.A3	Laufende Aktualisierung und Prüfung auf Schwachstellen und Schadcode	*keine*	*SYS.1.6.A14*
CIT.A4	Absicherung der Automation und des administrativen Fernzugriffs	*keine*	*SYS.1.6.A22,A23*

Tabelle 3.19.: Anforderungen an Continuous Integration

Kürzel	Beschreibung	Basis	Standard	erhöht
CDS.A1	Sicher automatisierte Installation	*APP.6.A5*	*SYS.1.6.A22*	*SYS.1.6.A30*
CDS.A2	Automatisierte Prüfung und Aktualisierung	*keine*	*keine*	*OPS.1.1.3.A14 SYS.1.6.A29*

Tabelle 3.20.: Anforderungen an Continuous Deployment

Kürzel	Beschreibung	Basis	Standard
TBC.A1	Trennung der Test- und Produktivumgebung und -daten	*CON.8.A7*	*OPS.1.1.6.A13 SYS.1.5.A10*
TBC.A2	Prüfung funktionaler und nichtfunktionaler Spezifikationen und Negativtests	*CON.8.A7*	*keine*

Tabelle 3.21.: Anforderungen an den Build- und Test-Cluster

Kürzel	Beschreibung	Basis	Standard	erhöht
MAL.A1	Einsatz professioneller Schadcode-Bekämpfung	*OPS.1.1.4.A1,3*	*keine*	*OPS.1.1.4.A11*
MAL.A2	Betrieb und regelmäßige Aktualisierung der Schadcode-Bekämpfung	*OPS.1.1.4.A5,A6*	*keine*	*keine*
MAL.A3	Sichere Isolierung von Schadcode	*keine*	*OPS.1.1.4.A9*	*OPS.1.1.4.A10*

Tabelle 3.22.: Anforderungen an die Schadcode-Bekämpfung

Kürzel	Beschreibung	Basis	Standard	erhöht
VUL.A1	Regelmäßige Prüfung und automatisierte Aktualisierung	*CON.8.A8*	*keine*	*SYS.1.6.A29*
VUL.A2	Überprüfung von externen Komponenten	*CON.8.A20*	*keine*	*keine*

Tabelle 3.23.: Anforderungen an die Schwachstellenerkennung

4. Entwurf der Systemarchitektur und Kernprozesse

4.1. Komponenten

Die Zahl der heute verfügbaren Optionen für jede Komponente im Bereich der Container-Plattformen ist überwältigend und wächst schnell weiter. Die CNCF[1] hat daher mit dem Projekt *Landscape* [CNCF21] ein Werkzeug etabliert, das es erlaubt, sich schnell einen Überblick zu verschaffen und die Auswahl mit Filtern einzugrenzen. Gleichzeitig bietet die CNCF Orientierung mit einem dreistufigen Reifegradmodell für die Projekte ihrer Mitglieder (Sandbox, Incubating, Graduation). Die OCI[2] fokussiert ihre Aktivitäten eng auf Container-Images und Container-Runtimes und strebt Zertifizierungen in diesem Bereich an.

Für die Eingrenzung der Auswahl gibt es mehrere Strategien. Die erste und einfachste bestünde darin, mechanistisch *alle* bekannten Optionen aufzulisten und zu prüfen, welche Anforderungen jeweils erfüllt sind. Der Nachteil besteht darin, dass die Prüfung pro Komponente aufwendig ist und ein Großteil des Aufwands für ausgeschlossene Optionen entstehen würde. Diese sind allerdings nicht voneinander unabhängig. Die Wahl einer bestimmten Komponente hat bedingt durch unterschiedliche Interoperabilität großen Einfluss auf die Zahl der verbleibenden Optionen für weitere Komponenten. Diese Interdependenz wird durch entstehende Standards langsam verringert. Die Wahl des Controllers schränkt aber nach wie vor die Freiheitsgrade bei der Wahl anderer Komponenten signifikant ein. Beispielsweise ist eine Vielzahl der Optionen für das Netzwerk- und Traffic-Management auf den Einsatz in Kombination mit Kubernetes beschränkt, obwohl es für diesen Bereich den Standard CNI[3] gibt. Die Kombination aus Controller, Scheduler und Netzwerk-Management hängt sehr eng zusammen. Mit einer einzigen Ausnahme (HashiCorp Nomad und Consul) sind Controller und Scheduler vollständig in einem Produkt gebündelt. An den Scheduler werden keine besonderen Anforderungen gestellt. Daher wird zuerst ein Controller mit integriertem oder passendem Scheduler gewählt.

Die Angaben zu den nachfolgend vorgestellten Produkten und Projekten, insbesondere der jeweilige Hersteller und das teilweise nur geschätzte Alter, entstammen eigener Recherche, vor allem über Suchmaschinen oder über Webseiten der Anbieter. Es wurde versucht, möglichst alle plausiblen Optionen zu ermitteln und bei der Bewertung streng nach den entwickelten Kriterien vorzugehen. Ein Anspruch auf Vollständigkeit kann im Rahmen dieser Arbeit jedoch nicht bestehen. Mit keinem der benannten Unternehmen besteht eine vertragliche Bindung oder eine andere finanzielle Beziehung oder Verpflichtung. Produktbezeichnungen können eingetragene Warenzeichen der jeweiligen Anbieter sein. Dies wurde nur kenntlich gemacht, sofern Anbieter in den Produktinformationen entsprechende Kennzeichen angebracht haben.

Controller und Scheduler

Eine Sonderstellung im Bereich der Controller nimmt Kubernetes (kurz k8s) durch seine marktbeherrschende Position ein. Die *Landschaft* oder das *Ökosystem*, das die CNCF präsentiert, beruht im Kern auf dieser Orchestrierungslösung, die ursprünglich von Google Inc. entwickelt wurde [Lie19, § 11.1.1]. Dabei wird die quelloffene Software, die unter https://kubernetes.io/ abrufbar ist, gern als *Vanilla Kubernetes* bezeichnet. Von deren direkten Einsatz raten aber Arundel und Liebel eindringlich ab. „Etliche Container-Teams, die in den letzten Jahren versucht haben, eine unternehmenstaugliche Lösung basierend auf Vanilla Kubernetes und offenen Drittanbieter-Tools zu implementieren, winken schon seit Längerem berechtigterweise nur noch frustriert und genervt ab.“ [Lie21, S. 900] Und so hat „Cindy Sridharan [...] geschätzt [...],

[1] Cloud Native Computing Foundation (https://www.cncf.io/)
[2] Open Container Initiative (https://opencontainers.org/)
[3] Container Network Interface

Name	Support	Alter	k8s	ALL.A2	ALL.A3	ALL.A4	ALL.A5	CTL.A1	CTL.A2	CTL.A3	erfüllt
Banzai Cloud PKE	Banzai Cloud Ltd.	2	j	n	n	n	n	n	j	n	1
Charmed Kubernetes	Canonical Ltd.	5	j	j	n	j	j	n	j	n	4
Clear Linux Project	Monta Vista LLC	5	j	j	j	n	n	n	j	n	3
D2iQ Konvoy	D2iQ Inc.	2	j	j	n	j	n	n	j	n	3
Docker Swarm	Mirantis Inc.	5	n	n	n	n	j	n	j	n	2
kubeOne	Kubermatic GmbH	2	j	n	n	n	n	n	j	n	1
kubermatic	Kubermatic GmbH	5	j	n	n	n	n	n	j	n	1
kubic	SUSE GmbH	5	j	n	j	n	n	n	j	n	2
Lokomotive	Kinvolk GmbH	4	j	j	j	n	n	n	j	n	3
Magnum (k8s on OpenStack)	Red Hat Inc.	6	j	j	n	n	n	n	j	n	2
Mesosphere DC/OS Platform	D2iQ Inc.	5	n	j	n	j	n	n	j	n	3
metalk8s	Scality	3	j	n	n	j	n	n	j	n	2
Mirantis Kubernetes Engine	Mirantis Inc.	4	j	j	n	j	j	n	j	n	4
Nomad und Consul	HashiCorp Inc.	6	n	j	n	j	j	j	j	n	5
OpenNebula	OpenNebula Systems	13	n	n	n	n	n	n	j	n	1
Rancher Kubernetes Engine	SUSE GmbH	3	j	n	n	n	n	n	j	n	1
RedHat OpenShift	Red Hat Inc.	7	j	j	n	j	j	n	j	n	4
RKE 2 (Government)	SUSE GmbH	1	j	j	n	n	j	n	j	n	3
talos	Talos Systems	3	j	j	j	n	j	n	j	n	4
Triton	Joyent Inc.	6	n	j	n	j	n	n	j	n	3
VMware Tanzu Kubernetes Grid	VMware Inc.	3	j	j	n	n	n	n	j	n	2
weave works Kubernetes Platform	Weaveworks Inc.	6	j	n	n	n	n	n	j	n	1

Tabelle 4.1.: Vergleich möglicher Controller

dass es etwa eine Million Dollar an Techniker-Gehältern kostet, um Kubernetes in einer produktiven Konfiguration aus dem Nichts heraus aufzusetzen und lauffähig zu haben.“ [ADD19, S. 41] Daher kommen nur paketierte, von der CNCF zertifizierte *Distributionen* (67) und *Installer* (20) infrage, die dafür sorgen, dass Kubernetes und weitere prinzipiell interoperable Komponenten in jeweils überprüft kompatiblen Versionen installiert und geeignet konfiguriert werden.

Neben der Vielzahl von k8s Varianten gibt es nur wenige weitere Optionen für den Controller. Davon bestehen die meisten aus traditionellen Virtualisierungslösungen, die um eine Anbindung an Container-Engines ergänzt wurden und auch andere Nutzlasten orchestrieren können.

Die Auswahl wird grundsätzlich auf Optionen beschränkt, mit denen die Anforderung ALL.A1 erfüllt wird (*professioneller Support und Stabilität*). Von einzelnen oder sehr wenigen zentralen Individuen getriebene Projekte und solche mit Marktpräsenz unter einem Jahr werden daher nicht aufgeführt. Vielversprechend unter den derart vorab ausgeschlossenen Optionen ist allein die Amazon EKS-D Kubernetes Distribution.

Die Auflistung in Tabelle 4.1 erfolgt (wie auch in allen folgenden Tabellen) alphabetisch nach Produktname. Anforderungen gelten als erfüllt, wenn sie integraler Bestandteil des Produktdesigns sind und in der Standardeinstellung verwendet werden.

Tabelle 4.1 zeigt, dass die Anforderungen CTL.A2 und CTL.A3 nicht als Kriterium zur Entscheidung beitragen können, weil CTL.A2 von allen und CTL.A3 von keiner der Optionen erfüllt

wird. Für die Absicherung der Automatisierung (CTL.A3) müssen also in jedem Fall zusätzliche Maßnahmen ergriffen werden. Die einzige Option, die standardmäßig einen sicheren Betrieb des internen DNS gewährleistet, ist Consul. Die Anforderung ALL.A3 wird von der Kombination Nomad und Consul nicht erfüllt, denn es werden keine spezifischen, reduzierten Betriebssystem-Komponenten vorausgesetzt. Clear Linux, kubic, Lokomotive und talos erfüllen das Prinzip durch eigene Betriebssysteme, die teilweise auch in anderen Konfigurationen eingesetzt werden können. Obwohl Kubernetes die bekannteste Option darstellt und weitverbreitet ist, werden Consul und Nomad aus folgenden Gründen ausgewählt:

- Umsetzung des Prinzips Aufgabentrennung
- gegenüber Kubernetes deutlich reduzierte Komplexität
- stabile Software-Release-Zyklen
- DNS-Absicherung ist mit Kubernetes ist nicht trivial
- für Consul existieren ausgereifte Datensicherungskonzepte und Anbindungen an einen Sicherheitskern
- Consul erlaubt die Segmentierung des dynamischen Netzwerks der Nutzlast
- Nomad orchestriert auch Nutzlasten jenseits von Containern und ermöglicht damit vereinfachte Migrationspfade vorhandener Anwendungen

Die Auswahl der weiteren Komponenten orientiert sich an den Optionen, für die eine Integration ins *Nomad Ökosystem* (https://www.nomadproject.io/docs/ecosystem) besteht.

Betriebssysteme

Die Festlegung des Controllers schränkt die Optionen für die Betriebssysteme ein (Tabelle 4.2). Vorab entfallen alle Lösungen, die nicht auf einer Linux-Distribution beruhen, insbesondere IBM z/OS mit Container Extensions[4], Joyent SmartOS (OpenSolaris basiert), vmware Tanzu, Microsoft Windows, DC/OS und Red Hat Enterprise Linux CoreOS (RHCOS), das nur in Verbindung mit OpenShift verfügbar ist. Während die Liste der Linux-Distributionen sehr lang ist, gibt es nur wenige mit direktem, professionellem Support. Einen Migrationspfad zu einer nach CC zertifizierten Lösung bieten nur RHEL,[5] CentOS (über RHEL) und Ubuntu Server. Allein RHEL verfügt in Version 7.1 über die Zertifizierung der Stufe EAL4 oder höher, die vom BSI für Netzwerkkomponenten empfohlen wird. Deshalb wird für den initialen Entwurf CentOS als Betriebssystem ausgewählt. Sollte professioneller Support tatsächlich erforderlich sein, wäre ein Umstieg auf RHEL einfach möglich.

Name	Support	Alter	ALL.A2	ALL.A3	ALL.A4	ALL.A5	OSY.A1	OSY.A2	OSY.A3	OSY.A4	erfüllt
CentOS / RHEL	Red Hat® Inc.	19	j	n	j	n	n	j	n	n	3
ClearLinux	Intel® Corporation	6	j	j	n	n	n	n	j	n	3
debian	z.B. credativ GmbH (Berater)	25	n	n	n	n	n	n	n	n	0
FlatCar Linux	Kinvolk GmbH	3	j	j	j	n	n	n	j	n	4
SUSE Linux	SUSE GmbH	>20	n	n	n	n	n	n	n	n	0
Ubuntu Server	Canonical Ltd.	17	j	n	n	n	j	j	n	n	3

Tabelle 4.2.: Vergleich möglicher Betriebssysteme

[4] Abgesehen von der fehlenden Kompatibilität kostet eine z/OS-Großrechner-Anlage mit 40 Kernen für Container-Lasten ca. 570 000 USD in der Anschaffung.

[5] Red Hat® Enterprise Linux®

Virtualisierung

Die Zahl möglicher Virtualisierungslösungen ist klein (Tabelle 4.3). Nach CC zertifiziert ist KVM[6] in Verbindung mit RHEL oder Ubuntu Server. KVM/firecracker, als vielversprechende Alternative zu KVM/qemu, kommt mangels Support und aufgrund des Alters nicht infrage. Die Anforderungen ALL.A5, VIR.A1, VIR.A2 und VIR.A4 können nicht durch inhärente Eigenschaften einer Virtualisierungslösung erfüllt und daher nicht als Auswahlkriterium herangezogen werden.

Alle Optionen sind bereits sehr lange verfügbar und wurden initial nicht in Hinblick auf eingebaute Sicherheit entworfen. Eine zusätzliche Härtung der Systemkonfiguration nach den Anforderungen VIR.A1, VIR.A2 und VIR.A4 ist daher für jede dieser Virtualisierungslösungen notwendig. Die Anbindung an einen zentralen Sicherheitskern muss ebenfalls über passende Konfigurationen oder Ergänzungen realisiert werden. Um keine zusätzliche Komponente einzuführen, wird KVM/qemu als integraler Bestandteil des verwendeten Betriebssystems gewählt.

Name	Support	Alter	ALL.A2	ALL.A3	ALL.A4	VIR.A3	VIR.A5	erfüllt
KVM/qemu	RedHat Inc., Canonical Ltd	14	n	n	n	j	j	2
VMware ESX	vmware Inc.	20	n	n	n	n	n	0
Xen	Oracle Inc., Citrix Inc.	18	n	j	n	n	n	1

Tabelle 4.3.: Vergleich möglicher Virtualisierungen

Internetanbindung

Bei höheren Schutzbedarfen empfiehlt das BSI für Netzwerkkomponenten und Firewalls den Einsatz von Produkten, die nach den CC mindestens der Stufe EAL4 zertifiziert wurden. Dies betrifft alle Komponenten der verlangten P-A-P-Struktur.

Die Optionen für Paketfilter (p) und Application-Level-Gateways (a) ergeben sich aus der Liste unter https://www.commoncriteriaportal.org/products/ (Tabelle 4.4). Die Anforderungen SGW.A1–A5 sind Gegenstand der CC-Zertifizierung und für alle Produkte erfüllt. Die Anforderungen ALL.A1–A2 und ALL.A4–A5 sind für alle und ALL.A3 ist für kein Produkt erfüllt. Als ALG wird die f5 BIG-IP eingesetzt, weil sowohl eine Anbindung an Consul als auch virtualisierte Versionen verfügbar sind, die in der entwickelten Plattform eingesetzt werden können. Das BSI empfiehlt für erhöhten Schutzbedarf, die Paketfilter von unterschiedlichen Anbietern zu beziehen. Als Vorschlag wird für den äußeren Paketfilter genuscreen und für den inneren Paketfilter eine Sophos Firewall eingesetzt. Andere Kombinationen sind denkbar und gleichwertig.

Registry

Ausgewählt wurde *Harbor*, die einzige von der CNCF als ausgereiftes (*graduated*) Projekt geführte Registry. Professionelle Unterstützung kann vmware Inc. liefern. Die Möglichkeit, mehrere Schwachstellenscanner einfach einzubinden, gibt hier den Ausschlag gegenüber der zweiten Lösung (Red Hat Quay), mit der die Anforderungen nach einer sicheren Anbindung und nach Support gleichzeitig erfüllt werden können.

Rechteverwaltung

Obwohl die Rechteverwaltung eine generische Komponente ist und einen zentralen Baustein des Sicherheitskerns darstellt, gibt es nur wenige Optionen für den unabhängigen Betrieb. Die

[6] Kernel-based Virtual Machine

Name	Support	Alter	p / a
Check Point R80.30	Check Point Ltd.	>8	p
f5 BIG-IP® ADF-Base 11.5	F5, Inc.	17	a
Forcepoint Data Guard 3.0	Forcepoint LLC	>5	a
genugate 9.0 (ALG und innerer PF)	genua GmbH	>14	pa
genuscreen 7.0	genua GmbH	>14	p
Sophos Firewall OS 17.01	Sophos Ltd.	>10	p
STORMSHIELD Firewall 2.2.6	STORMSHIELD SAS	7	p
WatchGuard Fireware OS 12.3.1	WatchGuard Inc.	>20	p

Tabelle 4.4.: Nach CC zertifizierte Gateway-Komponenten (Stufe EAL4 oder höher)

Name	Support	Alter	ALL.A2	ALL.A3	ALL.A4	ALL.A5	IAM.A1	IAM.A2	IAM.A3	IAM.A4	IAM.A5	erfüllt
Apache Syncope	Tirasa S.r.l.	>10	n	n	n	n	j	j	n	n	n	2
didmos	DAASI GmbH	20	n	j	n	n	j	j	n	n	n	3
FusionAuth	FusionAuth	3	j	j	n	n	j	j	n	n	n	4
gluu	gluu Inc.	12	n	n	n	n	j	j	n	n	n	2
IAM Suite	_betasystems AG	>7	n	n	n	n	j	j	n	n	n	2
midPoint	Evolveum s.r.o.	10	n	n	n	n	j	j	j	n	n	3
OpenIAM	OpenIAM LLC	13	n	j	n	n	j	j	n	n	n	2
WSO2 Identity Server	WSO2 Inc	>7	n	n	n	n	j	j	n	n	j	3
XTON Access Manager	Xton Technologies, LLC.	17	j	n	j	j	j	j	n	j	j	7

Tabelle 4.5.: Produkte zur Rechteverwaltung

weitaus meisten Angebote sind heutzutage selbst Software-as-a-Service und werden unter dem Begriff *Identity Provider* vermarktet. Aus den aufgeführten Optionen (Tabelle 4.5) wurde *midPoint* ausgewählt, weil ausgereifte Anbindungen an viele gängige Produkte existieren und die Software auch ohne vertragliche Bindungen evaluiert werden kann. Nur bei dieser Option konnte zweifelsfrei festgestellt werden, dass die Kompatibilität mit allen anderen Komponenten gegeben ist (IAM.A3). Mit Abstand die meisten Kriterien erfüllt jedoch der *XTON Access Manager* und sollte für den Einsatz in einem produktiven System in Betracht gezogen werden.

Schlüsselverwaltung

Zur Verwaltung kryptografischer Schlüssel wird das KMS *Vault* des Anbieters HashiCorp Inc. verwendet. Alle allgemeinen und spezifischen Anforderungen werden erfüllt (ALL.A1–A5 sowie KMS.A1–A2). Insbesondere können alle notwendigen Zertifikate einer PKI[7] für den verschlüsselten Verkehr zwischen den Services von *Consul* vollautomatisch erstellt und erneuert werden. Die Aufteilung der zentralen Schlüssel auf mehrere Teile (*Shamir's Secret Sharing*) wird ebenso unterstützt wie deren Auslagerung in ein HSM. Als hochverfügbaren Datenspeicher kann *Vault* unter anderem *Consul* direkt verwenden. Dieses Szenario wird eingesetzt.

Zeitquelle

Als Zeitquelle wird der vom gewählten Betriebssystem bereitgestellte NTP-Server *chrony* verwendet. Weitere Implementierungen wie OpenNTPD erfüllen die Anforderungen ebenfalls.

[7] Public Key Infrastructure

Netzwerk- und Traffic-Management

Das Netzwerk- und Traffic-Management übernimmt die als Controller gewählte Komponente *Consul*. Die Anforderungen NTM.A1 und NTM.A2 werden in der Enterprise-Version unterstützt.

Worker-, Build- und Test-Cluster

Die Möglichkeiten, für die Container-Engine als zentrale Komponente einer Container-Plattform professionellen Support zu erhalten, sind erstaunlicherweise sehr stark begrenzt. Aufgrund der Wahl von Betriebssystem und Controller gibt es als einzige Option die Kombination aus containerd (ein Projekt der CNCF vom Reifegrad *graduated*) und der Container-Runtime *runc*. Zwar wird auch *podman* von RHEL und CentOS unterstützt, jedoch fehlen wesentliche Funktionen in der Anbindung an *Nomad* und der Treiber ist laut Dokumentation noch nicht produktionsreif. Andere Container-Engines werden rein informativ aufgeführt. Davon sind einige für die Zukunft vielversprechend, weil sie nach dem Security-by-Design-Ansatz (S-b-D) entworfen wurden. Es wird Kompatibilität zu den OCI-Spezifikationen und Optimierung für Kubernetes (k8s) vermerkt.

