

Sicherheitsanforderung

PostgreSQL Datenbanken

Deutsche Telekom Gruppe

Version	10.0
Datum	01.12.2025
Status	Freigegeben

Impressum

Herausgeber
Deutsche Telekom AG
Vorstandsbereich Technology & Innovation
Chief Security Officer

Reuterstrasse 65, 53315 Bonn
Deutschland

Dateiname	Dokumentennummer 3.60	Dokumententyp Sicherheitsanforderung
Version 10.0	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

Dieses Papier beschreibt Sicherheitsanforderungen an die Entwicklung / Betrieb von PostgreSQL Datenbanken.

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
2.	Allgemein	5
3.	Verzeichnis- und Dateiberechtigungen	6
4.	Passwörter	7
5.	Least Privilege Prinzip und Rollen/Rechte Trennung	10
6.	SQL Funktionen mit Betriebssystem und/oder Netzwerkaccess	14
7.	Datenkommunikation	16
8.	Logging und Monitoring	18
9.	Technischer Basisschutz von IT-/NT-Systemen	24
9.1.	Authentifizierung und Autorisierung	24

1. Einleitung

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

2. Allgemein

Req 1 In Produktionsumgebungen muss eine durch die Community bzw. kommerziellen Hersteller supportete Version verwendet werden.

Motivation: Die Entwickler liefern für nicht supportete Softwareversionen keine Sicherheitsupdates.

Umsetzungsbeispiel: Die exakte Versionsnummer kann mittels folgenden Befehls ermittelt werden:

`SELECT version();`

Die Clientversion kann mittels folgenden Befehls überprüft werden:

`psql -V`

Anschließend kann die Versionsnummer mit den auf der folgenden Webseite veröffentlichten Anbieterinformationen abgeglichen werden:

<https://www.postgresql.org/support/versioning/>

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: 3.60-1/10.0

Req 2 Auf einer Betriebssysteminstanz (mit physischer oder virtualisierter Hardware) dürfen nur dann mehrere PostgreSQL Instanzen zusammen betrieben werden, wenn diese den gleichen, niedrigen Schutzbedarf haben, unter einer einzigen Kundenhoheit stehen und administrativ vom gleichen Personenkreis betrieben werden.

Motivation: Durch den gemeinsamen Betrieb mehrerer Datenbankinstanzen steigt die Gefahr, dass Benutzer unberechtigterweise Zugriff auf Instanzen erhalten, für die sie weder fachlich noch administrativ zuständig sind.

Umsetzungsbeispiel: Zur Trennung der einzelnen Datenbankinstanzen können Administratoren physisch Serverhardware oder Virtualisierungslösungen einsetzen.

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: 3.60-2/10.0

3. Verzeichnis- und Dateiberechtigungen

Req 3	Ein Datenbankdienst darf nicht mit Root-Rechten oder anderen administrativen Rechten des Betriebssystems laufen.
-------	--

Motivation: Läuft ein Datenbankdienst mit hochprivilegierten Betriebssystemrechten (Root, Administrator), kann ein Angreifer durch die Ausnutzung einer Schwachstelle ein Datenbanksystem vollständig übernehmen (Zugriff auf das Filesystem bzw. Ausführung von Dateien).

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: 3.16-18/9.0

Req 4	Das PostgreSQL "data_directory" Verzeichnis und Konfigurationsdateien müssen exklusiv dem Betriebssystem Account der Datenbank zugewiesen werden. Anderen Systemuser müssen die Zugriffsrechte auf diese Daten entzogen werden.
-------	---

Die Berechtigungen der Verzeichnisse müssen wie folgt gesetzt werden:

Verzeichnispfad	Verzeichnisberechtigung
'data_directory'	'700:postgres:postgres'
'data_directory'/initdb.log	'600:postgres:postgres'
'data_directory'/other_subdirectories'	'700:postgres:postgres'

Zusätzlich muss die umask des postgres Benutzers auf „0077“ gesetzt werden, damit neue Dateien automatisch mit entsprechend restriktiven Berechtigungen erstellt werden.

Motivation: Zu lasch gesetzte Dateirechte führen zu einer erhöhten Angreifbarkeit.

Beispiel: Logfiles können durch Angreifer verändert werden.

Umsetzungsbeispiel: Bei PostgreSQL 15 können die Verzeichnisberechtigungen mit folgendem Befehl überprüft werden:

```
ls -la ~postgres/15
```

Falls erforderlich, können die Berechtigungen mit den Linux Kommandos „chmod“ und „chown“ angepasst werden.

Die umask des aktuellen Benutzers kann mit dem Befehl „umask“ überprüft werden.

Abhängig von der Benutzerumgebung von postgres kann die umask des postgres Benutzers in dessen „bash_profile“, „.profile“ oder „.bashrc“ Dateien mittels des Eintrags „umask 0077“ gesetzt werden.

Für diese Anforderung sind folgende Bedrohungen relevant:

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

ID: 3.60-4/10.0

4. Passwörter

Req 5	Falls Windows als Betriebssystem verwendet wird, müssen die Standardpasswörter geändert werden.
-------	---

Motivation: Häufig werden System aufgrund von Kenntnis der Standardpasswörter kompromittiert.

Umsetzungsbeispiel: Speziell beim "postgres" Account muss sichergestellt werden, dass das Passwort geändert wird.

