MAGAZIN FÜR PROFESSIONELLE INFORMATIONSTECHNIK

IT-Sicherheit: Neuer Entwurf des BSI-Bausteins zu Containern

20.04.2020 11:00 Uhr Christoph Puppe

(Bild: Fotolia - floraldeco)

Das BSI erweitert sein IT-Grundschutz-Kompendium um einen Baustein für den Einsatz von Containern.

Mit der steigenden Verbreitung von Containern gerät deren Sicherheit in den Fokus. Auch bei Behörden und Firmen in Deutschland sind Container heute eine der strategisch gesetzten Techniken. Das BSI hat daher Mitte 2018 den ersten Entwurf eines Bausteins für das IT-Grundschutz-Kompendium veröffentlicht, der den Nutzern einen Leitfaden an die Hand gibt, **wie sie Container sicher und zuverlässig einsetzen können [1]**. Die Anwender des IT-Grundschutzes waren gebeten, den Entwurf zu kommentieren, bevor er in das IT-Grundschutz-Komendium aufgenommen und für Zertifizierungen relevant wird.

Nach dem Eingang zahlreicher und umfassender Kommentare ist nun eine neue Version des Bausteins erschienen, die die Kommentare berücksichtigt, aber auch die veränderte Landschaft der Cloud-Stacks und des Betriebs von Containern beachtet. Auch die aktualisierte Version hat den Status "Public Draft". So können Nutzer ihre Anmerkungen noch einbringen, bevor der Baustein voraussichtlich in die Edition 2021 des IT-Grundschutz-Kompendiums einfließen wird.

Viele Anforderungen sind dabei unverändert, auch wenn sich ihre Reihenfolge geändert hat. Einige Anforderungen wurden in der neuen Version konkretisiert und der Themenbereich CI/CD und Orchestrierung

ist mehr in den Fokus gerückt. Dies hat zu sieben neuen Anforderungen geführt. Wer den Community Draft des Bausteins also schon produktiv genutzt hat, muss hier seine Modellierung ändern. Um dies zu vereinfachen, enthält die gedruckte Kreuzreferenztabelle auch die alten Nummern.

IX-TRACT

- Das IT-Grundschutz-Kompendium des BSI enthält Anforderungen an die Sicherheit von IT-Komponenten. Bundesbehörden müssen diese einhalten – und somit auch deren Auftragnehmer.
- Wegen des wachsenden Interesses hat das BSI einen Grundschutz-Baustein für den Umgang mit Software in Containern entworfen.
- Der Container-Baustein ist derzeit ein Entwurf, also nicht verpflichtend, soll aber in die nächste Version des Grundschutz-Kompendiums aufgenommen werden. Nutzer sind aufgerufen, ihn zu kommentieren.

Wie alle Bausteine seit der Novellierung des IT-Grundschutzes im August 2017 gliedert sich dieser Baustein in die Anforderungsstufen Basis, Standard und Erhöhter Schutzbedarf. Jede enthält Anforderungen, die das BSI für den Betrieb von Containern als notwendig erachtet. Zu jedem Baustein gehört außerdem eine Kreuzreferenztabelle, die zeigt, welche Anforderung welches Risiko abdeckt.

Bei der Modernisierung des IT-Grundschutzes hat das BSI das Vorgehen zur Risikoanalyse stark vereinfacht. Der Gefährdungskatalog enthält 47 elementare Gefährdungen, die für alle IT-Landschaften zu beachten sind. Dies reicht von physischen Risiken wie Hochwasser über Fehler der Anwender bis hin zum Angriff. Daneben enthält jeder Baustein spezifische Gefährdungen. Bei Containern sind dies vor allem Schwachstellen in Images, ungesicherte administrative Zugänge, potenzielle Schwachstellen der Orchestrierung, fehlende Persistenz und die Verwaltung von Zugangsdaten. Neu im aktuellen Community Draft ist die Gefährdung durch einen Ausbruch aus einem Container. Ein weiteres wichtiges Thema ist das Separieren der Anwendungen und eventuell der Mandanten.