Name	Alter	OCI	k8s	S-b-D	Sponsor / Anbieter	Website
containerd	5	j	n	n	CNCF	https://containerd.io
cri-o	4	j	j	n	CNCF	https://cri-o.io
gvisor	3	j	n	j	Google Inc.	https://gvisor.dev
katacontainers	3	j	n	j	mehrere	https://katacontainers.io
runc	7	j	n	n	OCI	https://github.com/opencontainers/runc
Singularity	5	n	n	j	Sylabs™	https://sylabs.io
SmartOS	10	n	n	j	Joyent™ Inc.	https://www.joyent.com/smartos

Speicherpool

In der CNCF Landscape sind 52 *Cloud-Native-Storage*-Angebote aufgelistet. Dies sind im Gegensatz zu klassischen Speicherpools zumeist verteilte Systeme, die sich horizontal auf sehr große Datenvolumina skalieren lassen, oft auch georedundant. Für den Einsatz in diesem Entwurf ist die Auswahl dennoch klein. Die meisten Angebote sind selbst SaaS und für kaum eine der verbleibenden Optionen gibt es professionellen Support. Der Speicherpool sollte sich einerseits als zentrales Dateisystem und andererseits als Objektspeicher für die Anwendung selbst über eine S3[8]-Schnittstelle nutzen lassen, die de facto Standard in diesem Bereich ist. Reine Objektspeicher wie *MinIO* und reine Blockspeicher wie *StorageOS* scheiden damit aus. Liebel nennt neben diesen und *GlusterFS* die ausgereifte und von Red Hat unterstützte Lösung *Ceph* und beschreibt den Aufbau und Einsatz ausführlich [Lie21, S. 1185–1211]. Im Projekt *Rook* wird *Ceph* in Containern implementiert. Es wird von Liebel ebenfalls erläutert [Lie21, S. 1211–1227] und stellt das einzige Projekt im Bereich Cloud-Native-Storage dar, das von der CNCF als *graduated* klassifiziert ist.

Im Gegensatz zu *Ceph* unterstützt die gewählte Lösung *RING* des Anbieters Scality die Verschlüsselung der Daten in Ruhe und in Transit in der Standardeinstellung. Es lässt sich direkt an ein Schlüsselverwaltungssystem anschließen und unterstützt Kerberos Authentifizierung. Damit stellt es das einzige System der Recherche dar, mit dem sich die Anforderung STO.A3 *Verschlüsselung und Zugriffsbeschränkung* ohne Weiteres erfüllen lässt.

[8] Simple Storage Service (Amazon)

API und GUI

Ein GUI wird nicht eingesetzt. Sowohl Nomad als auch Consul stellen ein API bereit, das die allgemeinen Anforderungen erfüllt.

Key-Value-Store

Die Funktion Key-Value-Store zur Speicherung des Cluster-Zustands übernimmt die als Controller gewählte Komponente *Consul*.

Continuous Integration, Continuous Deployment und Versionskontrolle

Für Continuous Integration und Continuous Deployment werden Lösungen gesucht, die eine ausgereifte Schnittstelle zum gewählten Controller und Scheduler bieten. Mehrere Angebote integrieren diese Funktionen sowie Werkzeuge zur Projektplanung und eine Versionskontrolle zu einer kompletten Entwicklungs-Plattform. „Wie so häufig ist das Problem nicht ein Fehlen verfügbarer Tools, sondern die schiere Masse möglicher Optionen. Es gibt eine Reihe von CD-Tools, die speziell für Cloud-Native-Anwendungen entworfen wurden, und schon lange verwendete klassische Build-Tools wie Jenkins haben mittlerweile Plug-Ins, damit sie mit Kubernetes und Containern arbeiten können." [ADD19, S. 272] Zur Verwendung mit *Nomad* bietet *GitLab* eine vollständige Anbindung und wird daher für Continuous Integration, Continuous Deployment und als Versionskontrolle gewählt.

Schadcodebekämpfung und Schwachstellenerkennung

Die Funktionen Schadcodebekämpfung und Schwachstellenerkennung werden unter dem Begriff *Container-Scanning* zusammengefasst und auch von den verfügbaren Angeboten kombiniert. Arundel nennt die Optionen *Clair*, *Aqua* und *Anchore Engine* [ADD19, S. 223–224]. Während *Aqua* eine integrierte und vollständige Plattform zur Absicherung von Cloud-Infrastrukturen darstellt, ist *Trivy* ein frei verfügbarer Container-Scanner des gleichen Anbieters. Dieser wird in der Standardkonfiguration von der gewählten Registry *Harbor* verwendet und daher auch hier eingesetzt. *Harbor* bietet für die drei genannten sowie weitere Container-Scanner Unterstützung, die auch parallel betrieben werden können.

4.2. Architektur

Sicherheit und Skalierbarkeit

Es gibt drei Grundprobleme beim Entwurf einer System-Architektur für Container-Plattformen, die den Anforderungen im *Kompendium* entsprechen soll.

Das erste Problem besteht darin, dass die vom BSI empfohlenen Architekturen regelmäßig davon ausgehen, dass diese sowohl Netzsegmente für Endgeräte (Clients) als auch für Server-Systeme enthalten. Differenzierte Referenzarchitekturen für die Segmentierung und Anbindung von Cloud-Diensten im reinen Rechenzentrumsbetrieb fehlen. Zwar gibt es in der ISi-Reihe Erweiterungen der Standard-Architektur für die Bereitstellung von Web-Services, diese gehen aber von einer einfachen dreistufigen Struktur aus. Die Aufteilung in Web-Server, Web-Application-Server und Datenbank-Server unterscheidet sich vom Aufbau einer Container-Plattform, bei der hauptsächlich zwischen den Ressourcentypen *Rechenressourcen* und *persistenter Speicher* differenziert wird.

Ein weiteres Problem rührt daher, dass vom BSI an die Art der Segmentierung sehr konkrete Anforderungen gestellt werden. Dabei sind Netzsegmente vorgegeben, die mit mehreren Paketfiltern an streng definierten Netzübergangspunkten physisch voneinander getrennt werden müssen. Bei typischen Container-Clustern fehlen Überlegungen und Möglichkeiten zu einer starken Segmentierung fast vollständig. Im Gegenteil wird davon ausgegangen, dass in eng

gekoppelten Cluster-Verbunden alle Kontroll-, Speicher- und Anwendungskomponenten mehr oder weniger ungehindert untereinander kommunizieren können. Dies begünstigt eine einfache horizontale Skalierung des Aufbaus.

Hier liegt das dritte Problem: Während Cluster-Verbunde ohne stärkere Segmentierung strukturell einfach auf hohe Bandbreiten skaliert werden können, stellen Netzübergangspunkte mit physischen Paketfiltern potenziell eine Grenze für die lieferbare Bandbreite dar. Für den hochverfügbaren Betrieb müssen sie redundant ausgelegt werden, wodurch die Kosten des Gesamtsystems signifikant steigen: Leistungsfähige Paketfilter sind meist teuer und es werden zusätzliche Netzwerkkomponenten wie Switch-Stacks benötigt.

ISi-Referenzarchitektur für Web-Services

Die ISi-Referenzarchitektur für Web-Services sieht deren Betrieb *innerhalb* einer P-A-P-Struktur vor. Beim Betrieb mit einem internen Netzsegment sind dabei mindestens fünf separate Paketfilter (PF1 bis PF5) vorgesehen, von denen einer (PF2) die Kopplung an das interne Netz übernimmt. Hinzu kommen drei weitere Paketfilter (PF8 bis PF10), mit denen die Management-Zone angebunden wird. Der eigentliche Web-Service wird in vier Komponenten aufgeteilt: Den Anwendungsserver (WWW AS), eine Datenbank (WWW DB), einen vorgelagerten Web-Server für die interne Verwendung (WWW intern) und einen für den öffentlichen Zugriff (WWW). Abb. 4.1 zeigt die Grundarchitektur gemäß BSI-Standard *ISi-Webserver* [BSI17b, S. 32], ergänzt um den *Virenschutz* (Schadcodebekämpfung) nach *ISi-LANA* [BSI14, S. 77].

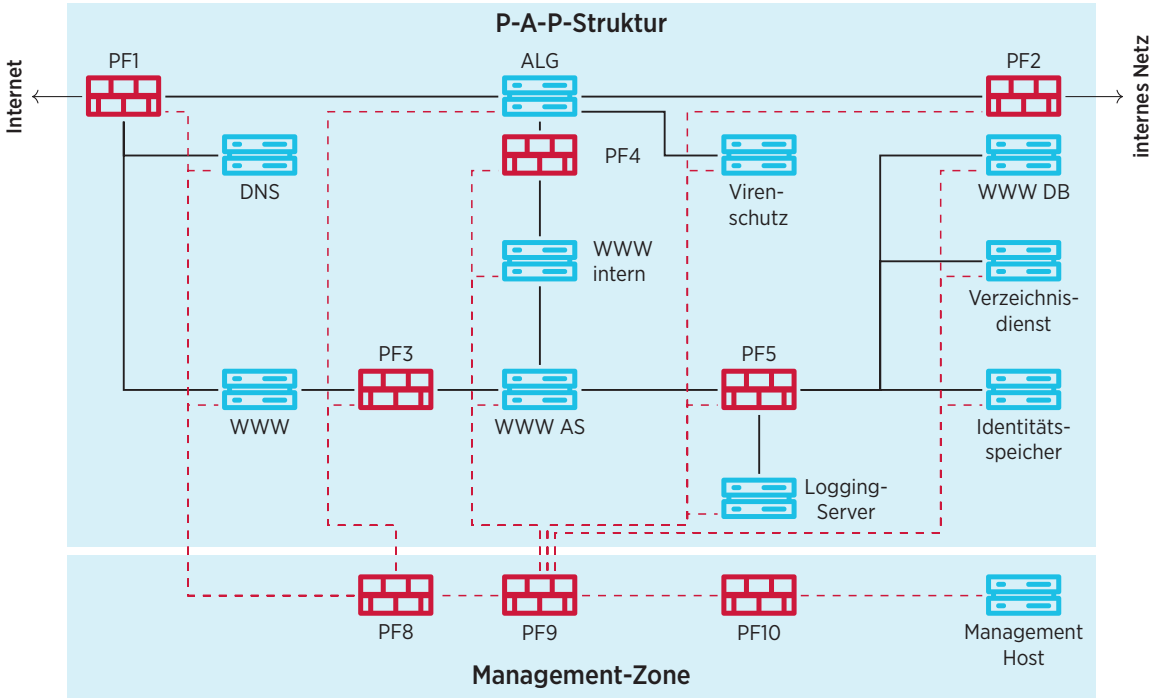


Abbildung 4.1.: ISi-Webserver Grundarchitektur

P-A-P-Struktur

Die P-A-P-Struktur ist vom *Kompendium* zwingend vorgesehen. Sie stellt die zentrale Kopplung zwischen internem und öffentlichem Netz dar. Der äußere Paketfilter (PF1), die DMZ[9]-Komponenten und der innere Paketfilter müssen physisch getrennt sein. Für normale Schutzbedarfe

[9] Demilitarisierte Zone

kann der innere Paketfilter mit dem des inneren Netzsegments zusammengelegt werden. Dies kommt nicht in Betracht, weil der Entwurf auch für erhöhte Schutzbedarfe ausgelegt sein soll. In einer Container-Plattform entspricht der Anwendungsserver dem Worker-Cluster und die Datenbank dem Speicherpool. Vorgelagerte (statische) Web-Server sind nicht vorgesehen, am ehesten entsprechen diese Komponenten dynamischen Proxys (bei Kubernetes *ingress controller* genannt). Gegenüber der Grundarchitektur können folgende Elemente entfallen:

- PF2, da kein internes Netzsegment existiert
- DNS, weil davon ausgegangen wurde, dass die öffentliche DNS-Auflösung von externen Dienstleistern übernommen wird und weitgehend statisch ist
- WWW, weil die Anbindung der Web-Anwendung in beide Richtungen über das ALG erfolgen kann, als zentraler eingehender und ausgehender Proxy
- PF3 weil DNS und WWW entfallen
- WWW intern, weil die Zwischenstufe ALG ausreichend ist: In der Grundarchitektur existiert ebenfalls nur eine weitere Komponente zwischen Anwendungsserver und öffentlichem Netzwerk.

Das bedeutet, dass sich eine geänderte P-A-P-Struktur aus PF1, ALG und PF4 ergibt, wobei ALG und Virenschutz Komponenten in der DMZ sind. Der Worker-Cluster bildet eine weitere, alleinstehende Zone, die über PF5 an den Bereich Überwachung (Logging als ein Bestandteil davon) und die Hintergrund-Komponenten angebunden ist. ALG und Virenschutz werden in einem virtualisierten Cluster betrieben, der jedoch für höhere Schutzbedarfe oder Lastprofile mit hohen Bandbreiten physisch getrennt werden sollte. Für die P-A-P-Struktur ergeben sich im hochverfügbaren Betrieb mindestens sechs Hardware-Komponenten ($PF1_1$, $PF1_2$, DMZ_1, DMZ_2, $PF4_1$, $PF4_2$). Abb. 4.2 zeigt eine mögliche Implementierung, wobei die beiden Paketfilter von unterschiedlichen Herstellern stammen und ebenso wie das ALG nach den CC vom BSI zertifiziert sind. Als Virenschutz wird MetaDefender ICAP[10] und core auf CentOS vorgeschlagen. CentOS kann durch RHEL ersetzt werden, um problemlos auf ein gemäß CC zertifiziertes Betriebssystem aufzurüsten.

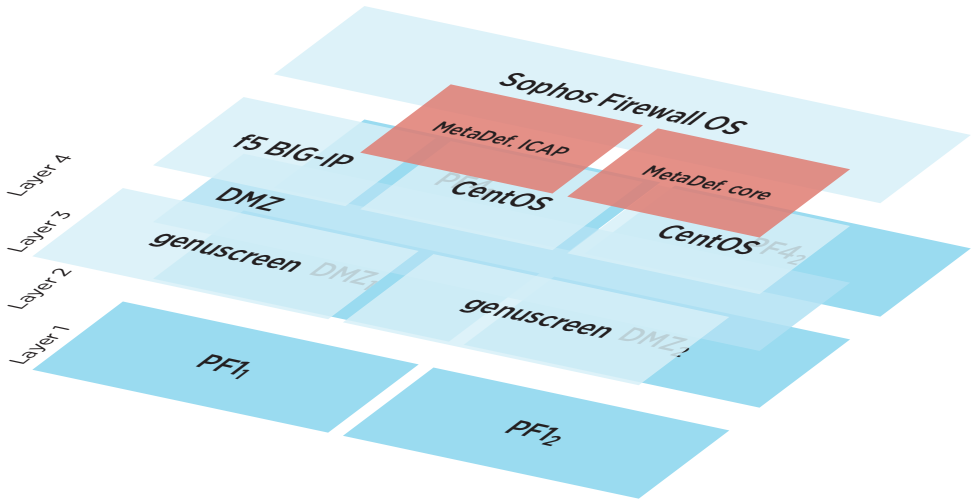


Abbildung 4.2.: P-A-P-Gateway

Worker-Cluster

Die Server innerhalb des Worker-Clusters werden mit einem minimalen Betriebssystem versehen, das sich schnell und atomar aktualisieren lässt und sind insgesamt nicht virtualisiert. Das

[10] Internet Content Adaptation Protocol

Austauschen und Aktualisieren bestehender und das Hinzufügen weiterer Knoten sollte möglichst schnell durchführbar sein, wobei die Größenordnung im Bereich von Stunden liegt. Es sind mindestens zwei Server einzuplanen ($WKN1_1$ und $WKN1_2$). In Richtung P-A-P-Struktur wird der Worker-Cluster über einen Switch-Verbund (SW1) angebunden, der mindestens zwei separate Switches enthält. Seine Dimensionierung muss ausreichen, um alle Server im Worker-Cluster und PF4 darüber anbinden zu können. Richtung Steuerebene und Speicherpool erfolgt die Anbindung über einen Paketfilterverbund (PF5). Dieser muss eine hohe Bandbreite für die Datenströme zwischen Worker-Cluster und Speicherpool bereitstellen und mit der Zahl der Worker-Knoten skaliert werden. Die Bandbreite in Richtung Steuerebene dürfte dagegen zu vernachlässigen sein.

Build- und Test-Cluster

Der Build- und Test-Cluster wird auf mindestens einem separaten Server betrieben. Abgesehen von der Nutzung lokaler Speicherkomponenten ist die Konfiguration identisch zu den Knoten des Worker-Clusters. Für eine hochverfügbare Grundausstattung sollten mindestens zwei Server bereitgestellt werden ($TB1_1$ und $TB1_2$). Von der Steuerebene aus wird ein direkter Zugriff über SW2 ohne zwischengeschalteten Paketfilter hergestellt.

Steuerebene und Entwicklung

In der Steuerebene werden die zentralen Komponenten Key-Value-Store, Controller und Scheduler betrieben. Dabei übernimmt Consul die Aufgaben des hochverfügbaren Key-Value-Stores und des Netzwerk- und Traffic-Managements. Consul basiert auf dem Konsens-Protokoll Raft und benötigt daher mindestens drei Knoten im Regelbetrieb. Ein einzelner Knoten darf ausfallen oder kann für Aktualisierungen temporär außer Betrieb genommen werden. Das Gleiche gilt für das Schlüsselverwaltungssystem Vault. Die Steuerebene wird daher auf mindestens drei physischen Servern implementiert (CTL_1–CTL_3). Für kleinere Installationen kann diese Hardware auch für Komponenten der Entwicklung und für die Basisdienste verwendet werden und wird insgesamt virtualisiert. Bei erhöhten Schutzbedarfen und für größere Systeme wird derart skaliert, dass diese drei Server ausschließlich für Consul und Vault verwendet und andere Komponenten der Steuerebene und der Entwicklung auf separaten virtualisierten Servern betrieben werden.

Speicherpool

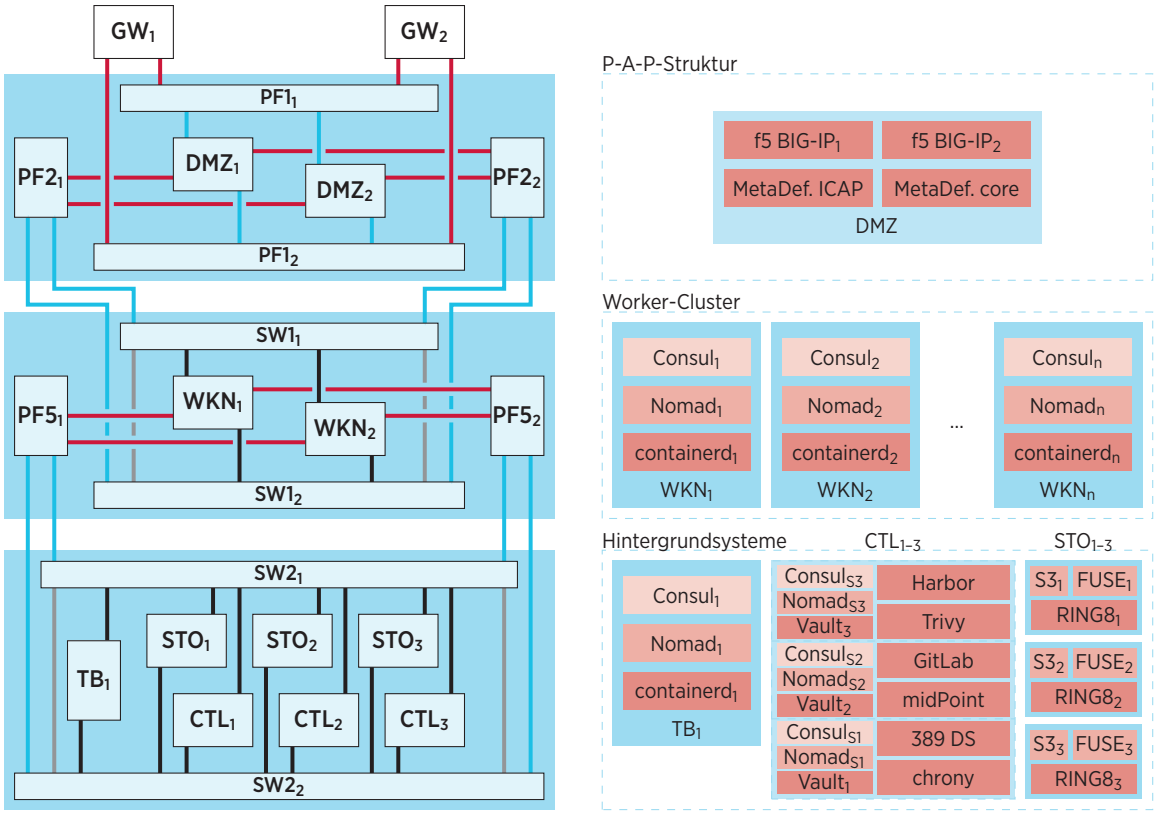
Der verwendete Dateisystem- und Objektspeicher-Cluster benötigt im Regelbetrieb mindestens drei aktive Knoten (STO_1–STO_3). Durch Hinzufügen weiterer Paare von Servern lassen sich neben der Kapazität auch die Ausfallsicherheit und der mögliche Auslastungsgrad erhöhen. Die Server im Speicherpool werden ausschließlich für die Datenhaltung verwendet und nicht virtualisiert. Das Betriebssystem wird entweder auf vollständig separaten Datenträgern abgelegt oder bei Bedarf (während des Bootvorgangs) aus der Steuerebene abgerufen. Der Speicherpool sollte sich ebenso schnell wie der Worker-Cluster skalieren lassen.

Minimalkonfiguration der Hardware

Insgesamt werden für den betrachteten Informationsverbund 21 separate Hardware-Komponenten bereits in der Minimalkonfiguration benötigt. Sie umfasst sechs Paketfilter ($PF1_{1–2}$, $PF4_{1–2}$ und $PF5_{1–2}$), fünf universelle Virtualisierungsserver ($DMZ_{1–2}$ und $CTL_{1–3}$), zwei Worker-Nodes mit minimalem lokalem Speicher ($WKN_{1–2}$), einen Build- und Test-Server (TB_1) sowie vier Switch-Elemente ($SW1_{1–2}$ und $SW2_{1–2}$). In der Management-Zone ist zusätzlich mindestens ein Sprung- und Installationsserver vorzusehen. Er wird nicht gezeigt, hier reicht in der Minimalkonfiguration ein virtuelles System, das der Hosting-Anbieter bereitstellt.

Netzwerkdesign und Gesamtsystem

Entsprechend NET.1.1.A2, NET.1.1.A13, NET.1.1.A16–A17 und NET.1.1.A25 muss der Netzwerkaufbau sorgfältig geplant und genau dokumentiert werden. Um den Anforderungen vollständig gerecht zu werden, umfasst dies alle Netzwerk-Layer, inklusive eventueller logischer Segmentierung in VLANs[11] und aller IP-Adressbereiche. Die Topologie der höheren Layer lässt sich dabei deutlich leichter anpassen als die Segmentierung auf der physischen Ebene. Abschließend wird daher ein Netzwerkplan auf Layer 1 inklusive der Anbindung an die externen Gateways des Hosting-Anbieters (GW_{1-2}) sowie alle Komponenten auf den Layern 4–6 gezeigt (Abb. 4.3). Der Layer 2 (Virtualisierung) wurde angedeutet, der Layer 3 (Betriebssysteme) der Übersichtlichkeit halber weggelassen. Die Verbindungen sind farblich gekennzeichnet: äußere Anbindungen der Paketfilter *rot* und innere *blau*, Verbindungen zwischen Servern ohne vermittelnden Paketfilter *schwarz* und interne Kopplungen von Switch-Stacks *grau*.