Folgender Befehl dient zur Änderung eines Passwortes:
\password <username>

Diese Passwortänderung wird nicht in dem Logfile gespeichert.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Abstreiten von durchgeführten Aktionen

ID: 3.60-5/10.0

Req 6	Passwörter in PostgreSQL müssen unter Einsatz eines für diesen Zweck als ausreichend sicher anerkannten "Password Hashing"-Verfahrens gehashed gespeichert und übertragen werden.
-------	---

Durch die Verwendung der Authentisierungsmethode "scram-sha-256" kann diese Anforderung erfüllt werden.

Die Authentisierungsmethode „md5“ erfüllt diese Anforderung nicht.

Motivation: Die Verwendung von Hashes (sowohl bei Speicherung als auch Übertragung) erhöht die Sicherheit deutlich.

Umsetzungsbeispiel: In der Datei "postgresql.conf" muss der Parameter "password_encryption" auf "scram-sha-256" gesetzt werden:
password_encryption = scram-sha-256

Desweiteren müssen in der Datei "pg_hba.conf" die betroffenen Einträge entsprechend der nachfolgenden Beispiele konfiguriert werden:

```
# TYPE DATABASE USER ADDRESS METHOD
# "local" is for Unix domain socket connections only
local database_1 user_1 scram-sha-256
```

```
# IPv4 local connections:
host database_2 user_2 127.0.0.1/32 scram-sha-256
```

```
# IPv6 local connections:
host database_2 user_3 ::1/128 scram-sha-256
```

```
# IPv4 remote connections:
hostssl database_1 user_4 10.23.24.123/32 scram-sha-256
```

Für die kodierte Speicherung eines Passwortes kann z. B. der nachfolgende Befehl verwendet werden:
CREATE USER user_1 PASSWORD '<password>;'

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: 3.60-6/10.0

Req 7	Falls Passwörter als Authentisierungsmerkmal genutzt werden, müssen diese mindestens 12 Zeichen lang sein und drei der folgenden Zeichenkategorien beinhalten: Kleinbuchstaben, Großbuchstaben, Ziffern und Sonderzeichen.
-------	--

Ein System darf nur Passwörter akzeptieren, welche die folgende Komplexität erfüllen:

- Mindestlänge 12 Zeichen
- Bestehend aus mindestens drei der folgenden vier Zeichenkategorien:
 - Kleinbuchstaben
 - Großbuchstaben
 - Ziffern
 - Sonderzeichen

Die maximal mögliche Länge für Passwörter sollte 25 Zeichen oder mehr betragen. Dies unterstützt Endbenutzer bei Bedarf mit mehr Freiraum, um individuelle merkbare Passwörter bilden zu können, und hilft so, unerwünschte Verhaltensweisen im Umgang mit Passwörtern zu vermeiden, welche durch stark eingeschränkte maximale Passwortlängen provoziert werden können.

PostgreSQL bietet nativ jedoch keine sichere Funktion an, um die Einhaltung der Passwortregeln zu überprüfen. Somit muss die Einhaltung entweder mittels organisatorischer Maßnahmen oder durch die Verwendung von externen Authentifizierungsmethoden sichergestellt werden.

Bei Verwendung eines zentralen Systems für die Authentisierung der Benutzer [vgl. Wurzel-Anforderungsdokument "[i]3.69 IAM (Identity Access Management) - Framework"] darf diese Aufgabe an das zentrale System weitergereicht bzw. dorthin ausgelagert werden.

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

ID: 3.60-7/10.0

Req 8	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 In-

frastrukturen 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 9	Falls Passwörter als Authentisierungsmerkmal genutzt werden, muss nach spätestens 12 Monaten ein Passwort-Wechsel erfolgen.
-------	---

Die maximal zulässige Nutzungsdauer für Passwörter beträgt 12 Monate.
Erreicht ein Passwort die maximal zulässige Nutzungsdauer, muss es geändert werden.

Das System muss hierzu den Benutzer automatisch bei dessen nächster Systemanmeldung auf die abgelaufene Nutzungsdauer hinweisen und ihn unmittelbar durch einen Passwort-Änderungsdialog führen. Der Zugriff auf das System darf ohne eine erfolgreich abgeschlossene Passwort-Änderung nicht weiter zugelassen werden.

Für technische Benutzerkonten (M2M bzw. Machine-2-Machine), die für die Authentifizierung und Autorisierung von Systemen untereinander oder von Anwendungen auf einem System genutzt werden, müssen ebenfalls automatisierte Lösungen zur Einhaltung der zulässigen Nutzungsdauer für Passwörter implementiert werden.

Falls eine automatisierte Abbildung des Prozesses zur Passwort-Änderung nicht umsetzbar ist, muss ersatzweise eine wirksame organisatorische Maßnahme angewendet werden, die eine verbindliche manuelle Passwort-Änderung bei Ablauf der zulässigen Nutzungsdauer sicherstellt.

Motivation: Passwörter sind im Gegensatz zu moderneren Authentisierungsmerkmalen leichter angreifbar. Ohne spezifische Maßnahmen einer verlässlichen technisch automatisierten Kompromittierungserkennung kann das Risiko, dass ein Passwort von einem Angreifer ermittelt oder gebrochen wird, über die Zeit deutlich steigen.

