

Sicherheitsanforderung

Container

Deutsche Telekom Gruppe

Version	4.0
Datum	01.12.2023
Status	Freigegeben

Impressum

Herausgeber
Deutsche Telekom AG
Vorstandsbereich Technology & Innovation
Chief Security Officer

Reuterstrasse 65, 53315 Bonn
Deutschland

Dateiname	Dokumentennummer 3.64	Dokumententyp Sicherheitsanforderung
Version 4.0	Stand 01.12.2023	Status Freigegeben
Fachlicher Ansprechpartner Telekom Security psa.telekom.de	Gültigkeitsdauer 01.12.2023 - 30.11.2028	Freigegeben von Stefan Pütz, Leiter SEC-T-TST

Zusammenfassung

Dieses Dokument beschreibt die funktionalen Sicherheitsanforderungen, die zum Sichern von Containern verwendet werden, zusätzlich zu den CIS-Benchmarks, die auch implementierungsorientierte Anforderungen abdecken.

Copyright © 2023 by Deutsche Telekom AG.

Alle Rechte, auch die des auszugsweisen Nachdrucks, der fotomechanischen Wiedergabe (einschließlich Mikrokopie) sowie der Auswertung durch Datenbanken oder ähnliche Einrichtungen, vorbehalten.

Inhaltsverzeichnis

1.	Einleitung	4
2.	Requirements	5
2.1.	General	5
2.2.	Registry	12
3.	Technischer Basisschutz von IT-/NT-Systemen	15
3.1.	Schutz von Daten und Informationen	15
3.2.	Authentisierungsmerkmal "Passwort"	17
3.3.	Protokollierung	17

1. Einleitung

Dieses Dokument wurde auf Basis der Vorgaben aus den im Konzern gültigen Sicherheitsrichtlinien erstellt.

Die Sicherheitsanforderung dient u.a. als Grundlage zur Freigabe im PSA-Prozess. Sie dient darüber hinaus als Umsetzungsempfehlung für Einheiten, die nicht am PSA-Verfahren teilnehmen. Diese Anforderungen müssen bereits während der Planungs- und Entscheidungsprozesse berücksichtigt werden. Bei der Umsetzung der Sicherheitsanforderung ist das jeweils vorrangige nationale, internationale und supranationale Recht zu beachten.

Sollte die Einhaltung der beschriebenen Anforderungen in individuellen Fällen nicht oder nur eingeschränkt umsetzbar sein, sind gemeinsam mit einem Sicherheits- und/oder Datenschutzexperten (entsprechend der betroffenen Anforderungen) Risikobewertungen vorzunehmen und mögliche alternative Schutzmaßnahmen abzustimmen.

2. Requirements

2.1. General

Req 1 Container müssen als unveränderlich behandelt werden.

Container sollen unveränderlich sein, daher dürfen sie zur Laufzeit nicht verändert werden. Anstatt Container im Livebetrieb zu patchen, muss das Image gepatched und redeployed werden. Container Images müssen über eine vorhandene CI/CD Pipeline geändert werden. Diese Pipeline muss den Sicherheitsanforderungen für CI/CD folgen

Motivation: Container sollen unveränderlich sein, daher dürfen sie zur Laufzeit nicht verändert werden. Anstatt Container im Livebetrieb zu patchen, muss das Image gepatched und redeployed werden. Container Images müssen über eine vorhandene CI/CD Pipeline geändert werden. Diese Pipeline muss den Sicherheitsanforderungen für CI/CD folgen

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Modifikation von Daten

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 3.64-1/4.0

Req 2 Es müssen Fixe Tags verwendet werden um die Unveränderlichkeit zu gewährleisten.

Verwenden Sie das spezifischste verfügbare Tag. Wenn ein Image mehrere Tags hat, z.B. :8 und :8.0.1 oder: 8.0.1-alpine, müssen sie das letzte verwenden, da es die spezifischste Referenz ist. Vermeiden Sie generische Tags wie "latest". [Bei Verwendung von unspezifischen Tags kann sich der Container bei einem Redeployment verändern und die Funktionsfähigkeit der Software einschränken.]

Motivation: Es kann von Vorteil sein, Images selbst zu erstellen, da Sie die Kontrolle über alle Komponenten behalten die mitgeliefert werden.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Modifikation von Daten
- Störung der Verfügbarkeit

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 3.64-2/4.0

Req 3 Unnötige Pakete müssen vermieden werden.

Ein Image darf nur die Komponenten enthalten, die es zum Ausführen einer Anwendung benötigt.

Motivation: Dadurch wird vermieden, dass nicht benötigte Software in Containern ausgeführt wird, um die Angriffsfläche zu verringern.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Nutzung von Diensten oder Ressourcen

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 3.64-3/4.0

Req 4 Es muss ein Benutzer mit den geringsten Berechtigungen verwendet werden.

Wenn in einer Container-Datei kein Benutzer angegeben ist, wird der Container standardmäßig mit dem Root-Benutzer ausgeführt.
Der Container darf nicht mit dem Standard-Root-Benutzer ausgeführt werden.
Es muss ein spezifischer Benutzer verwendet werden.

Motivation: Verhindern Sie, dass Container als Standard-Root-Benutzer ausgeführt werden.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 3.64-4/4.0

Req 5 Container müssen auf Schwachstellen untersucht werden.

Images müssen in der Registry und zur Laufzeit auf bekannte Sicherheitslücken gescannt werden.

Motivation: Finden Sie bekannte Sicherheitslücken in Containern.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 3.64-5/4.0

Req 6 Ein Containerspezifisches Host-Betriebssystem muss verwendet werden.