(a) Layer 1: Hardware und Netzwerk

(b) Layer 4–6: Software der Container-Plattform

Abbildung 4.3.: Gesamtsystem, minimaler Aufbau ohne Management-Zone

Die vom BSI vorgesehene physische Segmentierung in mindestens drei Zonen wurde dabei eingehalten und sinngemäß umgesetzt. Die Zonen *Überwachung* und *Management* sind für einen produktiven Betrieb zwingend erforderlich, liegen aber außerhalb des definierten *Informationsverbunds*. Gemäß BSI-Referenzdesign kommen durch die Management-Zone vor allem weitere Paketfilter hinzu. Der Bereich *Überwachung* kann durch einen oder zwei virtualisierte Server-Cluster oder auf Basis eines weiteren Container- und Storage-Clusters realisiert werden.

Zur Skalierung des Systems müssen erst weitere Server im Worker- und Storage-Cluster hinzugefügt und später der innere Paketfilter sowie die beiden Switch-Stacks erweitert werden.

[11] Virtual Local Area Network

4.3. Kernprozesse

Zwei wesentliche Bedingungen für das Entstehen und Aufrechterhalten von Sicherheit sind Planbarkeit und Nachvollziehbarkeit. Daher sind neben dem Einsatz geeigneter Strukturen und der Absicherung der Komponenten auch die Betriebsprozesse von zentraler Bedeutung. Harich nennt als Kernprozesse des Betriebs von IT-Systemen: Zugriffskontrolle, Konfigurations- und Patchmanagement, Benutzerverwaltung, Standardisierung, Incident Management, Notfallmanagement [Har18, S. 128] sowie sichere Softwareentwicklung, Identitätsmanagement und Genehmigungsprozesse [Har18, S. 146–154].

Hier werden nur die Prozesse zum Betrieb des *Informationsverbunds* und damit verbundene Rollen skizziert. Dabei sind nach Anforderung ORP.4.A24 des *Kompendiums* für erhöhte Schutzbedarfe alle administrativen Rollen mindestens doppelt zu besetzen und das Vier-Augen-Prinzip ist für deren Tätigkeiten technisch und organisatorisch zu forcieren.

Änderungsmanagement

Das *geregelte* Änderungsmanagement ist ein zentraler Steuerungsprozess eines ISMS und gleichzeitig derjenige, der sich erfahrungsgemäß nachträglich nur schwer einführen lässt, weil sein Geschäftswert nicht erkannt wird. Deshalb wird sowohl im Standard ISO 27001 als auch im *Kompendium* zwingend vorausgesetzt, dass die Einführung eines ISMS vom Top-Level-Management einer Organisation aktiv gefördert, begleitet und durchgesetzt wird. Für eine Standardabsicherung nach BSI IT-Grundschutz betrifft dies die *gesamte* Organisation. Die Planung des Änderungsmanagements im IT-Betrieb und seine Einbindung in die Organisation beschreiben die Anforderungen OPS.1.1.3.A4–A6.

Nachfolgend werden die für den betrachteten *Informationsverbund* relevanten Anforderungen und deren Umsetzung beschrieben und damit OPS.1.1.3.A1 sinngemäß erfüllt. Die nach OPS.1.1.3.A2 festzulegenden *Zuständigkeiten* für Änderungen werden wie folgt an spezifische Rollen gebunden:

Rollenprofil	Zuständigkeitsbereich für Änderungen
Entwicklung	Software, die in der Plattform betrieben werden soll, inklusive der Definition und des Layouts der zu betreibenden Artefakte (Container-Images)
Qualitätssicherung	Versionen und Kombinationen der zum produktiven Betrieb freigegebenen Software
Anwendungsadministration	Parameter der Orchestrierung (insbesondere der Skalierung und Service-Definitionen), Umfang der Datensicherung, Parameter zur Überwachung der Anwendung
Systemadministration	Virtualisierung, Betriebssysteme, Software zum Betrieb der Plattform selbst
Netzwerkadministration	alle Netzwerkkomponenten, mindestens jedoch Switches, Paketfilter, Proxys und Application-Level-Gateways
Berechtigungsvergabe	Rechte-, Rollen- und Gruppenzuordnung im Teilprozess Identitäts- und Berechtigungsmanagement

Tabelle 4.6.: Zuständigkeitsbereiche und Rollenprofile für Änderungen

Die Rollenprofile *Entwicklung* und *Qualitätssicherung* sollten strikt getrennt vergeben werden, weil ein potenzieller Interessenkonflikt besteht. *System-, Anwendungs-* und *Netzwerkadministration* sind zu trennen, weil deren Kombination leicht dazu genutzt werden kann,

Trennungen und Schutzmechanismen im Regelbetrieb zu umgehen. Das Rollenprofil *Berechtigungsvergabe* ist grundsätzlich von administrativen Tätigkeiten zu trennen.

Mechanismen zur unüberwachten Aktualisierung (*Autoupdate-Mechanismen*) werden für alle Komponenten vollständig deaktiviert. Genutzte Versionen aller Komponenten werden deklarativ festgelegt und überwacht aktualisiert. Derart wird OPS.1.1.3.A3 erfüllt.

Einmal werktäglich wird überprüft, ob Schwachstellen oder Fehler der Komponenten bekannt geworden sind und ob Patches verfügbar sind, um diese zu adressieren. Die Erfassung wird anhand von Datenbanken der Lieferanten und bekannter Schwachstellen automatisiert, um die Sichtung zu systematisieren. Patches zur Beseitigung kritischer Schwachstellen und Fehler werden spätestens innerhalb von zwei Werktagen nach Bekanntwerden oder Verfügbarkeit eingespielt. Andere Patches folgen gemäß Priorisierung nach Leistungsumfang und Wirksamkeit innerhalb von drei Kalenderwochen. Das jeweils letzte Patchlevel wird als Konfigurationsparameter einer Automatisierungslösung (z. B. *ansible*) erfasst. Diese Parameter dienen sowohl zur Durchführung von Tests als auch zur Installation, zur Aktualisierung der Dokumentation und falls notwendig zum Wiederherstellen der vorigen Version. Damit sind die Anforderungen OPS.1.1.3.A5,A15 und OPS.1.1.6 erfüllt.

Bei *größeren* Änderungen muss gemäß OPS.1.1.6 darüber hinaus die/der ISB[12] beteiligt werden. Größere Änderungen am Informationsverbund werden wie folgt definiert:

- Eine intern oder extern an den Informationsverbund angebundene Komponente, Schnittstelle oder Softwarebibliothek wird hinzugefügt, strukturell geändert oder vollständig entfernt. Dies betrifft *nicht* die Skalierung, also das Ergänzen oder teilweise Weglassen gleichartiger bereits bestehender Elemente.
- Ein Element wird auf eine andere Hauptversion aktualisiert.
- Der Netzwerkaufbau eines beliebigen Netzwerk-Layers wird strukturell geändert.
- Das Rollen- und Rechtekonzept wird strukturell modifiziert.

Entsprechend APP.6.A4–A5 wird jede Software mit allen abhängigen Elementen und Bibliotheken grundsätzlich:

- „nur mit dem geringsten notwendigen Funktionsumfang installiert und ausgeführt“
- „mit den geringsten möglichen Berechtigungen ausgeführt“
- mit „datensparsamsten Einstellungen […] konfiguriert“
- unverändert ohne eigene Modifikationen und erst nach erfolgreicher Prüfung kryptografischer Signaturen der Lieferanten installiert

[BSI21, S. 442].

Die Konfiguration aller eingesetzten Software ist *ausschließlich* und *vollständig* über Konfigurationsparameter der gewählten Automatisierungslösung zu erfassen und ohne jegliche interaktive Konfigurationsschritte anzuwenden.

Die Regelungen bezüglich des Änderungsmanagements sind laut OPS.3.1.A12 den Nutzern des im Informationsverbund betriebenen SaaS-Angebots als Dokumentation zur Verfügung zu stellen.

Identitäts- und Berechtigungsmanagement

Nach Anforderung ORP.4.A15 sollten Teilprozesse definiert und umgesetzt werden, um Richtlinien, Identitätsprofile, Benutzerkennungen, Berechtigungsprofile und Rollen zu verwalten. Die Rollen für den Informationsverbund wurden unveränderlich festgelegt (Tabelle 4.6). Sie sind mit den Berechtigungsprofilen als Konfigurationsparameter der Automatisierung und im IAM zu hinterlegen.

[12] Informationssicherheitsbeauftragte/r

Im Rahmen der Einstellung von Personal erfolgt die Einrichtung neuer Identitäten. Diese sind von der Personalverwaltung vor Aufnahme der Beschäftigung im IAM zu hinterlegen. Die Zuordnung zu den relevanten Rollen muss Vertragsbestandteil sein und ist im IAM einzurichten.

Benutzerkennungen sind strukturiert, regelbasiert und automatisch im IAM zu erzeugen und einer Identität eineindeutig zuzuordnen. Sie sollten keine Rückschlüsse auf diese ermöglichen und dürfen nicht wiederverwendet werden. Spätestens einen Werktag nach Ende der Beschäftigung eines Individuums ist dessen Identität im IAM zu sperren und unter Einhaltung gesetzlicher Fristen zu löschen.

Datensicherung

Die Anforderungen an die Datensicherung werden im *Kompendium* unter CON.1.A2, CON.3.A2, CON.3.A4–A5, CON.3.A11,A13 und NET.1.2.A6 formuliert. Um diese zu erfüllen, werden Datensicherungen kontinuierlich erzeugt und verschlüsselt abgelegt. Als Ziel für die Speicherung wird eine weitere, separate *RING*-Installation verwendet, die in einem anderen Brandabschnitt oder einem entfernten Rechenzentrum betrieben wird.

Um einen angemessenen Schutz gegen Kompromittierung und Erpressung (*Ransomware*) zu erzielen, wird *RING* betrieben, dass für den Bereich des Objektspeichers ein Überschreiben einmal abgelegter Daten nicht mehr möglich ist (WORM[13]). Die Datenobjekte der SaaS-Anwendung werden in der Sicherungskopie jeweils versioniert geschrieben. Ein analoges Verfahren lässt sich für die Nutzung als Dateisystem einsetzen (Copy-on-Write). Dies erzeugt jedoch einen sehr hohen Speicherbedarf der Sicherungskopien.

System- und Anwendungsadministration

Die Trennlinie zwischen System- und Anwendungsadministration ist nicht leicht zu ziehen. Sie wird so definiert, dass die Systemadministration alle Bereiche umfasst, die dazu notwendig wären, die Plattform Drittpersonen als Dienst anzubieten (PaaS). So kann aus organisatorischer Sicht die SaaS-Anwendung prinzipiell auch bei anderen Anbietern einer Container-Plattform betrieben werden, ohne dass die Strukturen und Prozesse der Anwendung, ihrer Entwicklung und ihres Betriebs verändert werden müssten.

Die Administration wird weitestgehend automatisiert. Nicht reproduzierbare Tätigkeiten werden so weit wie möglich auf die Überwachung sowie die Reaktion auf nicht vorhersehbare fehlerhafte oder unerwünschte Betriebszustände reduziert. Der Soll-Zustand der Systeme und Komponenten wird konsequent in Form von Konfigurationsparametern versioniert erfasst. Änderungen werden durchgehend kryptografisch sicher signiert. Damit sind die Anforderungen OPS.1.1.2.A8, OPS.1.1.2.A15, SYS.1.5.A12 und SYS.1.6.A20,A22 erfüllt.

Netzmanagement

Das Netzmanagement teilt sich in zwei Bereiche auf: die automatisierte Anpassung der SaaS-Anwendung durch *Consul* sowie die Verwaltung des statischen Aufbaus auf Layer 1 und der Komponenten außerhalb der *Nutzlast*. Die Konfigurationsdaten *aller* Netzwerkkomponenten sind als Parameter in der Automatisierung zu erfassen. Zur Provisionierung werden ausschließlich verschlüsselte Protokolle eingesetzt, präferiert HTTPS zur Ansteuerung geeigneter APIs. Alle anderen über das Netzwerk erreichbaren GUIs oder sonstigen Konfigurationsmöglichkeiten sind zu deaktivieren. Lokal erreichbare Konsolen müssen physisch gesondert abgesichert werden. Sie dürfen nur für Eingriffe in Notfällen im Vier-Augen-Prinzip verwendet werden und ihre Benutzung muss gesondert protokolliert werden. So werden die Anforderungen NET.1.2.A1, NET.1.2.A13,A16,A22,A24 und NET.1.2.A31 erfüllt.

[13] Write Once Read Multiple

5. Überprüfung des Entwurfs

5.1. Prüfung des Systems entsprechend den Bausteinen

Eckert merkt an, dass die *allgemeine* Frage, ob ein System hinsichtlich gegebener Schutzzustände bestimmte Sicherheitseigenschaften besitzt, nicht *generell entscheidbar* ist, sondern „[...] von Fall zu Fall dedizierte Entscheidungsverfahren zu konstruieren sind." [Eck18, S. 249]

Das hier gewählte Verfahren beruht wie der IT-Grundschutz auf einem Soll-Ist-Vergleich. Die einzelnen Prüfschritte des Abgleichs zwischen den aufgestellten Anforderungen und deren Erfüllung durch Struktur und Komponenten im Detail aufzuführen, würde das Volumen der vorliegenden Arbeit mindestens verdoppeln. Zusammengefasst lässt sich aber feststellen:

- mit der entwickelten Architektur lassen sich die Anforderungen an die Struktur und die Segmentierung des Systems vollständig erfüllen
- die Kernprozesse sind vergleichsweise knapp beschreibbar und können vor allem durch Automatisierung ebenfalls den Anforderungen genügen
- schon bei der Recherche zur Auswahl der Komponenten wurde klar, dass eine vollständige Erfüllung der Anforderungen mit Standardkonfigurationen nicht erzielbar ist

Keiner der Controller erfüllt die Anforderung CTL.A3 *Sichere Automatisierung*. In der Standardkonfiguration wird *Kryptografie nach Stand der Technik* (ALL.A5, OSY.A4, IAM.A4, STO.A3) von keinem der Betriebssysteme, nur von einem IAM und einer Speicherlösung eingesetzt. Eine Härtung der Konfiguration ist für alle Virtualisierungen nötig.

Die Berücksichtigung von Sicherheitsaspekten nach dem Security-by-Design-Ansatz ist nur bei jüngeren Produkten zu finden, die zum großen Teil keinen Support durch einzelne Unternehmen mehr bieten. Das Entwicklungs- und Support-Modell vieler *Cloud-Native-Komponenten* verlagert sich auf offene Gemeinschaften und Konsortien mehrerer Unternehmen, sodass Support-Verträge nur mit unabhängigen Beratern, nicht jedoch mit den an der Entwicklung originär Beteiligten möglich sind.[1]

Die einfache Zusammenstellung geeigneter Komponenten und deren Kombination in einer an sich geeigneten Systemarchitektur reicht jedenfalls nicht aus, um die Anforderungen nach IT-Grundschutz ohne Weiteres zu erfüllen. Für die ausgewählten Komponenten liegt jedoch hinreichend Dokumentation vor, anhand derer die notwendigen zusätzlichen Maßnahmen umgesetzt werden können. Mit geeigneter Konfiguration lassen sich die technischen Anforderungen mindestens für normalen Schutzbedarf verwirklichen.

5.2. Einstufung in das Schutzniveau gemäß BSI

Ausschlaggebend für das erreichte Schutzniveau gemäß IT-Grundschutz ist die Erfüllung von Anforderungen bezüglich der Verschlüsselung von Daten, der konsequenten Segmentierung des Netzwerks sowie der größtmöglichen Einschränkung und Kontrolle von Zugriffsrechten.

Während sich die strukturellen Anforderungen durch die vorgeschlagene Architektur und den Einsatz geeigneter Netzwerkkomponenten auch für erhöhte Schutzbedarfe erfüllen lassen, gestaltet sich die Umsetzung des höheren Schutzniveaus für die einzelnen Komponenten aufwendiger. Insbesondere lässt sich dieses für die Betriebssysteme, die Virtualisierung und die Container-Engine nur durch sorgfältige zusätzliche Konfiguration erzielen, die gesondert zu überprüfen ist.

[1] An einigen Projekten lässt sich beobachten, dass dies nachteilig für deren Dauerhaftigkeit ist. Der Druck zur kommerziellen Auswertung der investierten Arbeitsleistung führt dazu, dass solche Projekte häufig kurz vor oder nach Erreichen der Produktionsreife von einzelnen Anbietern übernommen werden. Sie werden dann proprietär weitergeführt und vermarktet oder als unliebsame Konkurrenz zum bestehenden Produktportfolio vom Markt entfernt.

6. Fazit und Ausblick

6.1. Fazit: alternative Ansätze erforschen

Zusammenfassend kann gesagt werden, dass es ein ausgesprochen aufwendiges Unterfangen ist, den BSI IT-Grundschutz auf ein komplexes System wie eine Container-Plattform anzuwenden. Obwohl in dieser Arbeit zur Reduktion des Umfangs und Aufwands bereits zentrale, vor allem organisatorische Aspekte bewusst ausgelassen wurden, bleiben zahlreiche Anforderungen zu berücksichtigen.

Die meisten dieser technischen und organisatorischen Anforderungen sind mit geringem Aufwand zu erfüllen und die erforderlichen Konzepte und Unterlagen können knapp gehalten werden. Andere Anforderungen schränken die Anzahl der möglichen Optionen stark ein oder ziehen einen erheblichen Aufwand an Recherche nach sich.

Die vorliegende Arbeit trägt einen Teil dazu bei, diese Aufgabe zu vereinfachen. Dennoch bleiben für die konkrete Umsetzung umfangreiche Details zu beachten. Diese erschweren den ohnehin schon komplexen Aufbau und Betrieb einer Container-Plattform.

Die Umsetzung der hier beschriebenen Architektur kann daher nur Organisationen empfohlen werden, denen entsprechende personelle und finanzielle Mittel zur Verfügung stehen.[1] Es ist darüber hinaus dringend angeraten, *vor* der Umsetzung eines solchen Projekts ein funktionierendes und stabiles ISMS gemäß BSI IT-Grundschutz zu etablieren.

Daher dürfte es von besonderem Forschungsinteresse sein, ob sich mit alternativen Ansätzen, die nicht auf der Bearbeitung umfangreicher Kriterienkataloge basieren,[2] ein ähnliches Schutzniveau erreichen lässt. Zudem sollten im IT-Grundschutz vereinfachte Strukturen berücksichtigt werden, um speziell für Container-Plattformen einen adäquaten Schutz mit weniger Aufwand und deutlich geringeren initialen Kosten zu ermöglichen.

6.2. Ausblick: Zertifizierung nach ISO 27001

Der gewählte *Informationsverbund* wurde bewusst klein gehalten, um die Vorgaben zum Umfang dieser Arbeit erfüllen zu können. Für einen realen Systembetrieb ist aber der Bereich der Überwachung unverzichtbar und muss als Bestandteil eines größeren *Informationsverbunds* betrachtet werden. Die Vorgehensweise bleibt die gleiche, wobei sich der Umfang der Anforderungen aus den Bausteinen nicht wesentlich vergrößern würde. Es sind geeignete Kriterien für weitere neun Komponenten abzuleiten und mögliche Implementierungen zu bewerten.

Zusätzlich müssten die hier skizzierten Prozesse genauer dokumentiert und auf die Situation in der jeweiligen Organisation angepasst werden. Die umgesetzten Maßnahmen und Prozesse, inklusive des etablierten ISMS, müssen zuletzt noch auf die *Controls* der Norm ISO 27001 übersetzt werden. Für die Erteilung eines Zertifikats müssen sie von Auditoren untersucht werden, die selbst vom BSI zertifiziert sind.

Insgesamt stellt die Zertifizierung von Informationsverbunden nach ISO 27001 auf Basis des IT-Grundschutzes ein sehr anspruchsvolles Projekt dar, das sorgfältig geplant werden muss und einen erheblichen Bedarf an Personal und Ressourcen mit sich bringt. Bislang hat als einzige Organisation die]init[AG diesen Schritt auch für *skalierbare Container-Architekturen* unter der Zertifikatsnummer BSI-IGZ-0378–2019 vollzogen.

---

[1] Dies gilt besonders für hohe Schutzbedarfe, denn einerseits ist eine zusätzliche Risikoanalyse für alle entsprechenden Anforderungen vorgesehen, andererseits legen die Anforderungen für hohe Schutzbedarfe aus dem Baustein DER.1 den Einsatz eines SIEM und eines IPS nahe, deren Aufbau und Betrieb besonderer Expertise und erheblichen zusätzlichen Ressourcen bedarf, siehe Abschnitt 2.1.

[2] beispielsweise basierend auf einfachen Schutzprinzipien, automatisierten Strukturanalysen, algorithmischen Prüfverfahren, mathematischen Verfahren oder Machine-Learning-Ansätzen

A. Verzeichnisse

A.1. Literatur

[Abo17a] Ferri Abolhassan. „Security : Die echte Herausforderung für die Digitalisierung“. In: *Security Einfach Machen : IT-Sicherheit als Sprungbrett für die Digitalisierung*. Hrsg. von Ferri Abolhassan. Wiesbaden: Springer Fachmedien, 2017. Kap. 1, S. 1–10. ISBN: 978-3-658-14945-1.

[Abo17b] Ferri Abolhassan, Hrsg. *Security Einfach Machen : IT-Sicherheit als Sprungbrett für die Digitalisierung*. Wiesbaden: Springer Fachmedien, 2017. 142 S. ISBN: 978-3-658-14945-1.

[ADD19] John Arundel, Justin Domingus und Thomas Demmig. *Cloud Native DevOps mit Kubernetes : Bauen, Deployen und Skalieren moderner Anwendungen in der Cloud*. Heidelberg: dpunkt., 2019. 342 S. ISBN: 978-3-96088-828-4.

[Ade+18] Michael Adelmeyer u. a. *IT-Risikomanagement von Cloud-Services in Kritischen Infrastrukturen : HMD Best Paper Award 2017*. essentials. Wiesbaden: Springer Fachmedien, 2018. 38 S. ISBN: 978-3-658-22742-5.

[And20] Ross Anderson. *Security Engineering : A Guide to Building Dependable Distributed Systems*. 3. Aufl. Indianapolis: John Wiley & Sons, Inc., 2020. 1182 S. ISBN: 978-1-119-64283-1.