Relevante Gefährdungen für den Baustein Container

G 0.14	Ausspähen von Informationen (Spionage)
G 0.19	Offenlegung schützenswerter Informationen
G 0.20	Informationen oder Produkte aus unzuverlÃ¤ssiger 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	Softwareschwachstellen 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

Dieser Artikel wird nicht alle Anforderungen aufzählen oder erklären, aber für die wichtigsten Gefährdungen die relevanten Anforderungen aufzeigen, um einen Einblick in die Herangehensweise und den Umfang zu geben.

[2]

Container benötigen Standardinfrastruktur

Basis der Absicherung bilden die Container-Hosts. Deswegen sind immer die Bausteine "SYS.1.1 Allgemeiner Server" und "SYS.1.3 Server unter Linux und Unix" anzuwenden. Zusätzlich enthält der Baustein die Anforderungen "SYS.1.6.A4 Härtung des Host-Systems (B)" und "SYS.1.6.A7 Härtung der Software im Container (B)" in der Basisstufe.

An der Frage, welche Separierung angemessen ist, scheiden sich die Geister. Im Prinzip sind Container nur chroot-Umgebungen, die man bei Linux seit den 1990ern kennt, oder Jails, die BSD Anfang des Jahrtausends einführte. Auch war Sicherheit durch Trennung der Prozesse bei Containern nicht der Fokus der Entwicklung, so wie dies bei Solaris, BSD oder zOS der Fall war. Nebenbei sei erwähnt, **dass die Großrechnerwelt Container schon immer hatte [3]** . Letztlich sind sie, wie die Paravirtualisierung, die vor 15 Jahren aufkam, nur Prozesse in einer eingeschränkten Umgebung – vergleichbar mit der Sandbox von Java. Der Vergleich hinkt etwas, da bei Containern die Abschottung über cgroups (Namespaces des Kernels) geschieht. Auch können SECCOMP, SELinux und AppArmor die Prozesse im Container erheblich einschränken und so das Risiko eines Ausbruchs aus dem Container stark reduzieren. Der Aufwand für die Konfiguration ist allerdings erheblich.

Wer sich nicht durch Software helfen lässt, **hat hier wenig Chancen, die Sicherheitspotenziale auch zu nutzen [4]** . Im Baustein ist dies in den Anforderungen "SYS.1.6.A5 Separierung der Container (B)", "SYS.1.6.A29 Automatisierte Auditierung von Containern (CIA) (H)", "SYS.1.6.A31 Erstellung erweiterter

Richtlinien für Container (CIA) (H)" und "SYS.1.6.A32 Host Based Intrusion Detection für Container (CIA) (H)" enthalten. Der größte Vorteil der Container liegt allerdings in der Automatisierung. Deskriptive Konfiguration und Einbindung in automatisierte Prozesse zur Softwareverteilung bieten Chancen für die Sicherheit der Anwendungen. So sind viele Image-Sammlungen mittlerweile mit einem Schwachstellenmanagement ausgestattet, das die Images automatisch auf sicherheitsrelevante Fehler in der enthaltenen Software untersucht. Die Container lassen sich, wie in der Standard-Anforderung "SYS.1.6.A15 Unveränderlichkeit der Container (S)" gefordert, mit einem Read-only-Dateisystem betreiben. Das und die Anforderung "SYS.1.6.A12 Nur eine Anwendung bzw. ein Dienst pro Container (S)" erschwert das Ausnutzen von Schwachstellen erheblich. Wer hier alle Möglichkeiten ausschöpft, kann die Sicherheit vor einem Ausbruch aus dem Container so stark erhöhen, dass Container den Vergleich mit einem Typ1-Hypervisor bei der Eintrittshäufigkeit eines Containment-Ausbruchs und den Auswirkungen nicht mehr fürchten müssen. Besser als ein schlecht gepflegter Server auf einem IaaS ist eine gut gepflegte und automatisierte CI/CD-Pipeline allemal.

