

Sicherheitsanforderung

Linux OS für Server

Deutsche Telekom Gruppe

Version	9.1
Datum	01.12.2025
Status	Freigegeben

Impressum

Herausgeber
Deutsche Telekom AG
Vorstandsbereich Technology & Innovation
Chief Security Officer

Reuterstrasse 65, 53315 Bonn
Deutschland

Dateiname	Dokumentennummer 3.65	Dokumententyp Sicherheitsanforderung
Version 9.1	Stand 01.12.2025	Status Freigegeben
Fachlicher Ansprechpartner Telekom Security psa.telekom.de	Gültigkeitsdauer 01.12.2025 - 30.11.2030	Freigegeben von Stefan Pütz, Leiter SEC-T-TST

Zusammenfassung
Sicherheitsanforderungen für Linux OS für Server inklusive Anforderungen für IPTables, Mandatory Access Control
etc.

Copyright © 2025 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
1.1.	Gültigkeit	4
1.2.	Ausschluss	4
2.	Systemhärtung	5
2.1.	Netzwerkhärtung	12
2.2.	Systemaktualisierung	14
2.3.	Authentifizierung von Benutzern	19
2.4.	Administrativer Zugang	24
2.5.	Lokation	25
3.	Protokollierung	27
3.1.	Auditd	30
3.2.	Externe Protokollierung	34
3.2.1.	RSyslog	35
3.2.2.	Syslog-NG	36
4.	Authentisierungsmerkmal "Passwort" & Pluggable Authentication Modules	37
5.	Host-basierte Firewall	43
6.	Mandatory Access Control (MAC)	45
6.1.	SELinux	45
6.2.	AppArmor	47
7.	Regelmäßige Compliance Checks	49

1. Einleitung

Linux ist ein weit verbreitetes Betriebssystem für Server. Es wird auf einzelnen "Bare Metal" Servern wie auch hoch skalierbaren Cloud-Umgebungen genutzt. Dieses Dokument beinhaltet Sicherheitsanforderungen für das Basisbetriebssystem und auch für Applikationen, die Bestandteile des Betriebssystems sind, wie z.B. IPTables, Mandatory Access Control Lösung. Die Sicherheitsanforderungen sind für alle Linux-Distributionen gültig.

Dieses Dokument wurde auf Basis der Vorgaben aus den im Konzern gültigen Sicherheitsrichtlinien erstellt und basiert auf Best-Practice-Sicherheitsvorgaben und Herstellerempfehlungen.

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.

Ein Werkzeug zur Härtung eines Linux-Systems gemäß den Anforderungen in diesem Dokument finden Sie [hier](#).

1.1. Gültigkeit

Dieses Dokument ist gültig für Server-Systeme mit Linux Betriebssystem. Dies schließt die folgenden Server-Typen ein:

- Bare-Metal-Servers
- Virtualisierte Server (VMs)
- Virtualisierungs-Hosts

1.2. Ausschluss

Dieses Dokument kann nicht für Workstations mit Linux Betriebssystem und Container-Images basierend auf Linux verwendet werden. Auf Grund der unterschiedlichen Anwendungsbereiche, der damit verbundenen Bedrohungen und der verwendeten Software benötigen Workstations unterschiedliche Anforderungen. Container-Images müssen anders behandelt werden, da sie normalerweise kein komplettes Linux Betriebssystem beinhalten.

2. Systemhärtung

Req 1 Nicht benötigte Dienste müssen deaktiviert werden.

Nach der Installation von Systemen und Softwareprodukten sind häufig herstellerseitig voreingestellte, lokale oder aus dem Netz erreichbare Dienste aktiv, die für den Betrieb und die Funktionalität des konkreten Systems im vorgesehenen Einsatzumfeld nicht erforderlich sind.

Auf einem System dürfen jedoch grundsätzlich nur die tatsächlich benötigten Dienste aktiv sein.

Entsprechend müssen unmittelbar nach der Installation sämtliche auf einem System nicht erforderliche Dienste vollständig deaktiviert werden. Es muss sichergestellt werden, dass diese Dienste auch nach einem Neustart des Systems weiterhin deaktiviert bleiben.

Motivation: Aktive Dienste, die nicht benötigt werden, vergrößern unnötig die Angriffsfläche eines Systems und in unmittelbarer Folge das Risiko einer erfolgreichen Kompromittierung. Dieses Risiko kann zusätzlich nochmals erhöht sein, falls - wie bei nicht benötigten Diensten häufig beobachtet - eine gezielte Betrachtung und Optimierung der Konfiguration hinsichtlich Sicherheit nicht ausreichend stattfindet.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Nutzung von Diensten oder Ressourcen
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.01-5/9.0

Req 2 Die Erreichbarkeit von aktivierten Diensten muss eingeschränkt werden.

Ein bereitgestellter Dienst muss grundsätzlich auf allen Schnittstellen des Systems vollständig deaktiviert werden, über die eine Erreichbarkeit des Dienstes für den ordnungsgemäßen Betrieb des Systems nicht erforderlich ist. Die Deaktivierung ist primär durch eine entsprechende Konfiguration des Dienstes bzw. Betriebssystems umzusetzen. In Fällen, in denen die zu Verfügung stehenden Konfigurationsoptionen eine Deaktivierung auf einzelnen Schnittstellen nicht ermöglichen, darf stattdessen auch ein lokaler Filter ("Host Firewall") auf dem System eingesetzt werden, um Zugriffe auf den Dienst über nicht erforderliche Schnittstellen zu blockieren.

Die Erreichbarkeit eines Dienstes über die erforderlichen Schnittstellen muss zudem auf legitime Kommunikationspartner eingeschränkt werden. Die Einschränkung ist durch eine entsprechende Konfiguration des Dienstes bzw. Betriebssystems oder mittels eines lokalen Filters ("Host Firewall") umzusetzen. Alternativ darf diese Aufgabe an ein netzseitiges Filterelement ausgelagert werden, sofern sich das System in einem separaten Netzwerk-Segment befindet und die Kommunikation mit diesem Segment ausschließlich über das netzseitige Filterelement möglich ist.

Motivation: Mittels Deaktivierung von Diensten auf Schnittstellen, über die eine Erreichbarkeit nicht notwendig ist, sowie durch die Einschränkung möglicher Kommunikationspartner, kann die Angriffsfläche, die ein System bietet, stark reduziert werden.

Umsetzungsbeispiel: Ein SNMP-Dienst, über den ein System gemonitort wird, ist ausschließlich auf der dedizierten Management-Netzwerk-Schnittstelle des Systems aktiviert. Über eine Firewall ist zusätzlich reglementiert, dass einzig das legitime Monitoring System der Infrastruktur-Umgebung diesen Dienst erreichen kann.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Nutzung von Diensten oder Ressourcen
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.01-6/9.0

Req 3 Nicht benötigte Software darf nicht installiert oder muss deinstalliert sein.

Es kann Software bei der Installation eines Linux Betriebssystems mit installiert sein, die nicht für die Funktion des Servers notwendig ist. Solche Software sollte nicht installiert sein oder muss nach der Installation deinstalliert sein. Bei-

spiele für Software, die typischerweise nicht auf Servern notwendig ist, sind:

- inetd
- xinetd
- X Window System
- Avahi Server
- CUPS
- rsync service
- NIS server/client
- talk server/client
- telnet server/client
- tftp server/client
- ftp server/client
- rsh server/client

Hinweis: Es ist nicht erlaubt Software auf einem Server zu installieren, die nicht für den Betrieb, Wartung oder Funktion benötigt werden!

Motivation: Schwachstellen in der Software eines Systems vergrößern die Angriffsfläche. Durch die Deinstallation nicht benötigter Komponenten werden Angriffsvektoren eliminiert.

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

ID: 3.65-3/9.1

Req 4	Nicht benötigte Funktionen in der eingesetzten Software und Hardware müssen deaktiviert werden.
-------	---

Während der initialen Installation von Software können zunächst Funktionen standardmäßig aktiviert worden sein, die für den Betrieb und die Funktionalität des konkreten Systems nicht notwendig sind. Funktionen sind meistens ein fester Bestandteil der Software, der nicht einzeln gelöscht oder deinstalliert werden kann.

Solche Funktionen müssen unmittelbar nach der initialen Installation über die Konfigurationseinstellungen der Software deaktiviert werden, so dass sie auch nach einem Neustart des Systems dauerhaft deaktiviert bleiben.

Bereits vor der Auslieferung oder während der initialen Inbetriebnahme sind möglicherweise auch in der Hardware zunächst Funktionen standardmäßig aktiviert worden, die für den Einsatz des konkreten Systems nicht benötigt werden. Solche Funktionen, wie beispielsweise nicht benötigte Schnittstellen, müssen ebenfalls unmittelbar nach der initialen Inbetriebnahme dauerhaft deaktiviert werden.

Motivation: Die Hardware oder Software eines Systems enthält oft aktivierte Funktionen, die nicht genutzt werden. Solche Funktionen können eine unnötige Angriffsfläche für Manipulationen darstellen. Des Weiteren entsteht darüber ein Potential, dass unautorisiert auf Bereiche oder Daten des Systems zugegriffen werden kann.

Umsetzungsbeispiel: [Beispiel 1]

Deaktivierung von Debugging-Funktionen in einer Software, die im Falle der Entstörung genutzt werden, aber während des normalen Betriebs nicht aktiv sein müssen.

[Beispiel 2]

Deaktivierung von ungenutzten Netzwerk-Schnittstellen eines Servers.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Nutzung von Diensten oder Ressourcen
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.01-4/9.0

Req 5	Dedizierte Partitionen müssen für anwachsenden Inhalte genutzt werden, die die Verfügbarkeit des Systems beeinträchtigen können.
-------	--

Es muss eine eigene Partition für Verzeichnisse angelegt werden, in denen dynamische Inhalte gespeichert werden. Es wird empfohlen, eigene Partitionen für die folgenden Verzeichnisse zu nutzen:

- /tmp
- /var

In speziellen Fällen kann es sinnvoll sein die folgenden Partitionen zu nutzen:

- /var/log und /var/tmp (anstelle von /var)
- /home (für File-Server mit einer großen Benutzerzahl)

Motivation: Ein volles Dateisystem kann die Funktion eines Servers stören. Das kann bei Angriffen ausgenutzt werden, um die Verfügbarkeit eines Servers zu beeinträchtigen.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Störung der Verfügbarkeit

ID: 3.65-5/9.1

Req 6	Die Parameter nodev, nosuid und noexec müssen für bestimmte Partitionen gesetzt werden.
-------	---

Die genannten Mount-Optionen müssen für die folgenden Partitionen konfiguriert werden, wenn sie existieren:

- /tmp (nodev, nosuid)
- /var/tmp (nodev, nosuid, noexec)
- /home (nodev)

Hinweis: Für Installationen kann es notwendig sein, 'noexec' für die Partition '/tmp' zu entfernen, da diese Partition teilweise für die Ausführung von Skripten während der Installation genutzt wird.

Wenn separate Partitionen existieren, müssen auch die folgenden Partitionen konfiguriert werden:

- /dev/shm (RHEL, SLES) (nodev, nosuid, noexec)
- /run/shm (Ubuntu) (nodev, nosuid, noexec)

Motivation: Um potentiell bösartige Rechteauserweiterungen ("privilege escalation") zu verhindern müssen bestimmte Partitionen mit Einschränkungen eingehangen werden.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Störung der Verfügbarkeit

Req 7	Das automatische Einhängen ("mounting") von Dateisystemen mittels "autofs" muss deaktiviert werden.
-------	---

Das automatische Mounten von Dateisystemen muss deaktiviert werden, um zu verhindern, dass externe Laufwerke wie USB-Sticks und CD-ROMs automatisch eingebunden werden.

Motivation: Wenn automatisches Mounten aktiviert ist, werden externe Laufwerke automatisch eingebunden und können ggf. missbräuchlich genutzt werden.

Umsetzungsbeispiel: Debian-basiertes Linux:

```
$ sudo apt purge autofs
```

RedHat-basiertes Linux:

```
$ sudo yum remove autofs
```

ODER

```
$ sudo systemctl disable --now autofs
```

Für diese Anforderung sind folgende Bedrohungen relevant:

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

Req 8	USB-Storage muss deaktiviert werden, dies gilt für virtuelle sowie physische Server.
-------	--

Das Kernel-Modul "usb-storage" dient als Treiber für USB-Massenspeicher und darf nicht mittels Standardeinstellungen geladen werden. Dazu wird "modprobe" konfiguriert das Modul "usb-storage" bei Systemstart nicht zu laden. Bei Bedarf kann das Modul zur Laufzeit geladen, verwendet und anschließend wieder entfernt werden.

Motivation: USB-Massenspeicher finden für gewöhnlich kein Einsatz im produktiven Serverbetrieb. Aufgrund der Möglichkeit durch sie Schadsoftware einzubringen bzw. Daten zu entwenden müssen sie deaktiviert werden.

Umsetzungsbeispiel: /etc/modprobe.d/usb-storage.conf enthält:

```
install usb-storage /bin/true
```

Alternativ kann ebenso

```
install usb-storage /bin/false
```

genutzt werden, für den Fall, dass man negatives Feedback zurückgeben möchte Versuch "usb-storage" zu laden.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unbemerkt durchführbare Angriffe

Req 9	Die Nutzung von "at" und "cron" muss auf autorisierte Konten beschränkt werden.
-------	---

Die Nutzung der Programme "at" und "cron", welche zur planmäßigen Ausführung von Aufgaben dienen, muss auf autorisierte Konten beschränkt werden.

Motivation: Um Missbrauch vorzubeugen müssen die Programme eingeschränkt werden.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Nutzung von Diensten oder Ressourcen

ID: 3.65-9/9.1

Req 10 Das Sticky Bit muss für World-Writable-Verzeichnisse gesetzt werden.

Das Sticky-Bit in einem Verzeichnis bedeutet, dass eine Datei in diesem Verzeichnis nur vom zugehörigen Konto einer Datei, vom zugehörigen Konto des Verzeichnisses und von einem privilegierten Prozess umbenannt oder gelöscht werden kann.

Motivation: Das Setzen des Sticky Bit für World-Writable-Verzeichnisse verhindert, dass Benutzer Dateien in diesen Verzeichnissen löschen oder umbenennen, die ihnen nicht gehören.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten

ID: 3.65-10/9.1

Req 11 Es dürfen keine World-Writable-Dateien existieren.

World-Writable-Dateien sind Dateien, auf die von allen Konten eines Systems schreibend zugegriffen werden kann. Solche Dateien müssen identifiziert werden und die Rechte der entsprechenden Dateien müssen auf ein adäquates Niveau angepasst werden.

Motivation: Daten in World-Writable-Dateien können von jedem Konto eines Systems gelesen, verändert und ggf. kompromittiert werden.

Umsetzungsbeispiel: Auffinden von World-Writable-Dateien im gesamten Dateisystem mit:

```
sudo find / -type f -perm -002
```

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten

ID: 3.65-11/9.1

Req 12 Sitzungen müssen nach einer dem Verwendungszweck angepassten Zeit der Inaktivität automatisch beendet werden.