Containerspezifische Host-Betriebssysteme müssen anstelle von Allzweck-Betriebssystemen verwendet werden.
Containerspezifische Host-Betriebssysteme verringern die Angriffsfläche, die ein System bietet.
Es gibt weniger Möglichkeiten, ein Container spezifisches Host-Betriebssystem wie z. B. : RedHat Atomic oder Core anzugreifen

Motivation: Containerspezifische Betriebssysteme reduzieren die Angriffsfläche aufgrund ihres minimalistischen Ansatzes.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 3.64-6/4.0

Req 7	Falls ein CIS Benchmark für die verwendete Container Runtime verfügbar ist , muss dieser implementiert werden.
-------	--

Für Docker muss der CIS-Benchmark implementiert sein: <https://cisecurity.telekom.de/de/>

- CIS Level 1 muss umgesetzt werden
- CIS Level 2 sollte nach Möglichkeit umgesetzt werden.

Falls eine Level 2 Anforderung funktionale Behinderungen der Applikation oder unverhältnismässig hohe Aufwände in der Umsetzung erzeugt, darf sie ausgelassen werden

Motivation: Einrichtung einer sicheren Konfiguration für Docker.

Umsetzungsbeispiel: <https://cisecurity.telekom.de/de/cis/benchmarks/>

B e i s p i e l f ü r L X D C o n t a i n e r : h t t -
p s : / / c i s e c u r i t y . t e l e k o m . d e / d o w n l o a d / C I S / o p e r a t i n g _ s y s t e m s / L X D /
C I S _ U b u n t u _ 1 8 . 0 4 _ L X D _ C o n t a i n e r _ B e n c h m a r k 1 . 0 . 0 . p d f

B e i s p i e l f ü r D o c k e r : h t t -
p s : / / c i s e c u r i t y . t e l e k o m . d e / d o w n l o a d / C I S / s e r v e r _ s o f t w a r e / D o c k e r / C I S _ D o c k e r _ B e n c h m a r k 1 . 5 . 0 . p d f

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 3.64-7/4.0

Req 8	Die Namespace-Isolation muss gewährleistet sein.
-------	--

Ressourcen, mit denen ein Container interagieren kann, müssen begrenzt sein. Dies umfasst Dateisysteme, Netzwerkschnittstellen und Interprozess

Kommunikation, Hostnamen, Benutzerinformationen und Prozesse.

Stellen Sie sicher, dass sich Apps und Prozesse, die sich in einem Container befinden auf die ihm zugewiesenen physischen und virtuellen Ressourcen beschränken.

Die Namespace-Isolierung bietet jedem Container einen eigenen Netzwerkstapel mit eindeutigen Schnittstellen und IP-Adressen.

Motivation: Container isolieren

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 3.64-8/4.0

Req 9	Eine Secret Verwaltung muss zur Verfügung gestellt werden.
-------	--

Secrets müssen außerhalb von Images gespeichert und bei Bedarf zur Laufzeit dynamisch bereitgestellt werden.

Motivation: Die meisten Orchestratoren, wie Docker Swarm und Kubernetes, verfügen über eine native Verwaltung

von Secrets. Diese Orchestratoren sorgen nicht nur für die sichere Speicherung von Secrets und die "just in time" - Injektion in Container, sondern machen auch die Verwaltung einfacher. Der Orchestrator kann z.B. sicherstellen, dass nur der Webanwendungscontainer Zugriff auf ein bestimmtes Secret hat.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 3.64-9/4.0

Req 10 Sicherheitszonen müssen eingerichtet werden.

Container, die verschiedenen Sicherheitszonen zugeordnet sind (z. B. DMZ-Zone, Anwendungs- / Datenbankzone) müssen

auf verschiedenen Hosts / Host-Pools bereitgestellt werden. Ein Pool bezieht sich auf eine Gruppe von mindestens einer oder mehreren VMs. Im Falle der Verwendung von Container as a Service (CaaS) -Plattformen können diese Pools für mehrere Anwendungen gemeinsam genutzt werden. Für kritische Anwendungen müssen dedizierte Hosts / Host-Pools . Wenn der einzige exponierte Dienst ein Ingress-Controller (z. B. Nginx) ist, der hinsichtlich Sicherheit gehärtet ist, können Sie dies implementieren, indem Sie nur Container-spezifische Maßnahmen verwenden, wie z.B. Pod Security Policies / Network Security Policies

Darüber hinaus sollte für alle Container eine Überwachung zur Laufzeit vorhanden sein.

Motivation: Die Unterteilung von Containern nach Zweck und Bedrohungslage bietet eine zusätzliche Separierung. Durch das Gruppieren von Containern auf diese Weise ist es für einen Angreifer, der eine der Gruppen kompromittiert, schwieriger diesen Zugriff auszuweiten.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 3.64-10/4.0

Req 11 Produktionssysteme müssen strikt von Test- und Entwicklungssystemen getrennt sein.

Dies muss erfolgen, indem eine starke Mandantentrennung sichergestellt wird oder bestimmte Hosts/Hostpools für Produktion und Test verwendet werden.

Motivation: Das Verwalten mehrerer Umgebungen bietet eine bessere Sicherheit: Um die Integrität Ihrer Produktionsdaten zu schützen, sollten Sie den Zugriff darauf beschränken.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Nutzung von Diensten oder Ressourcen

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 3.64-11/4.0

Req 12 Pod Security Admissions / Pod Security Context oder 3rd party admission plugins müssen verwendet werden, um Container-Berechtigungen einzuschränken und Plattform Regeln bereitzustellen.

Container müssen ihren Security Context spezifizieren : <https://kubernetes.io/docs/tasks/configure-pod-container/security-context/>

Eine restriktive Policy muss aktiviert sein. Ressourcen und Volumes, auf die ein Pod zugreifen kann, müssen begrenzt sein.