[Bin20] Jan Bindig. *Das IT-Security-Mindset : Der Mittelstand auf dem digitalen Prüfstand*. 2. Aufl. München: Finanz Buch, 2020. 159 S. ISBN: 978-3-96092-316-9.

[BSI14] Bundesamt für Sicherheit in der Informationstechnik. *Sichere Anbindung von lokalen Netzen an das Internet (ISi-LANA). BSI-Standards zur Internet-Sicherheit (ISi-S)*. Version 2.1. 2014. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Internetsicherheit/isi_lana_studie_pdf (besucht am 04. 03. 2021).

[BSI17] Bundesamt für Sicherheit in der Informationstechnik. *Informationssicherheit und IT-Grundschutz : BSI-Standards 200-1, 200-2, 200-3*. 3. Aufl. Köln : Bundesanzeiger Verlag, 2017. 325 S. ISBN: 978-3-8462-0816-8.

[BSI17b] Bundesamt für Sicherheit in der Informationstechnik. *Sicheres Bereitstellen von Web-Angeboten (ISi-Webserver). BSI-Standards zur Internet-Sicherheit (ISi-Reihe)*. Version 1.1. 2017. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Internetsicherheit/isi_web_server_studie.pdf (besucht am 13. 03. 2021).

[BSI18] Bundesamt für Sicherheit in der Informationstechnik. *IT-Grundschutz-Kompendium*. 1. Aufl. Köln: Bundesanzeiger, 2018. ISBN: 978-3-8462-0906-6.

[BSI20a] Bundesamt für Sicherheit in der Informationstechnik. *IT-Grundschutz-Kompendium*. 3. Aufl. Köln: Reguvis Fachmedien, 2020. 816 S. ISBN: 978-3-8462-0906-6.

[BSI20b] Bundesamt für Sicherheit in der Informationstechnik. *SYS.1.6: Container*. Community Draft 2, in Überarbeitung. 2020. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/Drafts/Community_Draft/SYS_1_6_Container_CD.pdf (besucht am 04. 03. 2021).

[BSI21] Bundesamt für Sicherheit in der Informationstechnik. *IT-Grundschutz-Kompendium*. 4. Aufl. Köln: Reguvis Fachmedien, 2021. 810 S. ISBN: 978-3-8462-0906-6.

[BSIMM20] Sammy Migues. *Everything You Need to Know About the BSIMM*. Hrsg. von Synopsis Inc. 2020. URL: https://www.bsimm.com/content/dam/bsimm/ebook/everything-know-bsimm-eb.pdf (besucht am 02. 02. 2021).

[CLASP06] Open Web Application Security Project, Hrsg. *Comprehensive, Lightweight Application Security Process*. Version 1.2. 2006. URL: https://owasp.org/www-pdf-archive/Us_owasp-clasp-v12-for-print-lulu.pdf (besucht am 02. 02. 2021).

[CNCF21] The Linux Foundation®, Hrsg. *CNCF Cloud Native Landscape*. 2021. URL: https://landscape.cncf.io/images/landscape.pdf (besucht am 22. 03. 2021).

[Dun08] Jürgen Dunkel, Hrsg. *Systemarchitekturen für Verteilte Anwendungen : Client-Server, Multi-Tier, SOA, Event Driven Architectures, P2P, Grid, Web 2.0*. München: Hanser, 2008. 292 S. ISBN: 978-3-4464-1321-4.

[Eck18] Claudia Eckert. *IT-Sicherheit : Konzepte - Verfahren - Protokolle*. 10. Aufl. De Gruyter Oldenbourg Studium. Berlin: De Gruyter Oldenbourg, 2018. 1004 S. ISBN: 978-3-11-056390-0.

[Har18] Thomas W. Harich. *IT-Sicherheitsmanagement : Praxiswissen für IT Security Manager*. Hrsg. von mitp Verlags-GmbH & Co. KG. 2. Aufl. Frechen: mitp, 2018. ISBN: 978-3-95845-274-9.

[HB19] Christoph Haar und Erik Buchmann. „IT-Grundschutz für die Container-Virtualisierung mit dem neuen BSI-Baustein SYS. 1.6“. In: *INFORMATIK 2019: 50 Jahre Gesellschaft für Informatik – Informatik für Gesellschaft*. Hrsg. von Klaus David u. a. Bonn: Gesellschaft für Informatik e.V., 2019, S. 479–492. DOI: 10.18420/inf2019_65. URL: https://doi.org/10.18420/inf2019_65.

[Her17] Lars Herrmann. „Ineinandergreifen : Container und Virtualisierung ergänzen sich“. In: *iX Kompakt - Container und Virtualisierung : Handreichungen für Administratoren und Anwender*. Hrsg. von iX-Redaktion. Hannover: Heise Media, 2017, S. 54–56. ISBN: 978-3-9578-8185-4.

[iX-17] iX-Redaktion, Hrsg. *iX Kompakt - Container und Virtualisierung : Handreichungen für Administratoren und Anwender*. Hannover: Heise Media, 2017. ISBN: 978-3-9578-8185-4.

[Lie19] Oliver Liebel. *Skalierbare Container-Infrastrukturen : das Handbuch für Administratoren*. 2. Aufl. eBook ohne Seitenzahlen. Bonn: Rheinwerk, 2019. ISBN: 978-3-8362-6387-0.

[Lie21] Oliver Liebel. *Skalierbare Container-Infrastrukturen : das Handbuch für Administratoren*. 3. Aufl. Bonn: Rheinwerk, 2021. 1260 S. ISBN: 978-3-8362-7773-0.

[MG11] Peter M. Mell und Timothy Grance. *SP 800-145. The NIST definition of cloud computing*. Techn. Ber. Gaithersburg, MD, USA, 2011. DOI: 10.6028/NIST.SP.800-145. URL: https://doi.org/10.6028/NIST.SP.800-145.

[Mou17] Adrian Mouta. „Klare Linie : Container absichern und beschränken“. In: *iX Kompakt - Container und Virtualisierung : Handreichungen für Administratoren und Anwender*. Hrsg. von iX-Redaktion. Hannover: Heise Media, 2017, S. 148–154. ISBN: 978-3-9578-8185-4.

[Mül18] Klaus-Rainer Müller. *IT-Sicherheit mit System : Integratives IT-Sicherheits-, Kontinuitäts- und Risikomanagement – Sichere Anwendungen – Standards und Practices*. 6., erweiterte und überarbeitete Auflage. Wiesbaden: Springer Fachmedien, 2018. 866 S. ISBN: 978-3-658-22065-5.

[Pil17] Ernst Piller. *Beschaffung unter Berücksichtigung der IT-Sicherheit : Wichtigkeit, Herausforderungen und Maßnahmen*. essentials. Wiesbaden: Springer Fachmedien, 2017. 58 S. ISBN: 978-3-658-18599-2.

[Pog17] Werner Poguntke. *Basiswissen IT-Sicherheit : das Wichtigste für den Schutz von Systemen und Daten*. 3. Aufl. Informatik. Berlin: Springer Campus, 2017. 308 S. ISBN: 978-3-96149-013-4.

[SAMM2020] Open Web Application Security Project, Hrsg. *Software Assurance Maturity Model*. Version 2.0. 2020. URL: https://github.com/OWASP/samm/blob/master/Supporting%20Resources/v2.0/OWASP-SAMM-v2.0.pdf (besucht am 02. 02. 2021).

[Sch17] Peter Schaar. „Datenschutz-Empowerment“. In: *Security Einfach Machen : IT-Sicherheit als Sprungbrett für die Digitalisierung*. Hrsg. von Ferri Abolhassan. Wiesbaden: Springer Fachmedien, 2017. Kap. 3, S. 23–28. ISBN: 978-3-658-14945-1.

[Sim18] Eric Simmon. *SP 500-322. Evaluation of cloud computing services based on NIST SP 800-145*. Techn. Ber. Gaithersburg, MD, USA, 2018. DOI: 10.6028/NIST.SP.500-322. URL: https://doi.org/10.6028/NIST.SP.500-322.

[Wen18] Steffen Wendzel. *IT-Sicherheit für TCP/IP- und IoT-Netzwerke : Grundlagen, Konzepte, Protokolle, Härtung*. Wiesbaden: Springer Vieweg, 2018. 354 S. ISBN: 978-3-658-22603-9.

A.2. Glossar

Artefakt wörtlich: etwas von Menschenhand Geschaffenes; in der Software-Technik das greifbare und archivierbare Ergebnis (meistens: Nebenprodukt) eines Prozessschritts oder eines Prozesses, beispielsweise ein ausführbares, binäres Kompilat von Quellcode oder eine Programmbibliothek. 1, 3, 5, 7, 9, 10, 35, 48, IV

Baustein „Das IT-Grundschutz-Kompendium enthält für unterschiedliche Vorgehensweisen, Komponenten und IT-Systeme Erläuterungen zur Gefährdungslage, Sicherheitsanforderungen und weiterführende Informationen, die jeweils in einem Baustein zusammengefasst sind. Das IT-Grundschutz-Kompendium ist aufgrund der Baustein-Struktur modular aufgebaut und legt einen Fokus auf die Darstellung der wesentlichen Sicherheitsanforderungen in den Bausteinen. Die grundlegende Struktur des IT-Grundschutz-Kompendiums unterteilt die Bausteine in prozess- und systemorientierte Bausteine, zudem sind sie nach Themen in ein Schichtenmodell einsortiert." [BSI21, S. 31] 18

Circuit Gateway Netzwerkkomponente, die den durchfließenden Datenverkehr auf Protokoll-Ebene normalisiert, um den unerwünschten Abfluss von Daten über Seitenkanäle zu verhindern. 15

Container Siehe Definition 8: Ein Container ist eine einzelne Instanz eines abgeschlossenen Software-Artefakts (Container-Image), die gemeinsam mit anderen Containern auf einem virtuellen oder physischen Container-Host läuft und dessen Betriebssystem nutzt. Der Container enthält alle notwendigen binären Artefakte und Bibliotheken, um die Software auszuführen, aber im Gegensatz zu einer virtuellen Maschine weder einen Betriebssystemkern noch Bibliotheken zum direkten oder indirekten Zugriff auf die darunter liegende Hardware. 5, IV, V

Container-Engine Eine Container-Engine ist Software, die Benutzer- oder Maschinenanfragen zum Erzeugen, Aktualisieren, Starten, Stoppen, und Löschen von Containern entgegennimmt. Sie interpretiert das Format von Container-Images und Konfigurationsparametern und wandelt diese in ein geeignetes Format um, das mit einer Container-Runtime ausgeführt werden kann. Die Begriffe Container-Engine und Container-Runtime werden oft synonym gebraucht, wenn ihre Implementierung oder Aufgaben vollständig übereinstimmen. 5, IV

Container-Image Als Container-Image wird ein Software-Artefakt bezeichnet, das in einem spezifizierten Format alle binären Artefakte und Bibliotheken enthält, um Software auszuführen, die einen Service über ein Netzwerk bereitstellt. Im Gegensatz zu einem Virtual-Machine-Image enthält es nur die minimal notwendigen Komponenten, um diese Aufgabe zu erfüllen, es enthält insbesondere keine Betriebssytem-Bestandteile (Kernel, Hardware-Emulation, Initialisierungsdienste). Moderne Container-Image Formate sind geschichtet aufgebaut, damit unterschiedliche Container-Images, die gemeinsam genutzte Komponenten enthalten, platzsparend gespeichert und mit geringer Bandbreite übertragen und aktualisiert werden können. 3, 5, 37, IV

Container-Runtime Eine Container-Runtime ist eine Software-Komponente, die typischerweise in einer Container-Engine verwendet wird (zum Teil aber auch mit dieser identisch ist). Die Referenzimplementierung des OCI Runtime Standards ist *runc*. Eine Reihe üblicher Container-Engines verwendet *runc* direkt oder indirekt. Die Container-Runtime stellt die Schnittstelle zwischen ruhenden Container-Images und dem Betriebssystem-Kernel

dar. Sie alloziiert die notwendigen Ressourcen, konfiguriert die Parameter für die Isolierung und die Anbindung ans Netzwerk und steuert die notwendigen Systemprozesse, um Software in Form eines Containers auszuführen. 37, IV

Datendiode Netzwerkgerät, mit dem garantiert werden kann, dass Daten zwischen zwei Netzsegmenten unterschiedlicher Vertrauensstellung nur in eine Richtung fließen können. Datendioden unterschiedlicher Hersteller gehören zu den wenigen Geräten, die nach den höchsten Stufen der Common Criteria zertifiziert wurden. 15

Fuzzing Beim Fuzzing wird versucht, Software durch die Verarbeitung automatisiert erzeugter, ungültiger, defekter oder teilweise defekter Daten zu einer Fehlfunktion zu bringen. 10

Informationsverbund „Unter einem Informationsverbund ist die Gesamtheit von infrastrukturellen, organisatorischen, personellen und technischen Objekten zu verstehen, die der Aufgabenerfüllung in einem bestimmten Anwendungsbereich der Informationsverarbeitung dienen. Ein Informationsverbund kann dabei als Ausprägung die gesamte Institution oder auch einzelne Bereiche, die durch organisatorische Strukturen (z. B. Abteilungen) oder gemeinsame Geschäftsprozesse bzw. Anwendungen (z. B. Personalinformationssystem) gegliedert sind, umfassen.“ [BSI21, S. 34] 18

konform Als *konform* werden in diesem Kontext Organisationen, Abläufe und Systeme bezeichnet, wenn sie mindestens die Basis- und Standard-Anforderungen der für sie relevanten Bausteine des *Kompendium*s vollständig erfüllen. 17, 25

P-A-P-Struktur Struktur zur Kopplung von Netzsegmenten unterschiedlicher Vertrauensstufen, bei der ein äußerer Paketfilter auf der Seite der niedrigen Vertrauensstufe, ein ALG (Application-Layer-Gateway) und ein innerer Paketfilter auf der Seite der hohen Vertrauensstufe in Serie geschaltet werden. Der gesamte Netzwerkverkehr (eingehend und ausgehend) muss dabei alle drei Elemente passieren. Die Paketfilter arbeiten auf den OSI-Layern 3 und 4, das ALG arbeitet auf den OSI-Layern 5–7, terminiert und analysiert auch verschlüsselte Datenströme und befindet sich in einem Netzsegment, das *demilitarisierte Zone* (DMZ) genannt wird und weder vom äußeren noch vom inneren Netzsegment aus direkt angesprochen werden darf. 15, 28, 32, 40, 44 ff.

Zielobjekt „Zielobjekte sind Teile des Informationsverbunds, denen im Rahmen der Modellierung ein oder mehrere Bausteine aus dem IT-Grundschutz-Kompendium zugeordnet werden können. Zielobjekte können dabei physische Objekte sein, z. B. IT-Systeme. Häufig sind Zielobjekte jedoch logische Objekte, wie beispielsweise Organisationseinheiten, Anwendungen oder der gesamte Informationsverbund.“ [BSI21, S. 39] 18

A.3. Abkürzungen

ALG Application-Level-Gateway 15, 40, 45

API Application Programming Interface 3, 7, 10, 21, 34, 43, 50

ASVS Application Security Verification Standard 13

BSI Bundesamt für Sicherheit in der Informationstechnik (https://www.bsi.bund.de/) 1, 12, 15–20, 22, 26, 27, 32, 39, 40, 43 ff., 47, 48, 52, IX, X

BSIMM Building Security In Maturity Model 12

CC Common Criteria for Information Technology Security Evaluation 13, 29, 39, 40, 45

CIA Confidentiality, Integrity, Availability 13

CLASP Comprehensive Lightweight Application Security Process 12

CLI Command Line Interface 34

CNCF Cloud Native Computing Foundation (https://www.cncf.io/) 37, 38, 40, 42

CNI Container Network Interface 37

CSC Cloud Service Consumer 2

CSP Cloud Service Provider 2

CTCPEC Canadian Trusted Computer Product Evaluation Criteria 13

DLP Data Loss Prevention 15

DMZ Demilitarisierte Zone 44

DNS Domain Name Service 34, 35

DSGVO Datenschutz-Grundverordnung 13, XXIII

DSK Die Konferenz der unabhängigen Datenschutzbehörden des Bundes und der Länder (kurz: Datenschutzkonferenz) XXIII

FOSS Free and Open Source Software 29

GUI Graphical User Interface 7, 34, 43, 50

HIDS Host based Intrusion Detection System 14

HSM Hardware Security Module 15, 41

IaaS Infrastructure-as-a-Service 2

IAM Identity and Access Management 32, 49 ff.

ICAP Internet Content Adaptation Protocol 45

IDS Intrusion Detection System 9, 14

IP Internet Protocol 27, 34, 47

IPS Intrusion Prevention System 9, 14, 15, 52

ISB Informationssicherheitsbeauftragte/r 49

ISMS Information Security Management System 1, 13, 25, 48, 52

ISO International Organization for Standardization (https://www.iso.org/) 1

ITSEC Information Technology Security Evaluation Criteria 13

KMS Key Management System 32, 41

KVM Kernel-based Virtual Machine 40

LDAP Lightweight Directory Access Protocol 15

MTA Mail Transport Agent 26

NIDS Network based Intrusion Detection System 14

NIST National Institute of Standards and Technology (https://www.nist.gov/) 2

NTP Network Time Protocol 32, 33, 41

OCI Open Container Initiative (https://opencontainers.org/) 37, 42, IV

OSI Open Systems Interconnection 5, 15, 27

OWASP Open Web Application Security Project (https://owasp.org/) 12

PaaS Plattform-as-a-Service 2, 50

PDCA Plan — Do — Check — Act 12

PKI Public Key Infrastructure 41

RFC Request for Comment 19

RHEL Red Hat® Enterprise Linux® 39, 40, 42, 45

S3 Simple Storage Service (Amazon) 42

SaaS Software-as-a-Service 1–4, 6, 21, 23, 25, 26, 34, 42, 49, 50, XXIII, XXV

SAMM Software Assurance Maturity Model 12

SAN Storage Area Network 34

SDL Security Development Lifecycle 12

SDM Standard-Datenschutzmodell XXIII

SIEM Security Information and Event Management 8, 14, 52

SOC Security Operation Center 9

TCP Transmission Control Protocol 27

TCSEC Trusted Computer System Evaluation Criteria 13

VLAN Virtual Local Area Network 47

VPN Virtual Private Network 7, 23

VSAN Virtual Storage Area Network 34

WORM Write Once Read Multiple 50

A.4. Abbildungen und Tabellen

Abbildungsverzeichnis

Tabellenverzeichnis

B. Material zum IT-Grundschutz-Kompendium

B.1. SYS.1.6 Container: Community Draft

Bundesamt
für Sicherheit in der
Informationstechnik

Community Draft

SYS: IT-Systeme

SYS.1.6: Container

1 Beschreibung

1.1 Einleitung

Der Begriff „Container“ bezeichnet eine Technik, bei der ein Wirtssystem mehrere Anwendungen parallel in separierten Umgebungen ausführt (Operating Stem Level Virtualization). In den meisten Fällen erfolgt die Überwachung, das Starten und Beenden und die weitere Verwaltung der Container durch eine Verwaltungssoftware, die somit die sogenannte Orchestrierung übernimmt. Die Orchestrierung erfolgt dabei zumeist in Gruppen von gemeinsam verwalteten Container-Hosts in einem oder mehreren sogenannten Clustern. Ohne die automatisierte Orchestrierung mit der Verwaltungssoftware ist ein Betrieb von Containern zwar möglich, aber der manuelle Betrieb oder der Betrieb über eigens erstellte Skripte ist in der Praxis nur selten anzutreffen.

Um Container zu betreiben und zu verwalten, haben sich mehrere Produkte etabliert, die es erlauben, auch sehr große Umgebungen zu bedienen. Hier ist zwischen der eigentlichen Container-Runtime, die die Prozesse auf den Container-Hosts betreibt, und der Orchestrierung, die die Runtimes auf mehreren Container-Hosts steuert, zu unterscheiden.

Der Betrieb von Containern benötigt eine spezialisierte Infrastruktur, zu der z. B. Cluster-Betriebssoftware, Image Registries, Automatisierungswerkzeuge, Verwaltungsserver, Speichersysteme und virtuelle Netze sowie Server gehören.

1.2 Zielsetzung

Ziel dieses Bausteins ist der Schutz von Informationen, die in Containern verarbeitet, angeboten oder darüber übertragen werden. Der Baustein behandelt, wie Container grundsätzlich abgesichert, wie sie orchestriert und wie die verwendeten Images verwaltet werden können. Dabei wird zwischen dem eigentlichen Container-Dienst und der Cluster-Betriebssoftware, also der Software, die für Betrieb und Verwaltung der Container zuständig ist, und den Anwendungsdiensten, die in den Containern ausgeführt werden, unterschieden.

1.3 Abgrenzung und Modellierung

Der Baustein SYS.1.6 *Container* ist immer anzuwenden, wenn Serverdienste und -anwendungen in Containern betrieben werden.

Dieser Baustein betrachtet Container und ihre Orchestrierung unabhängig vom verwendeten Produkt, die Anforderungen orientieren sich aber an den Fähigkeiten derzeit im Markt befindlicher Produkte.

Der Baustein enthält grundsätzliche Anforderungen zur Einrichtung, zum Betrieb und zur Orchestrierung von Containern sowie zur Image Registry, dem Infrastrukturdienst für die Verwaltung

und Bereitstellung von Container-Images. Die weiteren im Container-Umfeld üblichen Dienste, wie z. B. Automatisierung für CI/CD-Pipelines und Codeverwaltung in GIT, behandelt dieser Baustein nicht in der Tiefe.

Er konkretisiert und ergänzt die Aspekte, die in den Bausteinen SYS.1.1 *Allgemeiner Server* und SYS.1.3 *Server unter Unix* sowie SYS.1.2.2 *Windows Server 2012* behandelt werden, um Spezifika von Containern. Die Anforderungen dieser Bausteine sollten von den Container-Wirten (Hosts) erfüllt werden, unabhängig davon, ob diese selbst auf physischen Servern ausgeführt werden oder virtualisiert sind.

Container werden unabhängig vom Einsatzzweck der darin betriebenen Dienste bzw. Anwendungen betrachtet. Sicherheitsanforderungen möglicher Dienste, wie z. B. Webserver (APP.3.2 *Webserver*) oder Server für Groupware (siehe APP.5.1 *Groupware*), sind Gegenstand eigener Bausteine. Das Thema Virtualisierung wird im Baustein SYS.1.5 *Server-Virtualisierung* beleuchtet.

Der Schwerpunkt des Bausteins liegt auf dem Betrieb von Serverdiensten und -anwendungen. Die Isolation von Anwendungen, wie Browsern auf Clients, wird nicht betrachtet.

2 Gefährdungslage

Folgende spezifische Bedrohungen und Schwachstellen sind im Bereich Container von besonderer Bedeutung:

2.1 Schwachstellen in Images

Container werden auf Basis von vorgefertigten Images erstellt, die häufig aus dem Internet bezogen oder selbst erstellt werden. In diesen Images ist die Software enthalten, aus denen der IT-Betrieb eigene Images erstellt oder die er um die zu betreibende Software ergänzt.