Spätestens seit Rowhammer, Meltdown und Spectre in all seinen Varianten sollte jedem klar sein, dass geteilte Hardware immer ein Sicherheitsrisiko ist. Auch Ausbrüche aus Virtualisierung und Sandboxes gibt es ausreichend. Das soll nicht heißen, dass Container keine Separierung von Anwendungen ermöglichen. Aber die Entscheidung, welcher Separierungslevel ausreicht, muss bei der Planung immer im Hinblick auf die Sicherheitsanforderungen von Daten und Anwendungen erfolgen. Der Baustein folgt diesem Prinzip und enthält daher die Basis-Anforderungen "SYS.1.6.A1 Planung des Container-Einsatzes (B)" und "SYS.1.6.A2 Planung der Separierung (B)". Anwender sind hier gefordert, im Kontext des eigenen Schutzbedarfs festzulegen, was ihnen genügt und was nicht. Der Baustein enthält derzeit keine konkrete Anforderung für beispielsweise das Nutzen von Taints und Tolerations zum physischen Separieren auf unterschiedliche Container-Hosts oder Virtualisierungsserver. Die Umsetzungshinweise werden nach der Veröffentlichung hier zwar noch etwas Hilfestellung geben, aber die finale Entscheidung verbleibt immer beim Inhaber der Daten.

Es ist immer zu bedenken, dass ein moderner, agiler und kostengünstiger Betrieb von Anwendungen nur so lange kostengünstig ist, wie die Daten nicht plötzlich im Internet stehen und die Kosten für die Ausfälle und Entschädigungen die gesparten Betriebskosten schnell um ein Mehrfaches übersteigen.

Ein anderer Aspekt der Separierung ist das Verschlüsseln der Daten. Daten in Ruhe, also gespeicherte, kann das Betriebssystem des Container-Hosts meist schnell und zuverlässig verschlüsseln. Aber es hat die Schlüssel während des Betriebs im Zugriff, was ein Angreifer ausnutzen kann. Für den Transport bietet sich die Verschlüsselung der virtuellen Netze an, die die Container miteinander verbinden. Hierfür enthält der Baustein die Anforderungen bei erhöhtem Schutzbedarf "SYS.1.6.A35 Verschlüsselte Datenhaltung bei Containern (H)" und "SYS.1.6.A36 Verschlüsselung der Netzkommunikation zwischen Containern (H)".

Bei den Netzwerken der Administration und der Anwendungen hat sich in den letzten drei Jahren einiges Neues ergeben. Separierung ist weiterhin die Grundlage, wie dies "SYS.1.6.A18 Absicherung der Wirk- und Administrations-Netze (S)" und "SYS.1.6.A19 Verwendung vorgelagerter Ein- und Ausgangssysteme (S)" fordert. Allerdings sind mit mTLS in Service Meshes auch harte Authentifizierung, Verschlüsselung und regelbasierte Steuerung – wer mit wem reden darf – möglich. Dies findet sich in der neuen Anforderung "SYS.1.6.A33 Mikro-Segmentierung von Containern (H)".

Images, sicherer Fernzugriff und weitere wichtige Faktoren

Üblicherweise lädt man Images von einer zentralen Sammlung. Docker bietet selbst Docker-Hub an. GitHub, GitLab und weitere Anbieter haben ebenfalls solche Sammlungen erstellt. Die Qualität dieser Images ist leider sehr schwankend. Die Anforderung "SYS.1.6.A6 Verwendung sicherer Images (B)" definiert, worauf ein Nutzer hierbei achten muss. Da sie in den Basis-Anforderungen enthalten ist, muss jeder sie beachten und umsetzen. Die Standard-Anforderungen "SYS.1.6.A13 Freigabe von Images und Konfigurationen (S)" und "SYS.1.6.A14 Updates von Containern (S)" regeln den Umgang mit der Freigabe. Für erhöhten Schutzbedarf gibt es die Anforderungen "SYS.1.6.A29 Automatisierte Auditierung von Containern (H)" und "SYS.1.6.A30 Eigene Trusted Registry für Container (H)".

Cloud-Infrastruktur beinhaltet im Lieferzustand häufig keine Authentifizierung oder gar rollenbasierte Autorisierung. Dies ist in gut gesicherten Netzen vielleicht praktisch, bei der Verwaltung großer Umgebungen sollte aber der Schutz vor Angreifern immer durch möglichst viele Schichten erfolgen. Daher enthält der Baustein die Basis-Anforderung "SYS.1.6.A23 Administrativer Fernzugriff auf Container (S)" sowie die Standard-Anforderungen "SYS.1.6.A24 Identitäts- und Berechtigungsmanagement für die Container-Verwaltung (S)", damit kein ungesicherter Socket Befehle von nicht authentifizierten oder autorisierten Personen annimmt.