Es gibt verschiedenen Möglichkeiten Policies zu implementieren:

- Pod Security Admissions <https://kubernetes.io/docs/concepts/security/pod-security-admission/>
- Open Policy Agent / Gatekeeper <https://www.openpolicyagent.org/>
- Kyverno <https://kyverno.io/>

Policies müssen limitieren:

Control Aspect	Field Names
Running of privileged containers	privileged
Usage of host namespaces	hostPID, hostIPC
Usage of host networking and ports	hostNetwork, hostPorts
Usage of volume types	volumes
Usage of the host filesystem	allowedHostPaths
Allow specific FlexVolume drivers	allowedFlexVolumes
Allocating an FSGroup that owns the pod's volumes	fsGroup
Requiring the use of a read only root file system	readOnlyRootFilesystem
The user and group IDs of the container	runAsUser, runAsGroup, supplementalGroups
Restricting escalation to root privileges	allowPrivilegeEscalation, defaultAllowPrivilegeEscalation
Linux capabilities	defaultAddCapabilities, requiredDropCapabilities, allowedCapabilities
The SELinux context of the container	seLinux
The Allowed Proc Mount types for the container	allowedProcMountTypes
The sysctl profile used by containers	forbiddenSysctls, allowedUnsafeSysctls

Motivation: Ressourcenzugriffe und Rechte von Container beschränken und die Angriffsfläche minimieren.

ID: 3.64-12/4.0

Req 13 Linux-Kernel-Funktionen müssen innerhalb von Containern eingeschränkt werden.

Standardmäßig starten Container mit einem eingeschränkten Satz von Linux-Kernel-Funktionen. Das bedeutet, dass jedem Prozess die erforderlichen Fähigkeiten gewährt werden können, anstatt ihm Root zu geben. Unter Verwendung der Linux-Kernel-Funktionen müssen Prozesse im Allgemeinen nicht als Root ausgeführt werden. Alle Fähigkeiten, die für die korrekte Funktion des Behälters nicht erforderlich sind müssen entfernt werden. Insbesondere sollte in dem von Docker bereitgestellten Standardfunktionssatz die NET_RAW-Funktion entfernt werden, wenn dies nicht ausdrücklich erforderlich ist.

Übersicht der Linux Kernel Funktionen für Docker : <https://docs.docker.com/engine/security/#linux-kernel-capabilities>

Motivation: Der Linux-Kernel ist in der Lage, die Privilegien des Root-Benutzers in verschiedene Einheiten aufzuteilen, die als Capabilities bezeichnet werden.

Minimierung der Angriffsfläche durch das entfernen von Capabilities, die für den Betrieb des Containers nicht benötigt werden.

Umsetzungsbeispiel: Beispiel für Docker Docker :

```
$ docker run --cap-drop ALL
```

Danach können via **-cap-add** flag neue Capabilities hinzugefügt werden.

ID: 3.64-13/4.0

Req 14	Eingesetzte Software muss aus vertrauenswürdigen Quellen bezogen und auf Integrität geprüft werden.
--------	---

Die auf dem System eingesetzte Software muss aus vertrauenswürdigen Quellen bezogen und vor der Installation auf ihre Integrität geprüft werden.

Diese Anforderung bezieht sich auf sämtliche Arten von Software:

- Firmware sowie Microcode für Hardware-Komponenten
- Betriebssysteme
- Software-Bibliotheken
- Applikationen und Anwendungssoftware
- vorintegrierte Anwendungslösungen, wie z.B. Software Appliances oder Container

sowie weitere Software, die ggf. zum Einsatz kommt.

Vertrauenswürdige Quellen

Als vertrauenswürdige Bezugsquellen gelten allgemein:

- die offiziellen Vertriebs- und Bereitstellungskanäle des Herstellers
 - dritte Distributoren, sofern sie durch den Hersteller autorisiert und legitimer Bestandteil von dessen Bereitstellungskanälen sind
 - Internet-Downloads, sofern diese von offiziellen Bereitstellungs-Servern des Herstellers oder autorisierter Distributoren erfolgen
- (1) Bietet der Bereitstellungs-Server verschiedene Formen von Downloads an, müssen durch Verschlüsselung bzw. kryptografische Signaturen geschützte Downloads jenen ohne einen solchen Schutz vorgezogen werden.
- (2) Sichert der Bereitstellungs-Server die Transport-Schicht mittels kryptografischer Protokolle (z. B. https, sftp), müssen die zugehörig präsentierten Server-Zertifikate oder Server-Keys/Fingerprints bei jedem Download validiert werden, um die Identität des Bereitstellungs-Servers zu bestätigen; schlägt die Validierung fehl, ist der Download abubrechen und der Bereitstellungs-Server als nicht vertrauenswürdige Quelle zu bewerten.

Integritätsprüfung

Die Integritätsprüfung soll sicherstellen, dass die erhaltene Software frei von Manipulationen und Schadsoftware-Befall ist. Sofern vorhanden, müssen zur Prüfung die vom Hersteller implementierten Mechanismen verwendet werden.

Valide Mechanismen sind:

- physikalische Siegel bzw. fest aufgebrachte Echtheitszertifikate (bei Bereitstellung der Software auf physikalischen Datenträgern)
- Abgleich von kryptografischen Hash Werten (z.B. SHA256, SHA512) der erhaltenen Software gegen Soll-

Werte, die der Hersteller separat benennt

- Prüfung von kryptografischen Signaturen (z. B. GPG, Zertifikate), mit denen der Hersteller seine Software versieht

Zusätzlich ist eine Überprüfung der Software mittels eines Anti-Virus- bzw. Anti-Malware Scanners empfohlen (falls der Hersteller keinen der zuvor erwähnten Integritätsschutz-Mechanismen für seine Software implementiert hat, ist diese Überprüfung verpflichtend).

Erweiterte Integritätsprüfung bei Bezug von Software aus öffentlichen Registries

Öffentliche Registries ermöglichen es Entwicklern, beliebige eigene Software-Projekte zur Nutzung bereitzustellen. Die Bandbreite umfasst sowohl Projekte von namhaften Anbietern mit kontrollierten Entwicklungsprozessen als auch von kleineren Anbietern oder Amateur-Entwicklern.