Die in den Images enthaltene Software könnte verwundbar und die aus dem Image erstellten Serverdienste könnten somit angreifbar sein. Solche Schwachstellen sind oft dem IT-Betrieb nicht bekannt, da die in den Images enthaltene Software nicht in der eigenen Software-Verwaltung erfasst ist.

Sind neue Schwachstellen in der enthaltenen Software vorhanden, ist es nur mit zusätzlichen Werkzeugen möglich, diese zu erkennen und in das Schwachstellenmanagement der Institution aufzunehmen.

2.2 Administrative Zugänge ohne Absicherung

Um Container-Dienste zu verwalten, benötigen die Administratoren und die toolgestützte Orchestrierung administrative Zugänge. Diese Zugänge sind entweder als Sockets oder Ports für Netzzugänge ausgeführt. Mechanismen zur Authentisierung und Verschlüsselung der administrativen Zugänge sind häufig vorhanden, aber nicht bei allen Produkten standardmäßig aktiviert.

Wenn Unbefugte auf das Datennetz oder auf die Container-Hosts zugreifen, können sie über ungeschützte administrativen Zugänge Befehle ausführen, die der Verfügbarkeit, Vertraulichkeit und Integrität der verarbeiteten Daten schaden.

2.3 Tool-basierte Orchestrierung ohne Absicherung

Sofern eine größere Anzahl von Containern in Betrieb ist, wird zumeist eine Software zur Orchestrierung zur Verwaltung der Container eingesetzt. Diese Software kann selbst über Schwachstellen verfügen oder nicht ausreichend gegen unbefugte Nutzung abgesichert sein.

Auf diese Weise kann ein Angreifer Befehle auf den Container-Hosts mit administrativen Berechtigungen ausführen. Dienste können abgeschaltet, Daten gelöscht oder eingesehen werden.

2.4 Ausbruch aus dem Container

Sollte ein Angreifer in der Lage sein, im Container eigenen Code auszuführen, kann er möglicherweise aus dem Container ausbrechen und auf den Container-Host zugreifen. Dieser Angriff kann z. B. über

Schwachstellen in Prozessoren, im Betriebssystem-Kernel oder in lokal angebotenen Infrastruktur-Diensten wie z. B. DNS oder SSH erfolgen.

In der Folge könnte ein Angreifer die Kontrolle über den Container-Host oder andere Server aus der Infrastruktur übernehmen und so unbefugt Daten einsehen, verändern und löschen oder andere Container, Hosts oder Infrastrukturdienste angreifen.

2.5 Datenverluste durch fehlende Persistenz

Container sind von ihrem Aufbau her dafür gedacht, nur eine bestimmte Zeit lang ausgeführt zu werden, und die Verwaltungssoftware kann sie jederzeit abschalten. Wird dies nicht beachtet, könnten Anwendungen in Containern Daten speichern, die sich ausschließlich im Container befinden. Wird eine neue Instanz des Containers gestartet, beispielsweise bei einem Update des Images oder der betriebenen Anwendung, sind alle diese Daten verloren.

Nutzdaten der Anwendung werden in der Regel geeignet gesichert. Bei dateibasierten Protokolldaten oder Zwischenergebnissen der Verarbeitung fällt eine fehlende Datensicherung nur dann auf, wenn ein Container beendet und entfernt ist und die enthaltenen Daten unwiderruflich verloren sind. Sind die Protokolldaten oder Zwischenergebnisse verloren, kann die Verarbeitung nicht lückenlos dokumentiert und deren Ergebnisse können unter Umständen nicht mehr nachvollzogen werden.

2.6 Vertraulichkeitsverlust von Zugangsdaten

Aufbau und Erstellung von Images für Container machen es oft notwendig, dass Zugangsdaten im Container verfügbar sind, z. B. für Datenbanken. Oft liegen sie dann ungeschützt im Image. Über die Images selbst, die Skripte zur Erstellung der Images oder die Versionskontrolle der Skripte könnten diese Zugangsdaten in unbefugte Hände gelangen.

3 Anforderungen

Im Folgenden sind die spezifischen Anforderungen des Bausteins SYS.1.6 *Container* aufgeführt. Grundsätzlich ist der IT-Betrieb für die Erfüllung der Anforderungen zuständig. Der Informationssicherheitsbeauftragte (ISB) ist bei strategischen Entscheidungen stets einzubeziehen. Außerdem ist der ISB dafür zuständig, dass alle Anforderungen gemäß dem festgelegten Sicherheitskonzept erfüllt und überprüft werden. Zusätzlich kann es noch andere Rollen geben, die weitere Zuständigkeiten bei der Umsetzung von Anforderungen haben. Diese sind dann jeweils explizit in eckigen Klammern in der Überschrift der jeweiligen Anforderungen aufgeführt.

Grundsätzlich zuständig	IT-Betrieb
Weitere Zuständigkeiten	-

3.1 Basis-Anforderungen

Die folgenden Anforderungen MÜSSEN für den Baustein *SYS.1.6 Container* vorrangig erfüllt werden.

SYS.1.6.A1 Planung des Container-Einsatzes (B)
Bevor Container eingesetzt werden, MÜSSEN alle sicherheitsrelevanten Aspekte der Installation, des Betriebs und der Außerbetriebnahme geplant werden. Diese Planung SOLLTE angemessen dokumentiert werden.

SYS.1.6.A2 Planung der Separierung der Anwendungen in Containern (B)
Vor der Inbetriebnahme MUSS geplant werden, wie die in Containern betriebenen Anwendungen und deren unterschiedlichen Test- und Produktions-Betriebsumgebungen separiert werden. Auf Basis des Schutzbedarfs der Anwendungen, des Netzzonenkonzepts und einer Risikobetrachtung MUSS die Planung festlegen, welche Architektur angemessen auf die Risiken eingeht.

SYS.1.6.A3 Planung der Verwaltung und Orchestrierung (B)

Die Verwaltung und Orchestrierung der Container DARF NUR nach einer geeigneten Planung erfolgen. Die Planung MUSS den gesamten Lebenszyklus von Inbetrieb- bis Außerbetriebnahme inklusive Betrieb und Updates umfassen. Die Orchestrierung MUSS die Container überwachen, starten, stoppen und nach festgelegten Regeln verschieben, um so die Verfügbarkeit der Dienste auch bei Ausfall eines Container-Hosts zu gewährleisten. Sie SOLLTE Schnittstellen für Automatisierungs-Software (CI/CD) anbieten.

SYS.1.6.A4 Härtung des Host-Systems (B)

Es DÜRFEN NUR für den Einsatz als Container-Host benötigte Dienste und Anwendungen auf dem Host-System installiert sein. Die Konfiguration des Host-Systems MUSS angemessen gehärtet werden.

Server für den Betrieb als Container-Host DÜRFEN KEINE Dienste betreiben, die nicht für den Betrieb der Container notwendig sind.

SYS.1.6.A5 Separierung der Container (B)

Der Betriebssystem-Kernel MUSS über Namespaces (wie Linux cgroups) oder andere geeignete Mechanismen die Container voneinander und von anderen Prozessen auf dem Container-Host trennen. Die Trennung MUSS dabei mindestens Prozess-IDs, Inter-Process-Kommunikation, Benutzer-IDs, Dateisystem und Netz inklusive Hostname umfassen.

Wenn der Container-Dienst oder die Cluster-Betriebssoftware mehrere Container miteinander in einer Einheit betreibt, dürfen diese Einheiten sich einen Namespace teilen.

SYS.1.6.A6 Verwendung sicherer Images (B)

Es MUSS sichergestellt sein, dass Images nur aus vertrauenswürdigen Verzeichnissen (Registries) stammen. Sie MÜSSEN unverändert und frei von bekannten Schwachstellen sind.

Signaturen MÜSSEN jedes Image gegen Veränderung und falsche Herausgeber absichern. Die Quelle MUSS danach ausgewählt werden, dass der Anbieter die enthaltene Software regelmäßig auf Sicherheitsprobleme prüft, diese behebt und dies seinen Kunden zusichert.

Die verwendete Version von Basis-Images DARF NICHT abgekündigt („deprecated“) sein. Es MÜSSEN Versionsnummern angegeben sein. Um Updates nicht zu behindern, SOLLTE die Versionsangabe NICHT die Minor-Version enthalten.

SYS.1.6.A7 Härtung der Software im Container (B)

Alle nicht benötigten Bestandteile der Anwendung bzw. des Dienstes, die bzw. der im Container ausgeführt wird, MÜSSEN deinstalliert werden. Die Konfiguration der Anwendung bzw. des Dienstes MUSS angemessen gehärtet werden.

SYS.1.6.A8 Persistenz von Protokollierungsdaten (B)

Die Protokollierung MUSS den Betrieb der Container-Hosts, der Containern und der Orchestrierung vollständig erfassen. Notwendige Protokollierungsdaten MÜSSEN persistent außerhalb der Container gespeichert werden.

SYS.1.6.A9 Persistenz von Nutzdaten (B)

Nutzdaten, auf die durch Anwendungen bzw. Dienste im Container zugegriffen wird oder die sie dauerhaft abspeichern, MÜSSEN persistent außerhalb des Containers gespeichert werden.

SYS.1.6.A10 Speicherung von Zugangsdaten (B)

Zugangsdaten MÜSSEN so gespeichert und verwaltet werden, dass nur berechtigte Personen und Container hierauf zugreifen können. Insbesondere MUSS sichergestellt sein, dass Zugangsdaten nur an zugangsgeschützten Orten und nicht in den Images liegen. Die von der Verwaltungssoftware bereitgestellten Verwaltungsmechanismen für Zugangsdaten SOLLTEN eingesetzt werden.

Folgende Zugangsdaten MÜSSEN mindestens berücksichtigt werden:

- Passwörter jeglicher Accounts,

- API-Keys für von der Anwendung genutzte Dienste sowie
- Private Schlüssel bei Public-Key Authentisierung.

3.2 Standard-Anforderungen

Gemeinsam mit den Basis-Anforderungen entsprechen die folgenden Anforderungen dem Stand der Technik für den Baustein SYS.1.6 *Container*. Sie SOLLTEN grundsätzlich erfüllt werden.

SYS.1.6.A11 Richtlinie für Betrieb und Images (S)
Es SOLLTE eine Richtlinie existieren, die die Anforderungen an den Betrieb der Container, der Cluster-Betriebssoftware und die Container-Images festlegt. Die Richtlinie SOLLTE auch Anforderungen an die Automatisierung der Verwaltung, der Bereitstellung der Images und des Betriebs enthalten.

SYS.1.6.A12 Nur eine Anwendung bzw. ein Dienst pro Container (S)
Jeder Container SOLLTE jeweils nur eine Anwendung bzw. einen Dienst bereitstellen. Eine Gruppierung von mehreren Anwendungen bzw. Diensten in eine funktionale Einheit SOLLTE erfolgen, indem die Orchestrierung die Container als eine Gruppe betreibt.

SYS.1.6.A13 Freigabe von Images und Konfigurationen (S)
Alle Images für den produktiven Betrieb SOLLTEN einen geeigneten Freigabeprozess durchlaufen. Änderungen an den Konfigurationsdateien, die Betrieb, Architektur, Images und Datennetze definieren, SOLLTEN ebenfalls in den Freigabeprozess integriert werden.

SYS.1.6.A14 Updates von Containern (S)
Wenn sicherheitsrelevante Updates der zugrundeliegenden Images oder der betriebenen Software bzw. des betriebenen Dienstes erscheinen, SOLLTEN die Images für die Container neu erstellt und daraus neue Container instanziiert werden. Container auf Basis veralteter Images SOLLTEN dann beendet werden.

Beim Start eines Containers SOLLTE der Container-Dienst immer auf die aktuell verfügbare Version des Images prüfen und eine vorhandene neue Version herunterladen. Auf dem Container-Host zwischengespeicherte alte Versionen DÜRFEN dann NICHT gestartet werden.

SYS.1.6.A15 Unveränderlichkeit der Container (S)
Die Container SOLLTEN ihr Dateisystem während der Laufzeit nicht verändern können.

SYS.1.6.A16 Limitierung der Ressourcen pro Container (S)
Für jeden Container SOLLTEN Ressourcen auf dem Host-System, wie CPU sowie flüchtiger und persistenter Speicher, angemessen limitiert werden.

SYS.1.6.A17 Einbinden von Massenspeichern in Container (S)
Die Container SOLLTEN NUR auf die für den Betrieb notwendigen Massenspeicher und Verzeichnisse zugreifen können. Wenn Schreibrechte nicht benötigt werden, SOLLTEN diese entfernt werden. Sofern Container lokale Speicher einbinden, SOLLTEN die Zugriffsrechte im Dateisystem auf den Service-Account des Containers eingeschränkt sein. Bei Netzspeicher SOLLTE diese Berechtigung auf dem Netzspeicher gesetzt sein.

SYS.1.6.A18 Absicherung der Wirk- und Administrations-Netze (S)
Die Netze für die Administration der Hosts, die Administration des Container-Dienstes und die einzelnen Netze der Anwendungsdienste SOLLTEN separiert werden.

Es SOLLTEN NUR die für den Betrieb notwendigen Netzports der Container in die dafür vorgesehenen Produktivnetze freigegeben werden.

Die zur Administration der Container-Hosts, der Container-Dienste und der Cluster-Betriebssoftware notwendigen Netzports SOLLTEN NUR aus dem Administrationsnetz erreichbar sein, Ausnahme sind hier die Container der Cluster-Betriebssoftware inklusive der Container des Netz-Managements („CNI“), die per SSH, mit lokalen Agenten der Cluster-Betriebssoftware und der Datenhaltung der

Cluster-Betriebssoftware oder dem Container-Host kommunizieren.

SYS.1.6.A19 Verwendung vorgelagerter Ein- und Ausgangssysteme (S)
Sofern mehrere Container-Hosts in einem Verbund arbeiten („Cluster"), SOLLTEN dedizierte Container-Hosts die Ein- und Ausgabe zu anderen Netzen übernehmen. Die anderen Container-Hosts SOLLTEN NICHT aus anderen Netzen außer dem Administrationsnetz erreichbar sein.

SYS.1.6.A20 Absicherung von Konfigurationsdaten und Automatisierung (S)
Die Beschreibung der Container-Konfigurationsdaten SOLLTE versioniert und annotiert erfolgen. Zugangsrechte auf die Verwaltungssoftware der Konfigurationen SOLLTEN minimal vergeben werden. Nur der notwendige Kreis von Personen SOLLTE die Berechtigung haben, Prozesse der Automatisierung auszulösen.

SYS.1.6.A21 Container-Ausführung ohne Privilegien (S)
Alle Anwendungsdienste in Containern SOLLTEN nur unter einem nicht privilegierten Account gestartet werden. Sie SOLLTEN NICHT über erweiterte Privilegien für die Container-Dienste oder die Cluster-Betriebssoftware verfügen.

SYS.1.6.A22 Absicherung von Hilfsprozessen der Automatisierung (S)
Alle Prozesse der Automatisierungssoftware SOLLTEN nur mit minimalen Rechten arbeiten. Wenn unterschiedliche Benutzergruppen über die Automatisierungssoftware die Konfiguration verändern können, SOLLTE dies für jede Gruppe durch eigene Prozesse durchgeführt werden, die nur die für die jeweilige Benutzergruppe notwendigen Rechte besitzen.

SYS.1.6.A23 Administrativer Fernzugriff auf Container (S)
Es SOLLTE sichergestellt sein, dass der administrative Fernzugriff nur auf die Cluster-Betriebssoftware oder den Container-Host und nur über diese auf die Container selbst erfolgen kann. Container SOLLTEN selbst keine Dienste für administrativen Fernzugriff enthalten.

SYS.1.6.A24 Identitäts- und Berechtigungsmanagement für die Container-Verwaltung (S)
Die Verwaltungssoftware SOLLTE jede Aktion eines Benutzers oder im automatisierten Betrieb einer entsprechenden Software authentifizieren und autorisieren, unabhängig davon, ob die Aktionen über eine Weboberfläche oder über eine API erfolgt. Aktionen SOLLTEN NICHT anonym erfolgen.

SYS.1.6.A25 Service-Accounts für Container (S)
Container SOLLTEN jeweils eigene Service-Accounts nutzen, um miteinander und mit den Diensten der Cluster-Betriebssoftware authentifiziert zu kommunizieren, Gruppen von Containern können, wenn sie gleiche Aufgaben haben, einen gemeinsamen Service-Account nutzen. Berechtigungen für die Service-Accounts SOLLTEN nur minimal vergeben sein.

Jeder Dienst, der einen Service-Account nutzt, SOLLTE ein eigenes Token erhalten.

SYS.1.6.A26 Accounts der Anwendungsdienste in Containern (S)
Die Accounts der Prozesse in den Containern SOLLTEN keine Berechtigungen auf dem Container-Host haben. Wenn dies dennoch notwendig ist, SOLLTEN diese Berechtigungen nur für unbedingt notwendigen Daten gelten.

SYS.1.6.A27 Überwachung der Container (S)
Jedes Image SOLLTE einen Health-Check für den Start und den Betrieb („readiness" und „liveness") definieren. Diese Checks SOLLTEN Auskunft über die Verfügbarkeit der Anwendung im Container geben. Sie SOLLTEN fehlschlagen, wenn die Anwendung nicht in der Lage ist, ihre Aufgaben ordnungsgemäß wahrzunehmen.

Der Container-Dienst oder die Cluster-Betriebssoftware SOLLTEN diese Checks überwachen und Container, bei denen die Checks fehlschlagen, beenden und durch neue Instanzen ersetzen.

SYS.1.6.A28 Absicherung der Registry für Images (S)
Sofern eine eigene Registry für Images eingesetzt wird, SOLLTE diese ausreichend abgesichert sein.

Dabei SOLLTEN beachtet werden:

- Verwendung von personenbezogenen und Service-Accounts für den Zugang,
- minimale Vergabe der Berechtigungen,
- Anbindung der Software für die Überwachung der Images auf Verwundbarkeiten,
- Protokollierung der Veränderungen der Images und
- die Datensicherung.

3.3 Anforderungen bei erhöhtem Schutzbedarf

Im Folgenden sind für den Baustein SYS.1.6 *Container* exemplarische Vorschläge für Anforderungen aufgeführt, die über das dem Stand der Technik entsprechende Schutzniveau hinausgehen und BEI ERHÖHTEM SCHUTZBEDARF in Betracht gezogen werden SOLLTEN. Die konkrete Festlegung erfolgt im Rahmen einer Risikoanalyse.

SYS.1.6.A29 Automatisierte Auditierung von Containern (H)
Die gesamte Software in den Images SOLLTE automatisiert katalogisiert werden. Sie SOLLTE mindestens täglich mit aktualisierten Datenbanken über bekannte Verwundbarkeiten abgeglichen werden. Auch die Einstellungen des Containers selbst sowie die des betriebenen Anwendungsdienstes SOLLTEN automatisiert mit einer Liste der erlaubten Einstellungen abgeglichen werden. Nur Container, die geeignet überprüft wurden, SOLLTEN für den Einsatz im Produktivbetrieb freigegeben werden. Die Orchestrierung SOLLTE diese Prozesse automatisieren.

SYS.1.6.A30 Eigene Trusted Registry für Container (H)
Images SOLLTEN nur in einem eigenen Verzeichnis (Registry) bereitgestellt werden. Es SOLLTE durch technische Maßnahmen sichergestellt sein, dass nur Images aus dieser Registry eingesetzt werden.

SYS.1.6.A31 Erstellung erweiterter Richtlinien für Container (H)
Erweiterte Richtlinien SOLLTEN die Berechtigungen der Container und der betriebenen Anwendungsdienste einschränken. Die Richtlinien SOLLTEN folgende Zugriffe einschränken:

- Netzverbindungen,
- Dateisystem-Zugriffe und
- Kernel-Anfragen (Syscalls).

SYS.1.6.A32 Host Based Intrusion Detection für Container (H)
Die Container und die betriebenen Anwendungsdienste SOLLTEN überwacht werden. Abweichungen vom normalen Verhalten SOLLTEN erkannt und gemeldet werden. Verdächtige Container SOLLTEN automatisch beendet und neu gestartet werden.

Das zu überwachende Verhalten SOLLTE umfassen:

- Netzverbindungen,
- Dateisystem-Zugriffe und
- Kernel-Anfragen (Syscalls).

SYS.1.6.A33 Mikro-Segmentierung von Containern (H)
Die Container SOLLTEN nur über die notwendigen Netzports miteinander kommunizieren können. Es SOLLTEN innerhalb der virtuellen Netze Regeln existieren, die alle bis auf die für den Betrieb notwendigen Netzverbindungen unterbinden. Die Regeln SOLLTEN Quelle und Ziel der Verbindungen genau definieren und dafür mindestens die Service-Namen, Meta-Daten („Labels“) oder die Service-Accounts verwenden. Sofern möglich SOLLTEN die Regeln eine zertifikatsbasierte Authentifizierung vorschreiben und nur die in den Zertifikaten hinterlegten Identitäten für die Definition der erlaubten Verbindungen nutzen.

Alle Kriterien, die als Bezeichnung für diese Verbindung dienen, SOLLTEN so abgesichert sein, dass nur

berechtige Personen und Verwaltungs-Dienste diese Kriterien setzen dürfen.

SYS.1.6.A34 Hochverfügbarkeit von Containern (H)
Der Containerbetrieb SOLLTE so aufgebaut sein, dass bei Ausfall eines Rechenzentrums die Anwendungen in den Containern in kurzer Zeit an einem anderen Standort neu anlaufen können. Dafür SOLLTEN alle notwendigen Konfigurationsdateien, Images, Nutzdaten, Netzverbindungen und sonstige für den Betrieb benötigten Ressourcen inklusive der zum Betrieb nötigen Hardware an diesem Standort verfügbar sein.

SYS.1.6.A35 Verschlüsselte Datenhaltung bei Containern (H)
Die Dateisysteme mit den persistenten Daten der Anwendungsdienste SOLLTEN verschlüsselt sein.

SYS.1.6.A36 Verschlüsselung der Netzkommunikation zwischen Containern (H)
Daten, die über virtuelle oder physische Netze zwischen den Containern übertragen werden, SOLLTEN verschlüsselt werden. Die Verbindungen SOLLTEN mit Zertifikaten authentifiziert werden.