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
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.01-30/9.0

5. Least Privilege Prinzip und Rollen/Rechte Trennung

Req 10 Die Rechte SUPERUSER, CREATEROLE oder CREATEDB darf nur der Datenbankadministrator besitzen.

Motivation: Unnötige Rechte erhöhen die Gefahr einer Kompromittierung und genereller nicht gewollter Datenverluste.

Umsetzungsbeispiel: Folgender Befehl kann verwendet werden, um Rollen mit entsprechenden Rechten zu ermitteln:
`SELECT rolname,rolsuper,rolcreatorole,rolcreatedb FROM pg_roles WHERE rolsuper IS TRUE OR rolcreatorole IS TRUE or rolcreatedb IS TRUE;`

Folgende Befehle können verwendet werden, um die entsprechenden Rechte zu entfernen:

```
ALTER ROLE role_name WITH NOSUPERUSER;  
ALTER ROLE role_name WITH NOCREATOROLE;  
ALTER ROLE role_name WITH NOCREATEDB;
```

Für diese Anforderung sind folgende Bedrohungen relevant:

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

ID: 3.60-10/10.0

Req 11 Rechte an der Tabelle `pg_catalog.pg_authid` darf nur der Datenbankadministrator besitzen.

Motivation: Die `pg_catalog.pg_authid` Tabelle enthält Anmeldedaten wie Benutzername und Password.

Umsetzungsbeispiel: Folgender Befehl kann verwendet werden, um Benutzer mit Rechten an der Tabelle `pg_catalog.pg_authid` zu ermitteln:
`\dp pg_catalog.pg_authid`

Folgender Befehl kann verwendet werden, um alle Rechte an der Tabelle `pg_catalog.pg_authid` zu entfernen:
`REVOKE ALL PRIVILEGES ON pg_catalog.pg_authid FROM user_name;`

Für diese Anforderung sind folgende Bedrohungen relevant:

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

ID: 3.60-11/10.0

Req 12 Die Authentisierungsmethode "trust" darf nicht verwendet werden.

Motivation: Diese Methode erlaubt ein Login von einem Host ohne Kenntnis eines jeweiligen Passworts.

Umsetzungsbeispiel: In der "`pg_hba.conf`" Datei ist jedweder Eintrag mit "trust" auszukommentieren.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Unbemerkt durchführbare Angriffe

ID: 3.60-12/10.0

Req 13	Die Authentisierungsmethode "peer authentication" darf nur für administrative oder technische Accounts verwendet werden.
--------	--

Falls nicht vermeidbar (z. B. für technische Accounts und z. B. Wartungszwecke), ist sehr sorgsam mit diesem Feature umzugehen.

Motivation: Administrative Zugriffe müssen maximal abgesichert werden, aufgrund des Schadenspotentials.

Umsetzungsbeispiel: Die Auskommentierung oder Änderung der entsprechenden Einträge in der Datei "pg_hba.conf" von "peer" zu einer anderen Authentisierungsmethode (z. B. scram-sha-256) erfüllt die Anforderung.

Für diese Anforderung sind folgende Bedrohungen relevant:

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

ID: 3.60-13/10.0

Req 14	Die standardmäßig vergebene Berechtigung „CREATE“ auf das Schema „public“ muss allen Benutzern in allen Datenbanken entzogen werden.
--------	--

Die Standardeinstellungen von PostgreSQL Versionen vor Version 15 ermöglichen es Benutzern, durch das Erstellen gleichnamiger Objekte im „public“ Schema das Verhalten von Abfragen für andere Benutzer zu ändern (siehe CVE-2018-1058).

Hinweis: Durch ein Upgrade von einer früheren Version wird die Schwachstelle CVE-2018-1058 nicht automatisch behoben.

Motivation: Der Entzug der „CREATE“ Berechtigung für das „public“ Schema verhindert, dass ein Benutzer die in CVE-2018-1058 beschriebene Schwachstelle ausnutzen kann.

Umsetzungsbeispiel: Mit folgendem Befehl können die standardmäßig gesetzten „CREATE“ Berechtigungen für alle Benutzer in einer Datenbank entfernt werden:

`REVOKE CREATE ON SCHEMA public FROM PUBLIC;`

Mit folgendem Befehl kann in einer Datenbank überprüft werden, ob andere Benutzer als „postgres“ „CREATE“ Berechtigungen für das „public“ Schema besitzen:

```
SELECT u.username, pg_catalog.has_schema_privilege(u.username, 'public', 'CREATE') AS "create"
FROM pg_user u
WHERE u.username != 'postgres' AND pg_catalog.has_schema_privilege(u.username, 'public', 'CREATE') = 't';
```

Mit folgendem Befehlen kann überprüft werden, welche Funktionen im "public" Schema erstellt wurden:

`\df public.*`

Für diese Anforderung sind folgende Bedrohungen relevant:

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

ID: 3.60-14/10.0

Req 15	Privilegien auf Datenbankobjekte dürfen nicht an „PUBLIC“ vergeben werden.
--------	--

Es sollte beachtet werden, dass PostgreSQL beim Erstellen von Objekten die folgenden Standardrechte an „PUBLIC“ vergibt:

- "CONNECT" und "TEMPORARY" Berechtigungen für Datenbanken,
- "EXECUTE" Berechtigung für Funktionen und Prozeduren und

- "USAGE" Berechtigung für Sprachen und Datentypen (inkl. Domains).

Diese Standardberechtigungen müssen nach dem Erstellen des Objekts entzogen werden.