Es ist notwendig, dass Sitzungen auf einem System nach einer vorgegebenen Zeit der Inaktivität automatisch beendet werden.

Aus diesem Grund muss ein Time-Out für Sitzungen gesetzt werden. Der hier zu wählende Zeitraum richtet sich nach der Verwendung des Systems und ggf. nach der physischen Umgebung. So ist beispielsweise der Time-Out für eine Anwendung, die in einer ungesicherten Umgebung genutzt wird, kürzer (wenige Minuten) zu wählen als der Time-Out für eine Anwendung, die von Betriebsmitarbeitern für System-Monitoring-Aufgaben in einem zugangsgeschützten Bereich genutzt wird (60 Minuten und länger).

Motivation: Für eine offene, aber ungenutzte Sitzung besteht ein Risiko, dass ein nicht legitimer Benutzer diese unbemerkt übernimmt und weiterführt, um im Namen des betroffenen Benutzers einen nicht autorisierten Zugriff auf das System und enthaltene Daten auszuüben.

Umsetzungsbeispiel: Für Systeme, die auf sh, mksh, bash, zsh, ksh oder rsh setzen enthält /etc/profile.d/session-timeout.sh:

```
TMOUT=900
readonly TMOUT
export TMOUT
```

Req 13 Der Standard für "umask" muss 027 oder restriktiver sein.

Die Einstellung der "umask" legt fest, welche Berechtigungen Dateien und Verzeichnisse bekommen, wenn sie erzeugt werden. Standardeinstellung diverser Distributionen sind nicht restriktiv genug. Daher muss "umask" konfiguriert werden.

Eine "umask" von 027 wird gefordert. Dies definiert die Berechtigungen "Lesen, Schreiben, Ausführen" (0) für Anwender, "Lesen, Ausführen" (2) für Gruppen und keine Berechtigung (7) für Andere.

Motivation: Mit einer eingeschränkten "umask" kann der unautorisierte Zugriff auf Dateien und Verzeichnisse verhindert werden.

Umsetzungsbeispiel: Für Systeme mit ausschließlich bash und sh enthält `/etc/profile.d/set_umask.sh`:

```
umask 027
```

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen

Req 14 Nicht benötigte SUID und SGID müssen von ausführbaren Dateien entfernt werden.

Dateien mit SUID und SGID werden mit erweiterten Berechtigungen ausgeführt und können ein Sicherheitsrisiko darstellen. Daher muss die Menge von ausführbaren Dateien mit SUID und SGID auf das notwendige Minimum reduziert werden.

Eine Alternative ist es granulare Berechtigungen mittels POSIX Capabilities zu realisieren. Diese Lösung ermöglicht einer Binärdatei nur erlaubte Systemfunktionen zu zuweisen und nicht komplette Root-Berechtigung wie mit SUID/SGID. Es wird empfohlen, wenn möglich POSIX Capabilities anstelle von SUID/SGID zu nutzen!

Die folgenden ausführbaren Dateien dürfen mit SUID und SGID ausgeführt werden, wenn Posix Capabilities nicht benutzt werden können:

- `/bin/ping`
- `/sbin/pam_timestamp_check`
- `/sbin/unix_chkpwd`
- `/usr/bin/at`
- `/usr/bin/gpasswd`
- `/usr/bin/locate`
- `/usr/bin/newgrp`
- `/usr/bin/passwd`
- `/usr/bin/ssh-agent`
- `/usr/libexec/utempter/utempter`
- `/usr/sbin/lockdev`
- `/usr/sbin/sendmail.sendmail`
- `/usr/bin/expiry`
- `/bin/ping6`
- `/usr/bin/traceroute6.iputils`
- `/sbin/mount.nfs`

- /sbin/umount.nfs
- /sbin/mount.nfs4
- /sbin/umount.nfs4
- /usr/bin/crontab
- /usr/bin/wall
- /usr/bin/write
- /usr/bin/screen
- /usr/bin/mlocate
- /usr/bin/chage
- /usr/bin/chfn
- /usr/bin/chsh
- /bin/fusermount
- /bin/fusermount3
- /usr/bin/pkexec
- /usr/bin/su
- /usr/bin/sudo
- /usr/bin/sudoedit
- /usr/sbin/postdrop
- /usr/sbin/postqueue
- /usr/sbin/suexec
- /usr/sbin/ccreds_validate
- /usr/lib/dbus-1.0/dbus-daemon-launch-helper
- /usr/lib/policykit-1/polkit-agent-helper-1

Motivation: Wenn für eine ausführbare Datei das SUID- oder SGID-Bit gesetzt ist, wird das Programm mit den Rechten des zugehörigen Kontos oder der Gruppe der Datei ausgeführt und nicht mit den Rechten des Kontos, welches das Programm ausführt, oder seiner Standardgruppe. Ausführbare Dateien mit SUID und SGID stellen ein Risiko dar, da die darin enthaltenen Schwachstellen möglicherweise zu einer Kompromittierung des Systems führen können.

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

ID: 3.65-14/9.1

Req 15 Core Dumps müssen deaktiviert und Software zur Fehlerberichterstattung muss deinstalliert werden.

Ein Core Dump ist ein Speicherabbild eines abgestürzten oder terminierten ausführbaren Programms. Es kann schutzbedürftige Daten beinhalten, wie z. B. Passwörter oder kryptografische Schlüssel. Softwares zur Fehlerberichterstattung überschreiben teilweise die Konfiguration zu Core Dumps (z. B. "Apport", "abrt") und sie können automatisch Fehlerberichte zum Hersteller senden, wodurch ggf. schutzbedürftige Daten ungewollt preisgegeben werden.

Core Dumps müssen deaktiviert werden für den normalen Betrieb und dürfen nur zur Fehleranalyse aktiviert werden. Ggf. vorhandene Software zur Fehlerberichterstattung muss deinstalliert werden, ist die Deinstallation nicht möglich muss sie deaktiviert werden.

Auf Systemen, auf denen Core Dumps notwendig sind, müssen Core Dumps für Prozesse mit SETUID (`fs.suid_dumpable=0`) deaktiviert werden.

Motivation: Core Dumps und Fehlerberichte können schutzbedürftige Daten enthalten. Um die versehentliche Preisgabe von schützenden Daten zu verhindern, müssen Core Dumps und Fehlerberichterstattung deaktiviert werden.

Umsetzungsbeispiel: 1) Software zur Fehlerberichterstattung entfernen (sie könnte `kernel.core_pattern` sonst ggf. überschreiben) :

Debian :

```
$ sudo apt remove reportbug
```

Ubuntu :

```
$ sudo apt remove apport
```

Redhat :

```
$ yum remove abrt
```

2) Finden & entfernen ggf. vorhandener "kernel.core_pattern" Einstellung und anschließendes konfigurieren:

```
sudo grep -r 'kernel.core_pattern' /etc/sysctl.*
```

`/etc/sysctl.d/50-coredump.conf` enthält:

```
kernel.core_pattern=|/bin/false
```

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten

ID: 3.65-15/9.1

Req 16	Schutz gegen Buffer Overflow Angriffe muss aktiviert sein.
--------	--

Eine Funktion zum Schutz gegen Buffer Overflow Angriffe muss auf Linux Servern aktiviert sein. Folgendes Kernel-option muss konfiguriert sein:

- `kernel.randomize_va_space=2` (siehe kernel.org, Standardeinstellung bei den meisten Linux Distributionen)

Voraussetzung dafür sind die Prozessormerkmale

- NoExecute (NX) bzw.
- eXecute Disable (XD)

welche im System-BIOS aktiviert sein müssen. Sofern Prelink vorhanden ist muss es bei gleichzeitiger Nutzung des ASLR Features PIE (Position-independent Executable) deaktiviert werden. Dabei handelt es sich um ein Tool, mit dem sich ELF shared libraries und ELF dynamically linked libraries modifizieren lassen. In modernen Distributionen wie RHEL und SLES ist Prelink obsolet und nicht mehr inkludiert.

Selbst-kompilierte Software muss mit Unterstützung für diese Sicherheitsmerkmale kompiliert werden. Die Standardeinstellungen von GCC erfüllen dies.

Motivation: Buffer Overflow Angriffe können genutzt werden, um unberechtigt Code auf einem System auszuführen, um die Verfügbarkeit zu beeinträchtigen oder um kompletten Zugriff zu einem System zu erhalten.

Umsetzungsbeispiel: `/etc/sysctl.conf` enthält:

```
kernel.randomize_va_space=2
```

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Störung der Verfügbarkeit

ID: 3.65-16/9.1

2.1. Netzwerkhärtung

Der IPv4-Protokollstack des Linux Server muss gehärtet werden. Hierfür muss die folgende Konfiguration umgesetzt werden:

- IPv4 Forwarding muss deaktiviert werden.
- IPv4 Redirects dürfen nicht akzeptiert werden.
- Secure IPv4 Redirects dürfen nicht akzeptiert werden.
- Senden von IPv4 Packet Redirect muss deaktiviert werden.
- IPv4 Source Routed Pakete dürfen nicht akzeptiert werden.
- Verdächtige Pakete müssen geloggt werden.
- Broadcast ICMP Requests müssen ignoriert werden.
- Bogus ICMP Responses müssen ignoriert werden.
- Reverse Path Filtering muss aktiviert werden.
- TCP SYN Cookies müssen aktiviert werden.
- TCP Timestamps müssen einen zufälligen Zeitversatz nutzen (seit Kernel 4.12) oder deaktiviert werden (vor Kernel 4.12).
- Ein ICMP Ratelimit muss konfiguriert werden.
- ARP muss eingeschränkt werden.

Motivation: Ein nicht gehärteter IPv4-Protokollstack ist ungeschützt gegen diverse Angriffe wie beispielsweise Denial-of-Service, Traffic High-Jacking.

Umsetzungsbeispiel: Beispiel für Debian-basierte Distributionen mit Kernel >= 4.12:

```
vim /etc/sysctl.d/80-ipv4-hardening.conf

net.ipv4.ip_forward=0
net.ipv4.conf.all.accept_redirects=0
net.ipv4.conf.default.accept_redirects=0
net.ipv4.conf.all.secure_redirects=0
net.ipv4.conf.default.secure_redirects=0
net.ipv4.conf.all.shared_media=0
net.ipv4.conf.default.shared_media=0
net.ipv4.conf.all.send_redirects=0
net.ipv4.conf.default.send_redirects=0
net.ipv4.conf.all.accept_source_route=0
net.ipv4.conf.default.accept_source_route=0
net.ipv4.conf.all.log_martians=1
net.ipv4.conf.default.log_martians=1
net.ipv4.icmp_echo_ignore_broadcasts=1
net.ipv4.icmp_ignore_bogus_error_responses=1
net.ipv4.conf.all.rp_filter=1
net.ipv4.conf.default.rp_filter=1
net.ipv4.tcp_syncookies=1
net.ipv4.tcp_timestamps=1
net.ipv4.icmp_ratelimit=100
net.ipv4.icmp_ratemask=88089
net.ipv4.conf.all.arp_ignore=2
net.ipv4.conf.default.arp_ignore=2
net.ipv4.conf.all.arp_announce=2
net.ipv4.conf.default.arp_announce=2
net.ipv4.conf.all.arp_notify=0
net.ipv4.conf.default.arp_notify=0
net.ipv4.conf.all.arp_accept=0
net.ipv4.conf.default.arp_accept=0
```

Für diese Anforderung sind folgende Bedrohungen relevant:

- Störung der Verfügbarkeit

ID: 3.65-17/9.1

Req 18	Falls IPv6 verwendet wird, muss der IPv6-Protokollstack sicher konfiguriert werden, andernfalls muss IPv6 explizit deaktiviert werden.
--------	--

Falls IPv6 nicht verwendet wird, muss es vollständig deaktiviert werden. Andernfalls muss der IPv6-Stack auf Linux-Servern gehärtet werden. Dazu muss folgende Konfiguration umgesetzt werden:

- IPv6 Forwarding muss deaktiviert werden.
- IPv6 Redirects dürfen nicht akzeptiert werden.
- IPv6 Source Routed Pakete dürfen nicht akzeptiert werden.
- IPv6 Router Advertisements dürfen nicht akzeptiert werden.
- IPv6 Router Solicitations Nachrichten dürfen nicht versendet werden.
- IPv6 Autoconfiguration muss deaktiviert werden.

Motivation: Ein nicht gehärteter IPv6-Protokollstack ist ungeschützt gegen diverse Angriffe wie beispielsweise Denial-of-Service, Traffic Hijacking.

Umsetzungsbeispiel: Beispielkonfiguration zur Härtung von IPv6 für eine Debian-basierte Linux-Distribution:

`/etc/sysctl.d/80-ipv6-hardening.conf`

```
net.ipv6.conf.all.forwarding=0
net.ipv6.conf.all.accept_redirects=0
net.ipv6.conf.all.accept_source_route=0
net.ipv6.conf.all.accept_ra=0
net.ipv6.conf.all.router_solicitations=0
net.ipv6.conf.all.autoconf=0
net.ipv6.conf.default.forwarding=0
net.ipv6.conf.default.accept_redirects=0
net.ipv6.conf.default.accept_source_route=0
net.ipv6.conf.default.accept_ra=0
net.ipv6.conf.default.router_solicitations=0
net.ipv6.conf.default.autoconf=0
```

Beispielkonfiguration zur Deaktivierung von IPv6 für eine Debian-basierte Linux-Distribution:

`/etc/sysctl.d/80-ipv6-disable.conf`

```
net.ipv6.conf.all.disable_ipv6=1
net.ipv6.conf.default.disable_ipv6=1
```

Für diese Anforderung sind folgende Bedrohungen relevant:

- Störung der Verfügbarkeit

ID: 3.65-18/9.1

2.2. Systemaktualisierung

Req 19	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 Bereitstellungs Kanäle des Herstellers
- dritte Distributoren, sofern sie durch den Hersteller autorisiert und legitimer Bestandteil von dessen Bereitstellungs Kanälen 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 Sicher-

heits-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.

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

ID: 3.01-2/9.0

Req 20	Es müssen Versionen von Linux Betriebssystemen eingesetzt werden, welche aktiv gepflegt werden und deren Distributionen Support erhalten. Zwingend erforderlich ist Support für die Behebung von Sicherheitsschwachstellen ("security updates").
--------	--

Linux Betriebssysteme und Software aus Repositorien der Distribution mit aktiver Pflege durch Anbieter*Innen bzw. Lieferant dürfen eingesetzt werden. Distributionen, welche im Lebenszyklus End-of-Support (EOS), End-of-Life (EOL) oder dergleichen angekommen sind, dürfen nicht eingesetzt werden. Versionen mit Langzeitunterstützung ("long term support", LTS) sind zu bevorzugen.

Ein solcher Support muss umfassen, dass

- das verwendete Software kontinuierlich auf eine Betroffenheit von Sicherheitsschwachstellen beobachtet und analysiert,
- über Art, Schweregrad und Ausnutzbarkeit von entdeckten Schwachstellen umgehend informiert

- zeitnah Aktualisierungen oder wirksame Workarounds zur Behebung der Schwachstellen bereitstellt.