Beispiele für solche Registries sind:

- Code Registries (z. B. GitHub, Bitbucket, SourceForge, Python Package Index)
- Container Registries (z. B. Docker Hub)

Software aus öffentlichen Registries muss vor dem Einsatz einer erweiterten Integritätsprüfung unterzogen werden. Zusätzlich zu den im vorherigen Abschnitt beschriebenen Bestandteilen der Integritätsprüfung soll die erweiterte Prüfung explizit sicherstellen, dass die Software ihre Funktion tatsächlich wie beschrieben erfüllt, keine inhärenten Sicherheits-Risiken wie etwa gezielt implementierte Malware-Funktionen enthält und nicht von bekannten Sicherheits-Schwachstellen betroffen ist. Sofern die Software direkte Abhängigkeiten zu dritten Software-Projekten aufweist (sog. "Dependencies", sehr typisch in Open Source Software), die zum Einsatz der Software ebenfalls bezogen und installiert werden müssen, sind diese in die erweiterte Integritätsprüfung mit einzubeziehen.

Geeignete Methoden für eine erweiterte Integritätsprüfung können beispielsweise sein:

- strikte Validierung der Projekt-/Paket-Namen (Vermeidung der Verwechslung mit maliziösen, gezielt nachgeahmten Software-Projekten)
- dynamische Code Analyse / strukturierte Funktionsprüfung in einer Test-Umgebung
- statische Code Analyse mittels eines Linters (z. B. Splint, JSLint, Pylint)
- Untersuchung mittels eines Sicherheitsschwachstellen-Scanners (z. B. Qualys, Nessus)
- Untersuchung mittels eines Container Security Scanners (z. B. JFrog Xray, Harbor, Clair, Docker Scan)
- Untersuchung mittels eines SCA (Software Composition Analysis) Tools bzw. Dependency-Scanners (z. B. OWASP Dependency Check, Snyk)

Die Prüfungsmethoden sind passend zur exakten Form der Software-Bereitstellung (Quellcode, Binaries/Artefakte, Container) auszuwählen und angemessen zu kombinieren.

Motivation: Softwarelieferketten enthalten diverse Angriffsvektoren. Ein Angreifer kann an verschiedenen Stellen ansetzen, um Software zu manipulieren oder gezielt eigene Routinen einzubringen und die Zielumgebung, in der die Software anschließend zum Einsatz gebracht wird, zu schädigen oder unter seine Kontrolle zu bringen. Der Angriff kann auf dem Transport- bzw. Übertragungsweg geschehen oder auch auf die Bereitstellungsquelle selbst ausgeübt werden. Entsprechend erleichtert wird ein Angriff, falls Software nicht über offizielle und kontrollierte Quellen bezogen wird oder eine Integritätsprüfung ausgelassen wird.

Besonderes Risiko zeigt sich für Software, die aus öffentlichen Registries bezogen wird, da diese jedermann für die Bereitstellung von Software-Projekten offenstehen. Perfide Angriffsmethoden sind inzwischen bekannt, in denen der Angreifer zunächst eine Zeitlang sicherheitstechnisch völlig unauffällige, funktionale Software bereitstellt und sobald diese sich etabliert und eine gewisse Verbreitung gefunden hat, in zukünftigen Versionen gezielt versteckten Schadcode integriert. Andere Methoden setzen auf ähnlich klingende Projektnamen zu weit verbreiteten Bestandsprojekten oder überstimmende Versionsnummern, um manipulierte Software in darauf aufbauende Lösungen zu injizieren.

Umsetzungsbeispiel: Bezug der Software über die offiziellen Bereitstellungskanäle des Herstellers. Nach Erhalt der Software umgehend Prüfung auf Integrität mittels kryptografischer Checksummen, wie vom Hersteller bereitgestellt, sowie Freiheit von bekannter Schadsoftware mittels Anti-Malware- / Anti-Virus-Scanner. Ablage der geprüften Software auf einer internen, geschützten Dateiablage und weitere Nutzung (z.B. Rollout auf die Zielsysteme) nur noch von dort.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Modifikation von Daten
- Unbemerkt durchführbare Angriffe
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 3.01-2/7.0

Req 15	Bekannte Schwachstellen in der Software oder Hardware des Systems müssen behoben oder gegen Missbrauch abgesichert werden.
--------	--

Bekannte Schwachstellen in Software- und Hardware Komponenten müssen mittels Installation von verfügbaren Systemaktualisierungen des Herstellers (z. B. Patches, Updates/Upgrades) behoben werden. Alternativ ist der Einsatz von Workarounds (Akut-Lösungen, welche die Schwachstelle nicht beheben, aber die Ausnutzung wirksam verhindern) zulässig. Workarounds sollten grundlegend nur temporär Verwendung finden und sobald möglich durch eine reguläre Systemaktualisierung ersetzt werden, um die Schwachstellen vollständig zu schließen.

Komponenten, die bekannte, nicht behebbare Schwachstellen enthalten, dürfen in einem System nicht eingesetzt werden.

Die Behandlung von neu bekannt werdenden Schwachstellen muss zudem für die gesamte Einsatzphase des Systems fortlaufend sichergestellt und in den kontinuierlichen Betriebsprozessen eines Security Patch Managements umgesetzt werden.

Motivation: Der Einsatz von Komponenten ohne eine Behebung enthaltener Schwachstellen erhöht deutlich das Risiko einer erfolgreichen Kompromittierung. Der Angreifer wird zusätzlich dadurch begünstigt, dass i. d. R. nicht nur detaillierte Informationen zu bereits bekannt gewordenen Schwachstellen offen verfügbar sind, sondern oft auch bereits angepasste Angriffswerkzeuge, die eine aktive Ausnutzung erleichtern.

Umsetzungsbeispiel: Im Anschluss an die initiale Installation eines Betriebssystems von einem offiziellen Installationsmedium werden alle aktuell verfügbaren Patches und Sicherheits-Updates eingespielt.

Zusätzliche Informationen:

Primäre Quellen für bereits bekannt gewordene Schwachstellen in einer Software/Hardware sind Auflistungen in den Versionshinweisen sowie die Sicherheitshinweise aus den offiziellen Meldekanälen des Herstellers oder unabhängiger CERTs. Insbesondere die Meldekanäle werden sinnvollerweise fest in kontinuierliche Prozesse des Security Patch Managements für ein System eingebunden, damit neu bekanntwerdende Schwachstellen zeitnah registriert und in operative Behebungsmaßnahmen geführt werden können.