Die standardmäßigen Berechtigungseinstellungen können mit dem Befehl „ALTER DEFAULT PRIVILEGES“ geändert werden.

Motivation: Durch die Vergabe von Privilegien auf Datenbankobjekte an „PUBLIC“ erhalten alle Rollen/ Benutzer (auch zukünftig angelegte) diese Berechtigungen. Dies kann dazu führen, dass Benutzer unberechtigt Privilegien erhalten, die missbräuchlich verwendet werden könnten (siehe z. B. CVE-2018-1058 und CVE-2020-14349).

Umsetzungsbeispiel: Mithilfe folgender Befehle kann geprüft werden, ob Berechtigungen auf nicht Standardobjekten an "PUBLIC" vergeben wurden:

Für Datenbanken:

\l

Bei Einträgen, die mit „=“ beginnen, wurden die benannten Berechtigungen an "PUBLIC" vergeben.

Für Tabellen:

```
SELECT * FROM information_schema.table_privileges WHERE grantee = 'PUBLIC' AND table_schema != 'pg_catalog' AND table_schema != 'information_schema';
```

Für Funktionen und Prozeduren:

```
SELECT * FROM information_schema.routine_privileges WHERE grantee = 'PUBLIC' AND specific_schema != 'pg_catalog' AND specific_schema != 'information_schema';
```

Für Sprachen und Datentypen:

```
SELECT * FROM information_schema.usage_privileges WHERE grantee = 'PUBLIC' AND object_schema != 'pg_catalog' AND object_schema != 'information_schema';
```

Vergebene Berechtigungen an "PUBLIC" können mit folgendem Befehl entzogen werden:

```
REVOKE ALL ON <object> "<object_name>" FROM PUBLIC;
```

„<object>“ und „<object_name>“ müssen mit dem entsprechenden Objekttyp und Objektname ersetzt werden.

Für diese Anforderung sind folgende Bedrohungen relevant:

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

ID: 3.60-15/10.0

Req 16 Privilegien auf Objekte dürfen nicht mit "WITH GRANT OPTION" vergeben werden.

Motivation: Wenn Privilegien mit "WITH GRANT OPTION" vergeben wurden, kann der Empfänger diese Privilegien ebenfalls mit „GRANT“ an andere Benutzer vergeben. Dies kann zu einer Verletzung der Zuteilung von Rechten führen.

Umsetzungsbeispiel: Im "psql" Client kann der folgende Befehl zur Identifikation von solchen Privilegien verwendet werden (Privilegien mit „WITH GRANT OPTION“ werden dort mit "*" angezeigt.):

\dp

Mit dem folgenden Befehl können diese Berechtigungen entfernt werden:

```
REVOKE GRANT OPTION FOR ALL ON table_name FROM user_name;
```

Für diese Anforderung sind folgende Bedrohungen relevant:

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

- Unbemerkt durchführbare Angriffe

ID: 3.60-16/10.0

Req 17 SQL Funktionen mit SECURITY DEFINER müssen sehr restriktiv verwendet werden.

Motivation: Die Security Definer Funktion erlaubt es Funktionen mit den Rechten des Eigentümers auszuführen und nicht mit den Rechten des Benutzers. Dieses Verhalten ist vergleichbar mit "setuid" unter Linux Systemen.

Umsetzungsbeispiel: Die folgende Abfrage gibt Funktionen mit Security Definer zurück:

```
SELECT pg_proc.proname, pg_namespace.nsname, pg_user.username FROM pg_proc JOIN pg_namespace ON  
pg_proc.pronamespace=pg_namespace.oid JOIN pg_user ON pg_proc.proowner=pg_user.usesysid WHERE pro-  
secdef='t';
```

Folgender Befehl löscht einzelne Funktionen:

```
DROP FUNCTION function_name(param_1 TEXT, param_2 TEXT);
```

Für diese Anforderung sind folgende Bedrohungen relevant:

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

ID: 3.60-17/10.0

6. SQL Funktionen mit Betriebssystem und/oder Netzwerkaccess

Req 18 Das Flag "TRUSTED" darf nur dann bei prozeduralen Sprachen gesetzt werden, wenn diese keinen Zugriff auf die Interna des Datenbankservers oder das Dateisystem erlauben.

Durch das Setzen des Flag "TRUSTED" können auch nicht privilegierte Datenbankbenutzer die prozedurale Sprache verwenden, um Funktionen und Prozeduren zu erstellen.

Da prozedurale Sprachen innerhalb des Datenbankservers und damit mit dessen Berechtigungen ausgeführt werden, sollte das Flag "TRUSTED" nur bei Sprachen gesetzt werden, die keinen Zugriff auf die Interna des Datenbankservers oder des Dateisystems erlauben.

Standardmäßig werden in PostgreSQL die prozeduralen Sprachen „SQL“ und „PL/pgSQL“ mit dem Flag "TRUSTED" angeboten. Zusätzlich gelten auch die Sprachen „PL/TCL“ und „PL/Perl“ als vertrauenswürdig und können bei Bedarf verwendet werden. Bei anderen Sprachen darf das Flag "TRUSTED" nicht gesetzt sein.