Neu in dieser Version des Bausteins ist der Fokus auch auf die Automatisierung und Steuerung. So wurde "SYS.1.6.A3 Planung der Verwaltung und Orchestrierung (B)" zu einer Basis-Anforderung und durch einige neue Anforderungen für die Absicherung der CI/CD-Pipelines und Codeverwaltung ergänzt. Auch das Verfügbarkeitsmanagement hat mit der Standard-Anforderung "SYS.1.6.A27 Überwachung der Container (S)" Einzug in den Baustein gehalten. Der Baustein versucht dabei nicht, einen DevOps-Baustein zu simulieren, sondern konzentriert sich auf die Absicherung der für den Betrieb relevanten Aspekte wie Konfigurationsdateien, Service-Accounts und Hilfsprozesse, wie sie etwa Jenkins Runner oder Helm benötigen. Diese neuen Anforderungen sind hinzugekommen: "SYS.1.6.A20 Absicherung von Konfigurationsdaten und Automatisierung (S)", "SYS.1.6.A22 Absicherung von Hilfsprozessen der Automatisierung (S)" und "SYS.1.6.A28 Absicherung der Registry für Images (S)".

Benutzermanagement ist auch für die Anwendungen im Container relevant, denn diese laufen unter einer Benutzerkennung des Hosts und einem Service-Account in der Verwaltungssoftware – heutzutage eigentlich überall Kubernetes. Die Rechte hängen wie bei Prozessen außerhalb von Containern von den Rechten dieses Benutzers ab. Zwar sollte ein Prozess im Container nicht auf die Ressourcen außerhalb seines Namensraums zugreifen können, aber wie bei jeder Sandbox ist es nur eine Frage der Zeit, bis jemand eine Möglichkeit zum Ausbruch findet – sofern das nicht schon passiert ist. Daher fordert der Baustein als Standard-Anforderung in "SYS.1.6.A26 Accounts der Anwendungsdienste in Containern (S)" und "SYS.1.6.A21 Container-Ausführung ohne Privilegien (S)" die für die Absicherung von Diensten unter Unix üblichen Sicherheitsvorkehrungen. Auch die Service-Accounts der Container finden Beachtung mit der Standard-Anforderung "SYS.1.6.A25 Service-Accounts für Container (S)".

Akten sollten nichtflüchtig sein

Daten dauerhaft zu speichern, ist in einer Umgebung, in der die Anwendungen flüchtig sind – die Container können jederzeit starten, stoppen und in der Zahl variieren –, ein besonderer Aspekt, der sich stark von der Nicht-Containerwelt unterscheidet. Der Baustein hat dafür die Basis-Anforderungen "SYS.1.6.A8 Persistenz von Protokollierungsdaten (B)", "SYS.1.6.A9 Persistenz von Nutzdaten (B)" und "SYS.1.6.A17 Einbinden von Massenspeichern in Container (S)". Die Anforderungen sind wie alle unspezifisch für die eingesetzte Technik, erst die später veröffentlichten Umsetzungshinweise gehen auf diese ein.

Die meisten Anwendungen, sei es ein Microservice oder ein Monolith, benötigen Zugangsdaten, etwa für Datenbanken oder andere Dienste, von denen sie Daten beziehen oder an die sie senden. Wenn eine Anwendung Dienste über öffentliche Netze einbindet, wird die Authentifizierung noch wichtiger, denn der Schutz eines Perimeters fällt weg und der Dienst ist aus dem gesamten Internet erreich- und damit angreifbar.

Hier bieten alle Hersteller von Containern und Orchestrierung Werkzeuge an, mit denen sich diese Zugangsdaten spezifisch für jede Umgebung verwalten lassen. Ein Passwort für die Datenbank muss für die Entwicklung ein anderes sein als für Staging und Produktion. Der Baustein enthält dazu die Basis-Anforderung "SYS.1.6.A10 Speicherung von Zugangsdaten (B)".