4 Weiterführende Informationen

4.1 Wissenswertes

Weiterführende Informationen zu Gefährdungen und Sicherheitsmaßnahmen im Bereich Container finden sich unter anderem in folgenden Veröffentlichungen:

- NIST 800-190
  https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-190.pdf
- CIS Benchmark Docker
  https://www.cisecurity.org/benchmark/docker/
- CIS Benchmark Kubernetes
  https://www.cisecurity.org/benchmark/kubernetes/
- OCI – Open Container Initiative
  https://www.opencontainers.org/
- CNCF – Cloud Native Computing Foundation
  https://www.cncf.io/

5 Anlage: Kreuzreferenztabelle zu elementaren Gefährdungen

Die Kreuzreferenztabelle enthält die Zuordnung von elementaren Gefährdungen zu den Anforderungen. Anhand dieser Tabelle lässt sich ermitteln, welche elementaren Gefährdungen durch welche Anforderungen abgedeckt sind. Durch die Umsetzung der aus den Anforderungen abgeleiteten Sicherheitsmaßnahmen wird den entsprechenden elementaren Gefährdungen entgegengewirkt. Die Buchstaben in der zweiten Spalte (C = Vertraulichkeit, I = Integrität, A = Verfügbarkeit) zeigen an, welche Grundwerte der Informationssicherheit durch die Anforderung vorrangig geschützt werden.

Die folgenden elementaren Gefährdungen sind für Container von Bedeutung.

G 0.14 Ausspähen von Informationen / Spionage

G 0.19 Offenlegung schützenswerter Informationen

G 0.20 Informationen oder Produkte aus unzuverlässiger Quelle

G 0.21 Manipulation von Hard- oder Software

G 0.23 Unbefugtes Eindringen in IT-Systeme

G 0.25 Ausfall von Geräten oder Systemen

G 0.27 Ressourcenmangel

G 0.28 Software-Schwachstellen oder -Fehler

G 0.30 Unberechtigte Nutzung oder Administration von Geräten und Systemen

G 0.37 Abstreiten von Handlungen

G 0.39 Schadprogramme

G 0.45 Datenverlust

G 0.46 Integritätsverlust schützenswerter Informationen

Elementare Gefährdugen Anforderungen	CIA	G 0.14	G 0.19	G 0.20	G 0.21	G 0.23	G 0.25	G 0.27	G 0.28	G 0.30	G 0.37	G 0.39	G 0.45	G 0.46
SYS.1.6.A1		X	X	X	X	X		X	X	X	X	X	X	X
SYS.1.6.A2		X			X			X				X		X
SYS.1.6.A3				X	X					X				
SYS.1.6.A4		X			X	X				X		X		
SYS.1.6.A5		X				X						X		X
SYS.1.6.A6				X	X				X					
SYS.1.6.A7		X			X	X			X			X		
SYS.1.6.A8									X		X			
SYS.1.6.A9													X	X
SYS.1.6.A10		X				X				X				
SYS.1.6.A11		X	X	X	X	X		X		X	X	X	X	X
SYS.1.6.A12			X											
SYS.1.6.A13				X	X	X		X				X	X	X
SYS.1.6.A14		X				X			X			X		
SYS.1.6.A15		X				X						X		
SYS.1.6.A16								X						
SYS.1.6.A17			X										X	
SYS.1.6.A18		X	X			X				X				
SYS.1.6.A19		X			X	X				X			X	X
SYS.1.6.A20				X	X			X		X			X	X
SYS.1.6.A21						X			X				X	X
SYS.1.6.A22		X				X				X				
SYS.1.6.A23		X				X				X				
SYS.1.6.A24						X				X	X			
SYS.1.6.A25		X			X	X				X				
SYS.1.6.A26		X			X	X				X				
SYS.1.6.A27								X					X	X
SYS.1.6.A28				X		X				X	X			
SYS.1.6.A29	CIA	X		X		X		X	X	X		X	X	X
SYS.1.6.A30	CIA			X					X			X		
SYS.1.6.A31	CIA	X	X			X							X	X
SYS.1.6.A32	CIA	X	X			X			X	X				
SYS.1.6.A33	CI	X	X			X			X					
SYS.1.6.A34	A						X							
SYS.1.6.A35	C	X	X											X
SYS.1.6.A36	C	X	X											X

B.2. SYS.1.6 Container: CD Mapping

Mapping der Anforderungen zwischen aktuellem und ersten Community Draft des Bausteins SYS.1.6 Container

Neue Anforderungen sind fett markiert.

Aktueller Community Draft	Erster Community Draft
A1 Planung des Container-Einsatzes (B)	A1 Planung des Container-Einsatzes
A2 Planung der Separierung der Anwendungen in Containern (B)	A2 Planung der Separierung
A3 Planung der Verwaltung und Orchestrierung (B)	A19 Planung der Verwaltung und Orchestrierung
A4 Härtung des Host-Systems (B)	A3 Härtung des Host-Systems
A5 Separierung der Container (B)	-
A6 Verwendung sicherer Images (B)	A7 Verwendung sicherer Images
A7 Härtung der Software im Container (B)	A4 Härtung der Software im Container
A8 Persistenz von Protokollierungsdaten (B)	A5 Persistenz von Protokollierungsdaten
A9 Persistenz von Nutzdaten (B)	A6 Persistenz von Nutzdaten
A10 Speicherung von Zugangsdaten (B)	A8 Speicherung von Zugangsdaten
A11 Richtlinie für Betrieb und Images (S)	-
A12 Nur eine Anwendung bzw. ein Dienst pro Container (S)	A18 Nur ein Dienst pro Container
A13 Freigabe von Images und Konfigurationen (S)	A12 Freigabe von Images
A14 Updates von Containern (S)	A13 Updates von Containern
A15 Unveränderlichkeit der Container (S)	-
A16 Limitierung der Ressourcen pro Container (S)	A20 Limitierung der Ressourcen pro Container (A)
A17 Einbinden von Massenspeichern in Container (S)	A10 Einbinden von Volumes
A18 Absicherung der Wirk- und Administrations-Netze (S)	A9 Separierung der Netze
A19 Verwendung vorgelagerter Ein- und Ausgangssysteme (S)	-
A20 Absicherung von Konfigurationsdaten und Automatisierung (S)	-
A21 Container-Ausführung ohne Privilegien (S)	A17 Container-Ausführung ohne privilegierten Account
A22 Absicherung von Hilfsprozessen der Automatisierung (S)	-

A23 Administrativer Fernzugriff auf Container (S)	A11 Administrativer Fernzugriff auf Container
A24 Identitäts- und Berechtigungsmanagement für die Container-Verwaltung (S)	A15 Identitätsmanagement der Administratoren
A25 Service-Accounts für Container (S)	-
A26 Accounts der Anwendungsdienste in Containern (S)	A16 Accounts der Anwendungsdienste
A27 Überwachung der Container (S)	-
A28 Absicherung der Registry für Images (S)	-
A29 Automatisierte Auditierung von Containern (H)	A21 Automatisierte Auditierung (CIA)
A30 Eigene Trusted Registry für Container (H)	A22 Eigene Trusted Registry (CIA)
A31 Erstellung erweiterter Richtlinien für Container (H)	A24 Erstellung erweiterter Richtlinien für Container (CIA)
A32 Host Based Intrusion Detection für Container (H)	A25 Host Based Intrusion Detection (CIA)
A33 Mikro-Segmentierung von Containern (H)	-
A34 Hochverfügbarkeit von Containern (H)	A26 Hochverfügbarkeit (A)
A35 Verschlüsselte Datenhaltung bei Containern (H)	A27 Verschlüsselte Datenhaltung (C)
A36 Verschlüsselung der Netzkommunikation zwischen Containern (H)	A14 Verschlüsselung der Netzkommunikation
Abgedeckt durch SYS.1.6.A21 und SYS.1.6.A26	A23 Reduzierte Rechte (CIA)

B.3. Liste der Bausteine und Anforderungen

Nachfolgend werden *alle* Bausteine des *Kompendiums* (4. Ausgabe, Edition 2021) in der dort publizierten Reihenfolge aufgeführt.[1] Es wird tabellarisch dargestellt, welche Bausteine und Anforderungen für den Aufbau und Betrieb von Container-Plattformen relevant sind und welche davon in dieser Arbeit berücksichtigt wurden. Für Bausteine, die insgesamt nicht berücksichtigt wurden (siehe Abschnitt 3.2), werden keine Anforderungen einzeln aufgeführt. Diese können im frei verfügbaren *Kompendium* nachgelesen werden [BSI21].

Anforderungen, die mit *N* gekennzeichnet sind, haben keine direkte Relevanz für eine Container-Plattform. *O* bezeichnet Anforderungen, die ausgeschlossen wurden, weil sie rein organisatorische Maßnahmen beschreiben. Anforderungen, die mit *J* gekennzeichnet sind, wären auf eine Container-Plattform anwendbar. Sie wurden nicht berücksichtigt, weil sie auf Komponenten außerhalb des Informationsverbundes anzuwenden sind. Mit + markierte Anforderungen wurden im Rahmen dieser Arbeit verwendet.

ISMS:Sicherheitsmanagement

Der Baustein ISMS.1: *Sicherheitsmanagement* wird nicht betrachtet und seine Anforderungen müssen vollständig erfüllt werden. Dies wird als gegeben vorausgesetzt.

ORP:Organisation und Personal

Die Bausteine ORP.1: *Organisation*, ORP.2: *Personal*, ORP.3: *Sensibilisierung und Schulung zur Informationssicherheit* und ORP.5: *Compliance Management (Anforderungsmanagement)* werden nicht betrachtet und ihre Anforderungen müssen vollständig erfüllt werden. Dies wird als gegeben vorausgesetzt.

Gegenüber der 2. Ausgabe (Edition 2019) wurden die beiden Anforderungen ORP.4.A22: *Regelung zur Passwortqualität* und ORP.4.A23: *Regelung für Passwort-verarbeitende Anwendungen und IT-Systeme* nun als Basis-Anforderung qualifiziert und sie erscheinen daher an anderer Position.

ORP.4		**Identitäts- und Berechtigungsmanagement**	
ORP.4.A1	B	Regelung für die Einrichtung und Löschung von Benutzern und Benutzergruppen	O
ORP.4.A2	B	Einrichtung, Änderung und Entzug von Berechtigungen	O
ORP.4.A3	B	Dokumentation der Benutzerkennungen und Rechteprofile	O
ORP.4.A4	B	Aufgabenverteilung und Funktionstrennung	+
ORP.4.A5	B	Vergabe von Zutrittsberechtigungen	O
ORP.4.A6	B	Vergabe von Zugangsberechtigungen	O
ORP.4.A7	B	Vergabe von Zugriffsrechten	O
ORP.4.A8	B	Regelung des Passwortgebrauchs	O
ORP.4.A9	B	Identifikation und Authentisierung	+
ORP.4.A22	B	Regelung zur Passwortqualität	O
ORP.4.A23	B	Regelung für Passwort-verarbeitende Anwendungen und IT-Systeme	+
ORP.4.A10	S	Schutz von Benutzerkennungen mit weitreichenden Berechtigungen	+
ORP.4.A11	S	Zurücksetzen von Passwörtern	O
ORP.4.A12	S	Entwicklung eines Authentisierungskonzeptes für IT-Systeme und Anwendungen	+
ORP.4.A13	S	Geeignete Auswahl von Authentisierungsmechanismen	+

[1] Die laufend nummerierten Anforderungen wurden zwischen den Editionen teilweise neu qualifiziert und erscheinen daher nicht immer in aufsteigender Reihenfolge, weil sie gruppiert nach Basis-, Standard- und erhöhten Anforderungen aufgeführt werden.

ORP.4.A14	S	Kontrolle der Wirksamkeit der Benutzertrennung am IT-System bzw. Anwendung	O
ORP.4.A15	S	Vorgehensweise und Konzeption der Prozesse beim Identitäts- und Berechtigungsmanagement	+
ORP.4.A16	S	Richtlinien für die Zugriffs- und Zugangskontrolle	O
ORP.4.A17	S	Geeignete Auswahl von Identitäts- und Berechtigungsmanagement-Systemen	O
ORP.4.A18	S	Einsatz eines zentralen Authentisierungsdienstes	+
ORP.4.A19	S	Einweisung aller Mitarbeiter in den Umgang mit Authentisierungsverfahren und -mechanismen	O
ORP.4.A20	H	Notfallvorsorge für das Identitäts-und Berechtigungsmanagement-System	O
ORP.4.A21	H	Mehr-Faktor-Authentisierung	+
ORP.4.A24	H	Vier-Augen-Prinzip für administrative Tätigkeiten	+

CON:Konzepte und Vorgehensweisen

Die einzige Anforderung im Baustein CON.2: *Datenschutz* verlangt, das Standard-Datenschutzmodell (SDM[2]) der DSK[3] umzusetzen. Diese ist organisatorischer Art und der Baustein wird daher nicht berücksichtigt, obwohl im Rahmen eines SaaS-Angebots regelmäßig davon ausgegangen werden kann, dass dieser Baustein relevant ist, weil personenbezogene Daten verarbeitet werden.[4]

Der Baustein CON.4: *Auswahl und Einsatz von Standard-Software* wurde überarbeitet und als APP.6: *Allgemeine Software* neu verfasst, ebenso wie der alte Baustein CON.5: *Entwicklung und Einsatz von Individualsoftware* als neuer Baustein APP.7: *Entwicklung von Individualsoftware* erscheint.

Der Baustein CON.6: *Löschen und Vernichten* betrifft den Umgang mit Datenträgern und wird nicht berücksichtigt, weil angenommen wurde, dass die Hardware von einem Hosting-Provider sachgemäß bereitgestellt wird.

Die Bausteine CON.7: *Informationssicherheit auf Auslandsreisen* und CON.9: *Informationsaustausch* beschreiben ausschließlich organisatorische Anforderungen und werden ebenfalls nicht berücksichtigt.

Der neue Baustein CON.10: *Entwicklung von Webanwendungen* wird trotz seiner grundsätzlichen Relevanz für SaaS-Lösungen nicht betrachtet, weil die Software-Entwicklung selbst nicht Gegenstand dieser Arbeit ist.

CON.1		**Kryptokonzept**	
CON.1.A1	B	Auswahl geeigneter kryptografischer Verfahren	+
CON.1.A2	B	Datensicherung bei Einsatz kryptografischer Verfahren	+
CON.1.A3	S	Verschlüsselung der Kommunikationsverbindungen	+
CON.1.A4	S	Geeignetes Schlüsselmanagement	+
CON.1.A5	S	Sicheres Löschen und Vernichten von kryptografischen Schlüsseln	+
CON.1.A6	H	Bedarfserhebung für kryptografische Verfahren und Produkte	O
CON.1.A7	H	Erstellung einer Sicherheitsrichtlinie für den Einsatz kryptografischer Verfahren und Produkte	O

[2] Standard-Datenschutzmodell

[3] Die Konferenz der unabhängigen Datenschutzbehörden des Bundes und der Länder (kurz: Datenschutzkonferenz)

[4] Dies gilt besonders vor dem Hintergrund, dass die DSGVO den Begriff der personenbezogenen Daten sehr weit fasst.

CON.1.A9	H	Auswahl eines geeigneten kryptografischen Produkts	+
CON.1.A10	H	Entwicklung eines Kryptokonzepts	+
CON.1.A11	H	Sichere Konfiguration der Kryptomodule	+
CON.1.A12	H	Sichere Rollenteilung beim Einsatz von Kryptomodulen	+
CON.1.A13	H	Anforderungen an die Betriebssystem-Sicherheit beim Einsatz von Kryptomodulen	+
CON.1.A14	H	Schulung von Benutzern und Administratoren	O
CON.1.A15	H	Reaktion auf praktische Schwächung eines Kryptoverfahrens	O
CON.1.A16	H	Physische Absicherung von Kryptomodulen	D
CON.1.A17	H	Abstrahlsicherheit	D
CON.1.A18	H	Kryptografische Ersatzmodule	O
CON.3		**Datensicherungskonzept**	
CON.3.A1	B	Erhebung der Einflussfaktoren für Datensicherungen	O
CON.3.A2	B	Festlegung der Verfahrensweise für die Datensicherung	+
CON.3.A3	B	entfallen	E
CON.3.A4	B	Erstellung eines Minimaldatensicherungskonzeptes	+
CON.3.A5	B	Regelmäßige Datensicherung	+
CON.3.A6	S	Entwicklung eines Datensicherungskonzepts	O
CON.3.A7	S	Beschaffung eines geeigneten Datensicherungssystems	O
CON.3.A8	B	entfallen	E
CON.3.A9	S	Voraussetzungen für die Online-Datensicherung	O
CON.3.A10	S	Verpflichtung der Mitarbeiter zur Datensicherung	O
CON.3.A11	S	Sicherungskopie der eingesetzten Software	+
CON.3.A12	S	Geeignete Aufbewahrung der Datenträger von Datensicherungen	O
CON.3.A13	H	Einsatz kryptografischer Verfahren bei der Datensicherung	+
CON.8		**Software-Entwicklung**	
CON.8.A2	B	Auswahl eines Vorgehensmodells	O
CON.8.A3	B	Auswahl einer Entwicklungsumgebung	O
CON.8.A4	B	entfallen	E
CON.8.A5	B	Sicheres Systemdesign	+
CON.8.A6	B	Verwendung von externen Bibliotheken aus vertrauenswürdigen Quellen	+
CON.8.A7	B	Durchführung von entwicklungsbegleitenden Softwaretests	+
CON.8.A8	B	Bereitstellung von Patches, Updates und Änderungen	+
CON.8.A9	B	entfallen	E
CON.8.A10	B	Versionsverwaltung des Quellcodes	+
CON.8.A20	B	Überprüfung von externen Komponenten	+
CON.8.A1	B	Definition von Rollen und Zuständigkeiten	O
CON.8.A11	S	Erstellung einer Richtlinie für die Software-Entwicklung	O
CON.8.A12	S	Ausführliche Dokumentation	O
CON.8.A13	S	entfallen	E
CON.8.A14	S	Schulung des Projektteams zur Informationssicherheit	O
CON.8.A15	S	entfallen	E
CON.8.A16	S	Geeignete Steuerung der Software-Entwicklung	O
CON.8.A21	S	Bedrohungsmodellierung	O
CON.8.A22	S	Sicherer Software-Entwurf	O
CON.8.A17	H	Auswahl vertrauenswürdiger Entwicklungswerkzeuge	O
CON.8.A18	H	Regelmäßige Sicherheitsaudits für die Entwicklungsumgebung	O
CON.8.A19	H	Regelmäßige Integritätsprüfung der Entwicklungsumgebung	O

OPS:Betrieb

Der Baustein OPS.1.1.1 *Allgemeiner IT-Betrieb* wurde noch 2018 als *Community-Draft* öffentlich diskutiert, wurde aber nicht in das *Kompendium* aufgenommen. *OPS.1.2.2* und OPS.1.2.4: *Telearbeit* wurden ausdrücklich von der Betrachtung ausgeschlossen, es ist aber davon auszugehen, dass beide auf einen typischen SaaS-Anbieter anwendbar sind. Ein Szenario, das die Anwendung von OPS.2.2: *Cloud-Nutzung* erfordert, wurde ausgeschlossen. Auf den Baustein OPS.1.2.1 wird bis zur 2. Ausgabe (Edition 2019) einmal als *Patch- und Änderungsmanagement* und einmal als *Änderungsmanagement* verwiesen, dieser Baustein wurde aber nie publiziert. Der Baustein OPS.1.2.3: *Informations- und Datenträgeraustausch* erschien letztmalig in der 2 .Ausgabe (Edition 2019) und wurde in der 3. Ausgabe (Edition 2020) sinngemäß durch CON.9: *Informationsaustausch* ersetzt, allerdings ohne dass diese Änderung explizit in der Einleitung erwähnt wurde.

OPS.1.1.2		**Ordnungsgemäße IT-Administration**	
OPS.1.1.2.A1	B	entfallen	E
OPS.1.1.2.A2	B	Vertretungsregelungen und Notfallvorsorge	O
OPS.1.1.2.A3	B	Geregelte Einstellung von IT-Administratoren	O
OPS.1.1.2.A4	B	Beendigung der Tätigkeit als IT-Administrator	O
OPS.1.1.2.A5	B	Nachweisbarkeit von administrativen Tätigkeiten	+
OPS.1.1.2.A6	B	Schutz administrativer Tätigkeiten	+
OPS.1.1.2.A7	S	Regelung der IT-Administrationstätigkeit	O
OPS.1.1.2.A8	S	Administration von Fachanwendungen	+
OPS.1.1.2.A9	S	Ausreichende Ressourcen für den IT-Betrieb	O
OPS.1.1.2.A10	S	Fortbildung und Information	O
OPS.1.1.2.A11	S	Dokumentation von IT-Administrationstätigkeiten	O
OPS.1.1.2.A12	S	Regelungen für Wartungs- und Reparaturarbeiten	O
OPS.1.1.2.A13	S	entfallen	E
OPS.1.1.2.A20	S	Verwaltung und Inbetriebnahme von Geräten	O
OPS.1.1.2.A14	H	Sicherheitsüberprüfung von Administratoren	O
OPS.1.1.2.A15	H	Aufteilung von Administrationstätigkeiten	+
OPS.1.1.2.A16	H	Zugangsbeschränkungen für administrative Zugänge	+
OPS.1.1.2.A17	H	IT-Administration im Vier-Augen-Prinzip	+
OPS.1.1.2.A18	H	Durchgängige Protokollierung administrativer Tätigkeiten	J
OPS.1.1.2.A19	H	Berücksichtigung von Hochverfügbarkeitsanforderungen	+
OPS.1.1.3		**Patch- und Änderungsmanagement**	
OPS.1.1.3.A1	B	Konzept für das Patch- und Änderungsmanagement	+
OPS.1.1.3.A2	B	Festlegung der Zuständigkeiten	+
OPS.1.1.3.A3	B	Konfiguration von Autoupdate-Mechanismen	+
OPS.1.1.3.A15	B	Regelmäßige Aktualisierung von IT-Systemen und Software	+
OPS.1.1.3.A16	B	Regelmäßige Suche nach Informationen zu Patches und Schwachstellen	O
OPS.1.1.3.A4	S	entfallen	E
OPS.1.1.3.A5	S	Umgang mit Änderungsanforderungen	+
OPS.1.1.3.A6	S	Abstimmung von Änderungsanforderungen	+
OPS.1.1.3.A7	S	Integration des Änderungsmanagements in die Geschäftsprozesse	O
OPS.1.1.3.A8	S	Sicherer Einsatz von Werkzeugen für das Patch- und Änderungsmanagement	+
OPS.1.1.3.A9	S	Test- und Abnahmeverfahren für neue Hardware	D