Motivation: Prozedurale Sprachen mit dem Flag "TRUSTED" können von allen Datenbankbenutzern verwendet werden, um Funktionen und Prozeduren zu erstellen. Falls eine nicht limitierte Sprache als vertrauenswürdig gekennzeichnet wird, kann ein nicht privilegierter Benutzer mithilfe dieser Sprache auf Daten zugreifen, auf die dieser ansonsten keinen Zugriff hätte.

Umsetzungsbeispiel: Mit dem folgenden Befehl können alle Sprachen ermittelt werden, bei denen das Flag "TRUSTED" gesetzt wurde:

```
SELECT lanname AS language, lanpltrusted AS trusted FROM pg_language WHERE lanpltrusted = 't';
```

Falls das Ergebnis nicht limitierte Sprachen enthält, muss bei diesen entweder der Wert von „lanpltrusted“ auf „f“ geändert oder die Sprache entfernt werden.

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: 3.60-18/10.0

Req 19 Nur „C“ und „Internal“ dürfen als nicht vertrauenswürdige („untrusted“) prozedurale Sprachen verwendet werden.

Prozedurale Sprachen, die als „untrusted“ gekennzeichnet wurden, können nur von Datenbank-Superusern zur Erstellung von Funktionen verwendet werden. Generell sollten prozedurale Sprachen immer im „untrusted“ Modus installiert werden, wenn bei den Sprachen nicht sichergestellt werden kann, dass diese keinen Zugriff auf Daten gewähren, auf die der Benutzer ansonsten keinen Zugriff hätte.

Die prozeduralen Sprachen „C“ und „Internal“ werden in PostgreSQL standardmäßig als „untrusted“ angeboten. Falls andere „untrusted“ prozeduralen Sprachen benötigt werden, muss deren Verwendung beschrieben und begründet werden.

Motivation: Prozedurale Sprachen, die im „untrusted“ Modus installiert werden, können verwendet werden, um Zugriff auf den Datenbankserverprozess oder auf Betriebssystemebene zu erhalten. Um die Angriffsfläche zu reduzieren, sollte die Anzahl der prozeduralen Sprachen auf ein Minimum beschränkt werden.

Umsetzungsbeispiel: Mit dem folgenden Befehl können alle als „untrusted“ gekennzeichneten Sprachen ermittelt werden:

```
SELECT lanname AS language, lanpltrusted AS trusted FROM pg_language WHERE lanpltrusted = 'f';
```

Folgender Befehl entfernt eine nicht benötigte Sprache:

```
DROP LANGUAGE proclang_name;
```

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Unbemerkt durchführbare Angriffe

ID: 3.60-19/10.0

Req 20	Nicht benötigte prozedurale Sprachen müssen gelöscht werden.
--------	--

Folgende prozedurale Sprachen können verbleiben, da es sich um Standards handelt:

- internal
- c
- sql
- plpgsql

Motivation: Jede zusätzliche nicht benötigte prozedurale Sprache vergrößert unnötig die Angriffsfläche.

Umsetzungsbeispiel: Folgender Befehl listet die vorhandenen Sprachen auf:

```
SELECT lanname AS language FROM pg_language;
```

Nicht benötigte Sprachen können mit dem folgenden Befehl gelöscht werden:

```
DROP LANGUAGE plang_name;
```

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unbemerkt durchführbare Angriffe

ID: 3.60-20/10.0

7. Datenkommunikation

Req 21	In der pg_hba.conf muss jede Client-Verbindung zum Datenbankserver mittels eines eigenen Eintrags implementiert werden. Pro Verbindungseintrag darf nur eine Datenbank, ein Datenbankbenutzer und ein möglichst restriktiver IP-Adressbereich definiert werden.
--------	---

Bei der Definition der Verbindungseinträge ist die Angabe einzelner IP-Adressen der Angabe eines IP-Adressbereiches vorzuziehen. Ist dies nicht möglich, muss der IP-Adressbereich so restriktiv definiert werden, dass dieser nur Clients enthält, die den jeweiligen Zugriff benötigen.

Ausnahme: Im Verbindungseintrag für den Datenbankadministrator darf im Feld „database“ der Wert „all“ verwendet werden.

Alternativ kann diese Anforderung auch mit anderen geeigneten Filterelementen umgesetzt werden (z. B. mittels eines entsprechend konfigurierten „Connection Poolers“). In diesem Fall muss die eingesetzte Alternative benannt werden.

Motivation: Die Aufteilung von Privilegien reduziert Angriffsfläche.

Umsetzungsbeispiel: # Verbindungseintrag 1
hostssl database_1 user_1 10.23.24.123/32 scram-sha-256
Verbindungseintrag 2
hostssl database_1 user_2 10.23.24.11/32 scram-sha-256
Verbindungseintrag 3
hostssl database_2 user_1 10.23.24.123/32 scram-sha-256

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unbemerkt durchführbare Angriffe

ID: 3.60-21/10.0

Req 22	Transport Layer Security (TLS/SSL) muss in der PostgreSQL Konfiguration aktiviert und entsprechend dem Anforderungsdokument 3.50 "Kryptographische Algorithmen und Sicherheitsprotokolle" konfiguriert werden.
--------	--

Motivation: Wenn TLS/SSL auf dem Server nicht aktiviert und richtig konfiguriert ist, erhöht dies das Risiko, dass die übertragenen Daten kompromittiert werden.