Als ergänzende Maßnahme zur Aufdeckung von potenziell noch enthaltenen, grundsätzlich bereits bekannt gewordenen Arten von Schwachstellen können gezielte Schwachstellenuntersuchungen des Systems durchgeführt werden. Zu diesem Zweck eignen sich besonders spezialisierte Werkzeuge wie automatisierte Schwachstellen-Scanner. Beispiele hierfür sind: Tenable Nessus, Qualys Scanner Appliance.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Störung der Verfügbarkeit
- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 3.01-10/7.0

2.2. Registry

Req 16 Registries müssen über eine Benutzerverwaltung verfügen.

Für jede Registry muss eine Mandantenfähigkeit gewährleistet sein, entweder durch die Verwendung dedizierter Umgebungen oder durch eine Benutzerverwaltung auf der gleichen Instanz.

Motivation: Trennen und verwalten des Zugangs zu Registries.

ID: 3.64-16/4.0

Req 17 Images von externen Registries müssen zuerst in eine interne Registry heruntergeladen werden.

Images von externen Registries müssen zuerst in eine interne Registry heruntergeladen werden, damit sie auf Schwachstellen überprüft werden können

Innerhalb der DT können Sie die Magenta Trusted Registry <https://mtr.devops.telekom.de/> verwenden.

Motivation: Fortlaufende, ständig aktualisierte, zentralisierte Berichterstellung von Images. Eine Überprüfung auf Schwachstellen und Compliance.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Modifikation von Daten

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 3.64-17/4.0

Req 18 Verbindungen zu Registries müssen verschlüsselt sein.

Registries müssen so konfiguriert werden, dass sie von Entwicklungstools, Orchestratoren und Container-Runtimes nur verschlüsselt erreicht werden.

Motivation: Ziel ist es, sicherzustellen, dass alle Daten zwischen vertrauenswürdigen Endpunkten während des Transits verschlüsselt sind.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 3.64-18/4.0

Req 19 Es muss eine ausreichende Authentifizierung und Autorisierung verwendet werden.

Jeder Schreibzugriff auf eine Registry erfordert eine Authentifizierung. Entwickler sollten Images nur in für Sie bestimmten Repositories speichern können.

Images müssen von autorisierten Personal signiert und erst danach in eine Registry verschoben werden.

Motivation: Stellen Sie sicher, dass nur Images von vertrauenswürdigen Stellen zur Registry hinzugefügt werden können.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 3.64-19/4.0

3. Technischer Basisschutz von IT-/NT-Systemen

3.1. Schutz von Daten und Informationen

Req 20	Gespeicherte schutzbedürftige Daten müssen vor nicht autorisierter Einsichtnahme, Veränderung und Löschung geschützt werden.
--------	--

Der Schutzbedarf für gespeicherte Daten ist abhängig von deren Einstufung (beispielsweise nach geltenden gesetzlichen Datenschutzvorgaben, regulatorischen Auflagen, vertraglichen Verpflichtungen), dem Schadenspotential bei deren Missbrauch, sowie weiteren relevanten Faktoren (beispielsweise dem Ort der Speicherung). Art und Umfang von Schutzmaßnahmen müssen entsprechend angemessen gewählt werden.

Gespeicherte Authentisierungsmerkmale wie beispielsweise Passwörter, Private Keys, Tokens oder Zertifikate gelten grundsätzlich als schutzbedürftig. Ebenfalls als grundsätzlich schutzbedürftig gelten Daten, die Funktionalität und sicherheitsrelevantes Verhalten eines Systems bestimmen (z. B. System-Konfigurationsdateien, Betriebssysteme und Kernel, Treiber).

Die Einhaltung der Schutzziele Vertraulichkeit, Integrität und Verfügbarkeit muss für gespeicherte schutzbedürftige Daten durchgängig gewährleistet werden. Dies gilt auch während einer lediglich kurzfristigen Speicherung (z. B. bei der Zwischenspeicherung in einem Web Cache oder in einem temporären Ordner innerhalb einer Datenverarbeitungskette).

Grundlegend muss der Zugriff auf schutzbedürftige Daten in einem System auf Basis von technisch implementierten Berechtigungsvergaben und -kontrollen vollumfassend reglementiert werden.

Reicht eine solche technische Zugriffskontrolle allein nicht mehr aus, um den erforderlichen Schutzbedarf gespeicherter Daten zu gewährleisten, oder kann deren Wirksamkeit nicht durchgängig sichergestellt werden, müssen zusätzlich kryptografische Verfahren (z. B. Verschlüsselung, Signierung, Hashing) implementiert werden. Bei der Speicherung von Daten eingesetzte kryptografische Verfahren müssen für diesen Zweck geeignet sein und dürfen keine bekannten Schwachstellen aufweisen.

Motivation: Die Speicherung von Daten auf einem System ohne einen adäquaten Schutz ermöglicht es einem Angreifer, diese unberechtigt einzusehen, zu nutzen, zu verbreiten, zu verändern, oder zu vernichten. Dies eröffnet potenziell zusätzliche Angriffsvektoren auf die unmittelbaren sowie angeschlossene weitere Systeme und kann zu erheblichen Ausfällen, Kontrollverlusten und Schäden sowie resultierenden Strafzahlungsforderungen und Reputationsverlusten gegenüber Kunden und Geschäftspartnern führen.

Umsetzungsbeispiel: [Beispiel 1]

Ein System exportiert Daten für einen Transport auf einen mobilen Datenträger. Da die technische Zugriffskontrolle des Systems auf Dateiberechtigungsebene nicht mehr greift, sobald der mobile Datenträger aus dem System entfernt wird, müssen zusätzliche Maßnahmen zum Schutz der Daten getroffen sein. Bevor das System die Daten auf den mobilen Datenträger schreibt, werden sie entsprechend mit einem passenden Verfahren verschlüsselt. Der zugehörige Schlüssel wird auf einem separaten Kanal ausgetauscht, so dass die Daten im legitimen Zielsystem wieder entschlüsselt und verarbeitet werden können. Ein Angreifer, der den mobilen Datenträger in seinen Besitz bringt, hat hingegen keinen Zugriff auf die Daten.