Der Sicherheitsschwachstellen-Support muss für die gesamte Verwendungsdauer gegeben sein.

Support-Phasen mit eingeschränktem Leistungsumfang

Einige Distributionen bieten verlängerte Supportphasen mit eingeschränktem Leistungsumfang an. Erlaubt sind nur jene mit Sicherheitsschwachstellen-Support für sämtliche verwendete Software. Werden Sicherheitsaktualisierungen nur für den Linux-Kernel geboten ist die Verwendung der Distribution in jener Supportphase nicht gestattet.

Open Source Software

Open Source Produkte werden häufig von freien Organisationen oder Communities entwickelt; entsprechend ist ein vertraglich vereinbarter Sicherheitsschwachstellen-Support unter Umständen nicht verfügbar. Grundsätzlich muss auch hier sichergestellt sein, dass von der Organisation/Community (oder einem offiziell durch diese eingesetzten Dritten) für das betroffene Produkt ein umfassendes Sicherheitsschwachstellen-Management betrieben wird, welches die oben genannten Kriterien erfüllt und als zuverlässig etabliert gilt.

Motivation: Ein sicherer Betrieb kann nur mit Software gestaltet werden, welche Sicherheitsaktualisierungen erhält. Bekannte Schwachstellen und Proof-of-Concepts machen Software ohne verfügbare Sicherheitsaktualisierungen zu einfachen Zielen.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Störung der Verfügbarkeit
- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe

ID: 3.65-20/9.1

Req 21	Falls erforderlich, müssen benötigte, aktivierte, Lizenzen zur Sicherstellung von Sicherheits-Updates für das System hinterlegt sein.
--------	---

Die Anforderung, benötigte, aktivierte Softwarelizenzen zu hinterlegen, stellt sicher, dass die gesamte Software, die innerhalb der Organisation verwendet wird, berechtigt ist, die neuesten Sicherheitsupdates und Patches vom Anbieter zu erhalten. Dies ist entscheidend, um Schwachstellen zu minimieren und Systeme vor potenziellen Cyberbedrohungen zu schützen. Durch die Beibehaltung gültiger Lizenzen kann das Unternehmen die kontinuierliche Einhaltung von Best Practices für die Sicherheit sicherstellen und das Risiko von Sicherheitsverletzungen verringern. Folglich unterstützt diese Vorgehensweise die allgemeine Integrität, Vertraulichkeit und Verfügbarkeit der IT-Infrastruktur des Unternehmens.

Motivation: Einige Hersteller von Betriebssystemen lizenzieren ihre Produkte und fordern den Erwerb von Lizenzen. Diese ermöglichen den Zugriff auf (Sicherheits-) Updates. Ohne Update-Möglichkeiten darf ein System nicht betrieben werden.

Umsetzungsbeispiel: Red Hat Enterprise Linux beispielsweise kann registriert werden mittels: `subscription-manager register`

ID: 3.65-21/9.1

Req 22	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

ID: 3.01-10/9.0

Req 23	GPG Check für das Repository Server muss aktiviert und die zugehörigen Schlüssel konfiguriert werden.
--------	---

Der GPG Check muss aktiviert und Schlüssel müssen konfiguriert werden, um die Integrität der Software während der Installation von einem Repository-Server verifizieren zu können. Bei RedHat Linux ist es außerdem notwendig, den GPG Check global zu aktivieren.

Motivation: Der GPG Check ist notwendig, um zu garantieren, dass die genutzte Quelle valide ist und um die Integrität der Software zu überprüfen. Ohne diesen Check kann ein Angreifer eventuell die Software während der Installation manipulieren.

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

Für diese Anforderung sind folgende Gewährleistungsziele relevant:

- Integrität

Req 24 Das Betriebssystem muss über eine Endpoint Detection and Response (EDR) Lösung verfügen.

Das Betriebssystem muss eine aktuelle Endpoint Detection and Response (EDR) Lösung einsetzen. Eine EDR Lösung sammelt sicherheitsrelevante Aktivitätsdaten von Prozessen und wertet diese zentral aus. Alarme können aus maliziosen Verhalten von Prozessen erzeugt werden oder auch durch eine bestimmte Signatur (wie klassische Virens Scanner). Zusätzlich ist es über eine zentrale Konsole möglich, direkt auf verdächtiges Programmverhalten zu reagieren. Das Betriebssystem bzw. der Client kann beispielsweise isoliert oder der maliziose Prozess beendet werden. Darüber hinaus wird eine Übersicht der Schwachstellen aller eingesetzten Betriebssysteme an eine zentrale Stelle geleitet.

Motivation: Normale Virens Scanner verlassen sich rein auf Signaturen, während eine EDR Lösung auch nach Anomalien im Prozessverhalten sucht. Signaturen haben dabei den Nachteil, dass für neue Virenvarianten erst neue Signaturen geschrieben werden müssen, bevor sie erkannt werden.

Umsetzungsbeispiel: Der Artikel [PSA Wiki - Sicherheitsanforderungen : Endpoint Detection and Response \(EDR\)](#) dient als Orientierungshilfe.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unbemerkt durchführbare Angriffe

ID: 3.65-24/9.1

2.3. Authentifizierung von Benutzern

Req 25 Benutzerkonten müssen die eindeutige Identifizierung des Benutzers sicherstellen.

Benutzer müssen eindeutig von einem System identifiziert werden können.

Dies wird typischerweise dadurch erreicht, dass für jeden Benutzer ein individuelles Benutzerkonto genutzt wird.

Sogenannte Gruppenkonten, die sich dadurch auszeichnen, dass sie von mehreren Personen gemeinsam genutzt werden, dürfen nicht verwendet werden. Dies gilt uneingeschränkt auch für privilegierte Benutzerkonten. Die meisten Systeme verfügen nach der Grundinstallation zunächst nur über ein einzelnes Benutzerkonto mit administrativen Berechtigungen. Soll das System von mehreren Personen administriert werden, muss jede dieser Personen ein persönliches, individuelles Benutzerkonto verwenden, dem entsprechende administrative Berechtigungen bzw. Rollen zugewiesen sind.

Eine Besonderheit stellen sogenannte technische Benutzerkonten dar. Diese werden für Authentifizierung und Autorisierung von Systemen untereinander oder von Anwendungen auf einem System genutzt und können damit nicht einer konkreten Person zugewiesen werden. Solche Benutzerkonten müssen individuell pro System oder pro Anwendung vergeben werden. Hierbei ist allerdings sicherzustellen, dass eine missbräuchliche Nutzung solcher Benutzerkonten nicht möglich ist.

Möglichkeiten, um eine missbräuchliche Nutzung solcher Benutzerkonten durch Personen zu verhindern, sind:

- Konfiguration eines den Sicherheitsanforderungen entsprechenden Passwortes, dass nur einem kleinstmöglichen Kreis von Administratoren bekannt ist.
- Konfiguration des entsprechenden Benutzerkontos, so dass dieses nur lokal verwendet werden kann und eine interaktive Anmeldung nicht möglich ist.
- Verwendung eines Verfahrens mit öffentlichem und privatem Schlüssel oder Zertifikaten für die Authentifizierung des entsprechenden Benutzerkontos.
- Einschränkung des Zugriffs über ein Netz auf legitimierte Systeme.

Weitere Lösungen sind im Einzelfall auf ihre Verwendbarkeit zu prüfen.

Motivation: Die eindeutige Identifizierung eines Benutzers ist Voraussetzung, um einem Benutzer Berechtigungen zuweisen zu können, die für die Durchführung seiner Aufgaben auf dem System notwendig sind. Nur so kann eine ad-

äquate Zugriffskontrolle auf Daten und Dienste eines Systems erreicht und der missbräuchliche Zugriff verhindert werden. Des Weiteren wird hierdurch ermöglicht, dass Tätigkeiten und Aktionen auf einem System protokolliert und einem Benutzer zugeordnet werden können.

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

ID: 3.01-22/9.0

Req 26	Systemkonten müssen non-login sein.
--------	-------------------------------------

Auf Linux Servern sind verschiedene Konten verfügbar, die für die Funktion von Applikationen notwendig sind. Diese Konten sind nicht für die Nutzung einer Shell gedacht, daher muss die Shell für diese Benutzerkonten deaktiviert werden.

Hinweis: Die Systemkonten root, sync, shutdown und halt sind von dieser Anforderung ausgenommen

Motivation: Systemkonten können somit nicht missbraucht werden um auf einer Shell Kommandos auszuführen.

Umsetzungsbeispiel: Um zu verhindern, dass solche Systemkonten zum Einloggen auf dem System genutzt werden, kann der Shell-Parameter in der Datei `/etc/passwd` auf `/usr/sbin/nologin` oder `/bin/false` gesetzt werden.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System

ID: 3.65-26/9.1

Req 27	Benutzerkonten müssen mit mindestens zwei Authentisierungsmerkmalen aus unterschiedlichen Faktoren geschützt werden.
--------	--

Sämtliche Benutzerkonten des Systems müssen vor einer unberechtigten Nutzung geschützt werden. Dies ist durch Verwendung von Authentisierungsmerkmalen umzusetzen, die eine eindeutige Identifizierung des legitimen Benutzers ermöglichen.

Authentisierungsmerkmale werden in übergeordneten Gruppen, sogenannten "Faktoren", differenziert:

- Faktor WISSEN: "etwas, das nur der legitime Benutzer weiss"
- Faktor BESITZ: "etwas, das nur der legitime Benutzer besitzt"
- Faktor INHÄRENZ: "etwas, das nur der legitime Benutzer ist"

Es müssen mindestens zwei Authentisierungsmerkmale miteinander kombiniert werden.

Diese müssen aus unterschiedlichen Faktoren stammen ("Multi-Faktor-Authentisierung"; kurz: "MFA").

Eine Kombination von Authentisierungsmerkmalen desselben Faktors (Beispiel: zwei unterschiedliche PINs) ist nicht zulässig.

Beispiele für zulässige Authentisierungsmerkmale

- aus dem Faktor WISSEN: PINs, Passphrases (Passwörter*)
- aus dem Faktor BESITZ: Smartcards, FIDO2 Level 2 Security Keys, Device-bound Passkeys, in einem TPM/HSM gespeicherte kryptografische Schlüssel, OTPs (One-Time Passwords) generiert in einer zum Einsatz in der Deutschen Telekom freigegebenen Authenticator App
- aus dem Faktor INHÄRENZ: biometrische Merkmale wie Fingerabdruck, Iris- bzw. Retina-Scans, Gesichtser-

***) Wichtiger Hinweis zum Einsatz von Passwörtern**

Passwörter müssen als Authentisierungsmerkmal strikt vermieden werden.

Sie sind nur dann zulässig, falls sicherere Alternativen nicht umgesetzt werden können.

Grundsätzlich nicht zulässig sind Authentisierungsmerkmale, die leicht gefälscht oder manipuliert werden können. Dazu gehören:

- Rufnummern und Anrufer-IDs
- IP-Adressen
- geografische Standorte
- VPN-Zugehörigkeiten und Netzwerkinformationen
- Status von Clients, etwa von Arbeitsplatzsystemen oder mobilen Endgeräten

Hinweis zu technischen Benutzerkonten

Für den maschinellen Zugriff eingesetzte technische Benutzerkonten (auch unter dem Begriff "Machine-to-Machine" bzw. "M2M"-Konten bekannt) sind von der Verpflichtung zur Multi-Faktor-Authentisierung ausgenommen, da eine solche dort i.d.R. nicht umsetzbar ist.

Motivation: Benutzerkonten, die nicht über angemessene Authentisierungsmerkmale geschützt sind, können von einem Angreifer missbraucht werden, um unberechtigt auf ein System und die darauf gespeicherten Daten und Anwendungen zuzugreifen. Aufgrund vielfältiger gängiger Angriffsmethoden ist der Schutz von Benutzerkonten durch ein einzelnes Authentisierungsmerkmal mittlerweile als schwach zu bewerten, insbesondere, falls hier das klassische Passwort zum Einsatz kommt. Eine Kombination von Authentisierungsmerkmalen aus unterschiedlichen Faktoren erhöht die Sicherheit für Benutzerkonten signifikant.

Umsetzungsbeispiel: Beispiele für MFA

MFA Kombination	Beschreibung	Anmerkung zur Anwendung
Telekom "myCard" plus PIN-Eingabe	1. Faktor: in Hardware implementiertes Zertifikat ("Besitz") 2. Faktor: PIN ("Wissen")	(keine)
FIDO2 Level 2 Security Key plus Fingerabdruck	1. Faktor: in Hardware gespeicherter Crypto-Schlüssel ("Besitz") 2. Faktor: Biometrisches Merkmal ("Inhärenz")	Die Biometrie-Funktion muss vorab konfiguriert werden, da der Biometrie-Sensor sonst nur als "Taste" funktioniert und somit der 2. Faktor ("Inhärenz") nicht gegeben wäre.
Device-bound Passkey in WH4B* auf Nucleus PC plus PIN	1. Faktor: in Hardware gespeicherter Crypto-Schlüssel ("Besitz") 2. Faktor: PIN ("Wissen")	(keine)
Device-bound Passkey in Secure Enclave auf iOS Device plus FaceID	1. Faktor: in Hardware gespeicherter Crypto-Schlüssel ("Besitz") 2. Faktor: Biometrisches Merkmal ("Inhärenz")	(keine)

Passwort plus freigegebene OTP App (z. B. MS Authenticator)	1. Faktor: Passwort ("Wissen") 2. Faktor: spezifisch generiertes One Time Password ("Besitz")	Passwörter sind nur dann zulässig, falls sicherere Alternativen nicht umgesetzt werden können. Die Synchronisation der Authenticator App über einen Clouddienst sowie das Teilen (z. B. über einen Passwortmanager) des OTP ist nicht zulässig.
--	--	--

*) WH4B: "Windows Hello for Business"

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

ID: 3.01-20/9.0

Req 28	Privilegierte Benutzerkonten müssen mit mindestens zwei Authentisierungsmerkmalen geschützt werden, die aus unterschiedlichen Faktoren stammen und erhöhte Resilienz gegen Phishing-Angriffe bieten.
--------	--

Ein privilegiertes Benutzerkonto ist ein Benutzerkonto, welches über erweiterte Berechtigungen innerhalb eines Systems verfügt. Erweiterte Berechtigungen ermöglichen einen Zugriff auf Konfigurationseinstellungen, Funktionen oder Datenbestände, der regulären Benutzern des Systems nicht zu Verfügung steht. In direkter Abhängigkeit zu den besonderen Aufgaben, die mittels eines privilegierten Benutzerkontos innerhalb eines Systems wahrgenommen werden, können die zugewiesenen erweiterten Berechtigungen spezifisch limitiert sein oder einen vollständig uneingeschränkten Systemzugriff umfassen.