Umsetzungsbeispiel: In der Datei "postgresql.conf" müssen die folgenden Parameter gesetzt sein:

```
# Changes require restart!  
# Enables SSL connections.  
ssl = on  
# Specifies the file containing the SSL server certificate.  
ssl_cert_file = '/path/to/server.crt'  
# Specifies the file containing the SSL server private key.  
ssl_key_file = '/path/to/server.key'  
# Specifies allowed SSL cipher suites  
ssl_ciphers = 'EECDH+AESGCM:EDH+AESGCM'  
# Sets the minimum allowed SSL/TLS protocol version.  
ssl_min_protocol_version = 'TLSv1.2'
```

```
password_encryption = scram-sha-256
```

Desweiteren muss jeder Eintrag für Remote-Verbindungen in der Datei "pg_hba.conf" mit dem Präfix "hostssl" beginnen, wie in dem folgenden Beispiel dargestellt:

```
hostssl database_1 user_1 10.23.24.123/32 scram-sha-256  
hostssl database_1 user_2 10.23.24.11/32 scram-sha-256
```

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten

ID: 3.60-22/10.0

8. Logging und Monitoring

Req 23	Logging Funktionen müssen angeschaltet, sauber konfiguriert und compliant sein zu dem lokalen Recht.
--------	--

Motivation: Nur über Logging Funktionen ist eine Verfolgung eines Angriffs überhaupt möglich.

Umsetzungsbeispiel: In der Datei "postgresql.conf" erfüllen folgende Einträge die technischen Anforderungen:

```
log_destination = 'syslog'
logging_collector = on
log_directory = '/path/to/log_directory'
log_truncate_on_rotation = on
log_connections = on
log_disconnections = on
log_duration = on
log_error_verbosity = default
log_hostname = on
log_line_prefix = 'db=%d,user=%u,app=%a,client=%h '
log_statement = 'ddl'
```

Falls "log_destination" einen anderen Wert als "syslog" enthält, muss "log_line_prefix" die folgenden Werte enthalten:
log_line_prefix = '%m [%p]: [%l-1] db=%d,user=%u,app=%a,client=%h '

Der Parameter "log_line_prefix" kann nach Bedarf um zusätzliche Werte erweitert werden.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.60-23/10.0

Req 24	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 Vereinba-

rungen 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, 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 Löschrufen für lokal erfasste sicherheitsrelevante Protokollierungsdaten auf einem beispielhaften TK-System 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 nach spätestens 24 Stunden

Für diese Anforderung sind folgende Bedrohungen relevant:

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

ID: 3.01-34/9.0

Req 25	Sicherheitsrelevante Protokollierungsdaten müssen umgehend nach deren Erzeugung an einen Log-Server weitergeleitet werden.
--------	--

Die sicherheitsrelevanten Protokollierungsdaten des Systems müssen umgehend nach deren Erzeugung an einen Log-Server weitergeleitet werden.

Im Sinne der Anforderung muss der dafür genutzte Log-Server die folgenden Kriterien erfüllen:

- Der Log-Server muss dediziert für die Vorhaltung von Protokollierungsdaten eingesetzt sein. Er darf keine weiteren Funktionen oder Dienste abbilden, die nicht in Relation zu dieser Aufgabe stehen. Eine zulässige Ausnahme ist die Kombination eines Log Servers mit einer SIEM (Security Information and Event Management) Lösung.
- Der Log-Server muss vom System separiert sein. Er darf nicht als zusätzliches Modul unmittelbar mit auf dem System betrieben werden.
- Der Log-Server muss zentral für die Einsatzumgebung bereitstehen, zu der auch das System gehört.

Zur Übertragung der Protokollierungsdaten an den Log-Server sollten bevorzugt auf diesen Zweck spezialisierte Standard-Protokolle (z. B. Syslog oder SNMPv3) zum Einsatz kommen.

Motivation: Werden Protokollierungsdaten nur lokal gespeichert, können diese im Fall einer erfolgreichen Kompromittierung des Systems durch einen Angreifer manipuliert werden, um dessen Angriff und auf dem System durchgeführte Aktionen zu verschleiern. Aus diesem Grund muss die Weiterleitung von Protokollierungsdaten an einen Log-Server sehr zeitnah nach dem Auftreten eines sicherheitsrelevanten Ereignisses im Quell-System erfolgen. So stehen die angefallenen Protokollierungsdaten separiert vom Quell-System und unabhängig von dessen Zustand jederzeit für Analysen bereit. Die Verwendung eines zentralen Log-Servers für eine Einsatzumgebung stellt sicher, dass auf die Logs sämtlicher IT/NT-Systeme dieser Einsatzumgebung schnell und einheitlich zugegriffen und der Zugriff selbst optimal in die Incident-Prozesse der Umgebung integriert werden kann.

Für diese Anforderung sind folgende Bedrohungen relevant:

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

ID: 3.01-35/9.0

Req 26	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

Req 27	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.
- Sicherheitsexperten sind aktiv involviert in den Betrieb der Lösung sowie die Auswertung von Alarmen.

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 oder Windows Event Log des Betriebssystems. 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.

Beispiel 3: Netzwerk-Komponenten stellen Ereignisse an ein SIEM bereit, die es ermöglichen, verdächtiges Kommunikationsverhalten wie Netzwerk-Scans, unzulässige Datenflüsse zwischen Netz-Segmenten oder aktive Verbindungen bzw. Verbindungsversuche zu verdächtigen Zielen im Internet zu entdecken. Im speziellen liefern Gateway Router aus dem CNDTAG "Netflow" Logs, während Forward Proxies Zugriffe auf Domänen und/oder IP-Adressen melden und Firewalls blockierte Verbindungsversuche loggen.