[Beispiel 2]

In der lokalen Benutzer-Datenbank eines Systems werden nur kryptografische Hashes der Passwörter gespeichert, die mit einem als sicher geltenden Password-Hashing Verfahren erzeugt wurden. Für das System sind diese Hashes ausreichend, um die Benutzer bei einer Anmeldung am System zu authentifizieren. Sollte ein Angreifer die Benutzer-Datenbank kopieren können, gelangt er somit jedoch nicht unmittelbar in den Besitz von Klartext-Passwörtern, mit denen er sich am System im Namen der Benutzer anmelden könnte.

[Beispiel 3]

Auf einem System sind die Konfigurationsdateien des Web-Servers ausschließlich für den legitimen Admin schreibbar, in dem im Dateisystem entsprechende Berechtigungen gesetzt wurden. Die Zugriffskontrolle des Betriebssystem-Kernels verweigert somit allen anderen Benutzern des Systems, Veränderungen an den Konfigurationsdateien des Web-Servers vorzunehmen; einschließlich dem Web-Server Dienste-Konto selbst, was auch die Angriffsfläche von außen bei Schwachstellen im Web-Server verkleinert.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Störung der Verfügbarkeit
- Unbemerkt durchführbare Angriffe
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 3.01-14/7.0

Req 21	Schutzbedürftige Daten müssen während der Übertragung vor nicht autorisierter Einsichtnahme und Veränderung geschützt werden.
--------	---

Der Schutzbedarf von zu übertragenden Daten ist abhängig von deren Einstufung (beispielsweise nach geltenden gesetzlichen Datenschutzvorgaben, regulatorischen Auflagen, vertraglichen Verpflichtungen), dem Schadenspotential bei deren Missbrauch, sowie weiteren relevanten Faktoren (beispielsweise der Übertragung über öffentliche Netze). Art und Umfang der Schutzmaßnahmen müssen entsprechend angemessen gewählt werden.

Authentisierungsmerkmale wie beispielsweise Passwörter oder Tokens gelten grundsätzlich als schutzbedürftige Daten. Ebenfalls grundsätzlich als schutzbedürftig gelten Daten, die Funktionalität und sicherheitsrelevantes Verhalten eines Systems bestimmen (z. B. Updates & Patches, Konfigurationsparameter, Fernwartungen, Steuerung über APIs).

Die Einhaltung der Schutzziele Vertraulichkeit und Integrität muss während der Übertragung von schutzbedürftigen Daten durchgängig gewährleistet werden.

I.d.R. erfordert dies eine Implementation von kryptografischen Verfahren (z. B. Verschlüsselung, Signaturen, Hashes). Kryptografische Verfahren können

- vor der Übertragung direkt auf die Daten angewendet werden, was eine anschließende Übertragung der Daten auch über unsichere Kanäle vertretbar machen kann
- auf dem Übertragungs-Kanal eingesetzt sein, um einen sicheren Kanal zu schaffen und beliebige Daten, die durch diesen geleitet werden, zu schützen
- oder auch als Kombination aus beidem umgesetzt sein.

Bei der Übertragung von Daten eingesetzte kryptografische Verfahren müssen für diesen Zweck geeignet sein und dürfen keine bekannten Schwachstellen aufweisen.

Motivation: Die Übertragung von Daten ohne einen adäquaten Schutz ermöglicht es einem Angreifer, diese unberechtigt abzufangen, zu nutzen, zu verbreiten, zu verändern oder aus der Übertragung zu entfernen. Dies eröffnet potenziell weitere Angriffsvektoren auf die unmittelbaren Zielsysteme sowie angeschlossene weitere Systeme und kann zu erheblichen Ausfällen, Kontrollverlusten und Schäden sowie resultierenden Strafzahlungsforderungen und Reputationsverlusten gegenüber Kunden und Geschäftspartnern führen.

Umsetzungsbeispiel: [Beispiel 1]

Vertrauliche Dokumente werden vor dem Versand per E-Mail an den Kunden verschlüsselt.

[Beispiel 2]

Ein Administrator konfiguriert eine neue Cloud-Anwendung über das Internet. Der Zugriff erfolgt über eine TLS-verschlüsselte Verbindung ("https").

[Beispiel 3]

Ein System bezieht automatische Software-Updates von einem Update-Server. Der Update-Server liefert die Software Updates kryptografisch signiert aus. Das System kann die empfangenen Software-Updates somit validieren und sicher ausschließen, dass diese während der Übertragung manipuliert wurden.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Störung der Verfügbarkeit

- Unbemerkt durchführbare Angriffe
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 3.01-15/7.0

3.2. Authentisierungsmerkmal "Passwort"

Req 22	Falls Passwörter als Authentisierungsmerkmal für technische Benutzerkonten genutzt werden, müssen diese mindestens 30 Zeichen lang sein und drei der folgenden Zeichenkategorien beinhalten: Kleinbuchstaben, Großbuchstaben, Ziffern und Sonderzeichen.
--------	--

Technische Benutzerkonten zeichnen sich dadurch aus, dass sie nicht von Personen verwendet werden. Stattdessen werden sie für die Authentifizierung und Autorisierung von Systemen untereinander oder von Anwendungen auf einem System genutzt.