Leider werden jeden Tag noch Anwendungen und Seiten gehackt, weil Programmierer die Zugangsdaten im Code speichern und diese mit dem Code zum Beispiel auf GitHub landen. Für Software in der Cloud ist das Thema daher von besonderem Interesse und die Entwicklung oder der Betrieb von Anwendungen sollte das Thema frühzeitig beachten und absichern.

Fazit

Der Baustein-Draft gibt einen erneuten Ausblick, was der zukünftig für die Zertifizierung von Unternehmen und die Bundesbehörden relevante Baustein Container enthalten wird. In seiner jetzigen Form ist er noch nicht bindend, da der aktuelle Entwurf noch einmal zur Kommentierung veröffentlicht wurde.

Der Baustein deckt alle Aspekte der Planung, des Betriebs und der Orchestrierung ab, Details zur Umsetzung sind allerdings noch nicht verfügbar, da die Umsetzungshinweise noch nicht verfügbar sind. Wer heute schon seinen Betrieb mit Containern am Grundschutz ausrichten will, hat zumindest einen Einstieg und eine Vorlage für einen gemäß Standard zu erstellenden Baustein.

Kreuzreferenz aus dem Baustein Container-Sicherheit 2020

		Nummer alt	**G 0.14**	**G 0.19**	**G 0.20**	**G 0.21**	**G 0.23**	**G 0.25**	**G 0.27**
Basis-Anforderungen									
A1	SYS.1.6.A1 Planung des Container-Einsatzes (B)	A1	X	X	X	X	X		X
A2	SYS.1.6.A2 Planung der Separierung der Anwendungen in Containern (B)	A2	X			X			X
A3	SYS.1.6.A3 Planung der Verwaltung und Orchestrierung (B)	A19			X	X			
A4	SYS.1.6.A4 Härtung des Host-Systems (B)	A3	X			X	X		
A5	SYS.1.6.A5 Separierung der Container (B)	NEU	X				X		

A6	SYS.1.6.A6 Verwendung sicherer Images (B)	A7			X	X			
A7	SYS.1.6.A7 Härtung der Software im Container (B)	A4	X			X	X		
A8	SYS.1.6.A8 Persistenz von Protokollierungsdaten (B)	A5							
A9	SYS.1.6.A9 Persistenz von Nutzdaten (B)	A6							
A10	SYS.1.6.A10 Speicherung von Zugangsdaten (B)	A8	X				X		
Standard-Anforderungen									
A11	SYS.1.6.A11 Richtlinie für Betrieb und Images (S)	NEU	X	X	X	X	X		X
A12	SYS.1.6.A12 Nur eine Anwendung bzw. ein Dienst pro Container (S)	A18		X					
A13	SYS.1.6.A13 Freigabe von Images und Konfigurationen (S)	A12			X	X	X		X
A14	SYS.1.6.A14 Updates von Containern (S)	A13	X				X		
A15	SYS.1.6.A15 Unveränderlichkeit der Container (S)	NEU	X				X		
A16	SYS.1.6.A16 Limitierung der Ressourcen pro Container (S)	A20							X
A17	SYS.1.6.A17 Einbinden von Massenspeichern in Container (S)	A10		X					
A18	SYS.1.6.A18 Absicherung der Wirk- und Administrations-Netze (S)	A9	X	X			X		
A19	SYS.1.6.A19 Verwendung vorgelagerter Ein- und Ausgangssysteme (S)	NEU	X			X	X		
A20	SYS.1.6.A20 Absicherung von Konfigurationsdaten und Automatisierung (S)	NEU			X	X			X
A21	SYS.1.6.A21 Container-Ausführung ohne Privilegien (S)	A17					X		

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

	Netzkommunikation zwischen Containern (C) (H)								

(**avr [5]**)

URL dieses Artikels:

https://www.heise.de/-4703039

Links in diesem Artikel:

[1] https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzKompendium/bausteine/bausteine_node.html

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

[3] https://www.heise.de/hintergrund/Mit-IBMs-z15-und-z-OS-2-4-zCX-Container-sicherer-betreiben-4617149.html

[4] https://www.heise.de/hintergrund/Tools-zur-Absicherung-von-Containern-4340132.html

[5] mailto:avr@ix.de