Beispiele für privilegierte Benutzerkonten:

- Konten für Administrations-, Wartungs- oder Entstörungsaufgaben
- Konten für Benutzer-Verwaltungsaufgaben (z. B. Anlegen/Löschen von Benutzern; Zuweisen von Berechtigungen; Passwort-Rücksetzungen)
- Konten, die zur Freigabe, Auslösung oder Unterbindung von geschäftskritischen Vorgängen berechtigt sind
- Konten mit Zugriff auf Daten, die im Interesse des DTAG-Konzerns, dessen Kunden oder der Öffentlichkeit als SCD (Sensitive Customer Data) eingestuft sind
- Konten, die in hohem Umfang auf lt. EU-DSGVO als „personenbezogen“ definierte Daten zugreifen können (z. B. Massenabruf von größeren Teilen oder des vollständigen Datenbestands)

Zum Schutz privilegierter Konten müssen mindestens zwei Authentisierungsmerkmale miteinander kombiniert werden.

Diese müssen aus unterschiedlichen Faktoren ("WISSEN", "BESITZ" oder "INHÄRENZ") stammen.

Eine Kombination von Authentisierungsmerkmalen desselben Faktors ist nicht zulässig.

Mindestens eines der Authentisierungsmerkmale darf nicht anfällig gegenüber Phishing-Angriffen sein.

Bei einem Phishing-Angriff wird der legitime Nutzer trickreich manipuliert, um seine Anmeldedaten dem Angreifer offenzulegen, beispielsweise mittels gefälschter Anmelde-Seiten. Ein Phishing-Angriff kann auch gegen eine Kombination aus mehreren Authentisierungsmerkmalen erfolgen.

Phishing-resiliente Authentisierungsmerkmale verhindern Angriffe anhand ihrer technischen Umsetzung. Diese Authentisierungsmerkmale nutzen kryptographische Verfahren und stellen eine starke Bindung zwischen dem Nutzer, dem Gerät und dem Dienst her. Sie erfordern während der Authentisierung zusätzlich eine manuelle Nutzerinteraktion.

on, die nicht automatisiert werden kann; beispielsweise, einen Fingerabdruck-Scan an einem Hardware Token abzugeben.

Als nicht anfällig gegenüber Phishing-Angriffen gelten

- FIDO2 Level 2 Security Keys
- in Hardware implementierte Zertifikate

Wichtiger Hinweis zum Einsatz von Passwörtern

Passwörter müssen als Authentisierungsmerkmal strikt vermieden werden.

Sie sind nur dann zulässig, falls sicherere Alternativen nicht umgesetzt werden können.

Hinweis zu technischen Benutzerkonten

Für den maschinellen Zugriff eingesetzte technische Benutzerkonten (auch unter dem Begriff "Machine-to-Machine" bzw. "M2M"-Konten bekannt) sind von der Verpflichtung zur Multi-Faktor-Authentisierung ausgenommen, da eine solche dort i.d.R. nicht umsetzbar ist.

Motivation: Privilegierte Benutzerkonten stellen ein erhöhtes Risiko für die Sicherheit eines Systems dar. Kompromittiert ein Angreifer erfolgreich ein solches Benutzerkonto, erhält er weitreichende Berechtigungen, mit denen er das System oder Systemanteile unter seine Kontrolle bringen, die Systemfunktionen stören, verarbeitete Daten einsehen/manipulieren oder geschäftskritische Vorgänge beeinflussen kann. Die Kombination von mehreren Authentisierungsmerkmalen unterschiedlicher Art minimiert das Risiko der Kompromittierung eines Benutzerkontos bereits signifikant, jedoch muss für privilegierte Benutzerkonten in besonderem Maße auf Resilienz gegen Phishing-Angriffe geachtet werden. Die zu beobachtenden Phishing-Angriffe werden zunehmend trickreicher und für Nutzer schwerer zu erkennen, so dass eine Multifaktor-Authentifizierung ohne erhöhte Resilienz gegen Phishing nicht mehr als ausreichend für privilegierte Benutzerkonten zu bewerten ist.

Umsetzungsbeispiel: Beispiele für MFA mit erhöhter Resilienz gegen Phishing-Angriffe

MFA Kombination	Beschreibung	Anmerkung zur Anwendung
Telekom "myCard" plus PIN-Eingabe	1. Faktor: in Hardware implementiertes Zertifikat ("Besitz") 2. Faktor: PIN ("Wissen")	(keine)
FIDO2 Level 2 Security Key plus Fingerabdruck	1. Faktor: in Hardware gespeicherter Crypto-Schlüssel ("Besitz") 2. Faktor: Biometrisches Merkmal ("Inhärenz")	Die Biometrie-Funktion muss vorab konfiguriert werden, da der Biometrie-Sensor sonst nur als "Taste" funktioniert und somit der 2. Faktor ("Inhärenz") nicht gegeben wäre.
FIDO2 Level 2 Security Key plus PIN-Eingabe	1. Faktor: in Hardware gespeicherter Crypto-Schlüssel ("Besitz") 2. Faktor: PIN ("Wissen")	(keine)

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

ID: 3.01-21/9.0

Req 29	Für den "Abgesicherten Modus" (Single-User Mode) und Bootloader (z.B. GRUB 2) muss eine Authentifizierung aktiviert sein. Zudem muss die Anzahl lokaler zum Betrieb notwendiger Konten minimiert werden.
--------	--

Für die Wiederherstellung des Systems existiert der "Abgesicherte Modus" oder "Single-User Modus". Dieser Modus kann während des Bootvorgangs manuell ausgewählt werden. Daher muss eine Authentifizierung für diesen Modus aktiviert sein. Dies ist vor allem dann relevant, wenn die Console der Maschine remote erreicht werden kann (z. B. über Lights-Out-Management oder über eine virtuelle Console bei virtuellen Maschinen).

Des Weiteren muss sichergestellt werden, dass alle nicht entfernbaren, ungenutzten lokalen Konten deaktiviert sind. Konten, bei denen eine Anmeldung in bestimmten Fällen mit einem Passwort (root, grubadmin) möglich sein muss, müssen Passwörter für jeden Rechner und jedes Konto individuell und ohne erkennbare Bildungsregel haben.

Motivation: Lokale Accounts sind zusätzliche Angriffspunkte, die durch Angreifer beziehungsweise unbefugte Personen missbraucht werden könnten. Dies erfordert, dass nur für den Betrieb unbedingt notwendige lokale Konten im Betriebssystem vorhanden sind. Ohne Authentifizierung kann ein nicht autorisierter Benutzer Root-Rechte erlangen.

Umsetzungsbeispiel: Beispiel für eine Debian-basierte Distribution:

- 1) Passwort für "Single-User Modus" setzen, was letztlich das Passwort für "root" ist: `$ sudo passwd root`
- 2) GRUB 2 Bootloader konfigurieren.
 - 2.1) Ein mit ausreichend vielen Runden gehashtes Passwort erzeugen: `$ grub-mkpasswd-pbkdf2 -c 640000`
 - 2.2) Benutzer und Hashwert in GRUB 2 Konfiguration schreiben:
`/etc/grub.d/40_custom` enthält:

```
set superusers="grubadmin"
password_pbkdf2 grubadmin grub.pbkdf2.sha512.640000.6B4ABEE1920FDE...
```
 - 2.3) Boot ohne Passwordeingabe konfigurieren:
`/etc/grub.d/10_linux` enthält Variable "CLASS" mit Parameter "--unrestricted":

```
CLASS="--class gnu-linux --class gnu --class os --unrestricted"
```
 - 2.4) GRUB 2 Konfiguration aktualisieren: `$ sudo update-grub`

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System

ID: 3.65-29/9.1

2.4. Administrativer Zugang

Req 30	Der Zugang zur Administration des Betriebssystems muss über eine vom Produktionsnetz unabhängige Schnittstelle erfolgen.
--------	--

Der administrative Zugang auf einen Server oder vergleichbares System darf nicht über die Netzschnittstelle erfolgen, auf der Dienste für Benutzer oder andere Systeme erbracht werden (Produktionsnetz). Weiterhin muss der Zugang auf legitime Systeme eingeschränkt werden. Über diesen Weg kann auch die Anwendungsadministration erfolgen.

Eine Einschränkung des Zugangs kann z. B. durch einen Filtermechanismus, lokale Access-Listen oder eine lokale Firewall realisiert werden. Die Beschränkung ist hierbei möglichst restriktiv auf einzelne IPs oder zumindest kleine IP-Adressbereiche zu setzen.

Motivation: Ein Angreifer kann im Falle eines erfolgreichen Angriffs auf vertrauliche Informationen oder sogar auf das komplette System zugreifen. Durch eine Einschränkung der netzseitigen Erreichbarkeit auf legitimierte Systeme kann der Kreis potentieller Angreifer verkleinert werden. Zudem müssen Systeme immer verwaltet werden können, auch wenn kundenseitige Netze gestört sind.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unbemerkt durchführbare Angriffe
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

Req 31	Administrative Dienste und Zugänge dürfen nur an die Schnittstellen gebunden werden, die für die Administration vorgesehen sind.
--------	--

Die Dienste zur Administration (interaktiv durch Personen oder Maschinen-Kommunikation, z. B. per SSH, HTTPS, RDP), müssen an die dafür vorgesehene(n) Schnittstelle(n) gebunden werden. Auf Grund der Trennung des Managementverkehrs vom Nutzdatenverkehr, ist dies üblicherweise die IP-Adresse der Management-Schnittstelle. Wird das System - oder Teile davon - über mehrere Schnittstellen administriert, so sind diese Dienste an die entsprechende geringstmögliche Anzahl Schnittstellen zu binden.

Je nach Erfordernissen des jeweiligen Dienstes ist der Zugriff darauf auf wenige, vertrauenswürdige, notwendige Ziel- bzw. Quell-Adressen einzuschränken.

Motivation: Mit Umsetzung dieser Maßnahme ist klar definiert, unter welcher Adresse diese Managementdienste erreichbar sind. Eine eindeutige Adresse ist wichtig für die Implementierung von Filtern und Firewall-Regeln, die den Zugriff auf diese Managementdienste auf legitime Adressen beschränken.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unbemerkt durchführbare Angriffe
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.37-11/9.0

Req 32	Zugänge über das Netz zur Administration des Betriebssystems müssen integritätsgeschützt, verschlüsselt und sicher authentifiziert erfolgen.
--------	--

Der Zugang darf nur mittels sicherer Protokolle (z. B. SSHv2, HTTPS, SNMPv3) erfolgen. Der Administrator muss sicherstellen, dass jede Netzverbindung zwischen dem Arbeitsplatzsystem oder einem Managementsystem und dem zu verwaltenden Betriebssystem sicher authentifiziert, verschlüsselt und manipulationsgeschützt ist.

Motivation: Überträgt der Administrator Änderungen an Konfigurationseinstellungen über unverschlüsselte oder unsichere Verbindungen, besteht die Gefahr, dass Unberechtigte durch Abhören Erkenntnisse gewinnen (Konfigurationseinstellungen, Zugangskennungen, etc.), die das Ausnutzen von Schwachstellen begünstigen.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Abstreiten von durchgeführten Aktionen

ID: 3.37-12/9.0

2.5. Lokation

Req 33	Falls sich das System nicht in einem abgesicherten Raum mindestens der Schutzklasse "hoch" (SK3) befindet, müssen das BIOS und, sofern vorhanden, andere lokale Management-Zugänge vor nicht autorisierten Zugriffen geschützt werden.
--------	--

Die Schutzklassen (SK) sind in der Sicherheitsanforderung "Physische Sicherheit" (Anlage zur Konzernrichtlinie "Sicherheit", siehe <https://richtlinien.telekom.de/>) definiert. Rechenzentren erfüllen in der Regel die Anforderungen nach SK3.

Server, die öffentlich oder in Räumlichkeiten von Kunden installiert sind, müssen besonders vor unautorisiertem Zugriff und Veränderungen geschützt werden: Es müssen die Einstellungen des BIOS' gegen Auslesen und Manipulation geschützt werden. Sind weitere Zugriffsmöglichkeiten auf die Systemkonfiguration möglich, z. B. durch Intel AMT, iLO, LOM u. ä., müssen diese ebenso geschützt werden. Bei Verwendung eines Passworts muss dieses exklusiv für

den einzelnen Server sein und darf keine Rückschlüsse auf ein Unterscheidungsmerkmal des Servers ermöglichen.

Das BIOS muss so konfiguriert sein, dass sich darüber ausschließlich das vorgesehene Betriebssystem von der dafür vorgesehenen Partition starten lässt.

Motivation: Das Ändern von BIOS-Einstellungen kann Angriffe begünstigen. Da beispielsweise dezentrale Technikräume selten einen Zugangsschutz zu Servern bieten, könnten Angreifer beim Starten des Servers im BIOS – ohne den beschriebenen Passwortschutz – die Startreihenfolge der Datenträger verändern. So ließe sich ein alternatives Betriebssystem starten, das die Sicherheitsmechanismen des implementierten Betriebssystems umgeht.

Für diese Anforderung sind folgende Bedrohungen relevant:

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

ID: 3.37-20/9.0

Req 34	Falls sich das System nicht in einem abgesicherten Raum mindestens der Schutzklasse "hoch" (SK3) befindet, müssen verwendete Datenträger vollständig verschlüsselt werden.
--------	--

Die Schutzklassen (SK) sind in der Sicherheitsanforderung "Physische Sicherheit" (Anlage zur Konzernrichtlinie "Sicherheit", siehe <https://richtlinien.telekom.de/>) definiert. Rechenzentren erfüllen in der Regel die Anforderungen nach SK3.

Als Datenträger gelten insbesondere alle in den Systemen befindliche Festplatten und Flash-Speicher.

Motivation: Der Zutritt zu Geräten, die außerhalb von Rechenzentren mit geschützten Zugängen betrieben werden, ist verhältnismäßig einfach möglich. Damit lassen sich dort vorhandene Datenträger leicht entwenden.

Umsetzungsbeispiel: Bei Windows Server 2008 kann der Administrator die Laufwerksverschlüsselung mittels BitLocker und Schlüsselspeicherung auf einem TPM aktivieren.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten

ID: 3.37-21/9.0

3. Protokollierung

Req 35 Protokollierung (auditing) muss beim Start durch Festlegen eines Kernelparameters aktiviert werden.

Jeder Prozess auf dem System trägt ein "auditable" Flag, das angibt, ob seine Aktivitäten protokolliert werden können. Obwohl "auditd" dafür sorgt, dass dies für alle Prozesse aktiviert wird, die nach "auditd" gestartet werden, wird durch Hinzufügen des Kernelparameters (audit=1) sichergestellt, dass es während des Startvorgangs für jeden Prozess festgelegt wird.

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.

Umsetzungsbeispiel: Für Debian-basierte Distributionen z. B. kann in der Datei /etc/default/grub zur Variable GRUB_CMDLINE_LINUX_DEFAULT der Kernelparameter mit "audit=1" hinzugefügt werden. Nach einer Änderung dieser Konfigurationsdatei sollte selbstverständlich ein update - grub erfolgen um die Änderungen zu übernehmen.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe

ID: 3.65-35/9.1

Req 36 Log Rotate für Log-Dateien muss konfiguriert werden.