Ein System darf für technische Benutzerkonten nur Passwörter verwenden, welche die folgende Komplexität erfüllen:

- Mindestlänge 30 Zeichen
- Bestehend aus mindestens drei der folgenden vier Zeichenkategorien:
 - Kleinbuchstaben
 - Großbuchstaben
 - Ziffern
 - Sonderzeichen

Motivation: Technische Benutzerkonten sind aufgrund ihrer Verwendung in Machine-to-Machine (M2M) Kommunikationsszenarien oft mit Berechtigungen ausgestattet, die für einen Angreifer von hohem Interesse sein können, um Infrastrukturen zu kompromittieren. Ohne Mechanismen einer umfassenden Kompromittierungserkennung kann dann das Risiko, dass ein Passwort von einem Angreifer ermittelt oder gebrochen wird, über die Zeit deutlich ansteigen. Eine signifikante Erhöhung der Passwort-Länge wirkt diesen Risiken entgegen und lässt sich zudem in M2M Szenarien besonders leicht umsetzen, da die Handhabung eines sehr langen Passworts für eine Maschine (im Gegensatz zu einer Person) keine besondere Herausforderung darstellt.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Abstreiten von durchgeführten Aktionen

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 3.01-27/7.0

3.3. Protokollierung

Req 23	Sicherheitsrelevante Ereignisse müssen mit einem exakten Zeitstempel und einer eindeutigen Systembezeichnung protokolliert werden.
--------	--

Systeme müssen das Auftreten von sicherheitsrelevanten Ereignissen protokollieren. Um eine Auswertbarkeit und Zuordnung der auftretenden Ereignisse zu ermöglichen, müssen diese zusammen mit einer eindeutigen Systembezeichnung (z. B. Hostname, IP- oder MAC-Adresse) und dem genauen Zeitpunkt des Auftretens ("Zeitstempel") protokolliert werden.

Von dieser Anforderung ausgenommen sind Systeme bei denen auf Grund ihrer Bauart, ihres Verwendungszwecks

oder Einsatzgebietes eine Protokollierung von sicherheitsrelevanten Ereignissen nicht sinnvoll realisierbar ist. Solche Systeme sind beispielsweise Kundenendgeräte wie Smartphones oder IADs/Homegateways (z. B. Speedport).

Der Zeitstempel eines protokollierten Ereignisses muss mindestens folgende Informationen beinhalten:

- Datum des Ereignisses (Jahr, Monat, Tag)
- Uhrzeit des Ereignisses (Stunden, Minuten, Sekunden)
- Zeitzone, auf die sich diese Zeitangaben beziehen

Bei der Protokollierung müssen die jeweils geltenden gesetzlichen und betrieblichen Bestimmungen beachtet werden. Letztere umfassen insbesondere auch Vereinbarungen, die mit den betrieblichen Sozialpartnern getroffen wurden. Diese Bestimmungen besagen unter anderem, dass die Protokollierung von Ereignissen nur zweckgebunden erfolgen darf. Die Protokollierung von Ereignissen zum Zweck der Arbeitskontrolle von Mitarbeitern ist nicht zulässig.

Zudem muss - wie für jegliche Daten, die von einem System verarbeitet werden - auch für Protokollierungsdaten ein angemessener Schutzbedarf berücksichtigt und umgesetzt werden; dies gilt für die Speicherung, die Übertragung sowie den Zugriff. Insbesondere falls Wirkdaten in den Protokollierungsdaten enthalten sind, muss derselbe Schutzbedarf berücksichtigt werden, der auch für die reguläre Verarbeitung dieser Wirkdaten innerhalb des Quell-Systems Anwendung findet.

Typische Ereignisse, die in den meisten Fällen sinnvoll sein können, sind:

Ereignis	Zu protokollierende Daten des Ereignisses
Fehlerhafte Anmeldeversuche	• Benutzerkonto, • Anzahl der Fehlversuche, • Quelle (IP-Adresse, Client-ID / Client-Name) bei entferntem Zugriff
Systemzugriff von Benutzerkonten mit Administrationsrechten	• Benutzerkonto, • Zugriffszeitpunkt, • Dauer des Zugriffs, • Quelle (IP-Adresse) bei entferntem Zugriff
Administration von Konten	• Benutzerkonto des Administrators, • Administriertes Benutzerkonto, • Durchgeführte Tätigkeit (Einrichten, Löschen, Aktivieren und Deaktivieren)
Änderung von Gruppenzugehörigkeiten von Konten	• Benutzerkonto des Administrators, • Administriertes Benutzerkonto, • Durchgeführte Tätigkeit (hinzugefügte oder entfernte Gruppe)
Kritischer Anstieg von Systemwerten wie Speicherauslastung, CPU-Auslastung über einen längeren Zeitraum	• Überschrittener Wert, • Erreichter Wert (Hierfür müssen abhängig vom System sinnvolle Schwellwerte definiert werden.)

Weitere Ereignisse für den jeweils verwendeten Systemtyp müssen im Einzelfall geprüft werden.

Motivation: Die Protokollierung von sicherheitsrelevanten Ereignissen ist eine Grundvoraussetzung, um laufende Angriffe oder auch Angriffe im Nachhinein erkennen zu können. Nur so ist es möglich, sinnvolle Maßnahmen zur Erhaltung oder auch Wiederherstellung der Systemsicherheit durchzuführen. Des Weiteren dienen die Protokollierungsdaten der Beweissicherung, um rechtlich gegen Angreifer vorgehen zu können.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 3.01-33/7.0

Req 24	Sicherheitsrelevante Protokollierungsdaten müssen umgehend nach deren Erzeugung an einen separaten Log-Server weitergeleitet werden.
--------	--

Protokollierungsdaten müssen umgehend nach deren Erzeugung an einen separaten Log-Server weitergeleitet werden. Standardisierte Protokolle wie Syslog, SNMPv3 sollten hierbei bevorzugt zum Einsatz kommen.

Motivation: Werden Protokollierungsdaten nur lokal gespeichert, können diese im Falle einer erfolgreichen Kompromittierung des Systems durch einen Angreifer manipuliert werden, um seinen Angriff und durchgeführte Manipulationen des Systems zu verschleiern. Aus diesem Grund muss die Weiterleitung zeitnah nach dem Auftreten des Ereignisses erfolgen.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Modifikation von Daten
- Störung der Verfügbarkeit
- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 3.01-35/7.0

Req 25	Für sicherheitsrelevante Protokollierungsdaten, die an den separaten Log Server weitergeleitet werden, muss die Einhaltung geltender Aufbewahrungszeiträume und Löschfristen sichergestellt werden.
--------	---

Folgende Grundregelung ist zu berücksichtigen:

- sicherheitsrelevante Protokollierungsdaten müssen auf dem separaten Log Server für einen Zeitraum von 90 Tagen aufbewahrt werden.
- nach Ablauf von 90 Tagen muss eine umgehende Löschung der auf dem separaten Log Server gespeicherten Protokollierungsdaten erfolgen.