Beispiel 4: Auf Basis von Cloud-nativen Funktionen entwickelte Systeme erzeugen Protokollierungsdaten mittels der Cloud-eigenen Protokollierungs- und Überwachungsinfrastruktur direkt aus diesen Funktionen heraus. In einem solchen System wird die Erzeugung sicherheitsrelevanter Protokollierungsdaten in den entsprechenden Konfigurationsoptionen der eingesetzten Cloud-nativen Funktionen gezielt aktiviert. Die entstehenden Protokollierungsdaten werden kontinuierlich zur weiteren Analyse, Angriffserkennung und Alarmbehandlung an ein SIEM (Security Information and Event Management) übertragen. Einige Cloud-Anbieter leisten eine gewisse Angriffsmustererkennung bereits in der Cloud-eigenen Überwachungsinfrastruktur und erzeugen daraus spezifische Cloud-Alarme. Diese werden zusätzlich an das SIEM übertragen.

Beispiel 5: Ein System basiert auf einer SaaS ("Software as a Service") Applikation eines Cloud-Anbieters. Typischerweise verantwortet im SaaS Modell der Cloud-Anbieter alle operativen Aspekte der Applikation, inkl. des Sicherheitsmanagements. Der Cloud-Kunde selbst hat keinen Einfluss auf die zugrunde liegende technische Infrastruktur oder den Betrieb der Applikation. Anhand von Vertragsvereinbarungen, anerkannten Zertifikaten und dokumentierten Selbstauskünften des Cloud-Anbieters wird formell überprüft, dass der Cloud-Anbieter eine Erkennungs- und Reaktions-

onslösung für Sicherheitsvorfälle implementiert hat. Einige Cloud-Anbieter erzeugen zusätzlich dedizierte Protokollierungsdaten über sicherheitsrelevante Aktivitäten innerhalb der separaten Kunden-Mandanten ihrer SaaS Applikation (beispielsweise zu Änderungen von Konfigurationseinstellungen, Berechtigungsvergaben oder Benutzer-Konten innerhalb der Mandanten) und stellen diese den Cloud-Kunden zu Verfügung. Diese sogenannten Audit-Logs werden kontinuierlich zur weiteren Analyse, Angriffserkennung und Alarmbehandlung an ein SIEM (Security Information and Event Management) übertragen.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unbemerkt durchführbare Angriffe

ID: 3.01-37/9.0

Req 28	Logfiles müssen kontinuierlich auf Missbrauch hin untersucht werden.
--------	--

Das Monitoring kann vollautomatisch erfolgen und sollte nur automatisiert Alarmer z. B. an die Betriebseinheit schicken. Auf jeden Fall ist die Compliance zu dem nationalen Recht inkl. Betriebsrat zu beachten.

Motivation: Ohne Monitoring kann verdächtiges Verhalten nicht entdeckt werden.

Umsetzungsbeispiel: Logfiles müssen auf verdächtige Aktivitäten, wie beispielsweise Zugriffsversuche auf schutzbedürftige Tabellen (u. a. "pg_shadow" und "pg_authid"), überprüft werden.

Ein exemplarisches Skript kann wie folgt aussehen:

```
grep -r 'permission denied' /path/to/log_directory/*.log
```

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unbemerkt durchführbare Angriffe

ID: 3.60-28/10.0

Req 29	Wichtige Datenbank-Dienste und -Instanzen müssen kontinuierlich auf Missbrauchsszenarien hin überwacht werden.
--------	--

Die Überwachung von Benutzeraktionen auf Missbrauch muss in Übereinstimmung mit dem jeweils geltenden nationalen Recht erfolgen.

Es muss eine Fallstudie zur Art des Dienstes/Nutzerdatens, der Anzahl der Einträge und möglichen Missbrauchsszenarien geben, um das notwendige Maß an Überwachung der Fälle zu definieren.

- Insider-Bedrohungen
- Anomalieerkennung
- Accidental exposure
- Fehlkonfiguration
- Verwundbare Anwendung
- Machine Language Models
- Unsichere Daten im Transit
- ...

Motivation: Die missbräuchliche Nutzung von Datenbanksystemen ist vielfältig denkbar. Benutzer erzeugen eine ungewöhnlich hohe Datenlast beziehungsweise agieren zu ungewöhnlichen Uhrzeiten. Angreifer nutzen bei Datenbankabfragen unübliche und kritische Befehle sowie Tools und Schadcode, um ihre Rechte zu erweitern. Um eine missbräuchliche Nutzung zu erkennen, sollten Datenbanksysteme kontinuierlich auf Missbrauchsszenarien überwacht werden etwa durch Datenbanktrigger und Logüberwachung.