Log Rotate für Log-Dateien muss aktiviert werden. Dateien müssen basierend auf der Dateigröße und der maximalen Lebenszeit der Datei rotiert werden.

Motivation: Angriffe können auf das Erzeugen von Log-Events abzielen, um so den verfügbaren Speicherplatz eines Servers überlaufen zu lassen. Dies kann zu einer Störung der Verfügbarkeit des Servers führen.

Umsetzungsbeispiel: logrotate Konfiguration enthält:

```
# restrict maximum size with maxsize and rotate
maxsize 10M
rotate 100

# control maximum age with daily rotation and maxage
daily
maxage 90

# permissions
create 640 syslog adm

# secure deletion
shred
```

Für diese Anforderung sind folgende Bedrohungen relevant:

- Störung der Verfügbarkeit

ID: 3.65-36/9.1

Req 37 Die Systemzeit muss zu einer exakten Referenzzeit (Zeitnormal) synchronisiert werden.

Als Zeitnormal ist eine Zeitgeber-Quelle heranzuziehen, die ein Zeitsignal auf Basis der koordinierten Weltzeit ("UTC" = "Universal Time Coordinated") liefert.

Anmerkung: die anhand der Zeitgeber-Quelle zu UTC synchronisierte Systemzeit darf auf dem lokalen System mittels entsprechender Zeitzonen-Konfiguration bei der Ausgabe von Zeitinformationen in die Ortszeit transformiert werden, sofern die so stattfindende Zeitzonenanpassung eindeutig erkennbar und nachvollziehbar ist.

Systeme, die zu einer gemeinsamen Sicherheits-Domäne gehören, müssen sich einheitlich an derselben Zeitgeber-Quelle synchronisieren.

Motivation: Zeitsynchronisation kann auf einem System eine technische Voraussetzung sein, um zeitkritische Abhängigkeiten in beispielsweise der Nutzung von Zertifikaten oder Authentisierungsmechanismen korrekt bedienen zu können. Dringend erforderlich sind außerdem exakte Zeitstempel in Protokollierungseinträgen, damit diese bei Sicherheitsvorfällen oder Problemanalysen in eindeutiger zeitlicher Relation auswertbar sind.

Umsetzungsbeispiel: Einige zulässige Zeitgeber-Quellen:

- **vertrauenswürdige** eingestufte NTP ("NetworkTimeProtocol") Server **erreichbar über ein vertrauenswürdige IP-Netzwerk**
- DCF77-Funksignal über physisch angebundenes Empfängermodul
- GPS-Funksignal über physisch angebundenes Empfängermodul

Wenn NTP verwendet wird, müssen entsprechende Clients wie (1) systemd-timesyncd, (2) ntpd oder (3) chrony für die Zeitsynchronisation verwendet werden.

AdminLAN NTP Server sind als Beispiel eine gültige Wahl.

Wenn NTP-Server nur über **nicht vertrauenswürdige** Netzwerke wie Kundenumgebungen oder das Internet verfügbar sind, muss NTP mit *Network Time Security* verwendet werden.

ID: 3.65-37/9.1

Req 38	Für sicherheitsrelevante Protokollierungsdaten, die lokal erfasst werden, müssen geltende Aufbewahrungszeiträume und Löschfristen eingehalten werden.
--------	---

Aus Sicht der IT-Sicherheit wird auf einem System die lokale Erfassung von sicherheitsrelevanten Protokollierungsdaten nicht zwingend gefordert. Da lokal erfasste Protokollierungsdaten bei Systemstörungen beschädigt oder von einem erfolgreichen Angreifer gezielt manipuliert werden können, sind sie für sicherheitstechnische bzw. forensische Analysen nur bedingt belastbar. Entsprechend ist für die IT-Sicherheit relevant, dass Protokollierungsdaten an einen separaten Log Server weitergeleitet werden.

Eine lokale Erfassung kann dennoch erfolgen; beispielsweise falls bei der Erzeugung der Protokollierungsdaten eine lokale Speicherung aufgrund technischer Abläufe zunächst unabdingbar ist oder es begründete betriebliche Interessen gibt, Protokollierungsdaten auch lokal vorzuhalten.

Folgende Grundregelung ist bei einer lokalen Erfassung von Protokollierungsdaten zu berücksichtigen:

- Sicherheitsrelevante Protokollierungsdaten müssen für einen Zeitraum von 90 Tagen aufbewahrt werden.
(Diese Anforderung gilt nur, falls auf dem System keine zusätzliche Weiterleitung an einen separaten Log Server umgesetzt ist und Protokollierungsdaten daher ausschließlich lokal erfasst werden.)
- Nach Ablauf von 90 Tagen muss eine umgehende Löschung von gespeicherten Protokollierungsdaten erfolgen.

Abweichungen

Abweichende Aufbewahrungszeiträume und Löschfristen können durch gesetzliche oder regulatorische Anforderungen (insbesondere im Zusammenhang mit personenbezogenen Daten) bestehen 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.

Motivation: Protokollierungsdaten sind ein immens wichtiges Werkzeug der IT-Sicherheit zur Abwendung, Detektie-

ung 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: Mittels logrotate können alte Logs gelöscht werden sobald 90 Tage erreicht werden. Die Option "shred" sorgt dafür, dass die Daten (auf Ebene des Hosts) sicher entfernt werden. Das folgende Beispiel kann i.d.R. nicht 1:1 zusätzlich zu Distro-Standards übernehmen werden. Die Optionen "daily" & "maxage" sowie "shred" sollen primär verdeutlichen wie eine Implementierung realisiert werden kann.

```
$ cat /etc/logrotate.d/any-log
/var/log/* {
    daily
    maxage 90
    shred
}
```

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten

ID: 3.65-38/9.1

Req 39	Das System muss über eine Kennzeichnung verfügen, die eine eindeutige Identifizierung ermöglicht.
--------	---

Der Hostname und eine IP-Adresse reichen häufig nicht aus, die Identität der Betriebssysteminstanz zweifelsfrei festzustellen. Bei maschineller Kommunikation mit anderen Systemen wird eine zuverlässigere Prüfung der Identität benötigt, z. B. beim Zugriff auf zentrale Managementsysteme. Hierfür wird für jeden Host individuell generiertes Schlüsselmaterial, i. d. R. ein Zertifikat, benötigt. Dies ist der Standard in Active Directory Umgebungen.

In Fällen, in denen nur eine Identifizierung aber keine Vertraulichkeit oder Integritätsschutz benötigt wird, können Daten zusammen mit einem Unique Identifier gesendet werden, z. B. eine UUID (RFC4122), die auf der Gegenseite bekannt ist.

Motivation: Eine starke Authentifizierung der Maschine anhand von Schlüsseln und/oder Zertifikaten ist gegenüber einer einfachen Identifizierung anhand des Hostnamen oder einer IP-Adresse zu bevorzugen.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen

ID: 3.37-28/9.0

Req 40	Sicherheitsrelevante Protokollierungsdaten müssen echtzeitnah an eine Erkennungs- und Reaktionslösung für Sicherheitsvorfälle bereitgestellt werden.
--------	--

Eine zulässige Erkennungs- und Reaktionslösung im Sinne dieser Anforderung muss die folgenden Kriterien erfüllen:

- Es erfolgt eine kontinuierliche Analyse bzw. Korrelation der aus angebundenen IT/NT-Systemen bereitgestellten Protokollierungsdaten.
- Die Identifizierung bedrohlicher Aktivitäten und laufender Angriffe findet auf Basis einer fortlaufend aktuell gehaltenen Erkennungsmuster-Datenbank statt.
- Integrierte Alarmierungsprozesse lösen im Fall eines Fundes die Einleitung von Gegenmaßnahmen aus.

Eine solche Lösung ist in ihrer Funktion nicht mit einem Log Server gleichzusetzen. Entsprechend muss die Bereitstellung an eine Erkennungs- und Reaktionslösung stets zusätzlich zu einer vorhandenen Weiterleitung an einen Log Server erfolgen. **Eine Ausnahme besteht, falls der genutzte Log Server seinerseits die Protokollierungsdaten umgehend nach Erhalt zusätzlich an eine Erkennungs- und Reaktionslösung bereitstellt. In diesem Fall ist es ausreichend, dass das System die Protokollierungsdaten nur an den Log Server weiterleitet.**

Die Protokollierungsdaten müssen ab dem Zeitpunkt ihrer Erzeugung im System innerhalb von 5 Minuten an die Erkennungs- und Reaktionslösung bereitgestellt werden ("echtzeitnah").

Die konkret zu unterstützenden technischen Schnittstellen und Protokolle für die Bereitstellung werden durch die im Einsatzumfeld des Systems verfügbare Erkennungs- und Reaktionslösung sowie die technische Basis des Systems bestimmt.

Motivation: IT/NT-Systeme sind permanent sicherheitstechnischen Bedrohungen und Angriffen ausgesetzt. Mittels einer zentralen Analyse und der Korrelation erfasster System-Ereignisse ganzer IT-Landschaften in nahezu Echtzeit können aktuelle Bedrohungen und laufende Angriffe auch über Systemgrenzen hinaus erkannt und zeitnah passende Gegenmaßnahmen eingeleitet werden. Die unverzügerte Bereitstellung von Protokollierungsdaten an die Erkennungs- und Reaktionslösung ist essenziell für deren Effektivität.

Umsetzungsbeispiel: **Beispiel 1:** Ein System auf Basis eines klassischen Servers erfasst sicherheitsrelevante Protokollierungsdaten im lokalen Syslog. Die Protokollierungsdaten werden kontinuierlich zur weiteren Analyse, Angriffserkennung und Alarmbehandlung an ein SIEM (Security Information and Event Management) weitergeleitet.

Beispiel 2: Auf einem System wird ein EDR ("Endpoint Detection and Response") Agent ausgeführt. Der EDR-Agent leitet die von ihm auf dem System erfassten sicherheitsrelevanten Protokollierungsdaten kontinuierlich an das zentrale Backend der EDR-Lösung aus, auf dem die weitere Analyse, Angriffserkennung und Alarmbehandlung erfolgt.

ID: 3.65-40/9.1

3.1. Auditd

Das Programm "Auditd" ist ein Zugangsprotokollierungs- und Accounting-System für Linux. Es kann genutzt werden, um granular Events zu definieren, die im Linux OS protokolliert werden sollen. Aus Sicherheitssicht können diese Events genutzt werden, um auffällige Ereignisse zu erkennen oder um potentielle Angriffe zu analysieren.

Req 41 Der Dienst "auditd" muss genutzt werden, um sicherheitsrelevante Events zu protokollieren.

Auf Linux Servern muss das Programm "auditd" installiert und konfiguriert werden, um sicherheitsrelevante Ereignisse zu loggen. Jedes sicherheitsrelevante Ereignis muss mit einem möglichst genauen Zeitstempel und einer eindeutigen Systembezeichnung protokolliert werden.

Logdaten von "auditd" zählen zu sicherheitsrelevanten Ereignissen, welche an einen zentralen Logserver weitergeleitet werden müssen, siehe auch Kapitel "externe Protokollierung".

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

ID: 3.65-41/9.1

Req 42 Syscalls "execve" (execute program) müssen protokolliert werden.

Auf Linux-Servern müssen die folgenden Events protokolliert werden:

Event-Typ	Beschreibung	Kategorie
Ausführung von Programmen	Alle Programmausführungen müssen protokolliert werden.	Verbindlich

Motivation: Forensik & Mitigation : Im Falle einer Kompromittierung eines Linux-Hosts muss nachvollziehbar sein, welche Aktionen ausgeführt wurden.

Umsetzungsbeispiel: Am Beispiel eines Ubuntu 20.04 Host kann die Lösung so aussehen:

```
$ cat /etc/audit/rules.d/execve.rules
-a exit,always -F arch=b64 -S execve
-a exit,always -F arch=b32 -S execve
```

Für diese Anforderung sind folgende Bedrohungen relevant:

- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe

ID: 3.65-42/9.1

Req 43 System-Events müssen protokolliert werden.

Auf Linux-Servern müssen die folgenden Events protokolliert werden:

Event-Typ	Beschreibung	Kategorie
System Start und Shutdown	Jeder Neustart oder Shutdown des Betriebssystems muss protokolliert werden.	Verbindlich
(De-)Installation von Software	Nach der Kommissionierung eines Servers muss jede Deinstallation und Installation von Software protokolliert werden.	Verbindlich
Änderung Systemzeit	Änderung der lokalen Systemzeit und der NTP-Konfiguration muss protokolliert werden.	Verbindlich
Verbindung von externen Geräten (Speicher)	Die Verbindungen von externen Geräten wie USB-Laufwerken, die auf dem Server gemountet werden können, müssen protokolliert werden.	Verbindlich
Ausführung privilegierter Kommandos	Die Nutzung privilegierter Kommandos mit SUID/SGID muss protokolliert werden.	Verbindlich
Laden und Stoppen von Kernel-Modulen	Das Laden und Stoppen von Kernel-Modulen muss protokolliert werden.	Verbindlich
Änderung von terminierten Jobs	Jobs, die periodisch ausgeführt werden, müssen protokolliert werden, wenn sie verändert oder gelöscht werden.	Optional

Motivation: Es ist ungewöhnlich, wenn nach der Inbetriebnahme Veränderungen an den Betriebssystemeinstellungen vorgenommen werden. Änderungen am System können mittels Logging nachvollzogen werden. Protokollierung von Systemevents ist daher notwendig, um Angriffe erkennen und nachverfolgen zu können.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe

ID: 3.65-43/9.1

Req 44 Zugriffs- und Authentifizierungs-Events müssen protokolliert werden.

Auf Linux-Servern müssen die folgenden Zugriffs- und Authentifizierungs-Events protokolliert werden:

Event-Typ	Beschreibung	Kategorie
An-/Abmeldung	An- oder Abmeldungen an/von Konten des Systems müssen protokolliert werden.	Verbindlich
Änderung von Passwörtern	Die Änderung oder das Zurücksetzen von Passwörtern muss protokolliert werden.	Verbindlich
Erweiterung von Privilegien	Ausführung von "sudo" oder Änderungen an der "sudoers" Datei müssen protokolliert werden.	Verbindlich
Änderung der DAC Berechtigungen	Die Änderung der DAC Berechtigungen muss protokolliert werden.	Verbindlich

Motivation: Die Protokollierung von Zugriffs- und Authentifizierungs-Events können hilfreich sein, um nachzuverfolgen wer Zugriff zu einem bestimmten Zeitpunkt hatte. Mittels dieser Protokollierungsdaten ist es beispielsweise möglich missbräuchlich verwendete Konten ausfindig zu machen.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe

ID: 3.65-44/9.1

Req 45 Konten- und Gruppen-Management-Events müssen protokolliert werden.