OPS.1.1.3.A10	S	Sicherstellung der Integrität und Authentizität von Softwarepaketen	+
OPS.1.1.3.A11	S	Kontinuierliche Dokumentation der Informationsverarbeitung	+
OPS.1.1.3.A12	H	Einsatz von Werkzeugen beim Änderungsmanagement	+
OPS.1.1.3.A13	H	Erfolgsmessung von Änderungsanforderungen	O
OPS.1.1.3.A14	H	Synchronisierung innerhalb des Änderungsmanagements	+
OPS.1.1.4		**Schutz vor Schadprogrammen**	
OPS.1.1.4.A1	B	Erstellung eines Konzepts für den Schutz vor Schadprogrammen	+
OPS.1.1.4.A2	B	Nutzung systemspezifischer Schutzmechanismen	+
OPS.1.1.4.A3	B	Auswahl eines Virenschutzprogrammes	n/a
OPS.1.1.4.A4	B	entfallen	E
OPS.1.1.4.A5	B	Betrieb und Konfiguration von Virenschutzprogrammen	+
OPS.1.1.4.A6	B	Regelmäßige Aktualisierung der eingesetzten Virenschutzprogramme und Signaturen	+
OPS.1.1.4.A7	B	Sensibilisierung und Verpflichtung der Benutzer	O
OPS.1.1.4.A8	S	entfallen	E
OPS.1.1.4.A9	S	Meldung von Infektionen mit Schadprogrammen	+
OPS.1.1.4.A10	H	Nutzung spezieller Analyseumgebungen	+
OPS.1.1.4.A11	H	Einsatz mehrerer Scan-Engines	+
OPS.1.1.4.A12	H	Einsatz von Datenträgerschleusen	D
OPS.1.1.4.A13	H	Umgang mit nicht vertrauenswürdigen Dateien	O
OPS.1.1.4.A14	H	Auswahl und Einsatz von Cyber-Sicherheitsprodukten gegen gezielte Angriffe	O
OPS.1.1.4.A15	H	entfallen	E
OPS.1.1.5		**Protokollierung**	
OPS.1.1.5.A1	B	Erstellung einer Sicherheitsrichtlinie für die Protokollierung	O
OPS.1.1.5.A2	B	entfallen	E
OPS.1.1.5.A3	B	Konfiguration der Protokollierung auf System- und Netzebene	J
OPS.1.1.5.A4	B	Zeitsynchronisation der IT-Systeme	+
OPS.1.1.5.A5	B	Einhaltung rechtlicher Rahmenbedingungen	J
OPS.1.1.5.A6	S	Aufbau einer zentralen Protokollierungsinfrastruktur	J
OPS.1.1.5.A7	S	entfallen	E
OPS.1.1.5.A8	S	Archivierung von Protokollierungsdaten	J
OPS.1.1.5.A9	S	Bereitstellung von Protokollierungsdaten für die Auswertung	J
OPS.1.1.5.A10	S	Zugriffsschutz für Protokollierungsdaten	J
OPS.1.1.5.A11	H	Steigerung des Protokollierungsumfangs	J
OPS.1.1.5.A12	H	Verschlüsselung der Protokollierungsdaten	J
OPS.1.1.5.A13	H	Hochverfügbare Protokollierungsinfrastruktur	J
OPS.1.1.6		**Software-Tests und -Freigaben**	
OPS.1.1.6.A1	B	Planung der Software-Tests	O
OPS.1.1.6.A2	B	Durchführung von funktionalen Software-Tests	+
OPS.1.1.6.A3	B	Auswertung der Testergebnisse	+
OPS.1.1.6.A4	B	Freigabe der Software	+
OPS.1.1.6.A5	B	Durchführung von Software-Tests für nicht funktionale Anforderungen	+
OPS.1.1.6.A11	B	Verwendung von anonymisierten oder pseudonymisierten Testdaten	O
OPS.1.1.6.A6	S	Geordnete Einweisung der Software-Tester	O
OPS.1.1.6.A7	S	Personalauswahl der Software-Tester	O

OPS.1.1.6.A8	S	entfallen	E
OPS.1.1.6.A9	S	entfallen	E
OPS.1.1.6.A10	S	Erstellung eines Abnahmeplans	+
OPS.1.1.6.A12	S	Durchführung von Regressionstests	+
OPS.1.1.6.A13	S	Trennung der Testumgebung von der Produktivumgebung	+
OPS.1.1.6.A15	S	Überprüfung der Installation und zugehörigen Dokumentation	O
OPS.1.1.6.A14	H	Durchführung von Penetrationstests	O
OPS.1.1.6.A16	H	Sicherheitsüberprüfung der Tester	O
OPS.1.2.5	**Fernwartung**		
OPS.1.2.5.A1	B	Planung des Einsatzes der Fernwartung	O
OPS.1.2.5.A2	B	Sicherer Verbindungsaufbau bei der Fernwartung von Clients	n/a
OPS.1.2.5.A3	B	Absicherung der Schnittstellen zur Fernwartung	J
OPS.1.2.5.A4	B	entfallen	E
OPS.1.2.5.A5	S	Einsatz von Online-Diensten	J
OPS.1.2.5.A6	S	Erstellung einer Richtlinie für die Fernwartung	O
OPS.1.2.5.A7	S	Dokumentation bei der Fernwartung	O
OPS.1.2.5.A8	S	Sichere Protokolle bei der Fernwartung	J
OPS.1.2.5.A9	S	Auswahl und Beschaffung geeigneter Fernwartungswerkzeuge	O
OPS.1.2.5.A10	S	Verwaltung der Fernwartungswerkzeuge	O
OPS.1.2.5.A11	S	entfallen	E
OPS.1.2.5.A12	S	entfallen	E
OPS.1.2.5.A13	S	entfallen	E
OPS.1.2.5.A15	S	entfallen	E
OPS.1.2.5.A16	S	entfallen	E
OPS.1.2.5.A17	S	Authentisierungsmechanismen bei der Fernwartung	J
OPS.1.2.5.A18	S	entfallen	E
OPS.1.2.5.A19	S	Fernwartung durch Dritte	n/a
OPS.1.2.5.A20	S	Betrieb der Fernwartung	J
OPS.1.2.5.A21	S	Erstellung eines Notfallplans für den Ausfall der Fernwartung	O
OPS.1.2.5.A24	S	Absicherung integrierter Fernwartungssysteme	J
OPS.1.2.5.A25	S	Entkopplung der Netzmanagement-Kommunikation bei der Fernwartung	J
OPS.1.2.5.A14	H	Dedizierte Clients bei der Fernwartung	n/a
OPS.1.2.5.A22	H	Redundante Kommunikationsverbindungen	D
OPS.1.2.5.A23	H	entfallen	E
OPS.2.1	**Outsourcing für Kunden**		
OPS.2.1.A1	B	Festlegung der Sicherheitsanforderungen für Outsourcing-Vorhaben	J
OPS.2.1.A2	S	Rechtzeitige Beteiligung der Personalvertretung	O
OPS.2.1.A3	S	Auswahl eines geeigneten Outsourcing-Dienstleisters	J
OPS.2.1.A4	S	Vertragsgestaltung mit dem Outsourcing-Dienstleister	O
OPS.2.1.A5	S	Festlegung einer Strategie zum Outsourcing	J
OPS.2.1.A6	S	Erstellung eines Sicherheitskonzepts für das Outsourcing-Vorhaben	O
OPS.2.1.A7	S	Festlegung der möglichen Kommunikationspartner	O
OPS.2.1.A8	S	Regelungen für den Einsatz des Personals des Outsourcing-Dienstleiters	O
OPS.2.1.A9	S	Vereinbarung über die Anbindung an Netze der Outsourcing-Partner	O

OPS.2.1.A10	S	Vereinbarung über Datenaustausch zwischen den Outsourcing-Partnern	O
OPS.2.1.A11	S	Planung und Aufrechterhaltung der Informationssicherheit im laufenden Outsourcing-Betrieb	O
OPS.2.1.A12	S	Änderungsmanagement	O
OPS.2.1.A13	S	Sichere Migration bei Outsourcing-Vorhaben	O
OPS.2.1.A14	S	Notfallvorsorge beim Outsourcing	O
OPS.2.1.A15	S	Geordnete Beendigung eines Outsourcing-Verhältnisses	O
OPS.2.1.A16	H	Sicherheitsüberprüfung von Mitarbeitern	O
OPS.3.1		**Outsourcing für Dienstleister**	
OPS.3.1.A1	B	Erstellung eines Grobkonzeptes für die Outsourcing-Dienstleistung	+
OPS.3.1.A2	S	Vertragsgestaltung mit den Outsourcing-Kunden	O
OPS.3.1.A3	S	Erstellung eines Sicherheitskonzepts für das Outsourcing-Vorhaben	+
OPS.3.1.A4	S	Festlegung der möglichen Kommunikationspartner	O
OPS.3.1.A5	S	Regelungen für den Einsatz des Personals des Outsourcing-Dienstleisters	O
OPS.3.1.A6	S	Regelungen für den Einsatz von Fremdpersonal	O
OPS.3.1.A7	S	Erstellung eines Mandantentrennungskonzeptes durch den Outsourcing-Dienstleister	O
OPS.3.1.A8	S	Vereinbarung über die Anbindung an Netze der Outsourcing-Partner	O
OPS.3.1.A9	S	Vereinbarung über Datenaustausch zwischen den Outsourcing-Partnern	O
OPS.3.1.A10	S	Planung und Aufrechterhaltung der Informationssicherheit im laufenden Outsourcing-Betrieb	O
OPS.3.1.A11	S	Zutritts-, Zugangs- und Zugriffskontrolle	D
OPS.3.1.A12	S	Änderungsmanagement	+
OPS.3.1.A13	S	Sichere Migration bei Outsourcing-Vorhaben	O
OPS.3.1.A14	S	Notfallvorsorge beim Outsourcing	O
OPS.3.1.A15	S	Geordnete Beendigung eines Outsourcing-Verhältnisses	O
OPS.3.1.A16	H	Sicherheitsüberprüfung von Mitarbeitern	O

DER:Detektion und Reaktion

Nicht betrachtet werden die Bausteine DER.2.1: *Behandlung von Sicherheitsvorfällen*, DER.2.2: *Vorsorge für die IT-Forensik*, DER.2.3: *Bereinigung weitreichender Sicherheitsvorfälle*, DER.3.1: *Audits und Revisionen*, DER.3.2: *Revisionen auf Basis des Leitfadens IS-Revision* und DER.4: *Notfallmanagement*.

Der Baustein DER.1 stellt eine Reihe von Anforderungen, die auf eine Container-Plattform insgesamt anwendbar wären. Aufgrund der Definition des Informationsverbunds braucht keine davon betrachtet werden.

DER.1		**Detektion von sicherheitsrelevanten Ereignissen**	
DER.1.A1	B	Erstellung einer Sicherheitsrichtlinie für die Detektion von sicherheitsrelevanten Ereignissen	O
DER.1.A2	B	Einhaltung rechtlicher Bedingungen bei der Auswertung von Protokolldaten	O
DER.1.A3	B	Festlegung von Meldewegen für sicherheitsrelevante Ereignisse	J
DER.1.A4	B	Sensibilisierung der Mitarbeiter	O
DER.1.A5	B	Einsatz von mitgelieferten Systemfunktionen zur Detektion	J

DER.1.A6	S	Kontinuierliche Überwachung und Auswertung von Protokolldaten	J
DER.1.A7	S	Schulung von Verantwortlichen	O
DER.1.A8	S	entfallen	E
DER.1.A9	S	Einsatz zusätzlicher Detektionssysteme	J
DER.1.A10	S	Einsatz von TLS-/SSH-Proxies	J
DER.1.A11	S	Nutzung einer zentralen Protokollierungsinfrastruktur für die Auswertung sicherheitsrelevanter Ereignisse	J
DER.1.A12	S	Auswertung von Informationen aus externen Quellen	O
DER.1.A13	S	Regelmäßige Audits der Detektionssysteme	O
DER.1.A14	H	Auswertung der Protokolldaten durch spezialisiertes Personal	J
DER.1.A15	H	Zentrale Detektion und Echtzeitüberprüfung von Ereignismeldungen	J
DER.1.A16	H	Einsatz von Detektionssystemen nach Schutzbedarfsanforderungen	J
DER.1.A17	H	Automatische Reaktion auf sicherheitsrelevante Ereignisse	J
DER.1.A18	H	Durchführung regelmäßiger Integritätskontrollen	O

APP:Anwendungen

Keine Relevanz haben die Bausteine APP.1.1: *Office-Produkte*, APP.1.2: *Webbrowser*, APP.1.4: *Mobile Anwendungen (Apps)*, APP.2.2: *Active Directory*, APP.2.3: *OpenLDAP*, APP.3.3: *Fileserver*, APP.3.4: *Samba*, APP.4.2: *SAP-ERP-System*, APP.4.6: *SAP ABAP-Programmierung*, APP.5.2: *Microsoft Exchange und Outlook*, und APP.5.3: *Allgemeiner E-Mail-Client und -Server*. Die Bausteine APP.1.3, APP.3.5, APP.4.1, APP.4.4 und APP.4.5 wurden nicht veröffentlicht. Der Baustein APP.5.1 wurde zu APP.5.3 verschoben und umbenannt. Unter der in Abschnitt 3.2 getroffenen Annahme braucht der Baustein APP.7: *Entwicklung von Individualsoftware* nicht berücksichtigt werden.

APP.2.1		**Allgemeiner Verzeichnisdienst**	
APP.2.1.A1	B	Erstellung einer Sicherheitsrichtlinie für Verzeichnisdienste	O
APP.2.1.A2	B	Planung des Einsatzes von Verzeichnisdiensten	+
APP.2.1.A3	B	Einrichtung von Zugriffsberechtigungen auf Verzeichnisdienste	O
APP.2.1.A4	B	Sichere Installation von Verzeichnisdiensten	+
APP.2.1.A5	B	Sichere Konfiguration und Konfigurationsänderungen von Verzeichnisdiensten	O
APP.2.1.A6	B	Sicherer Betrieb von Verzeichnisdiensten	+
APP.2.1.A7	S	Erstellung eines Sicherheitskonzepts für den Einsatz von Verzeichnisdiensten	O
APP.2.1.A8	S	Planung einer Partitionierung und Replikation im Verzeichnisdienst	+
APP.2.1.A9	S	Geeignete Auswahl von Komponenten für Verzeichnisdienste	+
APP.2.1.A10	S	entfallen	E
APP.2.1.A11	S	Einrichtung des Zugriffs auf Verzeichnisdienste	+
APP.2.1.A12	S	Überwachung von Verzeichnisdiensten	J
APP.2.1.A13	S	Absicherung der Kommunikation mit Verzeichnisdiensten	+
APP.2.1.A14	S	Geregelte Außerbetriebnahme eines Verzeichnisdienstes	O
APP.2.1.A15	S	Migration von Verzeichnisdiensten	O
APP.2.1.A16	H	Erstellung eines Notfallplans für den Ausfall eines Verzeichnisdienstes	O
APP.3.1		**Webanwendungen**	
APP.3.1.A1	B	Authentisierung bei Webanwendungen	+
APP.3.1.A2	B	entfallen	E
APP.3.1.A3	B	entfallen	E

APP.3.1.A4	B	Kontrolliertes Einbinden von Daten und Inhalten bei Webanwendungen	+
APP.3.1.A5	B	entfallen	E
APP.3.1.A6	B	entfallen	E
APP.3.1.A7	B	Schutz vor unerlaubter automatisierter Nutzung von Webanwendungen	+
APP.3.1.A14	B	Schutz vertraulicher Daten	+
APP.3.1.A16	B	entfallen	E
APP.3.1.A19	B	entfallen	E
APP.3.1.A8	S	Systemarchitektur einer Webanwendung	+
APP.3.1.A9	S	Beschaffung, Entwicklung und Erweiterung von Webanwendungen	O
APP.3.1.A10	S	entfallen	E
APP.3.1.A11	S	Sichere Anbindung von Hintergrundsystemen	+
APP.3.1.A12	S	Sichere Konfiguration von Webanwendungen	+
APP.3.1.A13	S	entfallen	E
APP.3.1.A15	S	entfallen	E
APP.3.1.A17	S	entfallen	E
APP.3.1.A18	S	entfallen	E
APP.3.1.A21	S	Sichere HTTP-Konfiguration bei Webanwendungen	+
APP.3.1.A22	S	Penetrationstest und Revision	O
APP.3.1.A23	S	entfallen	E
APP.3.1.A20	H	Einsatz von Web Application Firewalls	+
APP.3.1.A24	H	entfallen	E
APP.3.1.A25	H	entfallen	E
APP.3.2	**Webserver**		
APP.3.2.A1	B	Sichere Konfiguration eines Webservers	+
APP.3.2.A2	B	Schutz der Webserver-Dateien	+
APP.3.2.A3	B	Absicherung von Datei-Uploads und -Downloads	+
APP.3.2.A4	B	Protokollierung von Ereignissen (B)	J
APP.3.2.A5	B	Authentisierung	+
APP.3.2.A6	B	entfallen	E
APP.3.2.A7	B	Rechtliche Rahmenbedingungen für Webangebote	O
APP.3.2.A11	B	Verschlüsselung über TLS	+
APP.3.2.A8	S	Planung des Einsatzes eines Webservers	O
APP.3.2.A9	S	Festlegung einer Sicherheitsrichtlinie für den Webserver	O
APP.3.2.A10	S	Auswahl eines geeigneten Webhosters	n/a
APP.3.2.A12	S	Geeigneter Umgang mit Fehlern und Fehlermeldungen	+
APP.3.2.A13	S	Zugriffskontrolle für Webcrawler	+
APP.3.2.A14	S	Integritätsprüfungen und Schutz vor Schadsoftware	+
APP.3.2.A16	S	Penetrationstest und Revision	O
APP.3.2.A20	S	Benennung von Ansprechpartnern	O
APP.3.2.A15	H	Redundanz	+
APP.3.2.A17	H	entfallen	E
APP.3.2.A18	H	Schutz vor Denial-of-Service-Angriffen	D
APP.3.2.A19	H	entfallen	E
APP.3.6	**DNS-Server**		
APP.3.6.A1	B	Planung des DNS-Einsatzes	+
APP.3.6.A2	B	Einsatz redundanter DNS-Server	+

APP.3.6.A3	B	Verwendung von separaten DNS-Servern für interne und externe Anfragen	+
APP.3.6.A4	B	Sichere Grundkonfiguration eines DNS-Servers	+
APP.3.6.A5	B	entfallen	E
APP.3.6.A6	B	Absicherung von dynamischen DNS-Updates	+
APP.3.6.A7	B	Überwachung von DNS-Servern	J
APP.3.6.A8	B	Verwaltung von Domainnamen	O
APP.3.6.A9	B	Erstellen eines Notfallplans für DNS-Server	O
APP.3.6.A10	S	Auswahl eines geeigneten DNS-Server-Produktes	+
APP.3.6.A11	S	Ausreichende Dimensionierung der DNS-Server	+
APP.3.6.A12	S	entfallen	E
APP.3.6.A13	S	Einschränkung der Sichtbarkeit von Domain-Informationen	+
APP.3.6.A14	S	Platzierung der Nameserver	+
APP.3.6.A15	S	Auswertung der Logdaten	J
APP.3.6.A16	S	Integration eines DNS-Servers in eine „P-A-P“-Struktur	+
APP.3.6.A17	S	Einsatz von DNSSEC	+
APP.3.6.A18	S	Erweiterte Absicherung von Zonentransfers	n/a
APP.3.6.A19	S	Aussonderung von DNS-Servern	O
APP.3.6.A20	H	Prüfung des Notfallplans auf Durchführbarkeit	O
APP.3.6.A21	H	Hidden-Master	n/a
APP.3.6.A22	H	Anbindung der DNS-Server über unterschiedliche Provider	O
APP.4.3		**Relationale Datenbanksysteme**	
APP.4.3.A1	B	Erstellung einer Sicherheitsrichtlinie für Datenbanksysteme	O
APP.4.3.A2	B	entfallen	E
APP.4.3.A3	B	Basishärtung des Datenbankmanagementsystems	+
APP.4.3.A4	B	Geregeltes Anlegen neuer Datenbanken	+
APP.4.3.A5	B	entfallen	E
APP.4.3.A6	B	entfallen	E
APP.4.3.A7	B	entfallen	E
APP.4.3.A8	B	entfallen	E
APP.4.3.A9	B	Datensicherung eines Datenbanksystems	+
APP.4.3.A10	S	entfallen	E
APP.4.3.A11	S	Ausreichende Dimensionierung der Hardware	D
APP.4.3.A12	S	Einheitlicher Konfigurationsstandard von Datenbankmanagementsystemen	+
APP.4.3.A13	S	Restriktive Handhabung von Datenbank-Links	n/a
APP.4.3.A14	S	entfallen	E
APP.4.3.A15	S	entfallen	E
APP.4.3.A16	S	Verschlüsselung der Datenbankanbindung	+
APP.4.3.A17	S	Datenübernahme oder Migration	O
APP.4.3.A18	S	Überwachung des Datenbankmanagementsystem	J
APP.4.3.A19	S	Schutz vor schädlichen Datenbank-Skripten	O
APP.4.3.A20	S	Regelmäßige Audits	O
APP.4.3.A21	H	Einsatz von Datenbank Security Tools	J
APP.4.3.A22	H	Notfallvorsorge	O
APP.4.3.A23	H	Archivierung	n/a
APP.4.3.A24	H	Datenverschlüsselung in der Datenbank	+
APP.4.3.A25	H	Sicherheitsüberprüfungen von Datenbanksystemen	O

APP.6	**Allgemeine Software**		
APP.6.A1	B	Planung des Software-Einsatzes	O
APP.6.A2	B	Erstellung eines Anforderungskatalogs für Software	O
APP.6.A3	B	Sichere Beschaffung von Software	O
APP.6.A4	B	Regelung für die Installation und Konfiguration von Software	+
APP.6.A5	B	Sichere Installation von Software	+
APP.6.A6	S	Berücksichtigung empfohlener Sicherheitsanforderungen	O
APP.6.A7	S	Auswahl und Bewertung potenzieller Software	+
APP.6.A8	S	Regelung zur Verfügbarkeit der Installationsdateien	O
APP.6.A9	S	Inventarisierung von Software	O
APP.6.A10	S	Erstellung einer Sicherheitsrichtlinie für den Einsatz der Software	O
APP.6.A11	S	Verwendung von Plug-ins und Erweiterungen	+
APP.6.A12	S	Geregelte Außerbetriebnahme von Software	O
APP.6.A13	S	Deinstallation von Software	+
APP.6.A14	H	Nutzung zertifizierter Software	+

SYS:IT-Systeme

Nicht betrachtet werden die Bausteine SYS.1.2.2: *Windows Server 2012*, SYS.1.7: *IBM Z-System*, sowie die Bausteine der Gruppen SYS.2: *Desktop-Systeme* (Client-Systeme), SYS.3: *Mobile Devices* (mobile Geräte), SYS.4: *Sonstige Systeme* (Endgeräte ohne universelle Ein-/Ausgabeschnittstelle). Der Bausteine SYS.1.4 wurde nicht veröffentlicht. Obwohl der Baustein SYS.1.6: *Container* ebenfalls unveröffentlicht ist, findet er hier Berücksichtigung. Einerseits ist er unmittelbar für diese Arbeit relevant, andererseits ist davon auszugehen, dass in Kürze sein *Final Draft* erscheinen wird und er in dieser oder wenig abgeänderter Form in der nächsten Ausgabe des *Kompendium*s aufgenommen werden wird.