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: 3.16-32/9.0

Req 30	Massendatenexfiltration muss erkannt werden, wenn mehr als 100.000 Kundendaten und DSK2 oder DSK3 vorhanden sind.
--------	---

Die Massendatenexfiltrationserkennung ist eine kritische Kontrolle, die überwacht, ob ungewöhnlich große Datenmengen intern übertragen oder das Unternehmensnetzwerk oder die Datenbanksysteme verlassen. Wenn mehr als 100.000 Kundendaten – insbesondere als DSK2 oder DSK3 klassifiziert – abgerufen, kopiert oder übertragen werden, sollten automatisierte Alarmer ausgelöst werden. Diese schwellenwertbasierte Erkennung sollte mit SIEM- und User and Entity Behavior Analytics (UEBA)-Tools integriert werden, um Abweichungen von normalen Zugriffsmustern zu identifizieren. Das System sollte den Zugriffskontext (Benutzerrolle, Zeit, Datensensitivität und Übertragungskanal) korrelieren, um legitime Massenoperationen von potenziellen Exfiltrationsversuchen zu unterscheiden. Nach Entdeckung sollten vordefinierte Vorfalreaktions-Workflows die Aktivität sofort isolieren, das Sicherheitsoperationszentrum informieren und forensische Beweise für die Untersuchung sichern.

Motivation: Kundendaten sind eines der wertvollsten Vermögenswerte des Unternehmens. Deshalb müssen wir alle Maßnahmen ergreifen, um massenhafte Datenlecks, böswillige Handlungen oder Missbrauch zu vermeiden. Dafür müssen wir wissen, wie wir unsere Datensätze handhaben und nutzen.

Umsetzungsbeispiel: Technische Leitlinien und Implementierungsdetails werden ebenfalls auf dieser Seite bereitgestellt und kontinuierlich gepflegt:

<https://yam-united.telekom.com/pages/psa/apps/wiki/wiki-sicherheitsanforderungen/list>

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.16-33/9.0

9. Technischer Basisschutz von IT-/NT-Systemen

9.1. Authentifizierung und Autorisierung

Req 31	Die Nutzung von schutzbedürftigen Funktionen des Systems sowie der Zugriff auf interne oder vertrauliche Daten darf nicht ohne vorhergehende Authentifizierung und Autorisierung ermöglicht werden.
--------	---

Die Nutzung von schutzbedürftigen Funktionen des Systems sowie der Zugriff auf Daten, die als intern oder vertraulich eingestuft sind, darf erst nach einer eindeutigen Identifizierung und erfolgreichen Authentifizierung des Benutzers anhand eines Benutzerkontos ermöglicht werden. Ein Benutzerkonto besteht aus einem eindeutigen Benutzernamen und zugehörigen Authentisierungsmerkmalen. Zudem muss verifiziert werden, dass der Benutzer innerhalb der ihm im System zugeordneten Benutzerrolle für den Zugriff auf die betroffenen Funktionen und Daten legitimiert ist.

Eine Ausnahme hiervon sind Funktionen und Daten, die uneingeschränkt öffentlich genutzt werden dürfen; beispielsweise der Bereich einer Web-Seite im Internet, in dem nur öffentliche Informationen bereitgestellt werden.

Beispiele für Funktionen, die eine vorhergehende Authentifizierung erfordern:

- Fernzugriff auf Netzdienste (wie SSH, SFTP, Web-Dienste)
- lokaler Zugriff auf die Management-Konsole
- lokale Nutzung von Betriebssystem und Anwendungen

Diese Anforderung gilt ohne Einschränkung auch für jegliche maschinellen Zugriffe auf das System (Hier erfolgt die Umsetzung in der Regel durch den Einsatz von sogenannten "technischen Benutzerkonten", auch "Machine-2-Machine" oder "M2M"-Konten genannt).

Motivation: Die eindeutige Authentifizierung und Autorisierung von Zugriffen auf ein System sind elementar, um Funktionen und Daten vor Missbrauch zu schützen.

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: 3.01-19/9.0

Req 32	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, Gesichtserkennung

*) 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 33	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 of-

fenzulegen, 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, 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

Req 34 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 adäquate 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

Req 35 Die Berechtigungen von Benutzern und Anwendungen müssen auf den zur Erfüllung ihrer Aufgaben benötigten Umfang beschränkt werden.

Die Berechtigungen auf einem System müssen so weit eingeschränkt werden, dass ein Benutzer nur auf Daten zugreifen und Funktionen nutzen kann, die er im Rahmen seiner Arbeit benötigt. Entsprechende Berechtigungen sind auch für den Zugriff auf Dateien zu vergeben, die Bestandteil des Betriebssystems oder von Anwendungen sind oder von diesen erzeugt werden (z. B. Konfigurations- und Protokollierungsdateien).

Neben dem Zugriff auf Daten muss auch die Ausführung von Anwendungen und deren Bestandteilen mit möglichst niedrigen Berechtigungen erfolgen. Anwendungen sollten nicht mit Administrator- oder Systemberechtigungen aus-

geführt werden.

Motivation: Werden einem Benutzer zu weitreichende Berechtigungen auf einem System gewährt, kann er in einem Umfang auf Daten und Anwendungen zugreifen, der zur Erfüllung der zugewiesenen Aufgaben nicht erforderlich ist. Dies erzeugt ein unnötig erhöhtes Risiko im Falle eines Missbrauchs, insbesondere auch, falls der Benutzer oder dessen Benutzerkonto durch einen Angreifer kompromittiert werden.

Anwendungen, die über zu weitreichende Berechtigungen verfügen, können von einem Angreifer missbraucht werden, um unbefugte Zugänge zu erhalten oder zu erweitern und damit einen Zugriff auf schutzbedürftige Daten und Systembereiche zu erlangen.

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: 3.01-23/9.0