Auf Linux-Servern müssen die folgenden Konten- und Gruppen-Management-Events protokolliert werden:

Event-Typ	Beschreibung	Kategorie
Erstellen, modifizieren und entfernen von Konten	Es muss protokolliert werden, wenn Konten erstellt, modifiziert oder gelöscht werden.	Verbindlich
Erstellen, modifizieren und entfernen von Gruppen	Es muss protokolliert werden, wenn Gruppen erstellt, modifiziert oder gelöscht werden.	Verbindlich

Motivation: Das Protokollieren von Konten- und Gruppen-Management-Events kann hilfreich sein um maliziöses Anlegen, Verändern oder Löschen von Benutzern und Gruppen zu erkennen.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe

ID: 3.65-45/9.1

Req 46 Änderungen der Konfiguration müssen protokolliert werden.

Auf Linux-Servern müssen die folgenden Änderungen der Konfiguration protokolliert werden:

Event-Typ	Beschreibung	Kategorie
Deaktivierung der Protokollierung	Es muss protokolliert werden, wenn der Protokollierungsdienst deaktiviert wird.	Verbindlich
Löschen und Verändern von Protokolldaten	Das Löschen von Events muss protokolliert werden. Die nicht autorisierte Änderung von Protokollierungsdaten muss protokolliert werden.	Verbindlich
Änderung der Konfiguration der Protokollierung	Es muss protokolliert werden, wenn die Konfiguration für die Protokollierung verändert wird.	Verbindlich
Änderungen der Konfiguration des Netzwerks	Es muss protokolliert werden, wenn die Konfiguration von Netzwerk(-schnittstellen) geändert wird.	Verbindlich
Änderungen des Authentifizierungssubsystems	Änderungen des Authentifizierungssubsystems (z. B. LDAP- oder Kerberos-Policy) muss protokolliert werden.	Optional
Kritische Datei-Änderungen	Abhängig von Einsatzzweck müssen kritische Änderungen von Dateien protokolliert werden.	Optional
Änderung der Konfiguration von SELinux	Wenn SELinux eingesetzt wird, müssen die entsprechenden Ereignisse und die Änderung der Konfiguration protokolliert werden.	Optional
Änderung der Konfiguration von AppArmor	Wenn AppArmor eingesetzt wird, müssen die entsprechenden Ereignisse und die Änderung der Konfiguration protokolliert werden.	Optional

Motivation: Konfigurationsänderungen haben einen großen Einfluss auf das Betriebssystem und sind daher ein Sicherheitsrisiko. Es ist notwendig alle wichtigen Konfigurationen eines Betriebssystems zu identifizieren und Änderungen zu protokollieren.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe

ID: 3.65-46/9.1

Req 47 Die Konfiguration von auditd muss auf "immutable" gesetzt werden.

Der "Immutable Mode" muss für den auditd Dienst gesetzt sein, um zu verhindern, dass Audit-Regeln mit dem Kommando "auditctl" verändert werden können.

Motivation: Wenn das Programm "auditd" nicht im sogenannten "Immutable Mode" ist, können unautorisiert Veränderungen vorgenommen werden, um böswillige Aktivitäten zu verschleiern.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe

ID: 3.65-47/9.1

3.2. Externe Protokollierung

Req 48	Sicherheitsrelevante Protokollierungsdaten müssen umgehend nach deren Erzeugung an ein separates System weitergeleitet werden.
--------	--

Sicherheitsrelevante Protokollierungsdaten müssen neben der lokalen Speicherung in den entsprechenden Protokolldateien auch an ein externes System weitergeleitet werden. Standardisierte Protokolle wie Syslog, SNMPv3 müssen hierbei bevorzugt zum Einsatz kommen. Die Übertragung sollte gesichert erfolgen, d. h. verschlüsselt und authentifiziert, wenn schutzbedürftige Daten übertragen werden. Zur Prüfung auf Konsistenz und Vollständigkeit sollen Sequenznummern verwendet werden und die Übertragung möglichst per TCP statt UDP erfolgen.

Hinweis: Mit dem Empfänger der Daten muss abgestimmt werden, wie die gelieferten Daten zu bewerten sind.

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.

Umsetzungsbeispiel: rsyslog Konfiguration enthält:

```
*.* @@host1.central-log.internal
*.* @@host2.central-log.internal
```

ID: 3.65-48/9.1

Req 49	Für sicherheitsrelevante Protokollierungsdaten, die an den Log Server weitergeleitet werden, muss die Einhaltung geltender Aufbewahrungszeiträume und Löschfristen sichergestellt werden.
--------	---

Folgende Grundregelung ist zu berücksichtigen:

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

ID: 3.01-36/9.0

3.2.1. RSyslog

Req 50	Falls RSyslog genutzt wird, muss für Protokolldateien eine Standardberechtigung von 640 oder restriktiver konfiguriert sein.
--------	--

Für RSyslog muss eine Standardberechtigung von 640 oder restriktiver für Protokolldateien in der Datei "rsyslog.conf" konfiguriert sein.

Motivation: Durch falsche Berechtigungen ist ggf. ein unberechtigter Zugriff auf Protokolldateien möglich. Dies erhöht das Risiko einer unbeabsichtigten Preisgabe schutzbedürftiger Daten, die solche Dateien enthalten können.

Umsetzungsbeispiel: rsyslog Konfiguration enthält:

```
FileOwner syslog
FileGroup adm
FileCreateMode 0640
DirCreateMode 0750
Umask 0027
PrivDropToUser syslog
PrivDropToGroup adm
```

Für diese Anforderung sind folgende Bedrohungen relevant:

- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe

ID: 3.65-50/9.1

Req 51	Falls RSyslog genutzt wird, muss ein zentraler Protokollierungsserver konfiguriert werden.
--------	--

Einer oder mehrere externe Protokollierungsserver müssen in der RSyslog Konfiguration konfiguriert werden.

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

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten

ID: 3.65-51/9.1

3.2.2. Syslog-NG

Req 52	Falls Syslog-NG genutzt wird, muss eine Standardberechtigung von 640 oder restriktiver konfiguriert werden.
--------	---

Für Syslog-NG muss eine Standardberechtigung von 640 oder restriktiver für neu generierte Protokollierungsdateien in der Datei "syslog-ng.conf" konfiguriert werden.

Motivation: Durch falsche Berechtigungen ist ggf. ein unberechtigter Zugriff auf Protokollierungsdateien möglich. Dies erhöht das Risiko einer unbeabsichtigten Preisgabe schutzbedürftiger Daten, die solche Dateien enthalten können.

Umsetzungsbeispiel: syslog-ng Konfiguration enthält:

```
create-dirs yes
dir-owner syslog
dir-group adm
dir-perm 0750
owner syslog
group adm
perm 0640
```

Für diese Anforderung sind folgende Bedrohungen relevant:

- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe

ID: 3.65-52/9.1

Req 53	Falls Syslog-NG genutzt wird, muss ein zentraler Protokollierungsserver konfiguriert werden.
--------	--

Einer oder mehrere externe Protokollierungsserver müssen in der Syslog-NG Konfiguration konfiguriert werden.

Motivation: Werden Protokollierungsdaten nur lokal gespeichert, können diese im Falle einer Kompromittierung des Systems gezielt manipuliert werden, um Manipulationen zu verschleiern. Aus diesem Grund muss die Weiterleitung zeitnah nach dem Auftreten des Ereignisses erfolgen.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe

ID: 3.65-53/9.1

4. Authentisierungsmerkmal "Passwort" & Pluggable Authentication Modules

PAM (Pluggable Authentication Modules) ist eine Lösung, um unabhängige Module für Authentifizierungsschemata für Programme zur Verfügung zu stellen. PAM wird standardmäßig in den meisten Linux-Distributionen verwendet.

Req 54	Falls Passwörter als Authentisierungsmerkmal genutzt werden, muss die "Shadow Password Suite" konfiguriert werden um Passwörter mittels sicherem kryptographischen Algorithmus zu schützen. Passwörter dürfen erst nach Ablauf von 1 Tag und müssen spätestens nach 365 Tagen geändert werden.
--------	--

Passwörter müssen immer als Hashes gespeichert werden. **sha512** mit 640.000 Runden und einem Salt von 96 Bit oder **yescrypt** (*shadow* >= 4 . 14 . 0) müssen genutzt werden, um die Passwörter zu speichern. **yescrypt** ist zu bevorzugen sofern es verfügbar ist. Passwörter müssen mindestens 1 Tag und maximal 365 Tage verwendet werden. Der Hintergrund zu dieser Anforderung ist beschrieben in Sicherheitsanforderung 3.01 Technischer Basisschutz von IT-/NT-Systemen Kapitel 8. Authentisierungsmerkmal Passwort.

Motivation: Wenn eine nicht autorisierte Person Zugriff auf die Passwortdatei erhält, können die Passwörter einfach missbraucht werden, wenn diese nicht ausreichend geschützt sind. Gezielt häufiges ändern von Passwörtern kann missbraucht werden um die Passworthistorie zu umgehen, daher ist ein Mindestalter erforderlich.

Umsetzungsbeispiel: Es kann mit folgendem Kommando geprüft werden, ob die installierte PAM-Version auf einem Host **yescrypt** unterstützt mittels Auszug aus dem Manual:

```
man pam_unix | grep -E '^\\W+yescrypt$'
```

z.B. Ubuntu 22.04 LTS "jammy" enthält in `/etc/login.defs`:

```
ENCRYPT_METHOD YESCRYPT
PASS_MAX_DAYS 365
PASS_MIN_DAYS 1
```

z.B. Ubuntu 20.04 LTS "focal" enthält in `/etc/login.defs`:

```
ENCRYPT_METHOD SHA512
SHA_CRYPT_MIN_ROUNDS 640000
PASS_MAX_DAYS 365
PASS_MIN_DAYS 1
```

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten

ID: 3.65-54/9.1

Req 55	Falls Passwörter als Authentisierungsmerkmal genutzt werden, muss "PAM" konfiguriert werden zur Verwendung von sicheren "salted password hash functions" und ausreichend vielen Berechnungsrunden um Passwörter zu schützen.
--------	--

PAM kann konfiguriert werden um die sichere kryptografische Hashfunktion mit Salt für Passwörter zu nutzen. Um die Sicherheit weiter zu erhöhen werden mehrere Runden der Hashfunktion ausgeführt. Standardeinstellung für SHA-512 sind 5000 Runden. Dies ist nicht ausreichend aufgrund günstiger und leicht verfügbarer Rechenleistung, welche Brute-Force-Angriffe denkbar machen. Daher muss die Anzahl an Runden auf mindestens 640.000 erhöht werden.

Neuere kryptographische Bibliotheken bieten die **yescrypt** Hash-Funktionen, welche SHA-512 deutlich überlegen ist. Sie wird für mehrere Distributionen angeboten und mittlerweile standardmäßig verwendet. **yescrypt** ist zu bevorzugen sofern es verfügbar ist.

Motivation: Wenn eine nicht autorisierte Person Zugriff auf die Passwortdatei erhält, können die Passwörter einfach missbraucht werden, wenn diese nicht ausreichend geschützt sind.

Umsetzungsbeispiel: Prüfung bzgl. **yescrypt** Unterstützung durch PAM:

```
man pam_unix | grep -E '^\\W+yescrypt$'
```

Ubuntu 22.04 Beispiel: Die Datei `/etc/pam.d/common-password` enthält:

```
# Req 3.65-55
password requisite pam_pwquality.so
retry=3 minlen=12 minclass=3 enforce_for_root difok=2 maxse-
quence=3 maxrepeat=3 lcredit=0 ucredit=0 dcredit=0 ocredit=0
dictcheck=1
# Req 3.65-57
password required pam_pwhistory.so
use_authtok enforce_for_root retry=3 remember=60
# Req 3.65-54
password [success=1 default=ignore] pam_unix.so
obscure yescrypt
password requisite pam_deny.so
password required pam_permit.so
```

Achtung! PAM-Konfiguration ist komplex. Einfaches Copy/Paste führt ggf. nicht zum Erfolg. Die Reihenfolge der Zeilen ist relevant und die Ausführung startet von oben beginnend. Ein Beispiel wie mittels Ansible "PAM Hardening" für Redhat 8/9, SLES 15, Ubuntu 20/22 realisiert werden kann gibt es hier: gitlab.devops.telekom.de/ansible/roles/hardening_pam

SSSD (System Security Services Daemon) wird dort genutzt als zentrales IAM, dazu lokale Authentifizierung als Ausweichmöglichkeit.

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

ID: 3.65-55/9.1

Req 56	Falls Passwörter als Authentisierungsmerkmal genutzt werden, muss "PAM" konfiguriert werden, dass Passwortkomplexitätsregeln angewendet werden, um die Verwendung von Passwörtern mit einer Mindestlänge von 12 Zeichen und einer Kombination aus drei der folgenden Kategorien zu erzwingen: Großbuchstaben, Kleinbuchstaben, Ziffern und Sonderzeichen.
--------	---

PAM muss so konfiguriert werden, dass nur Passwörter akzeptiert werden, die die folgende Komplexität erfüllen:

- Mindestlänge 12 Zeichen
- Bestehend aus mindestens drei der folgenden Kategorien: Klein-/Großbuchstaben, Ziffern und Sonderzeichen

Motivation: Passwörter mit der genannten Komplexität bieten eine hohe Robustheit gegen Angriffe bei gleichzeitig akzeptabler Benutzerfreundlichkeit. In der Praxis haben sich Passwörter mit dieser Komplexität bewährt. Triviale und zu kurze Passwörter sind anfällig gegen Brute-Force- und Wörterbuch-Angriffe und sind von einem Angreifer leicht zu ermitteln. Ein einmal ermitteltes Passwort kann dann von einem Angreifer für den unautorisierten Zugriff auf das System und dessen Daten genutzt werden.

Umsetzungsbeispiel: Beispiel für Debian-basierte Distro: Die Datei `/etc/pam.d/common-password` enthält:

```
# Req 3.65-55
password requisite pam_pwquality.so
retry=3 minlen=12 minclass=3 enforce_for_root difok=2 maxse-
quence=3 maxrepeat=3 lcredit=0 ucredit=0 dcredit=0 ocredit=0
dictcheck=1
# Req 3.65-57
password required pam_pwhistory.so
```

```

use_authok enforce_for_root retry=3 remember=60
# Req 3.65-54
password      [success=1 default=ignore]      pam_unix.so
obscure yescrypt
password      requisite                        pam_deny.so
password      required                        pam_permit.so

```

Hinweis: **Suse Enterprise Linux** nutzt das Modul `pam_cracklib` anstelle von `pam_pwquality`, siehe [hier](#).

Achtung! PAM-Konfiguration ist komplex. Einfaches Copy/Paste führt ggf. nicht zum Erfolg. Die Reihenfolge der Zeilen ist relevant und die Ausführung startet von oben beginnend. Ein Beispiel wie mittels Ansible "PAM Hardening" für Redhat 8/9, SLES 15, Ubuntu 20/22 realisiert werden kann gibt es hier: gitlab.devops.telekom.de/ansible/roles/hardening_pam

SSSD (System Security Services Daemon) wird dort genutzt als zentrales IAM, dazu lokale Authentifizierung als Ausweichmöglichkeit.