SYS.1.1	**Allgemeiner Server**		
SYS.1.1.A1	B	Geeignete Aufstellung	D
SYS.1.1.A2	B	Benutzerauthentisierung an Servern	+
SYS.1.1.A3	B	entfallen	E
SYS.1.1.A4	B	entfallen	E
SYS.1.1.A5	B	Schutz von Schnittstellen	+
SYS.1.1.A6	B	Deaktivierung nicht benötigter Dienste und Kennungen	+
SYS.1.1.A7	B	entfallen	E
SYS.1.1.A8	B	entfallen	E
SYS.1.1.A9	B	Einsatz von Virenschutz-Programmen auf Servern	+
SYS.1.1.A10	B	Protokollierung	J
SYS.1.1.A11	S	Festlegung einer Sicherheitsrichtlinie für Server	O
SYS.1.1.A12	S	Planung des Server-Einsatzes	D
SYS.1.1.A13	S	Beschaffung von Servern	D
SYS.1.1.A14	S	entfallen	E
SYS.1.1.A15	S	Unterbrechungsfreie und stabile Stromversorgung	D
SYS.1.1.A16	S	Sichere Grundkonfiguration von Servern	O
SYS.1.1.A17	S	entfallen	E
SYS.1.1.A18	S	entfallen	E
SYS.1.1.A19	S	Einrichtung lokaler Paketfilter	+
SYS.1.1.A20	S	entfallen	E
SYS.1.1.A21	S	Betriebsdokumentation für Server	D

SYS.1.1.A22	S	Einbindung in die Notfallplanung	O
SYS.1.1.A23	S	Systemüberwachung und Monitoring von Servern	+
SYS.1.1.A24	S	Sicherheitsprüfungen	O
SYS.1.1.A25	S	Geregelte Außerbetriebnahme eines Servers	D
SYS.1.1.A35	S	Erstellung und Pflege eines Betriebshandbuchs	D
SYS.1.1.A26	H	entfallen	E
SYS.1.1.A27	H	Hostbasierte Angriffserkennung	J
SYS.1.1.A28	H	Steigerung der Verfügbarkeit durch Redundanz	+
SYS.1.1.A29	H	entfallen	E
SYS.1.1.A30	H	Ein Dienst pro Server	+
SYS.1.1.A31	H	Application Whitelisting	+
SYS.1.1.A32	H	entfallen	E
SYS.1.1.A33	H	Aktive Verwaltung der Wurzelzertifikate	O
SYS.1.1.A34	H	Festplattenverschlüsselung	+
SYS.1.1.A36	H	Absicherung des Bootvorgangs	+
SYS.1.3	**Server unter Linux und Unix**		
SYS.1.3.A1	B	entfallen	E
SYS.1.3.A2	B	Sorgfältige Vergabe von IDs	+
SYS.1.3.A3	B	Kein automatisches Einbinden von Wechsellaufwerken	n/a
SYS.1.3.A4	B	Schutz vor Ausnutzung von Schwachstellen in Anwendungen	+
SYS.1.3.A5	B	Sichere Installation von Software-Paketen	+
SYS.1.3.A6	B	Verwaltung von Benutzern und Gruppen	+
SYS.1.3.A7	S	entfallen	E
SYS.1.3.A8	S	Verschlüsselter Zugriff über Secure Shell	+
SYS.1.3.A9	S	entfallen	E
SYS.1.3.A10	S	Verhinderung der Ausbreitung bei der Ausnutzung von Schwachstellen	+
SYS.1.3.A11	S	entfallen	E
SYS.1.3.A12	S	entfallen	E
SYS.1.3.A14	H	Verhinderung des Ausspähens von System- und Benutzerinformationen	+
SYS.1.3.A15	H	entfallen	E
SYS.1.3.A16	H	Zusätzliche Verhinderung der Ausbreitung bei der Ausnutzung von Schwachstellen	+
SYS.1.3.A17	H	Zusätzlicher Schutz des Kernels	+
SYS.1.5	**Virtualisierung**		
SYS.1.5.A1	B	entfallen	E
SYS.1.5.A2	B	Sicherer Einsatz virtueller IT-Systeme	+
SYS.1.5.A3	B	Sichere Konfiguration virtueller IT-Systeme	+
SYS.1.5.A4	B	Sichere Konfiguration eines Netzes für virtuelle Infrastrukturen	+
SYS.1.5.A5	B	Schutz der Administrationsschnittstellen	+
SYS.1.5.A6	B	Protokollierung in der virtuellen Infrastruktur	J
SYS.1.5.A7	B	Zeitsynchronisation in virtuellen IT-Systemen	+
SYS.1.5.A8	S	Planung einer virtuellen Infrastruktur	+
SYS.1.5.A9	S	Netzplanung für virtuelle Infrastrukturen	+
SYS.1.5.A10	S	Einführung von Verwaltungsprozessen für virtuelle IT-Systeme	O
SYS.1.5.A11	S	Administration der Virtualisierungsinfrastruktur über ein gesondertes Managementnetz	+

SYS.1.5.A12	S	Rechte- und Rollenkonzept für die Administration einer virtuellen Infrastruktur	+
SYS.1.5.A13	S	Auswahl geeigneter Hardware für Virtualisierungsumgebungen	D
SYS.1.5.A14	S	Einheitliche Konfigurationsstandards für virtuelle IT-Systeme	+
SYS.1.5.A15	S	Betrieb von Gast-Betriebssystemen mit unterschiedlichem Schutzbedarf	n/a
SYS.1.5.A16	S	Kapselung der virtuellen Maschinen	+
SYS.1.5.A17	S	Überwachung des Betriebszustands und der Konfiguration der virtuellen Infrastruktur	J
SYS.1.5.A18	S	entfallen	E
SYS.1.5.A19	S	Regelmäßige Audits der Virtualisierungsinfrastruktur	O
SYS.1.5.A20	H	Verwendung von hochverfügbaren Architekturen	+
SYS.1.5.A21	H	Sichere Konfiguration virtueller IT-Systeme bei erhöhtem Schutzbedarf	+
SYS.1.5.A22	H	Härtung des Virtualisierungsservers	+
SYS.1.5.A23	H	Rechte-Einschränkung der virtuellen Maschinen	+
SYS.1.5.A24	H	Deaktivierung von Snapshots virtueller IT-Systeme	+
SYS.1.5.A25	H	Minimale Nutzung von Konsolenzugriffen auf virtuelle IT-Systeme	+
SYS.1.5.A26	H	Einsatz einer PKI	+
SYS.1.5.A27	H	Einsatz zertifizierter Virtualisierungssoftware	+
SYS.1.5.A28	H	Verschlüsselung von virtuellen IT-Systemen	+
SYS.1.6	**Container**		
SYS.1.6.A1	B	Planung des Container-Einsatzes	+
SYS.1.6.A2	B	Planung der Separierung der Anwendungen in Container	+
SYS.1.6.A3	B	Planung der Verwaltung und Orchestrierung	+
SYS.1.6.A4	B	Härtung des Host-Systems	+
SYS.1.6.A5	B	Separierung der Container	+
SYS.1.6.A6	B	Verwendung sicherer Images	+
SYS.1.6.A7	B	Härtung der Software im Container	J
SYS.1.6.A8	B	Persistenz von Protokollierungsdaten	+
SYS.1.6.A9	B	Persistenz von Nutzdaten	+
SYS.1.6.A10	B	Speicherung von Zugangsdaten	+
SYS.1.6.A11	S	Richtlinie für Betrieb und Images	O
SYS.1.6.A12	S	Nur eine Anwendung bzw. ein Dienst pro Container	J
SYS.1.6.A13	S	Freigabe von Images und Konfigurationen	+
SYS.1.6.A14	S	Updates von Containern	+
SYS.1.6.A15	S	Unveränderlichkeit der Container	+
SYS.1.6.A16	S	Limitierung der Ressourcen pro Container	+
SYS.1.6.A17	S	Einbinden von Massenspeichern in Container	+
SYS.1.6.A18	S	Absicherung der Wirk- und Administrations-Netze	+
SYS.1.6.A19	S	Verwendung vorgelagerter Ein- und Ausgangssysteme	+
SYS.1.6.A20	S	Absicherung von Konfigurationsdaten und Automatisierung	+
SYS.1.6.A21	S	Container-Ausführung ohne Privilegien	+
SYS.1.6.A22	S	Absicherung von Hilfsprozessen der Automatisierung	+
SYS.1.6.A23	S	Administrativer Fernzugriff auf Container	+
SYS.1.6.A24	S	Identitäts- und Berechtigungsmanagement für die Container-Verwaltung	+
SYS.1.6.A25	S	Service-Accounts für Container	+

SYS.1.6.A26	S	Accounts der Anwendungsdienste in Containern	+
SYS.1.6.A27	S	Überwachung der Container	+
SYS.1.6.A28	S	Absicherung der Registry für Images	+
SYS.1.6.A29	H	Automatisierte Auditierung von Containern	+
SYS.1.6.A30	H	Eigene Trusted Registry für Container	+
SYS.1.6.A31	H	Erstellung erweiterter Richtlinien für Container	+
SYS.1.6.A32	H	Host Based Intrusion Detection für Container	J
SYS.1.6.A33	H	Mikro-Segmentierung von Containern	+
SYS.1.6.A34	H	Hochverfügbarkeit von Containern	+
SYS.1.6.A35	H	Verschlüsselte Datenhaltung bei Containern	+
SYS.1.6.A36	H	Verschlüsselung der Netzkommunikation zwischen Containern	+
SYS.1.8		**Speicherlösungen**	
SYS.1.8.A1	B	Geeignete Aufstellung von Speichersystemen	D
SYS.1.8.A2	B	Sichere Grundkonfiguration von Speicherlösungen	+
SYS.1.8.A3	B	entfallen	E
SYS.1.8.A4	B	Schutz der Administrationsschnittstellen	+
SYS.1.8.A5	B	entfallen	E
SYS.1.8.A6	S	Erstellung einer Sicherheitsrichtlinie für Speicherlösungen	O
SYS.1.8.A7	S	Planung von Speicherlösungen	D
SYS.1.8.A8	S	Auswahl einer geeigneten Speicherlösung	+
SYS.1.8.A9	S	Auswahl von Lieferanten für eine Speicherlösung	D
SYS.1.8.A10	S	Erstellung und Pflege eines Betriebshandbuchs	D
SYS.1.8.A11	S	Sicherer Betrieb einer Speicherlösung	D
SYS.1.8.A12	S	entfallen	E
SYS.1.8.A13	S	Überwachung und Verwaltung von Speicherlösungen	D
SYS.1.8.A14	S	Absicherung eines SANs durch Segmentierung	+
SYS.1.8.A15	S	Sichere Trennung von Mandanten in Speicherlösungen	D
SYS.1.8.A16	S	Sicheres Löschen in SAN-Umgebungen	n/a
SYS.1.8.A17	S	Dokumentation der Systemeinstellungen von Speichersystemen	D
SYS.1.8.A18	S	Sicherheitsaudits und Berichtswesen bei Speichersystemen	O
SYS.1.8.A19	S	Aussonderung von Speicherlösungen	D
SYS.1.8.A20	S	Notfallvorsorge und Notfallreaktion für Speicherlösungen	O
SYS.1.8.A21	H	Einsatz von Speicherpools zur Mandantentrennung	n/a
SYS.1.8.A22	H	Einsatz einer hochverfügbaren SAN-Lösung	D
SYS.1.8.A23	H	Einsatz von Verschlüsselung für Speicherlösungen	D
SYS.1.8.A24	H	Sicherstellung der Integrität der SAN-Fabric	D
SYS.1.8.A25	H	Mehrfaches Überschreiben der Daten einer LUN	D
SYS.1.8.A26	H	Absicherung eines SANs durch Hard-Zoning	D

NET:Netze und Kommunikation

Die Bausteine NET.2.1: *WLAN-Betrieb*, NET.2.2: *WLAN-Nutzung* und die Bausteine der Gruppe NET.4 (Telekommunikation und Fax) sind für den Betrieb einer Container-Plattform nicht relevant. Der Baustein NET.3.1: *Router und Switches* muss durch den Hosting-Anbieter umgesetzt werden, da er sich ausdrücklich nicht auf die im Rahmen einer Container-Plattform eingesetzten virtualisierten Pendants bezieht.

NET.1.1		**Netzarchitektur und -design**	
NET.1.1.A1	B	Sicherheitsrichtlinie für das Netz	O
NET.1.1.A2	B	Dokumentation des Netzes	+

NET.1.1.A3	B	Anforderungsspezifikation für das Netz	O
NET.1.1.A4	B	Netztrennung in Zonen	+
NET.1.1.A5	B	Client-Server-Segmentierung	n/a
NET.1.1.A6	B	Endgeräte-Segmentierung im internen Netz	n/a
NET.1.1.A7	B	Absicherung von schützenswerten Informationen	+
NET.1.1.A8	B	Grundlegende Absicherung des Internetzugangs	+
NET.1.1.A9	B	Grundlegende Absicherung der Kommunikation mit nicht vertrauenswürdigen Netzen	+
NET.1.1.A10	B	DMZ-Segmentierung für Zugriffe aus dem Internet	+
NET.1.1.A11	B	Absicherung eingehender Kommunikation vom Internet in das interne Netz	+
NET.1.1.A12	B	Absicherung ausgehender interner Kommunikation zum Internet	+
NET.1.1.A13	B	Netzplanung	+
NET.1.1.A14	B	Umsetzung der Netzplanung	O
NET.1.1.A15	B	Regelmäßiger Soll-Ist-Vergleich	O
NET.1.1.A16	S	Spezifikation der Netzarchitektur	+
NET.1.1.A17	S	Spezifikation des Netzdesigns	+
NET.1.1.A18	S	P-A-P-Struktur für die Internet-Anbindung	+
NET.1.1.A19	S	Separierung der Infrastrukturdienste	+
NET.1.1.A20	S	Zuweisung dedizierter Subnetze für IPv4/IPv6-Endgerätegruppen	+
NET.1.1.A21	S	Separierung des Management-Bereichs	+
NET.1.1.A22	S	Spezifikation des Segmentierungskonzepts	+
NET.1.1.A23	S	Trennung von Netzsegmenten	+
NET.1.1.A24	S	Sichere logische Trennung mittels VLAN	+
NET.1.1.A25	S	Fein- und Umsetzungsplanung von Netzarchitektur und -design	+
NET.1.1.A26	S	Spezifikation von Betriebsprozessen für das Netz	O
NET.1.1.A27	S	Einbindung der Netzarchitektur in die Notfallplanung	O
NET.1.1.A28	H	Hochverfügbare Netz- und Sicherheitskomponenten	+
NET.1.1.A29	H	Hochverfügbare Realisierung von Netzanbindungen	+
NET.1.1.A30	H	Schutz vor Distributed-Denial-of-Service	D
NET.1.1.A31	H	Physische Trennung von Netzsegmenten	D
NET.1.1.A32	H	Physische Trennung von Management-Netzsegmenten	D
NET.1.1.A33	H	Mikrosegmentierung des Netzes	+
NET.1.1.A34	H	Einsatz kryptografischer Verfahren auf Netzebene	D
NET.1.1.A35	H	Einsatz von netzbasiertem DLP	J
NET.1.1.A36	H	Trennung mittels VLAN bei sehr hohem Schutzbedarf	D
NET.1.2	**Netzmanagement**		
NET.1.2.A1	B	Planung des Netzmanagements	+
NET.1.2.A2	B	Anforderungsspezifikation für das Netzmanagement	O
NET.1.2.A3	B	entfallen	E
NET.1.2.A4	B	entfallen	E
NET.1.2.A5	B	entfallen	E
NET.1.2.A6	B	Regelmäßige Datensicherung	+
NET.1.2.A7	B	Grundlegende Protokollierung von Ereignissen	J
NET.1.2.A8	B	Zeit-Synchronisation	+
NET.1.2.A9	B	Absicherung der Netzmanagement-Kommunikation und des Zugriffs auf Netz-Management-Werkzeuge	+
NET.1.2.A10	B	Beschränkung der SNMP-Kommunikation	+

NET.1.2.A11	S	Festlegung einer Sicherheitsrichtlinie für das Netzmanagement	O
NET.1.2.A12	S	Ist-Aufnahme und Dokumentation des Netzmanagements	O
NET.1.2.A13	S	Erstellung eines Netzmanagement-Konzepts	+
NET.1.2.A14	S	Fein- und Umsetzungsplanung	O
NET.1.2.A15	S	Konzept für den sicheren Betrieb der Netzmanagement-Infrastruktur	O
NET.1.2.A16	S	Einrichtung und Konfiguration von Netzmanagement-Lösungen	+
NET.1.2.A17	S	Regelmäßiger Soll-Ist-Vergleich im Rahmen des Netzmanagements	O
NET.1.2.A18	S	Schulungen für Management-Lösungen	O
NET.1.2.A19	S	entfallen	E
NET.1.2.A20	S	entfallen	E
NET.1.2.A21	S	Entkopplung der Netzmanagement-Kommunikation	+
NET.1.2.A22	S	Beschränkung der Management-Funktionen	+
NET.1.2.A23	S	entfallen	E
NET.1.2.A24	S	Zentrale Konfigurationsverwaltung für Netzkomponenten	+
NET.1.2.A25	S	Statusüberwachung der Netzkomponenten	J
NET.1.2.A26	S	Alarming und Logging	J
NET.1.2.A27	S	Einbindung des Netzmanagements in die Notfallplanung	O
NET.1.2.A28	S	Platzierung der Management-Clients für das In-Band-Management	D
NET.1.2.A29	S	Einsatz von VLANs im Management-Netz	+
NET.1.2.A30	H	Hochverfügbare Realisierung der Management-Lösung	+
NET.1.2.A31	H	Grundsätzliche Nutzung von sicheren Protokollen	+
NET.1.2.A32	H	Physische Trennung des Managementnetzes	D
NET.1.2.A33	H	Physische Trennung von Management-Segmenten	D
NET.1.2.A34	H	entfallen	E
NET.1.2.A35	H	Festlegungen zur Beweissicherung	O
NET.1.2.A36	H	Einbindung der Protokollierung des Netzmanagements in eine SIEM-Lösung	J
NET.1.2.A37	H	Standort übergreifende Zeitsynchronisation	+
NET.1.2.A38	H	Festlegung von Notbetriebsformen für die Netzmanagement-Infrastruktur	O
NET.3.2		**Firewall**	
NET.3.2.A1	B	Erstellung einer Sicherheitsrichtlinie	O
NET.3.2.A2	B	Festlegen der Firewall-Regeln	+
NET.3.2.A3	B	Einrichten geeigneter Filterregeln am Paketfilter	+
NET.3.2.A4	B	Sichere Konfiguration der Firewall	+
NET.3.2.A5	B	entfallen	E
NET.3.2.A6	B	Schutz der Administrationsschnittstellen	+
NET.3.2.A7	B	Notfallzugriff auf die Firewall	D
NET.3.2.A8	B	Unterbindung von dynamischem Routing	+
NET.3.2.A9	B	Protokollierung	J
NET.3.2.A10	B	Abwehr von Fragmentierungsangriffen am Paketfilter	+
NET.3.2.A11	B	entfallen	E
NET.3.2.A12	B	entfallen	E
NET.3.2.A13	B	entfallen	E
NET.3.2.A14	B	Betriebsdokumentationen	+
NET.3.2.A15	B	Beschaffung einer Firewall	O
NET.3.2.A16	S	Aufbau einer „P-A-P“-Struktur	+
NET.3.2.A17	S	Deaktivierung von IPv4 oder IPv6	+

NET.3.2.A18	S	Administration über ein gesondertes Managementnetz	+
NET.3.2.A19	S	Schutz vor TCP SYN Flooding, UDP Paket Storm und Sequence Number Guessing am Paketfilter	+
NET.3.2.A20	S	Absicherung von grundlegenden Internetprotokollen	+
NET.3.2.A21	S	Temporäre Entschlüsselung des Datenverkehrs	+
NET.3.2.A22	S	Sichere Zeitsynchronisation	+
NET.3.2.A23	S	Systemüberwachung und -Auswertung	J
NET.3.2.A24	S	Revision und Penetrationstests	O
NET.3.2.A32	S	Notfallvorsorge für die Firewall	O
NET.3.2.A25	H	Erweiterter Integritätsschutz für die Konfigurationsdateien	+
NET.3.2.A26	H	Auslagerung von funktionalen Erweiterungen auf dedizierte Hardware	D
NET.3.2.A27	H	Einsatz verschiedener Firewall-Betriebssysteme und -Produkte in einer mehrstufigen Firewall-Architektur	O
NET.3.2.A28	H	Zentrale Filterung von aktiven Inhalten	+
NET.3.2.A29	H	Einsatz von Hochverfügbarkeitslösungen	+
NET.3.2.A30	H	Bandbreitenmanagement für kritische Anwendungen und Dienste	+
NET.3.2.A31	H	Einsatz von zertifizierten Produkten	O
NET.3.3	**VPN**		
NET.3.3.A1	B	Planung des VPN-Einsatzes	J
NET.3.3.A2	B	Auswahl eines VPN-Dienstleisters	n/a
NET.3.3.A3	B	Sichere Installation von VPN-Endgeräten	J
NET.3.3.A4	B	Sichere Konfiguration eines VPN	J
NET.3.3.A5	B	Sperrung nicht mehr benötigter VPN-Zugänge	O
NET.3.3.A6	S	Durchführung einer VPN-Anforderungsanalyse	O
NET.3.3.A7	S	Planung der technischen VPN-Realisierung	J
NET.3.3.A8	S	Erstellung einer Sicherheitsrichtlinie zur VPN-Nutzung	O
NET.3.3.A9	S	Geeignete Auswahl von VPN-Produkten	J
NET.3.3.A10	S	Sicherer Betrieb eines VPN	O
NET.3.3.A11	S	Sichere Anbindung eines externen Netzes	J
NET.3.3.A12	S	Benutzer- und Zugriffsverwaltung bei Fernzugriff-VPNs	J
NET.3.3.A13	S	Integration von VPN-Komponenten in eine Firewall	J

INF:Infrastruktur

Die Bausteine INF.3: *Elektrotechnische Verkabelung* und INF.4: *IT-Verkabelung* wurden in der 4. Ausgabe (Edition 2021) des *Kompendiums* zum Baustein INF.12: *Verkabelung* zusammengefasst.

Die Bausteine INF.1: *Allgemeines Gebäude*, INF.2: *Rechenzentrum sowie Serverraum*, INF.5: *Raum sowie Schrank für technische Infrastruktur*, INF.6: *Datenträgerarchiv* und INF.12: *Verkabelung* müssen durch den Hosting-Anbieter umgesetzt werden. Die Bausteine INF.7: *Büroarbeitsplatz*, INF.8: *Häuslicher Arbeitsplatz*, INF.9: *Mobiler Arbeitsplatz* und INF.10: *Besprechungs-, Veranstaltungs- und Schulungsräume* sind für den Betrieb und Aufbau einer Container-Plattform nur mittelbar relevant.