Abweichungen

Abweichende Aufbewahrungszeiträume und Löschfristen können durch gesetzliche oder regulatorische Anforderungen bestehen (insbesondere im Zusammenhang mit personenbezogenen Daten) oder durch vertragliche Vereinbarungen definiert sein. In diesen Fällen sind die geltenden Zeiträume individuell mit einem Project Security Manager (PSM) bzw. Datenschutzberater (DSB) abzustimmen oder werden durch diesen vorgegeben.

Log Server in Verantwortung eines Dritten

Sollte der ausgewählte separate Log Server nicht innerhalb derselben operativen Verantwortung liegen wie das Quellsystem der Protokollierungsdaten, muss sichergestellt werden, dass dem verantwortlichen Betreiber des Log Servers die gültigen Parameter für die zu empfangenden Protokollierungsdaten bekannt sind und durch diesen im Sinne der hier genannten Regelungen eingehalten werden.

Motivation: Protokollierungsdaten sind ein immens wichtiges Werkzeug der IT-Sicherheit zur Abwendung, Detektierung und Aufklärung von Systemstörungen, Sicherheits- und Datenschutzvorfällen. Auf der anderen Seite unterliegt

die Erfassung von Protokollierungsdaten wie jede andere Datenverarbeitung auch gesetzlichen und regulatorischen Auflagen. Entsprechend müssen Richtlinien eingehalten werden, die beides in Einklang bringen.

Umsetzungsbeispiel: Unter Berücksichtigung der gültigen Gesetzeslage und zutreffenden Datenschutz-Regularien werden folgende Löschfristen für ausgeleitete sicherheitsrelevante Protokollierungsdaten eines beispielhaften TK-Systems auf dem separaten Log Server umgesetzt:

- Standard System Logs: Löschung nach spätestens 90 Tagen
- Protokollierungen von öffentlichen IP-Adressen: Löschung (oder Anonymisierung) nach spätestens 7 Tagen
- Protokollierungen der Vergabe von dynamischen öffentlichen IP-Adressen durch die TK-Lösung: Löschung nach spätestens 7 Tagen
- Protokollierungen nicht abrechnungsrelevanter Call Detail Records: Löschung nach spätestens 7 Tagen
- Protokollierungen der Kommunikations-Inhalte von E-Mail und SMS: Löschung nach spätestens 24 Stunden
- Protokollierungen der behandelten Domain-Abfragen des DNS-Servers der TK-Lösung: Löschung spätestens nach 24 Stunden

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

ID: 3.01-36/7.0

Req 26	Das System muss Protokollierungsdaten bereitstellen, anhand derer die systemspezifisch relevanten Angriffsformen in einem SIEM detektierbar sind.
--------	---

Die Angriffsformen, die typischerweise für das vorliegende System zu erwarten sind, müssen systematisch analysiert und identifiziert werden.

Als strukturierte Orientierungshilfe während einer solchen Identifizierung kann die Attack Matrix des MITRE herangezogen werden (<https://attack.mitre.org>).

Es muss sichergestellt werden, dass das System entsprechende Protokollierungsdaten zu Ereignissen erzeugt, die im Zusammenhang mit diesen identifizierten Angriffsformen stehen oder stehen können und anhand derer ein stattfindender Angriff erkannt werden kann.

Die Protokollierungsdaten müssen unverzüglich nach dem Auftreten des System-Ereignisses an ein SIEM ausgeleitet werden.

SIEM (Security Information & Event Management) Lösungen erfassen Ereignisprotokolldaten aus diversen Quell-Systemen, korrelieren diese und werten sie automatisiert in Echtzeit aus, um anomale Aktivitäten wie beispielsweise laufende Angriffe auf IT/NT-Systeme zu entdecken und Alarmierungen oder Gegenmaßnahmen einleiten zu können. Der unverzügerte Eingang von System-Ereignissen im SIEM ist daher absolut entscheidend, damit dieses seine Schutzfunktionen erfüllen kann.

Hinweis:

Die unmittelbare Notwendigkeit, ein System an ein SIEM anzubinden, wird gezielt über die separaten Sicherheitsanforderungskataloge "Betrieb" geregelt.

Sollte das vorliegende System nicht unter diese Notwendigkeit fallen, darf die Anforderung als "nicht zutreffend" beantwortet werden.

Motivation: Ein SIEM als automatisiertes Detektionssystem für Angriffe kann nur dann effektiv sein, wenn es fortlaufend ausreichende und vor allem systemspezifisch relevante Ereignis-Meldungen aus den zu überwachenden Infrastrukturen und Systemen erhält. Allgemeine Standard Ereignis-Meldungen sind unter Umständen nicht ausreichend, um ein angemessenes Detektions-Niveau zu erreichen und ermöglichen nur rudimentäre Angriffserkennungen.

Umsetzungsbeispiel: Ein Beispiel-System ermöglicht Endbenutzern die Anmeldung mittels Benutzername und Passwort. Eine der typischen Angriffsformen für dieses System wäre, mittels automatisiertem Ausprobieren von Passwörtern (Wörterbuch- oder Brute-Force-Angriff) zu versuchen, Benutzerkonten mit schwachen oder häufig verwendeten Passwörtern zu entdecken und diese zu übernehmen. Das Beispiel-System wird konfiguriert, jedes fehlgeschlagene Login-Ereignis in Protokollierungsdaten ("Logs") festzuhalten. Indem diese Protokollierungsdaten parallel an ein SIEM ausgeleitet werden, kann das SIEM in Echtzeit erkennen, dass hier offensichtlich ein Angriff stattfindet, diesen alarmieren und somit unmittelbar Gegenmaßnahmen ermöglichen.

ID: 3.01-37/7.0