Für diese Anforderung sind folgende Bedrohungen relevant:

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

ID: 3.65-56/9.1

Req 57	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

ID: 3.01-27/9.0

Req 58	Falls Passwörter als Authentisierungsmerkmal verwendet werden, muss die Wiederverwendung bereits eingesetzter Passwörter unterbunden werden.
--------	--

Für jedes Benutzerkonto muss eine Historie der bisher genutzten Passwörter aufgezeichnet werden. Wird eine Passwort-Änderung für ein Benutzerkonto initiiert, ist das neue Passwort gegen diese Passwort-Historie abzugleichen. Bei erkannter Wiederverwendung eines Passworts muss die Passwort-Änderung abgelehnt werden. Die Umsetzung dieses Validierungsvorgangs muss im System auf Basis von technischen Maßnahmen erfolgen. Bei Verwendung eines zentralen IAM Systems für die Authentisierung der Benutzer darf die Umsetzung an das zentrale IAM System weitergereicht bzw. dorthin ausgelagert werden [vgl. Wurzel-Anforderungsdokument "3.69 IAM (Identity Access Management) - Framework"].

Generell soll mittels der Passwort-Historie sichergestellt werden, dass ein schon einmal verwendetes Passwort niemals erneut einsetzbar ist.

Aufgrund technischer Limitierungen kann eine Passwort-Historie jedoch in vielen IT/NT-Produkten nicht uneingeschränkt aufgezeichnet werden. In diesem Fall ist die folgende Grundregelung zu berücksichtigen:

- ein bereits genutztes Passwort darf für den Zeitraum von mindestens 60 Tagen nicht erneut einsetzbar sein (gemessen ab dem Zeitpunkt, an dem das betroffene Passwort durch ein anderes ersetzt wurde)
- in Systemen, in denen der Zeitraum von mindestens 60 Tagen nicht abgebildet werden kann, muss der längst mögliche Zeitraum konfiguriert werden. Zusätzlich muss durch einen Project Security Manager (PSM) bestätigt werden, dass der konfigurierte Zeitraum im Gesamtkontext des Systems noch ausreichend im Sinne der Sicherheitsanforderung ist.

Anmerkung:

Einige IT/NT-Produkte bieten keinen technischen Konfigurationsparameter an, mit dem die Passwort-Historie direkt an einen Zeitraum gebunden werden kann, sondern ermöglichen nur, die Anzahl der aufzuzeichnenden Passwörter zu definieren. In solchen Fällen darf der Zeitraum alternativ durch eine Verknüpfung der folgend beschriebenen, i. d. R. breit verfügbaren Konfigurationsparameter sichergestellt werden. Innerhalb der resultierenden Policy kann ein Benutzer nur einmal pro Tag sein Passwort ändern und aufgrund der Anzahl der aufgezeichneten Passwörter effektiv nach frühestens 60 Tagen ein altes Passwort erneut einsetzen.

- Minimum Passwort-Alter: 1 Tag
- Passwort-Historie: Aufzeichnung der letzten 60 benutzten Passwörter

In dieser Umsetzungsvariante ist zu beachten, dass das Minimum Passwort-Alter nicht mehr als einen Tag betragen sollte, um den Benutzer nicht unangemessen hinsichtlich der grundsätzlichen Notwendigkeit einzuschränken, das Passwort jederzeit selbstständig ändern zu können.

Motivation: Benutzer favorisieren in der Regel Passwörter, die sie sich gut merken können und verwenden diese in Folge oft wiederkehrend über lange Zeiträume, falls das System dies zulässt. Aus Sicht des Benutzers ist das Verhalten nachvollziehbar, führt aber effektiv zu einer erheblichen Reduzierung der Schutzwirkung dieses Authentisierungsparameters. Mit etwa entsprechenden Kenntnissen über den Benutzer oder erbeuteten Informationen aus zurückliegenden System-Kompromittierungen kann ein Angreifer so Zugriff auf vermeintlich geschützte Benutzerkonten erlangen. Insbesondere auch in Situationen, in denen im Rahmen einer akuten Risikobehandlung zentral neue Initialpasswörter vergeben werden, die Benutzer diese jedoch unmittelbar aus Bequemlichkeit auf ein vorheriges Passwort zurückändern, bestünde hohe Gefahr, dass ein Angreifer seinen illegalen Zugriff erneut aufnehmen kann. Es ist daher wichtig zu vermeiden, dass Benutzer alte Passwörter erneut einsetzen.

Umsetzungsbeispiel: Beispiel für Debian-basierte Distro: Die Datei `/etc/pam.d/common-password` enthält:

```
# Req 3.65-55
password requisite pam_pwquality.so
retry=3 minlen=12 minclass=3 enforce_for_root difok=2 maxsequence=3 maxrepeat=3 lcredit=0 ucredit=0 dcredit=0 ocredit=0
dictcheck=1
# Req 3.65-57
password required pam_pwhistory.so
use_authtok enforce_for_root retry=3 remember=60
```

```
# Req 3.65-54
password      [success=1 default=ignore]      pam_unix.so
obscure yescrypt
password      requisite                        pam_deny.so
password      required                        pam_permit.so
```

Achtung! PAM-Konfiguration ist komplex. **Achtung!** PAM-Konfiguration ist komplex. Einfaches Copy/Paste führt ggf. nicht zum Erfolg. Die Reihenfolge der Zeilen ist relevant und die Ausführung startet von oben beginnend. Ein Beispiel wie mittels Ansible "PAM Hardening" für Redhat 8/9, SLES 15, Ubuntu 20/22 realisiert werden kann gibt es hier: gitlab.devops.telekom.de/ansible/roles/hardening_pam

SSSD (System Security Services Daemon) wird dort genutzt als zentrales IAM, dazu lokale Authentifizierung als Ausweichmöglichkeit.

ID: 3.65-58/9.1

Req 59	Falls Passwörter als Authentisierungsmerkmal genutzt werden, muss "PAM" konfiguriert werden um einen Schutz gegen Online-Angriffe wie Wörterbuch- und Brute-Force-Attacken zu bieten, welcher das Erraten von Passwörtern stark erschwert.
--------	--

Wörterbuch- und Brute-Force-Angriffe können für automatische Angriffe zum Erraten von Passwörtern von Benutzer- und Maschinenkonten genutzt werden. Um sich gegen diese Angriffe zu schützen, muss eine Beschränkung von fehlerhaften Eingabeversuchen konfiguriert werden. Es wird empfohlen, ein Benutzerkonto nach 5 Fehleingaben für 10 Minuten zu sperren.

Motivation: Ohne entsprechenden Schutz kann ein Angreifer durch bloßes Durchprobieren von Wörterbuchlisten oder automatisch erzeugten Zeichenkombinationen versuchen, ein Passwort zu ermitteln, um so das entsprechende Benutzerkonto missbräuchlich zu nutzen.

Umsetzungsbeispiel: Ubuntu 22.04 Beispiel: Die Datei `/etc/pam.d/common-auth` enthält:

```
auth      required                        pam_faillock.so      preauth
silent even_deny_root audit deny=5 unlock_time=600
auth      [success=1 default=bad]        pam_unix.so
try_first_pass
auth      [default=die]                  pam_faillock.so      authfail
silent even_deny_root audit deny=5 unlock_time=600
auth      optional                       pam_cap.so
auth      sufficient                     pam_faillock.so      authsucc
silent even_deny_root audit deny=5 unlock_time=600
auth      required                       pam_deny.so
```

Hinweis: Ältere Distros setzen ggf. auf Das Modul `pam_tally2` anstelle von `pam_faillock`.

Achtung! PAM-Konfiguration ist komplex. Einfaches Copy/Paste führt ggf. nicht zum Erfolg. Die Reihenfolge der Zeilen ist relevant und die Ausführung startet von oben beginnend. Ein Beispiel wie mittels Ansible "PAM Hardening" für Redhat 8/9, SLES 15, Ubuntu 20/22 realisiert werden kann gibt es hier: gitlab.devops.telekom.de/ansible/roles/hardening_pam

SSSD (System Security Services Daemon) wird dort genutzt als zentrales IAM, dazu lokale Authentifizierung als Ausweichmöglichkeit.

Für diese Anforderung sind folgende Bedrohungen relevant:

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

ID: 3.65-59/9.1

Req 60	PAM muss so konfiguriert werden, dass Message-of-the-Day (motd) keine schutzbedürftigen Daten enthält.
--------	--

Message-of-the-Day (motd) darf keine vertraulichen Daten enthalten. Dies sind beispielsweise Angaben zu fehlenden

Patches, die genutzte Software oder Kernel Version. Die folgenden Dateien müssen auf solche Informationen hin überprüft werden:

- /etc/pam.d/login
- /etc/pam.d/sshd

Motivation: Vertrauliche Daten in Motd geben einem Angreifer Informationen, die hilfreich für die Planung weiterer Angriffe sein können.

Umsetzungsbeispiel: Beispiel für Debian-basierte Distro :

```
user@server:~# sudo rm /etc/update-motd.d/*
```

Für diese Anforderung sind folgende Bedrohungen relevant:

- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.65-60/9.1

5. Host-basierte Firewall

Eine Host-basierte Firewall wird empfohlen um die Erreichbarkeit von Diensten zu beschränken, wie es in Anforderung 2 abgeleitet vom Technischer Basisschutz von IT-/NT-Systemen auch gefordert wird.

Beispiele sind iptables, nftables, ufw.

Req 61	Falls iptables, nftables oder ufw genutzt wird, müssen Regeln für den Loopback-Verkehr konfiguriert werden.
--------	---

Loopback-Verkehr wird zwischen Serverprozessen genutzt. Eine Regel für den Verkehr des Loopback-Netzes (127.0.0.1/8, ::1) muss für jede Netzwerkschnittstelle des Servers konfiguriert werden.

Motivation: Der Verkehr zu Loopback-Netzwerken muss geschützt werden, um Angriffe mit gefälschten Adressen zu verhindern.

Umsetzungsbeispiel: Am Beispiel eines **Ubuntu 20.04** Host mit **iptables** kann die Lösung so aussehen

```
$ iptables -A INPUT -i lo -j ACCEPT
$ iptables -A OUTPUT -o lo -j ACCEPT
$ iptables -A INPUT -s 127.0.0.0/8 -j DROP
```

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Störung der Verfügbarkeit

ID: 3.65-61/9.1

Req 62	Falls iptables, nftables oder ufw genutzt wird, müssen Regeln für eingehende und (sofern genutzt) weiterleitende Verbindungen konfiguriert werden.
--------	--

Für eingehende und (ggf.) weiterleitende Verbindungen müssen Regeln konfiguriert werden.

- iptables: `tablesINPUT&FORWARD( https://ipset.netfilter.org/iptables.man.html )`
- nftables: `at least a table in net containing a base chain hook input & a base chain hook forward( https://manpages.ubuntu.com/manpages/focal/man8/nft.8.html )`
- ufw: `DIRECTIONincoming&routed( http://manpages.ubuntu.com/manpages/focal/man8/ufw.8.html )`

Motivation: Ohne Regeln wird jeglicher Verkehr durch die Standardeinstellung (default drop) geblockt und damit die Nutzung des Netzes gestört.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Störung der Verfügbarkeit

ID: 3.65-62/9.1

Req 63	Falls iptables, nftables oder ufw genutzt wird, müssen Regeln für alle offenen Ports existieren.
--------	--

Es ist notwendig, Regeln für alle verfügbaren Dienste (Ports) eines Servers zu konfigurieren. Wenn eine uneingeschränkte Erreichbarkeit nicht notwendig ist (z. B. bei Management-Diensten wie SSH) müssen zusätzlich die erlaubten IP-Adressen oder IP-Netze der Kommunikationspartner konfiguriert werden.

Motivation: Die Einschränkung der Erreichbarkeit von Diensten verringert die potentiellen Angriffsmöglichkeiten eines

Systems.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Störung der Verfügbarkeit

ID: 3.65-63/9.1

Req 64	Falls iptables, nftables oder ufw genutzt wird, muss die Standardrichtlinie (default policy) für eingehenden oder weiterleitenden Verkehr Pakete verwerfen.
--------	---

Die Standardeinstellungen von müssen dazu führen, dass ausschließlich explizit zugelassener Datenverkehr eingehend und weiterleitend erlaubt ist.

- iptables: `tables INPUT&FORWARD`(<https://ipset.netfilter.org/iptables.man.html>)
- nftables: `type filter hook input&forward`(<https://manpages.ubuntu.com/manpages/focal/man8/nft.8.html>)
- ufw: `DIRECTION incoming&routed`(<https://manpages.ubuntu.com/manpages/focal/man8/ufw.8.html>)

Motivation: Es besteht die Möglichkeit, dass Dienste unkontrolliert erreichbar sind, wenn keine Standardregel existiert.

Umsetzungsbeispiel:

```
$ sudo iptables -P INPUT DROP
$ sudo iptables -P FORWARD DROP

$ sudo nft chain inet filter input { policy drop \; }
$ sudo nft chain inet filter forward { policy drop \; }

$ sudo ufw default deny incoming
$ sudo ufw default deny routed
```

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Störung der Verfügbarkeit

ID: 3.65-64/9.1

6. Mandatory Access Control (MAC)

"Mandatory Access Control" (MAC) ist eine Methode, um festzulegen, welche Benutzer oder Programme auf Ihrem Computer auf welche Dateien, Ordner oder andere Ressourcen zugreifen dürfen. Im Fall von Linux ist es ein Sicherheitsmechanismus, der von einem Administrator festgelegt wird. Mit MAC können bestimmte Regeln erstellt werden, die bestimmen, welche Aktionen Benutzer oder Programme ausführen dürfen. Es ermöglicht eine genauere Kontrolle über die Sicherheit Ihres Computers, indem es die Zugriffe auf eine "need-to-know" Basis einschränkt.

Empfohlene Implementierung von MAC für Linux sind:

1. **SELinux (Security-Enhanced Linux)** wurde von der National Security Agency (NSA) entwickelt und ist in vielen Linux-Distributionen standardmäßig enthalten, darunter *Red Hat Enterprise Linux (RHEL)*, *CentOS* und *Fedora*. Es ermöglicht eine sehr feine Granularität bei der Steuerung von Zugriffsrechten auf Dateien, Prozesse, Ports und andere Ressourcen.
2. **AppArmor** wurde von Canonical für *Ubuntu* entwickelt und ist auch in anderen Linux-Distributionen verfügbar. Im Vergleich zu SELinux ist AppArmor oft als einfacher zu bedienen und einzurichten bekannt. Es arbeitet auf der Grundlage von "Profiles", die den Zugriff von Anwendungen auf Dateien und Systemressourcen einschränken.

Req 65	Falls ein System aus dem Internet erreichbare Dienste hat, es sich um einen Virtualisierungs- oder Container-Host handelt, so muss eine Implementierung für Mandatory Access Control (MAC) genutzt werden zur Einschränkung der Dienste, Gast-VMs bzw. Container.
--------	---

Mandatory Access Control (MAC) erzwingt erweiterte Sicherheitsrichtlinien für den Zugriff. Eine MAC-Lösung wie beispielsweise SELinux oder AppArmor muss auf, wie im Folgenden genannten, Systemen mit erhöhtem Schutzbedarf genutzt werden:

- Server mit Diensten, die aus dem Internet erreichbar sind.
- Host-Systeme mit Hypervisor-Virtualisierung
- Host-Systeme für Container-Lösungen

Hinweis: Auf Debian-basiertem Linux (z. B. Ubuntu) und SUSE Enterprise Linux wird standardmäßig AppArmor verwendet. Auf RedHat-basierten Linux (z. B. RHEL, Oracle Linux, CentOS) wird SELinux verwendet. Es wird empfohlen, die vorinstallierte MAC-Lösung der entsprechenden Linux-Distribution zu verwenden.

Motivation: Das strikte Zugriffsmodell von MAC-Implementierungen bietet einen besseren Schutz als Linux selber. Im Falle einer erfolgreichen Kompromittierung eines Dienstes wird der Angreifer daran gehindert weitere Teile des Systems zu erreichen.

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

ID: 3.65-65/9.1

6.1. SELinux

Req 66	Falls die MAC-Implementierung SELinux genutzt wird, darf die Funktion nicht in der Bootloader Konfiguration deaktiviert werden.
--------	---

In den meisten Distributionen wird die SELinux Infrastruktur vom Kernel standardmäßig geladen. Nur die Übergabe des Kernel-Parameter `selinux=0` würde zur Deaktivierung führen. SELinux muss also in der Grub-Bootloader-Konfiguration nicht deaktiviert sein, damit die entsprechende Funktion beim Systemstart ausgeführt wird.

Motivation: Das strikte Zugriffsmodell von MAC-Implementierungen bietet einen besseren Schutz als Linux selber. Im Falle einer erfolgreichen Kompromittierung eines Dienstes wird der Angreifer daran gehindert weitere Teile des Systems zu erreichen.

Umsetzungsbeispiel: Beispiel für Fedora- bzw. RedHat-basierte Distro:

```
$ sudo grubby --update-kernel=ALL --remove-args 'selinux=0 enforcing=0'
```

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

ID: 3.65-66/9.1

Req 67	Falls die MAC-Implementierung SELinux genutzt wird, muss es im Modus "enforcing" ausgeführt werden um tatsächlich konfigurierte Richtlinien durchzusetzen.
--------	--

Prozesse mit hoher Kritikalität sollten mittels Mandatory-Access-Control (MAC) wie SELinux geschützt werden um die Verwundbarkeit zu reduzieren. Software wie Hypervisor, Container Software oder aus dem Internet erreichbare Netzwerkdienste sind gute Beispiele zur Anwendung. Mittels Profilen schränkt SELinux Prozesse auf ihr erwartetes Verhalten hin ein. Entweder selbst oder vorkonfigurierte Profile müssen eingesetzt werden um Software hoher Kritikalität einzuschränken.

Für einige Linux Distributionen läuft SELinux per Standard im "permissive" Modus, welcher dazu führt, dass nur Warnung geloggt werden und Richtlinien nicht erzwungen. Aus diesem Grund muss außerdem geprüft werden ob der "enforcing" Modus konfiguriert ist.

Motivation: Das strikte Zugriffsmodell von MAC-Implementierungen bietet einen besseren Schutz als Linux allein. Im Angriffsfall wird eine weitergehende Kompromittierung signifikant erschwert.

Umsetzungsbeispiel: SELinux wird unter `/etc/selinux/config` konfiguriert. Für Debian-basierte Distros wird die Nutzung von Kernelparametern (`selinux=1 security=selinux enforcing=1`) empfohlen. Red Hat wiederum nutzt nur die Konfigurationsdatei. *In jedem Fall ist am besten die offizielle Dokumentation der Distribution zu rate zu ziehen!*

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

ID: 3.65-67/9.1

Req 68	Falls die MAC-Implementierung SELinux genutzt wird, muss eine Policy konfiguriert werden.
--------	---

Die SELinux Policy muss mindestens auf `"SELINUXTYPE=targeted"` konfiguriert werden. Die [Dokumentation auf redhat.com](#) bzgl. "Writing a custom SELinux policy" ist ein guter Ansatzpunkt.

Motivation: Das strikte Zugriffsmodell von MAC-Implementierungen bietet einen besseren Schutz als Linux allein. Im Angriffsfall wird eine weitergehende Kompromittierung signifikant erschwert.

Umsetzungsbeispiel: Datei `"/etc/selinux/config"` enthält:

```
SELINUXTYPE=targeted
```

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

ID: 3.65-68/9.1

Req 69	Falls die MAC-Implementierung SELinux genutzt wird, dürfen die Dienste SETroubleshoot und MCS Translation Service nicht installiert werden.
--------	---

Der Dienst „SETroubleshoot“ darf auf Servern nicht genutzt werden. Dies gilt insbesondere, da X-Windows nicht genutzt wird. Der Dienst „MCS Translation Service“ darf ebenfalls nicht genutzt werden, da er für den Betrieb von SELinux nicht benötigt wird. Beide Dienste dürfen nicht installiert werden oder müssen deinstalliert werden.

Motivation: Nicht benötigte Software und Dienste dürfen nicht auf einem System installiert werden, um potentiell fehlerhaften Code zu minimieren.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Störung der Verfügbarkeit

ID: 3.65-69/9.1

6.2. AppArmor

Req 70	Falls die MAC-Implementierung AppArmor genutzt wird, darf die Funktion nicht in der Bootloader Konfiguration deaktiviert werden.
--------	--

AppArmor muss in der Grub-Bootloader-Konfiguration aktiviert sein, damit die entsprechende Funktion beim Systemstart ausgeführt wird.

Motivation: Das strikte Zugriffsmodell von MAC-Implementierungen bietet einen besseren Schutz als Linux allein. Im Angriffsfall wird eine weitergehende Kompromittierung signifikant erschwert.

Umsetzungsbeispiel: GRUB 2 Konfiguration (/etc/default/grub) hat u.a. folgende Kernel Parameter für den Bootvorgang:

```
GRUB_CMDLINE_LINUX="apparmor=1 security=apparmor"
```

Nach Anpassung der GRUB 2 Konfiguration ist das Ausführen von "update - grub" notwendig.

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

ID: 3.65-70/9.1

Req 71	Falls die MAC-Implementierung AppArmor genutzt wird, muss der Status auf "enforced" gesetzt werden.
--------	---

Profile für AppArmor müssen für Dienste mit einem hohen Sicherheitsrisiko, wie z. B. Hypervisor, Container Software oder aus dem Internet erreichbare Netzwerkdienste auf "enforced" gesetzt werden.

Motivation: Das strikte Zugriffsmodell von MAC-Implementierungen bietet einen besseren Schutz als Linux allein. Im Angriffsfall wird eine weitergehende Kompromittierung signifikant erschwert.

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

ID: 3.65-71/9.1

7. Regelmäßige Compliance Checks

Dieses Kapitel beinhaltet Anforderungen, die dem Standardverhalten des Linux OS entsprechen. Allerdings ist die Nennung dieser Anforderungen auch wichtig, da es notwendig ist, diese in regelmäßigen Compliance-Checks zu überprüfen, um Fehlkonfigurationen zu identifizieren und um ein adäquates Sicherheitsniveau während des kompletten Lebenszyklus eines Systems zu garantieren.

Req 72	Es dürfen keine Legacy + Einträge in den Dateien passwd, shadows und group existieren.
--------	--

Das Zeichen "+" wird als ein Marker genutzt, um Daten in Konfigurationsdateien von NIS (Network Information Service) Maps einzufügen. Daher darf dieser Marker nicht in den Dateien '/etc/passwd', '/etc/shadows' und '/etc/group' vorkommen.

Motivation: Unautorisierte Erweiterung von Privilegien droht im Angriffsfall, wenn das Zeichen "+" in einer der Dateien '/etc/passwd', '/etc/shadows' oder '/etc/group' existiert.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Modifikation von Daten

ID: 3.65-72/9.1

Req 73	Die Home-Verzeichnisse von Konten müssen eine Berechtigung von 750 oder restriktiver aufweisen.
--------	---

Home-Verzeichnisse müssen ihren zugeordneten Konten gehören und "750" als Berechtigung aufweisen. In Betracht gezogen muss das für „root“ mit Home-Verz. "/root" und von Menschen oder Maschinen genutzte Konten i.d.R. mit Home-Verz. unter "/home". Nicht gefordert wird es für Systemkonten, welche u.a. durch das Betriebssystem vordefiniert und in /etc/passwd mit Home-Verzeichnissen wie „/dev“, „/bin“ usw zu finden sind. Dieser Zustand muss regelmäßig geprüft und ggf. wiederhergestellt werden. Ausgenommen von dieser Anforderung sind Home-Verzeichnisse von Benutzern, die für eine chroot-Umgebung für SFTP genutzt werden. Bei diesen Benutzern muss das Home-Verzeichnis im Besitz von root sein.

Motivation: Zur Unterbindung von zahlreichen Angriffsmöglichkeiten müssen fehlerhafte Berechtigungen ausgeschlossen werden.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten

ID: 3.65-73/9.1

Req 74	Die Standardgruppe des Kontos "root" muss GID 0 sein.
--------	---

Die Nutzung von GID 0 als Standardgruppe für das "root"-Konto ist Standard auf den meisten Linux Systemen. Diese darf nicht geändert werden und sollte regelmäßig in der Datei "/etc/passwd" überprüft werden.

Motivation: Die Veränderung der GID für "root" kann dazu führen, dass Dateien im Besitz von "root" für Angriffe missbraucht werden.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten

ID: 3.65-74/9.1

Req 75 "root" muss das einzige Konto mit UID 0 sein.

Jedes Konto mit UID 0 hat "root"-Rechte auf einem Linux-Host. Daher muss regelmäßig überprüft werden, dass "root" das einzige Konto mit UID 0 ist.

Motivation: Unprivilegierte Konten erlangen volle Systemrechte wie das "root"-Konto, wenn ihnen versehentlich die UID 0 zugeordnet wird.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten

ID: 3.65-75/9.1

Req 76 Alle Gruppen in der Datei /etc/passwd müssen auch in der Datei /etc/group existieren.

Es muss regelmäßig überprüft werden, ob alle Gruppen definiert in /etc/passwd auch in der /etc/group definiert sind.

Beispieleintrag aus /etc/passwd :

thomas:x:1000:1000:Thomas,,,:/home/thomas:/bin/bash

Die rot markierte Zahl zeigt die primäre Gruppe des Nutzers an und diese muss auch in der /etc/group definiert sein.

Motivation: Fehler im Gruppenmanagement können zu unspezifischem Verhalten bei der Authentifizierung und Autorisierung führen.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System

ID: 3.65-76/9.1

Req 77 Es dürfen keine doppelten UIDs und GIDs existieren.

Es muss regelmäßig überprüft werden, dass keine doppelten UIDs und GIDs auf dem Server existieren.

Motivation: Konten und Gruppen müssen eindeutige UIDs und GIDs zugewiesen werden, um einen angemessenen Zugriffsschutz sicherzustellen.

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

ID: 3.65-77/9.1

Req 78 Es dürfen keine doppelten Konten- oder Gruppennamen existieren.

Die Programme "useradd" und "groupadd" erlauben nicht, doppelte Konten- oder Gruppennamen zu vergeben. Allerdings ist es einem Administrator möglich, diesen Fehler durch manuelles Editieren der entsprechenden Konfigurationsdateien zu erzeugen. Aus diesem Grund müssen die Dateien "/etc/passwd" und "/etc/group" regelmäßig auf doppelte Einträge hin überprüft werden.

Motivation: Doppelte Konten- oder Gruppennamen bringend das Risiko eines unautorisierten Zugriffs mit sich.

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

ID: 3.65-78/9.1

Req 79 Falls eine Debian-basierte Linux-Distribution verwendet wird, muss die Gruppe "shadow" leer sein.

Bei Debian-basierten Linux-Distributionen ermöglicht die Gruppe "shadow" Systemprogrammen den lesenden Zugriff auf die Datei `/etc/shadow`. Es dürfen keine Konten dieser Gruppe zugeordnet werden. Die Gruppe "shadow" muss in der Datei `/etc/group` regelmäßig überprüft werden, ob ihr keine Konten zugeordnet sind.

Motivation: Unberechtigter Zugriff auf die Datei `/etc/shadow` ermöglicht es Passwort-Cracking-Angriffe auf die gespeicherten Passwort-Hashes durchzuführen.

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

ID: 3.65-79/9.1

Req 80 Es dürfen keine Dateien oder Verzeichnisse ohne zugehörigem Konto oder Gruppe existieren.

Wenn ein Konto oder eine Gruppe von einem System gelöscht wird, müssen die zugehörigen Dateien und Verzeichnisse auch gelöscht werden oder sie müssen einem anderen Konto oder Gruppe zugeordnet werden. Andernfalls existieren Dateien und Verzeichnisse ohne zugehöriges Konto oder Gruppe auf einem System. Das System muss regelmäßig auf verwaiste Dateien und Verzeichnisse überprüft werden.

Motivation: Existieren Dateien und Verzeichnisse ohne Konto oder Gruppe auf einem System kann es passieren, dass ein anderer Konto versehentlich darauf Zugriff erhält, sollte es die selbe UID oder GID wie das zuvor gelöschte Konto erhalten.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten

ID: 3.65-80/9.1

Req 81 Sicherheitsrelevante Konfigurationsdateien müssen dem richtigen Konto zugeordnet und die Berechtigungen eingeschränkt sein.

Die Berechtigungen von Konfigurationsdateien muss korrekt und Konto & Gruppe müssen richtig gesetzt sein. Dies betrifft Dateien der Shadow Password Suite (`passwd`, `group`, `shadow`), GRUB 2 Bootloader (`grub.cfg`, `user.cfg`), Cron-Daemon (`crontab`, `cron.d`, usw.) und Shells.

Motivation: In Konfigurationsdateien werden schutzbedürftige Informationen gespeichert. Falsche Berechtigungen erlauben Daten zu missbrauchen oder die Konfiguration zu manipulieren.

Umsetzungsbeispiel: Beispiel für eine Debian-basierte Distribution mittels `bash`:

```
sudo chown root:root /boot/grub/grub.cfg ; sudo chmod u-x,go-rwx /boot/grub/grub.cfg
sudo chown -R root:root /etc/*cron* ; sudo chmod -R go-rwx /etc/*cron*
```

```
# Wenn sich symbolische Links in Cron-Verzeichnissen befinden, ändern
Sie auch
# Besitz und Berechtigung referenzierter Dateien.
sudo chown root:root /etc/passwd* ; sudo chmod u-x,go-wx /etc/passwd*
sudo chown root:root /etc/group* ; sudo chmod u-x,go-wx /etc/group*
sudo chown root:root /etc/shadow* ; sudo chmod u-x,g-wx,o-rwx /etc/
shadow*
sudo chown root:root /etc/gshadow* ; sudo chmod u-x,g-wx,o-rwx /etc/
gshadow*
sudo chown root:root /etc/security/opasswd* ; sudo chmod u-x,go-rwx /
etc/security/opasswd*
sudo chown root:root /etc/shells ; sudo chmod u-x,go-wx /etc/shells
```

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten

ID: 3.65-81/9.1